



v2022-23.00

**e3Symbol.SetAsMaster( *on* )**

## Syntax

**Integer** SetAsMaster( [in]**Integer** *on* )

## Description

Sets the symbol item as a master symbol.

## Parameters

Type	Parameter	Description
		Value to set the master status to
[in] <b>Integer</b>	<i>on</i>	If 1, the symbol is set as the master symbol
		If 0, the symbol is unset as the master symbol

## Return Values

Value	Status	Description
1	Success	Previously the symbol was set as the master symbol
0	Success	Previously the symbol was not set as the master symbol
-1	Error	Error occurred

## Remarks

This function works on symbol items assigned to a device. The device item should have no component assigned to it but contain more than one symbol.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting gates in the project tree.

## Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set symbol = job.CreateSymbolObject()

Dim isMaster : isMaster = 1

symbolCount = job.GetTreeSelectedSymbolIds( symbolIds )           'get currently selected sym
If symbolCount > 0 Then
    For symbolIndex = 1 To symbolCount           'loop through selected symbols

        symbolId = symbol.SetId( symbolIds( symbolIndex ) )
        result = symbol.SetAsMaster( isMaster )
        Select Case result
        Case 1
            message = "Symbol " & symbolId & " set as master"
        Case 0
            message = "Symbol " & symbolId & " set as master"
        Case -1
            message = "Error occurred sending symbol " & symbolId & " as master"
        End Select
        e3Application.PutInfo 0, message           'output result of operation

    Next
End If

Set symbol = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

## See Also

- [e3Symbol - Overview](#)

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Symbol.SetAttributeValue( *name*, *value* )**

## Syntax

**Integer** SetAttributeValue( [in]**String** *name*, [in]**String** *value* )

## Description

Sets the symbol item's specified attribute value.

## Parameters

Type	Parameter	Description
[in] <b>String</b>	<i>name</i>	Name of the attribute
[in] <b>String</b>	<i>value</i>	Value of the attribute

## Return Values

Value	Status	Description
> 0	Success	Identifier of attribute
0	Failure	Error occurred

## Remarks

*name* must be the name of an existing attribute.

*value* maximum length is 252 characters.

A valid symbol item identifier value must be assigned using SetId(), otherwise 0 is returned.

Since v2014-1400 the value of 0 is returned if the symbol is locked or the attribute is not defined as changeable.

Adding a new attribute to the symbol item cannot be done using this function.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting symbols on a sheet using the given attribute.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set symbol = job.CreateSymbolObject()

Dim attributeName : attributeName = "Example"           'attribute should exist.
Dim attributeValue : attributeValue = "Legatus nec violatur, nec laeditur"

symbolCount = job.GetSelectedSymbolIds( symbolIds )     'get currently selected symbols
If symbolCount > 0 Then
    For symbolIndex = 1 To symbolCount

        symbolId = symbol.SetId( symbolIds( symbolIndex ) )
        If symbolId > 0 Then

            result = symbol.SetAttributeValue( attributeName, attributeValue )

            If result = 0 Then
                message = "Symbol " & symbolId & "; Attribute: " & attributeName & " value"
            Else
                message = "Symbol " & symbolId & "; Attribute: " & attributeName & " value"
            End If
            e3Application.PutInfo 0, message              'output result of operation

        End If
    Next
End If

Set symbol = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

Modified in v2014-14.00.

---

## See Also

- [e3Symbol - Overview](#)

- GetAttributeValue()
  - HasAttribute()
  - SetId()
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Symbol.SetBlockHatchColour( *value* )**

## Syntax

**Integer** SetBlockHatchColour( [in]**Integer** *value* )

## Description

Sets the symbol item's block hatch color.

## Parameters

Type	Parameter	Description
		Block hatch color value
[in] <b>Integer</b>	value	See <a href="#">Colors</a> for possible values

## Return Values

Value	Status	Description
0..255	Success	Previous block hatch color value See <a href="#">Colors</a> for possible values
-1	Inconclusive	Previous block hatch color was automatic or an error occurred

## Remarks

Due caution is recommended on relying on the return value of 0 meaning the previous block hatch color was automatic since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting symbols on a sheet.

Visual Basic Script

Examples

## SetAsMaster - e3Symbol

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set symbol = job.CreateSymbolObject()

Dim color : color = 128          'new color to apply; in this case 0, 218, 85 (greenish)

symbolCount = job.GetSelectedSymbolIds( symbolIds )          'get currently selected symbols
If symbolCount > 0 Then
    For symbolIndex = 1 To symbolCount          'loop through the selected symbols

        symbolId = symbol.SetId( symbolIds( symbolIndex ) )
        If symbolId > 0 Then

            result = symbol.SetBlockHatchColour( color )
            If result = -1 Then
                message = "Symbol " & symbolId & " block hatch color set from automatic to " & color
            Else
                message = "Symbol " & symbolId & " block hatch color set from " & result & " to " & color
            End If
            e3Application.PutInfo 0, message          'output result of operation

        End If
    Next
End If

Set symbol = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3Symbol - Overview](#)
  - [Colors](#)
  - [GetBlockHatchColour\(\)](#)
  - [IsBlock\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Symbol.SetBlockHatchDistance( value )**

## Syntax

*Double* SetBlockHatchDistance( [in]*Double* value )

## Description

Sets the symbol item's block hatch pattern distance.

## Parameters

Type	Parameter	Description
[in] <i>Double</i>	value	Block hatch pattern distance value

## Return Values

Value	Status	Description
$\geq 0.1$	Success	Previous block hatch pattern distance value
-1.0	Inconclusive	Previously no block hatch pattern assigned or an error occurred

## Remarks

This function operation is only valid for block symbol items.

The symbol item's block hatch pattern distance is in project measurement units.

Due caution is recommended on relying on the return value of -1.0 meaning previously there was no block hatch pattern assigned since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting symbol items on a sheet.



## Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set symbol = job.CreateSymbolObject()

Dim distance : distance = 2.0

symbolCount = job.GetSelectedSymbolIds( symbolIds )           'get currently selected symbols
If symbolCount > 0 Then
    For symbolIndex = 1 To symbolCount                         'loop through the selected symbols

        symbolId = symbol.SetId( symbolIds( symbolIndex ) )
        If symbolId > 0 Then

            result = symbol.SetBlockHatchDistance( distance )
            If result = -1 Then
                message = "Symbol " & symbolId & " block hatch pattern distance set from r
            Else
                message = "Symbol " & symbolId & " block hatch pattern distance set from "
            End If
            e3Application.PutInfo 0, message                     'output result of operation

        End If
    Next
End If

Set symbol = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

## See Also

- [e3Symbol - Overview](#)
- [GetBlockHatchDistance\(\)](#)
- [IsBlock\(\)](#)

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Symbol.SetBlockHatchPattern( *value*, *angle1*, *angle2* )**

## Syntax

**Integer** SetBlockHatchPattern( [in]**Integer** *value*, [in]**Double** *angle1*, [in]**Double** *angle2* )

## Description

Sets the symbol item's block hatch pattern.

## Parameters

Type	Parameter	Description
[in] <b>Integer</b>	<i>value</i>	Block hatch pattern to apply Angle value in degrees of a hatch line to apply
[in] <b>Double</b>	<i>angle1</i>	Value is only used if <i>value</i> is a line or cross hatch pattern value  The value range is from 90.0 (pointing upwards) to 0.0 (pointing rightwards) to -90.0 (pointing downwards) Angle value in degrees of a hatch line to apply
[in] <b>Double</b>	<i>angle2</i>	Value is only used if <i>value</i> is a cross hatch pattern value  The value range is from 90.0 (pointing upwards) to 0.0 (pointing rightwards) to -90.0 (pointing downwards)

## Return Values

Value	Status	Description
0	Inconclusive	No previous hatch pattern value set or error occurred
1, 2 or 4	Success	Previous hatch pattern value

## Remarks

The set of possible values for *value* and the successful return value are the following:

Value	Description
0	No hatch pattern
1	Solid hatch
2	Line hatch
4	Cross hatch

A value of 0 is returned if the block symbol item's type does not support the hatch functionality.

The 0 angle point for *angle1* and *angle2* is at 3 o'clock (on the right hand side of an imaginary circle). The range of the angle is from 90.0 (pointing upwards) to -90.0 (pointing downwards).

Due caution is recommended on relying on the return value of 0 meaning there is no block hatch pattern since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting block symbols on a sheet.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set symbol = job.CreateSymbolObject()

Const NO_HATCH = 0
Const SOLID_HATCH = 1
Const LINE_HATCH = 2
Const CROSS_HATCH = 4

Dim hatchPattern : hatchPattern = CROSS_HATCH
Dim hatchAngle1 : hatchAngle1 = 30.0
Dim hatchAngle2 : hatchAngle2 = -30.0

symbolCount = job.GetSelectedSymbolIds( symbolIds )           'get currently selected symbols
If symbolCount > 0 Then
    For symbolIndex = 1 To symbolCount                        'loop through the selected symbols
        symbolId = symbol.SetId( symbolIds( symbolIndex ) )
        If symbolId > 0 Then
    
```

## SetAsMaster - e3Symbol

```
result = symbol.SetBlockHatchPattern( hatchPattern, hatchAngle1, hatchAngle2 )
Select Case result
Case NO_HATCH
    currentHatchPattern = symbol.GetBlockHatchPattern( angle1, angle2 )
    If currentHatchPattern = 0 Then
        message = "Symbol " & symbolId & ": error setting hatch pattern"
    Else
        message = "Symbol " & symbolId & ": hatch pattern set from no hatch to " & hatchPattern
    End If

Case SOLID_HATCH
    message = "Symbol " & symbolId & ": hatch pattern set from solid hatch to " & hatchPattern
Case LINE_HATCH
    message = "Symbol " & symbolId & ": hatch pattern set from line hatch to " & hatchPattern
Case CROSS_HATCH
    message = "Symbol " & symbolId & ": hatch pattern set from cross hatch to " & hatchPattern
End Select
e3Application.PutInfo 0, message           'output result of operation

End If
Next
End If

Set symbol = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3Symbol - Overview](#)
  - [GetBlockHatchPattern\(\)](#)
  - [IsBlock\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Symbol.SetBlockHatchStyle( *value* )**

## Syntax

*Integer* SetBlockHatchStyle( [in]*Integer* *value* )

## Description

Sets the symbol item's block hatch line style.

## Parameters

Type	Parameter	Description
[in] <i>Integer</i>	<i>value</i>	Hatch line style value

## Return Values

Value	Status	Description
1..47	Success	Previous hatch line style value See <a href="#">Line Styles</a> for possible values
0 or >47	Success	Previous hatch line style value
-1	Failure	Error occurred

## Remarks

This function operation is only valid for block symbol items.

*value* and successful return values outside of the range of line style values represent a single line.

A value of -1 is returned if the symbol item's type does not support the hatch functionality.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting block symbols on a sheet.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set symbol = job.CreateSymbolObject()

Dim lineStyle : lineStyle = 27          'Big dashes

symbolCount = job.GetSelectedSymbolIds( symbolIds )          'get currently selected symbols
If symbolCount > 0 Then
    For symbolIndex = 1 To symbolCount          'loop through the selected symbols

        symbolId = symbol.SetId( symbolIds( symbolIndex ) )
        If symbolId > 0 Then

            result = symbol.SetBlockHatchStyle( lineStyle )
            If result = -1 Then
                message = "Symbol: " & symbolId & "; Error supplying block hatch line style"
            Else
                message = "Symbol: " & symbolId & ". Block hatch line style value set from"
            End If
            e3Application.PutInfo 0, message          'output result of operation

        End If
    Next
End If

Set symbol = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3Symbol - Overview](#)
- [Line Styles](#)
- [GetBlockHatchStyle\(\)](#)

- IsBlock()
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Symbol.SetBlockHatchWidth( *value* )**

## Syntax

*Double* SetBlockHatchWidth( in]*Double* *value* )

## Description

Sets the symbol item's block hatch line width.

## Parameters

Type	Parameter	Description
[in] <i>Double</i>	<i>value</i>	Hatch line width value to apply

## Return Values

Value	Status	Description
$\geq 0.1$	Success	Previous block hatch line width value
-1.0	Inconclusive	No block hatch line width assigned or an error occurred

## Remarks

This function operation is only valid for block symbol items.

The symbol item's block hatch line width is in project measurement units.

Due caution is recommended on relying on the return value of -1.0 meaning there is no block hatch line width is assigned since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting symbols on a sheet.

Visual Basic Script

Examples



## SetAsMaster - e3Symbol

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set symbol = job.CreateSymbolObject()

Dim hatchLineWidth : hatchLineWidth = 1.2          'in project measurement units

symbolCount = job.GetSelectedSymbolIds( symbolIds ) 'get currently selected symbols
If symbolCount > 0 Then
    For symbolIndex = 1 To symbolCount              'loop through the selected symbols

        symbolId = symbol.SetId( symbolIds( symbolIndex ) )
        If symbolId > 0 Then

            result = symbol.SetBlockHatchWidth( hatchLineWidth ) 'set the hatch line width
            If result = 0 Then
                message = "Symbol: " & symbolId & ". Hatch line width value could not be set"
            Else
                message = "Symbol: " & symbolId & ". Hatch line width value set from " & hatchLineWidth
            End If
            e3Application.PutInfo 0, message          'output result of operation

        End If
    Next
End If

Set symbol = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3Symbol - Overview](#)
  - [GetBlockHatchWidth\(\)](#)
  - [IsBlock\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Symbol.SetBlockOutlineColour( *value* )**

## Syntax

**Integer** SetBlockOutlineColour( [in]**Integer** *value* )

## Description

Sets the symbol item's block outline color.

## Parameters

Type	Parameter	Description
		Block outline color value
[in] <b>Integer</b>	<i>value</i>	See <a href="#">Colors</a> for possible values

## Return Values

Value	Status	Description
0..255	Success	Previous block outline color value See <a href="#">Colors</a> for possible values
-1	Inconclusive	Previous block outline color was automatic or an error occurred

## Remarks

Due caution is recommended on relying on the return value of 0 meaning the previous block outline color was automatic since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting block symbols on a sheet.

Visual Basic Script

Examples

## SetAsMaster - e3Symbol

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set symbol = job.CreateSymbolObject()

Dim color : color = 128          'new color to apply; in this case 0, 218, 85 (greenish)

symbolCount = job.GetSelectedSymbolIds( symbolIds )          'get currently selected symbols
If symbolCount > 0 Then
    For symbolIndex = 1 To symbolCount          'loop through the selected symbols

        symbolId = symbol.SetId( symbolIds( symbolIndex ) )
        If symbolId > 0 Then

            result = symbol.SetBlockOutlineColour( color )
            If result = -1 Then
                message = "Symbol " & symbolId & " block outline color set from automatic"
            Else
                message = "Symbol " & symbolId & " block outline color set from " & result
            End If
            e3Application.PutInfo 0, message          'output result of operation

        End If
    Next
End If

Set symbol = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3Symbol - Overview](#)
  - [Colors](#)
  - [GetBlockOutlineColour\(\)](#)
  - [IsBlock\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Symbol.SetBlockOutlineStyle( *value* )**

## Syntax

*Integer* SetBlockOutlineStyle( [in]*Integer* *value* )

## Description

Sets the symbol item's block outline line style.

## Parameters

Type	Parameter	Description
[in] <i>integer</i>	<i>value</i>	Outline block line style value

## Return Values

Value	Status	Description
1..47	Success	Previous block outline line style value See <a href="#">Line Styles</a> for possible values
0 or >47	Success	Previous block outline line style value
-1	Failure	Error occurred

## Remarks

This function operation is only valid for block symbol items.

*value* and successful return values outside of the range of block outline line style values represent a single line.

The value of -1 is returned if the symbol item's type does not support the outline functionality.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting block symbols on a sheet.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set symbol = job.CreateSymbolObject()

Dim lineStyle : lineStyle = 27          'Big dashes

symbolCount = job.GetSelectedSymbolIds( symbolIds )          'get currently selected symbols
If symbolCount > 0 Then
    For symbolIndex = 1 To symbolCount          'loop through the selected symbols

        symbolId = symbol.SetId( symbolIds( symbolIndex ) )
        If symbolId > 0 Then

            result = symbol.SetBlockOutlineStyle( lineStyle )
            If result = -1 Then
                message = "Symbol: " & symbolId & "; Error supplying block outline line st
            Else
                message = "Symbol: " & symbolId & ". Block outline line style value set fr
            End If
            e3Application.PutInfo 0, message          'output result of operation

        End If
    Next
End If

Set symbol = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3Symbol - Overview](#)
- [Line Styles](#)
- [GetBlockOutlineStyle\(\)](#)

- IsBlock()
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Symbol.SetBlockOutlineWidth( *value* )**

## Syntax

*Double* SetBlockOutlineWidth( in]*Double* *value* )

## Description

Sets the symbol item's block outline line width.

## Parameters

Type	Parameter	Description
[in] <i>Double</i>	<i>value</i>	Outline line width value to apply

## Return Values

Value	Status	Description
$\geq 0.1$	Success	Previous block outline line width value
-1.0	Inconclusive	No block outline line width assigned or an error occurred

## Remarks

This function operation is only valid for block symbol items.

The symbol item's block outline line width is in project measurement units.

Due caution is recommended on relying on the return value of -1.0 meaning there is no block outline line width assigned since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting block symbols on a sheet.

Visual Basic Script

Examples

## SetAsMaster - e3Symbol

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set symbol = job.CreateSymbolObject()

Dim outlineWidth : outlineWidth = 2.2          'in project measurement units

symbolCount = job.GetSelectedSymbolIds( symbolIds )      'get currently selected symbols
If symbolCount > 0 Then
    For symbolIndex = 1 To symbolCount          'loop through the selected symbols

        symbolId = symbol.SetId( symbolIds( symbolIndex ) )
        If symbolId > 0 Then

            result = symbol.SetBlockOutlineWidth( outlineWidth )      'set the outline w
            If result = 0 Then
                message = "Symbol: " & symbolId & ". Block outline width value could not b
            Else
                message = "Symbol: " & symbolId & ". Block outline width value set from "
            End If
            e3Application.PutInfo 0, message          'output result of operation

        End If
    Next
End If

Set symbol = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3Symbol - Overview](#)
  - [GetBlockOutlineWidth\(\)](#)
  - [IsBlock\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).





v2022-23.00

**e3Symbol.SetCharacteristic( *characteristic* )**

## Syntax

**Integer** SetCharacteristic( [in]**String** *characteristic* )

## Description

Sets the symbol item's characteristic value.

## Parameters

Type	Parameter	Description
[in] <b>String</b>	<i>characteristic</i>	New characteristic value

## Return Values

Value	Status	Description
1	Success	Characteristic value applied
0	Failure	Error occurred

## Remarks

A return value of 1 is returned if the *characteristic* value is the same as the symbol item's currently assigned characteristic value.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting symbol items on a sheet.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set symbol = job.CreateSymbolObject()
```

## SetAsMaster - e3Symbol

```
symbolCount = job.GetSelectedSymbolIds( symbolIds )           'get currently selected symbols
If symbolCount > 0 Then
    For symbolIndex = 1 To symbolCount                       'loop through the selected symbols

        symbolId = symbol.SetId( symbolIds( symbolIndex ) )
        If symbolId > 0 Then

            characteristicCount = symbol.GetValidCharacteristics( validCharacteristics )
            If characteristicCount = 0 Then
                e3Application.PutInfo 0, "Symbol " & symbolId & " has no valid characteristics"
            Else
                result = symbol.SetCharacteristic( validCharacteristics( 1 ) )
                If result = 0 Then
                    message = "Symbol " & symbolId & ": Error setting characteristic"
                Else
                    message = "Symbol " & symbolId & ": Characteristic set to " & validCharacteristics( 1 )
                End If
                e3Application.PutInfo 0, message               'output result of operation
            End If
        End If
    Next
End If

Set symbol = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3Symbol - Overview](#)
  - [GetCharacteristics\(\)](#)
  - [GetValidCharacteristics\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Symbol.SetDBTextSize( *txtsiz* )**

## Syntax

**Integer** SetDBTextSize( [in]**Boolean** *txtsiz* )

## Description

Sets a value indicating if the text size factor defined in the database is used for the symbol's text items when scaling.

## Parameters

Type	Parameter	Description
		New text size factor usage value
[in] <b>Boolean</b>	<i>txtsiz</i>	<p>If <b>True</b>, the text size factor defined in the database is used for the symbol's text items when scaling</p> <p>If <b>False</b>, the text size factor defined in the database is not used for the symbol's text items when scaling</p>

## Return Values

Value	Status	Description
1	Success	Previously the text size factor defined in the database was used for the symbol's text items when scaling
0	Inconclusive	Previously the text size factor defined in the database was not used for the symbol's text items when scaling or an error occurred

## Remarks

This value is shown in **E<sup>3</sup>.series** in the **Symbol Properties** dialog in **Display** ☐ **Scaling factor** ☐ **Text size defined in database** checkbox.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting symbols on a sheet.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set symbol = job.CreateSymbolObject()

Dim useDatabaseTextSize : useDatabaseTextSize = 1

symbolCount = job.GetSelectedSymbolIds( symbolIds )           'get currently selected symbols
If symbolCount > 0 Then
    For symbolIndex = 1 To symbolCount                         'loop through the selected symbols

        symbolId = symbol.SetId( symbolIds( symbolIndex ) )
        If symbolId > 0 Then

            result = symbol.SetDBTextSize( useDatabaseTextSize )
            If result = 0 Then
                currentDataBaseTextSizeUseage = symbol.GetDBTextSize()
                If currentDataBaseTextSizeUseage = useDatabaseTextSize Then
                    message = "Symbol " & symbolId & " usage of the text size factor defin
                Else
                    message = "Symbol " & symbolId & ": Error setting usage of the text si
                End If
            Else
                message = "Symbol " & symbolId & " usage of the text size factor defined i
            End If
            e3Application.PutInfo 0, message                    'output result of operation

        End If
    Next
End If

Set symbol = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3Symbol - Overview](#)
  - [GetDBTextSize\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Symbol.SetDeviceAssignment( *name* )**

## Syntax

**Integer** SetDeviceAssignment( [in]**String** *name* )

## Description

Sets the symbol item's higher level assignment value.

## Parameters

Type	Parameter	Description
[in] <b>String</b>	<i>name</i>	New higher level assignment value

## Return Values

Value	Status	Description
1	Success	Higher level assignment value assigned
0	Inconclusive	Higher level assignment value assigned or an error occurred

## Remarks

The assigned higher level assignment value is prefixed with "=" if *name* omits it.

Changing the higher level assignment will cause the symbol item to be assigned to an existing device whenever possible. If no device exists, the creation of a new device is attempted.

The higher level assignment value is assigned and 0 is returned under one of the following conditions:

- There is no device existing with the corresponding device name, higher level assignment and location
- There is no existing compatible gate available

Due caution is recommended on relying on the return value of 0 meaning a new higher level assignment value has been applied since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>**.series project and selecting symbols on a sheet.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set symbol = job.CreateSymbolObject()

Dim higherLevelAssignment : higherLevelAssignment = "=Shino"

symbolCount = job.GetSelectedSymbolIds( symbolIds )           'get currently selected symbols
If symbolCount > 0 Then
    For symbolIndex = 1 To symbolCount

        symbolId = symbol.SetId( symbolIds( symbolIndex ) )
        If symbolId > 0 Then

            result = symbol.SetDeviceAssignment( higherLevelAssignment )
            If result = 0 Then
                message = "Symbol " & symbolId & ": Higher level assignment set to " & hi
            Else
                message = "Symbol " & symbolId & ": Device higher level assignment set to
            End If
            e3Application.PutInfo 0, message                    'output result of operation

        End If
    Next
End If

Set symbol = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

## See Also

- [e3Symbol - Overview](#)

- SetDeviceAssignment()
  - SetDeviceLocation()
  - SetDeviceName()
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).





v2022-23.00

**e3Symbol.SetDeviceCompleteName( *name*, *ass*, *loc*, *onlygiven* )**

## Syntax

**Integer** SetDeviceCompleteName( [in]**String** *name*, [in]**String** *ass*, [in]**String** *loc*, [in][optional]**Boolean** *onlygiven* )

## Description

Sets all elements of the symbol item's device name value.

## Parameters

Type	Parameter	Description
[in] <b>String</b>	<i>name</i>	New device designation value
[in] <b>String</b>	<i>ass</i>	New higher level assignment value
[in] <b>String</b>	<i>loc</i>	New location value
		Indicates whether empty values in <i>name</i> , <i>ass</i> and <i>loc</i> should be ignored
[in][optional] <b>Boolean</b>	<i>onlygiven</i>	If <b>True</b> , empty values are ignored
		If <b>False</b> , empty values are set
		Default value is <b>True</b>

## Return Values

Value	Status	Description
1	Success	Device name values assigned
0	Inconclusive	New device name values assigned or an error occurred

## Remarks

Changing the device name will cause the symbol item to be assigned to an existing device whenever possible. If no device exists, the creation of a new device is attempted.

Prefix characters will be defined automatically in front of the individual name elements if they are not contained in the new name.

To assign a shield symbol to a shield within a cable, the shield's name should be appended to the *name* value, delimited by a double-colon (::).

The device name value are assigned and 0 is returned under one of the following conditions:

- There is no device existing with the corresponding device name, higher level assignment and location
- There is no existing compatible gate available

Due caution is recommended on relying on the return value of 0 meaning the device name values have been applied since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting symbols on a sheet.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set symbol = job.CreateSymbolObject()

Dim deviceDesignation : deviceDesignation = "-Manzo"
Dim higherLevelAssignment : higherLevelAssignment = "=Shino"
Dim location : location = "+Katsushiro"
Dim useOnlyGivenValues : useOnlyGivenValues = 1

symbolCount = job.GetSelectedSymbolIds( symbolIds )           'get currently selected symbols
If symbolCount > 0 Then
    For symbolIndex = 1 To symbolCount

        symbolId = symbol.SetId( symbolIds( symbolIndex ) )
        If symbolId > 0 Then

            result = symbol.SetDeviceCompleteName( deviceDesignation, higherLevelAssignment, location )
            If result = 0 Then
                message = "Symbol " & symbolId & ": Error setting device to " & deviceDesignation
            End If
        End If
    Next
End If

```

## SetAsMaster - e3Symbol

```
Else
    message = "Symbol " & symbolId & ": Device set to " & deviceDesignation &
End If
e3Application.PutInfo 0, message          'output result of operation

End If
Next
End If

Set symbol = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3Symbol - Overview](#)
  - [SetDeviceAssignment\(\)](#)
  - [SetDeviceLocation\(\)](#)
  - [SetDeviceName\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Symbol.SetDeviceLocation( *name* )**

## Syntax

**Integer** SetDeviceLocation( [in]**String** *name* )

## Description

Sets the symbol item's location value.

## Parameters

Type	Parameter	Description
[in] <b>String</b>	<i>name</i>	New location value

## Return Values

Value	Status	Description
1	Success	Location value assigned
0	Inconclusive	Location value assigned or an error occurred

## Remarks

The assigned location value is prefixed with "+" if *name* omits it.

Changing the location will cause the symbol item to be assigned to an existing device whenever possible. If no device exists, the creation of a new device is attempted.

The location value is assigned and 0 is returned under one of the following conditions:

- There is no device existing with the corresponding device name, higher level assignment and location
- There is no existing compatible gate available

Due caution is recommended on relying on the return value of 0 meaning a location value has been applied since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting symbols on a sheet.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set symbol = job.CreateSymbolObject()

Dim location : location = "+Manzo"

symbolCount = job.GetSelectedSymbolIds( symbolIds )           'get currently selected symbols
If symbolCount > 0 Then
    For symbolIndex = 1 To symbolCount

        symbolId = symbol.SetId( symbolIds( symbolIndex ) )
        If symbolId > 0 Then

            result = symbol.SetDeviceLocation( location )
            If result = 0 Then
                message = "Symbol " & symbolId & ": Location set to " & location
            Else
                message = "Symbol " & symbolId & ": Device location set to " & location
            End If
            e3Application.PutInfo 0, message                    'output result of operation

        End If
    Next
End If

Set symbol = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3Symbol - Overview](#)
- [SetDeviceAssignment\(\)](#)
- [SetDeviceCompleteName\(\)](#)

- SetDeviceName()
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Symbol.SetDeviceName( *name* )**

## Syntax

**Integer** SetDeviceName( [in]**String** *name* )

## Description

Sets the symbol item's device designation value.

## Parameters

Type	Parameter	Description
[in] <b>String</b>	<i>name</i>	New device designation value

## Return Values

Value	Status	Description
1	Success	Device designation value assigned
0	Inconclusive	Device designation value assigned or an error occurred

## Remarks

The assigned device designation value is prefixed with "-" if *name* omits it.

Changing the device designation will cause the symbol item to be assigned to an existing device whenever possible. If no device exists, the creation of a new device is attempted.

The device designation value is assigned and 0 is returned under one of the following conditions:

- There is no device existing with the corresponding device name, higher level assignment and location
- There is no existing compatible gate available

Due caution is recommended on relying on the return value of 0 meaning a device designation value has been applied since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>**.series project and selecting symbols on a sheet.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set symbol = job.CreateSymbolObject()

Dim deviceDesignation : deviceDesignation = "-Katsushiro"

symbolCount = job.GetSelectedSymbolIds( symbolIds )      'get currently selected symbols
If symbolCount > 0 Then
    For symbolIndex = 1 To symbolCount

        symbolId = symbol.SetId( symbolIds( symbolIndex ) )
        If symbolId > 0 Then

            result = symbol.SetDeviceName( deviceDesignation )
            If result = 0 Then
                message = "Symbol " & symbolId & ": Device designation set to " & deviceDesignation
            Else
                message = "Symbol " & symbolId & ": Device designation set to " & deviceDesignation
            End If
            e3Application.PutInfo 0, message                'output result of operation

        End If
    Next
End If

Set symbol = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

## See Also

- [e3Symbol - Overview](#)
- [SetDeviceAssignment\(\)](#)



- SetDeviceCompleteName()
  - SetDeviceLocation()
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Symbol.SetDisplayLength( *length* )**

## Syntax

**Integer** SetDisplayLength( [in]**Double** *length* )

## Description

Sets the symbol item's display length.

## Parameters

Type	Parameter	Description
[in] <b>Double</b>	<i>length</i>	Display length value

## Return Values

Value	Status	Description
1	Success	Display length value is assigned
0	Failure	Error occurred

## Remarks

The display length value is relevant to symbols with their symbol graphic value set to replicate or dynamic. Setting the symbol graphic can be done using the Symbol Properties dialog in the **E<sup>3</sup>** Database Editor.

*length* should be in project measurement units.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting symbols on a sheet.

Visual Basic Script

## SetAsMaster - e3Symbol

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set symbol = job.CreateSymbolObject()

Dim length : length = 50.0

symbolCount = job.GetSelectedSymbolIds( symbolIds )      'get currently selected symbols
If symbolCount > 0 Then
    For symbolIndex = 1 To symbolCount

        symbolId = symbol.SetId( symbolIds( symbolIndex ) )
        If symbolId > 0 Then

            result = symbol.SetDisplayLength( length )
            If result = 0 Then
                message = "Symbol " & symbolId & ": Error setting symbol length"
            Else
                message = "Symbol " & symbolId & ": length set to " & length
            End If
            e3Application.PutInfo 0, message                'output result of operation

        End If
    Next
End If

Set symbol = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3Symbol - Overview](#)
  - [GetDisplayLength\(\)](#)
  - [SetDisplayWidth\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Symbol.SetDisplayWidth( *width* )**

## Syntax

**Integer** SetDisplayWidth( [in]**Double** *width* )

## Description

Sets the symbol item's display width.

## Parameters

Type	Parameter	Description
[in] <b>Double</b>	<i>width</i>	Display width value

## Return Values

Value	Status	Description
1	Success	Display width value is assigned
0	Failure	Error occurred

## Remarks

The display width value is relevant to symbols with their symbol graphic value set to replicate or dynamic. Setting the symbol graphic can be done using the Symbol Properties dialog in the **E<sup>3</sup>** Database Editor.

*width* should be in project measurement units.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting symbols on a sheet.

Visual Basic Script

## SetAsMaster - e3Symbol

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set symbol = job.CreateSymbolObject()

Dim width : width = 50.0

symbolCount = job.GetSelectedSymbolIds( symbolIds )      'get currently selected symbols
If symbolCount > 0 Then
    For symbolIndex = 1 To symbolCount

        symbolId = symbol.SetId( symbolIds( symbolIndex ) )
        If symbolId > 0 Then

            result = symbol.SetDisplayWidth( width )
            If result = 0 Then
                message = "Symbol " & symbolId & ": Error setting symbol width"
            Else
                message = "Symbol " & symbolId & ": width set to " & width
            End If
            e3Application.PutInfo 0, message              'output result of operation

        End If
    Next
End If

Set symbol = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3Symbol - Overview](#)
  - [GetDisplayWidth\(\)](#)
  - [SetDisplayLength\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Symbol.SetGateId( *symid* )**

## Syntax

**Integer** SetGateId( [in]**Integer** *symid* )

## Description

Sets a gate as the current symbol item.

## Parameters

Type	Parameter	Description
[in] <b>Integer</b>	<i>symid</i>	Unique value identifying a gate item

## Return Values

Value	Status	Description
> 0	Success	Current gate item identifier
0	Failure	Error occurred

## Remarks

*symid* will remain the current symbol item until it is replaced.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting gate items in the project tree.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set symbol = job.CreateSymbolObject()
```

## SetAsMaster - e3Symbol

```
symbolCount = job.GetTreeSelectedSymbolIds( symbolIds )           'get selected symbols in th
If symbolCount > 0 Then
    For symbolIndex = 1 To symbolCount

        result = symbol.SetGateId( symbolIds( symbolIndex ) )
        If result = 0 Then
            message = "No symbol gate item is set"
        Else
            message = "Symbol gate " & result & " has been set"
        End If
        e3Application.PutInfo 0, message           'output result of operation
    Next
End If

Set symbol = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3Symbol - Overview](#)
  - [GetId\(\)](#)
  - [GetGateId\(\)](#)
  - [SetId\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Symbol.SetGID( *gid* )**

## Syntax

*String* SetGID( [in]*String* *gid* )

## Description

Sets a symbol item as the current item.

## Parameters

Type	Parameter	Description
[in] <i>String</i>	<i>gid</i>	Global identifier value of a symbol item

## Return Values

Value	Status	Description
"<GID>"	Success	Global identifier of the current symbol item
"<Empty>"	Failure	No symbol item

## Remarks

Global identifiers (GIDs) are used for item identification in multiuser projects.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting symbols on a sheet.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set symbol = job.CreateSymbolObject()
```



## SetAsMaster - e3Symbol

```
Set gidList = CreateObject( "System.Collections.ArrayList" )

symbolCount = job.GetSelectedSymbolIds( symbolIds )           'get currently selected symbols
If symbolCount > 0 Then
    For symbolIndex = 1 To symbolCount

        symbol.SetId symbolIds( symbolIndex )
        gidId = symbol.GetGID()
        gidList.Add gidId

    Next
End If

For Each gidId in gidList
    result = symbol.SetGID( gidId )
    If Len( "" & result ) = 0 Then
        message = "No symbol item is set"
    Else
        symbolId = symbol.GetId()
        symbolName = symbol.GetName()
        message = "Symbol " & symbolName & " ( " & symbolId & " ) has been set using GID "
    End If
    e3Application.PutInfo 0, message           'output result of operation
Next

Set gidList = Nothing
Set symbol = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2022-23.00.

---

## See Also

- [e3Symbol - Overview](#)
  - [GetGID\(\)](#)
  - [SetId\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Symbol.SetId( *id* )**

## Syntax

*Integer* SetId( [in]*Integer id* )

## Description

Sets a symbol item as the current item.

## Parameters

Type	Parameter	Description
[in] <i>Integer</i>	<i>id</i>	Unique value identifying a symbol item

## Return Values

Value	Status	Description
> 0	Success	Current symbol item identifier
0	Failure	Error occurred

## Remarks

*id* will remain the current symbol item until it is deleted or replaced.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting symbol items on a sheet.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set symbol = job.CreateSymbolObject()
```

## SetAsMaster - e3Symbol

```
symbolCount = job.GetSelectedSymbolIds( symbolIds )           'get selected symbols
If symbolCount > 0 Then
    For symbolIndex = 1 To symbolCount

        result = symbol.SetId( symbolIds( symbolIndex ) )
        If result = 0 Then
            message = "No symbol item is set"
        Else
            message = "Symbol " & result & " has been set"
        End If
        e3Application.PutInfo 0, message                       'output result of operation
    Next
End If

Set symbol = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3Symbol - Overview](#)
  - [GetGateId\(\)](#)
  - [GetId\(\)](#)
  - [SetGateId\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Symbol.SetLevel( *level* )**

## Syntax

**Integer** SetLevel( [in]**Integer** *level* )

## Description

Sets the symbol item's display level value.

## Parameters

Type	Parameter	Description
[in] <b>Integer</b>	<i>level</i>	Display level value to apply

## Return Values

Value	Status	Description
1..256	Success	Previous level value
0	Failure	Error occurred

## Remarks

*level* should be a value between 1 and 256. If the value of *level* is less than 1, the value of 1 is applied. If the value of *level* is greater than 256, the value of 256 is applied.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting symbols on a sheet.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set symbol = job.CreateSymbolObject()
```

## SetAsMaster - e3Symbol

```
Dim level : level = 5

symbolCount = job.GetSelectedSymbolIds( symbolIds )           'get currently selected symbols
If symbolCount > 0 Then
    For symbolIndex = 1 To symbolCount

        symbolId = symbol.SetId( symbolIds( symbolIndex ) )
        If symbolId > 0 Then

            result = symbol.SetLevel( level )
            If result = 0 Then
                message = "Symbol " & symbolId & ": Error setting level"
            Else
                message = "Symbol " & symbolId & ": Level set to " & level
            End If
            e3Application.PutInfo 0, message                    'output result of operation

        End If
    Next
End If

Set symbol = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3Symbol - Overview](#)
  - [GetLevel\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Symbol.SetName( *name* )**

## Syntax

**Integer** SetName( [in]**String** *name* )

## Description

Sets the symbol item's name.

## Parameters

Type	Parameter	Description
[in] <b>String</b>	<i>name</i>	Symbol name value to apply

## Return Values

Value	Status	Description
1	Success	Symbol name
0	Failure	Error occurred

## Remarks

A value of 0 is returned under one of the following criteria:

- No project is open
- The e3Symbol object has no symbol item set
- *name* value is "<Empty>"
- *name* value is over 12 characters in length
- *name* value is the same as the current symbol item name
- *name* value is already in use by another symbol item

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting symbols on a sheet.

## Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set symbol = job.CreateSymbolObject()

Dim symbolBaseName : symbolBaseName = "Sym"

symbolCount = job.GetSelectedSymbolIds( symbolIds )           'get currently selected symbols
If symbolCount > 0 Then
    For symbolIndex = 1 To symbolCount

        symbolId = symbol.SetId( symbolIds( symbolIndex ) )
        If symbolId > 0 Then

            symbolName = symbolBaseName & symbolId           'use a unique name
            result = symbol.SetName( symbolName )
            If result = 0 Then
                message = "Symbol " & symbolId & ": error setting symbol name"
            Else
                message = "Symbol " & symbolId & ": name set to " & symbolName
            End If
            e3Application.PutInfo 0, message                 'output result of operation

        End If
    Next
End If

Set symbol = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

## See Also

- [e3Symbol - Overview](#)
- [GetName\(\)](#)

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Symbol.OptionExpressions( *expressions* )**

## Syntax

*Integer* SetOptionExpressions( [in]*String Array expressions* )

## Description

Replaces the symbol item's option expressions.

## Parameters

Type	Parameter	Description
[in] <i>String Array</i>	<i>expressions</i>	Array of strings of all option names, boolean expressions or alias names of boolean expressions

## Return Values

Value	Status	Description
> 0	Success	Number of items in <i>expressions</i> assigned
0	Inconclusive	No assigned option expressions are supplied or an error occurred

## Remarks

Variant instances are currently available only for devices and wires.

*expressions* is a 0-based array.

All expressions already assigned to the symbol item are replaced by those in the *expressions* array.

The options expressions in the *expressions* array must be contained in the project.

Due caution is recommended on relying on the return value of 0 meaning no items in *expressions* were assigned since this also could mean an error has occurred. This could be the case if the intention is to remove all existing assigned option expressions by passing in an empty *expressions* array.



## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting symbols on a sheet.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set symbol = job.CreateSymbolObject()

Dim expressions(2)          'array with some option names
expressions(0) = "Option1"
expressions(1) = "Option2"

symbolCount = job.GetSelectedSymbolIds( symbolIds )          'get currently selected symbols
If symbolCount > 0 Then
    For symbolIndex = 1 To symbolCount

        symbolId = symbol.SetId( symbolIds( symbolIndex ) )
        If symbolId > 0 Then

            result = symbol.SetOptionExpressions( expressions )          'replace the field'
            If result = 0 Then
                e3Application.PutInfo 0, "Symbol " & symbolId & ": No assigned option expressions"
            Else
                expressionCount = symbol.GetAssignedOptionExpressionsEx( optionExpressions )
                If expressionCount > 0 Then
                    e3Application.PutInfo 0, "Symbol " & symbolId & ": assigned option expressions"
                    For expressionIndex = 1 To expressionCount
                        e3Application.PutInfo 0, "    " & optionExpressions( expressionIndex )
                    Next
                Else
                    e3Application.PutInfo 0, "Symbol " & symbolId & ": No assigned option expressions"
                End If
            End If
        End If
    Next
End If

Set symbol = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2010-9.12.

---

## See Also

- [e3Symbol - Overview](#)
  - [GetAssignedOptionExpressionsEx\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Symbol.SetScaling( *scale* )**

## Syntax

**Double** SetScaling( [in]**Double** *scale* )

## Description

Sets the symbol item's scaling factor value.

## Parameters

Type	Parameter	Description
[in] <b>Double</b>	scale	Scaling value to apply

## Return Values

Value	Status	Description
< 0.0	Success	Previous scaling factor value
0.0	Failure	Error occurred

## Remarks

Block symbols and dynamic symbols cannot be scaled using this function.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting symbols on a sheet.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set symbol = job.CreateSymbolObject()
```

## SetAsMaster - e3Symbol

```
Dim scale : scale = 2.5

symbolCount = job.GetSelectedSymbolIds( symbolIds )           'get currently selected symbols
If symbolCount > 0 Then
    For symbolIndex = 1 To symbolCount                         'loop through the selected symbols

        symbolId = symbol.SetId( symbolIds( symbolIndex ) )
        If symbolId > 0 Then

            result = symbol.SetScaling( scale )
            If result = 0.0 Then
                message = "Error setting scaling for symbol " & symbolId
            Else
                message = "Symbol " & symbolId & " scaling factor set from " & result & "
            End If
            e3Application.PutInfo 0, message                    'output result of operation

        End If
    Next
End If

Set symbol = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3Symbol - Overview](#)
  - [GetScaling\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Symbol.SetSelected( *newval* )**

## Syntax

**Integer** SetSelected( [in]**Boolean** *newval* )

## Description

Selects or deselects the symbol.

## Parameters

Type	Parameter	Description
		Indicates selection status of the symbol
[in] <b>Boolean</b>	<i>newval</i>	If <b>True</b> , the symbol will be selected
		If <b>False</b> , the symbol will be deselected

## Return Values

Value	Status	Description
1	Success	Symbol is selected or deselected
-1	Failure	No project open
-2	Failure	No symbol item set
-3	Failure	Symbol cannot be selected
-4	Failure	Selection exists on another sheet
-5	Failure	Symbol not displayed due to options/variants

## Remarks

The symbol is added to the selection if the current selection is on the same sheet otherwise the current selection is replaced.

# Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting sheets in the project tree containing symbols.

## Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set sheet = job.CreateSheetObject()
Set symbol = job.CreateSymbolObject()

sheetCount = job.GetTreeSelectedSheetIds( sheetIds )           'get all selected sheets in tr
If sheetCount > 0 Then
    For sheetIndex = 1 To sheetCount                           'loop through all selected sheets

        sheet.SetId( sheetIds( sheetIndex ) )
        embeddedsheetCount = sheet.GetEmbeddedSheetIds( embeddedSheetIds )
        If embeddedsheetCount > 0 Then
            sheet.SetId( sheetIds( 1 ) )                       'use the region sheet
        End If

        symbolCount = sheet.GetSymbolIds( symbolIds )           'get all symbols on the sheet
        If symbolCount > 0 Then

            sheetName = sheet.GetName()
            For symbolIndex = 1 To symbolCount                   'loop through all symbols

                symbolId = symbol.SetId( symbolIds( symbolIndex ) )
                If symbolId > 0 Then

                    symbolName = symbol.GetName()
                    result = symbol.SetSelected( True )         'select the symbol

                    Select Case result
                    Case 1
                        message = "Sheet: " & sheetName & " ; Symbol: " & symbolName & " s
                    Case -1
                        message = "No project open"
                    Case -2
                        message = "No symbol item set"
                    Case -3
                        message = sheetName & " : " & symbolName & " Symbol cannot be sele
                    Case -4
                        message = sheetName & " : " & symbolName & " Selection exists on a
                    Case -5
                        message = sheetName & " : " & symbolName & " Symbol not displayed
                    Case Else

```

## SetAsMaster - e3Symbol

```
message = sheetName & " : " & symbolName & " Unexpected return val
End Select
e3Application.PutInfo 0, message      'output result of operation

End If
Next
End If
Next
End If

Set symbol = Nothing
Set sheet = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2020-21.00.

---

## See Also

- [e3Symbol - Overview](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Symbol.SetSharedPinGroupState( *newval* )**

## Syntax

**Integer** SetSharedPinGroupState( [in]**Integer** *newval* )

## Description

Converts the symbol item between a normal symbol and a shared pin group.

## Parameters

Type	Parameter	Description
		Shared pin group value to apply
[in] <b>Integer</b>	<i>newval</i>	If 1, the symbol item is converted from a normal symbol to a shared pin group symbol
		If 0, the symbol item is converted from a shared pin group symbol to a normal symbol

## Return Values

Value	Status	Description
1	Success	Symbol item is converted to a shared pin group symbol
0	Inconclusive	Symbol item is converted to a normal symbol or an error occurred

## Remarks

Due caution is recommended on relying on the return value of 0 meaning the symbol is converted to a normal symbol since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting shared group pin symbols on a sheet.



## Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set symbol = job.CreateSymbolObject()

Dim convertToSharedPinGroup : convertToSharedPinGroup = 1

symbolCount = job.GetSelectedSymbolIds( symbolIds )           'get currently selected symbols
If symbolCount > 0 Then
    For symbolIndex = 1 To symbolCount                        'loop through the selected symbols

        symbolId = symbol.SetId( symbolIds( symbolIndex ) )
        If symbolId > 0 Then

            result = symbol.SetSharedPinGroupState( convertToSharedPinGroup )
            If result = 0 Then
                message = "Symbol " & symbolId & " set to a normal symbol"
            Else
                message = "Symbol " & symbolId & " set to a shared pin group"
            End If
            e3Application.PutInfo 0, message                    'output result of operation

        End If
    Next
End If

Set symbol = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2010-9.10.

## See Also

- [e3Symbol - Overview](#)
- [GetSharedPinGroupState\(\)](#)

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Symbol.SetStateId( *id* )**

## Syntax

**Integer** SetStateId( [in]**Integer** *id* )

## Description

Sets the symbols item's state.

## Parameters

Type	Parameter	Description
[in] <b>Integer</b>	<i>id</i>	Identifier of the state to apply

## Return Values

Value	Status	Description
1	Success	Symbol's state applied
-1	Failure	No project open or no symbol item set
-2	Failure	Invalid symbol type
-3	Failure	<i>id</i> is invalid
-4	Failure	Symbol is placed on a locked sheet
-5	Failure	Symbol is a view
-6	Failure	Symbol is not placed on a schematic sheet
-7	Failure	Placed symbol is in an inactive variant or option

## Remarks

Valid state identifiers can be retrieved using [GetStateIds\(\)](#).

The symbol's state can be retrieved using [GetStateId\(\)](#).

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting devices in the project tree.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set symbol = job.CreateSymbolObject()
Set state = job.CreateStateObject()

Const ACTIVE_STATE = 1

symbolCount = job.GetSelectedSymbolIds( symbolIds ) 'get currently selected symbols
If symbolCount > 0 Then
    For symbolIndex = 1 To symbolCount

        symbolId = symbol.SetId( symbolIds( symbolIndex ) )
        If symbolId > 0 Then
            symbolName = symbol.GetName()

            stateCount = symbol.GetStateIds( stateIds )
            If stateCount > 0 Then
                For stateIndex = 1 To stateCount
                    stateId = state.SetId( stateIds( stateIndex ) )
                    stateType = state.GetStateType()

                    If stateType = ACTIVE_STATE Then
                        result = symbol.SetStateId( stateId )
                        Select Case result
                            Case 1
                                message = "Symbol " & symbolName & " ( " & symbolId & " ) stat
                            Case -1
                                message = "Error setting symbol state: No project open or no s
                            Case -2
                                message = "Error setting symbol state: Symbol " & symbolName &
                            Case -3
                                message = "Error setting symbol state: Invalid state identifie
                            Case -4
                                message = "Error setting symbol state: Symbol " & symbolName &
                            Case -5
                                message = "Error setting symbol state: Symbol " & symbolName &
                            Case -6
                                message = "Error setting symbol state: Symbol " & symbolName &
                            Case -7
                                message = "Error setting symbol state: Symbol " & symbolName &
                        End Select
                    End If
                Next
            End If
        End If
    Next
End If

```

## SetAsMaster - e3Symbol

```
e3Application.PutInfo 0, message
```

```
'output result of operation
```

```
Exit For
End If
Next
End If
End If
Next
End If
```

```
Set state = Nothing
Set symbol = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2021-22.10.

---

## See Also

- [e3Symbol - Overview](#)
  - [e3State - Overview](#)
  - [GetStateId\(\)](#)
  - [GetStateIds\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Symbol.SetTableBreakTableAfter( *newval* )**

## Syntax

**Integer** SetTableBreakTableAfter( [in]**Boolean** *newval* )

## Description

Sets the table symbol item's break table after value.

## Parameters

Type	Parameter	Description
		Break table after value to apply
[in] <b>Boolean</b>	<i>newval</i>	If <b>True</b> , break table after value is activated
		If <b>False</b> , break table after value is deactivated

## Return Values

Value	Status	Description
0	Success	<i>newval</i> value is applied
-1	Failure	Error occurred

## Remarks

-1 will be returned if the current symbol item is not a table.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting table symbols on a sheet.

Visual Basic Script

## SetAsMaster - e3Symbol

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set symbol = job.CreateSymbolObject()

Dim useTableBreakAfter : useTableBreakAfter = 1

symbolCount = job.GetSelectedSymbolIds( symbolIds )      'get currently selected symbols
If symbolCount > 0 Then
    For symbolIndex = 1 To symbolCount

        symbolId = symbol.SetId( symbolIds( symbolIndex ) )
        If symbolId > 0 Then

            result = symbol.SetTableBreakTableAfter( useTableBreakAfter )
            If result = -1 Then
                message = "Error setting the break table after value for " & symbolId
            Else
                message = "Symbol " & symbolId & " break table after value set to " & useTableBreakAfter
            End If
            e3Application.PutInfo 0, message      'output result of operation

        End If
    Next
End If

Set symbol = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2018-19.00.

---

## See Also

- [e3Symbol - Overview](#)
  - [GetTableBreakTableAfter\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Symbol.SetTableBreakTableAfterNumberOfRows( *newval* )**

## Syntax

**Integer** SetTableBreakTableAfterNumberOfRows( [in]**Integer** *newval* )

## Description

Sets the number of rows displayed on the table symbol item before a table break occurs.

## Parameters

Type	Parameter	Description
[in] <b>Integer</b>	<i>newval</i>	Number of rows value to apply

## Return Values

Value	Status	Description
0	Success	New number of rows value was applied
-1	Failure	No symbol item set or the item is not a table symbol
-2	Failure	<i>newval</i> value lies outside the permitted range of 2 to 1000

## Remarks

Valid values for *newval* are between 2 and 1000.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>**.series project and selecting table symbols on a sheet.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
```

## SetAsMaster - e3Symbol

```
Set job = e3Application.CreateJobObject()
Set symbol = job.CreateSymbolObject()

Dim tableRows : tableRows = 12

symbolCount = job.GetSelectedSymbolIds( symbolIds )      'get currently selected symbols
If symbolCount > 0 Then
    For symbolIndex = 1 To symbolCount

        symbolId = symbol.SetId( symbolIds( symbolIndex ) )
        If symbolId > 0 Then

            result = symbol.SetTableBreakTableAfterNumberOfRows( tableRows )
            Select Case result
            Case 1
                message = "Symbol " & symbolId & " break table after " & tableRows & " rows"
            Case -1
                message = "No symbol or symbol " & symbolId & " not a table symbol"
            Case -2
                message = "Error table row value is invalid"
            End Select
            e3Application.PutInfo 0, message      'output result of operation

        End If
    Next
End If

Set symbol = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2018-19.00.

---

## See Also

- [e3Symbol - Overview](#)
  - [GetTableBreakTableAfterNumberOfRows\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).





v2022-23.00

**e3Symbol.SetTableOneRowForEachCore( *newval* )**

## Syntax

**Integer** SetTableOneRowForEachCore( [in]**Boolean** *newval* )

## Description

Sets the table symbol item's display one row for each core value.

## Parameters

Type	Parameter	Description
		Display one row for each core value to apply
[in] <b>Boolean</b>	<i>newval</i>	If <b>True</b> , display one row for each core is activated
		If <b>False</b> , display one row for each core is deactivated

## Return Values

Value	Status	Description
0	Success	Display one row for each core value was applied
-1	Failure	Error occurred

## Remarks

-1 will be returned if the current symbol item is not a table.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting table symbols on a sheet.

Visual Basic Script

## SetAsMaster - e3Symbol

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set symbol = job.CreateSymbolObject()

Dim useOneRowForEachCore : useOneRowForEachCore = 1

symbolCount = job.GetSelectedSymbolIds( symbolIds )      'get currently selected symbols
If symbolCount > 0 Then
    For symbolIndex = 1 To symbolCount

        symbolId = symbol.SetId( symbolIds( symbolIndex ) )
        If symbolId > 0 Then

            result = symbol.SetTableOneRowForEachCore( useOneRowForEachCore )
            If result = -1 Then
                message = "Error setting the one for each core value for " & symbolId
            Else
                message = "Symbol " & symbolId & " one for each core value set to " & useOneRowForEachCore
            End If
            e3Application.PutInfo 0, message                'output result of operation

        End If
    Next
End If

Set symbol = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2018-19.00.

---

## See Also

- [e3Symbol - Overview](#)
  - [GetTableOneRowForEachCore\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Symbol.SetTablePinsWithoutCores( *newval* )**

## Syntax

**Integer** SetTablePinsWithoutCores( [in]**Boolean** *newval* )

## Description

Sets the table symbol item's display pins without cores status value.

## Parameters

Type	Parameter	Description
		Display pins without cores status value to apply
[in] <b>Boolean</b>	<i>newval</i>	If <b>True</b> , display pins without cores is activated
		If <b>False</b> , display pins without cores is deactivated

## Return Values

Value	Status	Description
0	Success	Display pins without cores status value was applied
-1	Failure	Error occurred

## Remarks

-1 will be returned if the current symbol item is not a table.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting table symbols on a sheet.

Visual Basic Script

## SetAsMaster - e3Symbol

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set symbol = job.CreateSymbolObject()

Dim usePinsWithoutCores : usePinsWithoutCores = 1

symbolCount = job.GetSelectedSymbolIds( symbolIds )      'get currently selected symbols
If symbolCount > 0 Then
    For symbolIndex = 1 To symbolCount

        symbolId = symbol.SetId( symbolIds( symbolIndex ) )
        If symbolId > 0 Then

            result = symbol.SetTablePinsWithoutCores( usePinsWithoutCores )
            If result = -1 Then
                message = "Error setting the table pins without cores status value for " & symbolId
            Else
                message = "Symbol " & symbolId & " table pins without cores status value s
            End If
            e3Application.PutInfo 0, message      'output result of operation

        End If
    Next
End If

Set symbol = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2018-19.00.

---

## See Also

- [e3Symbol - Overview](#)
  - [GetTablePinsWithoutCores\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Symbol.SetType( *name*, *version* )**

## Syntax

**Integer** SetType( [in]**String** *name*, [in]**String** *version* )

## Description

Replaces the symbol item in the project for a symbol of the specified type name and version from the database.

## Parameters

Type	Parameter	Description
[in] <b>String</b>	name	Symbol type name from database
[in] <b>String</b>	version	Symbol type version from database

## Return Values

Value	Status	Description
> 0	Inconclusive	Identifier of the replacement symbol item
0	Failure	Error occurred

## Remarks

The symbol item's type name and version can only be exchanged with database symbols of the same symbol type.

If the identifier in the return value is the same as the original symbol item, the symbol item is not replaced.

If the identifier in the return value is different from the original symbol item, the symbol item has been replaced and the new symbol becomes the current item for the e3Symbol object.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting symbols on a sheet.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set symbol = job.CreateSymbolObject()

Dim newTypeName : newTypeName = "SOURCE_NODE_2"           'should be an existing symbol name
Dim newTypeVersion : newTypeVersion = "1"                 'should be an existing version of the symbol

symbolCount = job.GetSelectedSymbolIds( symbolIds )       'get currently selected symbols
If symbolCount > 0 Then
    For symbolIndex = 1 To symbolCount

        symbolId = symbol.SetId( symbolIds( symbolIndex ) )
        If symbolId > 0 Then

            result = symbol.SetType( newTypeName, newTypeVersion )
            If result = 0 Then
                message = "Error setting type for symbol " & symbolId
            Else
                If result = symbolId Then
                    message = "Symbol " & symbolId & " failed to be replaced"
                Else
                    message = "Symbol " & result & " of type " & newTypeName & ", version " & newTypeVersion
                End If
            End If
            e3Application.PutInfo 0, message                'output result of operation

        End If
    Next
End If

Set symbol = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3Symbol - Overview](#)
  - [GetType\(\)](#)
  - [GetVersion\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Symbol.UnGroup()**

## Syntax

Integer UnGroup()

## Description

Removes the symbol item from its group.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
1	Failure	Error occurred
0	Success	Symbol item ungrouped

## Remarks

0 is returned if the symbol item is not in a group.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting symbols on a sheet.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set symbol = job.CreateSymbolObject()
```



## SetAsMaster - e3Symbol

```
symbolCount = job.GetSelectedSymbolIds( symbolIds )           'get currently selected symbols
If symbolCount > 0 Then
    For symbolIndex = 1 To symbolCount

        symbolId = symbol.SetId( symbolIds( symbolIndex ) )
        If symbolId > 0 Then

            result = symbol.Ungroup()
            If result = 0 Then
                message = "Symbol " & symbolId & " removed from group"
            Else
                message = "Error removing symbol " & symbolId & " from group"
            End If
            e3Application.PutInfo 0, message                    'output result of operation

        End If
    Next
End If

Set symbol = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3Symbol - Overview](#)
  - [e3Group - Overview](#)
  - [GetGroupId\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

## e3TestPoint

# Description

Encapsulates the functionality for retrieving and modifying information for test point items.

## e3TestPoint Construction Functions

Function	Description
<u>e3Job.CreateTestPointObject()</u>	Creates an instance of e3TestPoint

## Item Creation/Destruction Functions

Function	Description
<u>Create()</u>	Creates a new test point item
<u>Delete()</u>	Deletes the current test point item from the project

## Retrieval Functions

Function	Description
<u>GetGID()</u>	Gets the global identifier of the current test point item
<u>GetId()</u>	Gets the identifier of the current test point item
<u>GetName()</u>	Gets the test point item's name

## Modification Functions

Function	Description
<u>SetGID()</u>	Sets a test point item as the current item
<u>SetId()</u>	Sets a test point item as the current item
<u>SetName()</u>	<b>Deprecated</b> Sets the test point item's name

## Attribute Functions

Function	Description
<u>AddAttributeValue()</u>	<b>Deprecated</b> Adds an attribute to the test point item
<u>DeleteAttribute()</u>	<b>Deprecated</b> Removes an attribute from the test point item
<u>DisplayAttributeValueAt()</u>	<b>Deprecated</b> Displays an attribute value from the test point item at a specific position
<u>GetAttributeCount()</u>	<b>Deprecated</b> Gets the number of the test point item's attributes
<u>GetAttributeIds()</u>	<b>Deprecated</b> Gets identifiers of the attributes assigned to the test point item
<u>GetAttributeValue()</u>	<b>Deprecated</b> Gets the test point item's specified attribute value
<u>HasAttribute()</u>	<b>Deprecated</b> Gets the number of the test point item's specified attributes
<u>SetAttributeValue()</u>	<b>Deprecated</b> Sets the test point item's specified attribute value
<u>SetAttributeVisibility()</u>	<b>Deprecated</b> Sets the visibility status of all text items representing the test point item's specified attribute value

## Process Functions

Function	Description
<u>Dump()</u>	Outputs information about the test point item to the Messages window
<u>Highlight()</u>	<b>Deprecated</b> Highlights the current test point item
<u>Search()</u>	Searches for a test point item matching the name

## Remarks

This interface is only relevant for use with **E<sup>3</sup>.logic**.

Test points are using to check the working state of a circuit and are applied to pins and signals.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting signals on a sheet or pins in the project tree.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set pin = job.CreatePinObject()
```

## SetAsMaster - e3Symbol

```
Set signal = job.CreateSignalObject()
Set testPoint = job.CreateTestpointObject()

Const DEVICE_PIN = 1
Const CONNECTOR_PIN = 2

pinCount = job.GetTreeSelectedPinIds( pinIds )           'get selected pins in the project tree
If pinCount > 0 Then
    For pinIndex = 1 To pinCount

        pinId = pin.SetId( pinIds( pinIndex ) )
        pinType = pin.GetTypeId()
        If pinType = DEVICE_PIN Or pinType = CONNECTOR_PIN Then

            pinName = pin.GetName()

            result = testPoint.Create( pinId )
            If result = 0 Then
                message = "Error creating test point item for pin " & pinName & " ( " & pinIndex & " ) "
            Else
                testPointName = testPoint.GetName()
                message = "Test point " & testPointName & " ( " & result & " ) created for pin " & pinName
            End If
            e3Application.PutInfo 0, message               'output result of operation

        End If
    Next
End If

signalCount = job.GetSelectedSignalIds( signalIds )      'get selected signal items
If signalCount > 0 Then
    For signalIndex = 1 To signalCount                    'loop through all signal items
        signalId = signal.SetId( signalIds( signalIndex ) )
        signalName = signal.GetName()

        result = testPoint.Create( signalId )
        If result = 0 Then
            message = "Error creating test point item for signal " & signalName & " ( " & signalIndex & " ) "
        Else
            testPointName = testPoint.GetName()
            message = "Test point " & testPointName & " ( " & result & " ) created for signal " & signalName
        End If
        e3Application.PutInfo 0, message                 'output result of operation

    Next
End If

Set testPoint = Nothing
Set signal = Nothing
```

```
Set pin = Nothing  
Set job = Nothing  
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [Classes - Overview](#)
  - [e3Attribute - Overview](#)
  - [e3Pin - Overview](#)
  - [e3Signal - Overview](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3TestPoint.AddAttributeValue( *name*, *value* )**

## Syntax

**Integer** AddAttributeValue( [in]**String** *name*, [in]**String** *value* )

## Description

Adds an attribute to the test point item.

Function has been deprecated.

## Parameters

Type	Parameter	Description
[in] <b>String</b>	<i>name</i>	Name of the new attribute
[in] <b>String</b>	<i>value</i>	Value of the new attribute

## Return Values

Value	Status	Description
> 0	Success	Identifier of the new attribute
0	Failure	Error occurred

## Remarks

This function is relevant for use with **E<sup>3</sup>.logic**.

*name* must be the name of an existing attribute.

*value* maximum length is 252 characters for attribute values in general. If the specific attribute value has a lower maximum length defined and *value* exceeds this, 0 is returned.

## Examples

The best results from the example can be achieved by:

- Ensuring the attribute exists or creating it with **E<sup>3</sup>.series** Database Editor if necessary
- Opening an **E<sup>3</sup>.series** project containing test points

Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set testPoint = job.CreateTestPointObject()

Dim attributeName : attributeName = "Example"           'attribute with this name should exist
Dim attributeValue : attributeValue = "Legatus nec violatur, nec laeditur"

testPointCount = job.GetTestPointIds( testPointIds )    'get test points
If testPointCount > 0 Then
    For testPointIndex = 1 To testPointCount

        testPointId = testPoint.SetId( testPointIds( testPointIndex ) )
        testPointName = testPoint.GetName()

        result = testPoint.AddAttributeValue( attributeName, attributeValue )
        If result = 0 Then
            message = "Error adding attribute value to test point item " & testPointName & "
        Else
            message = "Value " & attributeValue & " added to attribute " & result & " of test point " & testPointName
        End If
        e3Application.PutInfo 0, message                'output result of operation

    Next
End If

Set testPoint = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

Deprecated in v2009-8.52.

---

## See Also

- [e3TestPoint - Overview](#)
  - [DeleteAttribute\(\)](#)
  - [GetAttributeValue\(\)](#)
  - [HasAttribute\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).





v2022-23.00

**e3TestPoint.Create( *ani* )**

## Syntax

**Integer** Create( [in]**Integer** *ani* )

## Description

Creates a new test point item.

## Parameters

Type	Parameter	Description
[in] <b>Integer</b>	<i>ani</i>	Identifier of the target item

## Return Values

Value	Status	Description
> 0	Success	Identifier of the created test point item
0	Failure	Error occurred

## Remarks

This function is relevant for use with **E<sup>3</sup>.logic**.

*ani* should be an item identifier of one of the following item types:

- Connector pin
- Device pin
- Signal

Test point items can be deleted using Delete().

## Examples

## SetAsMaster - e3Symbol

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting signals on a sheet or pins in the project tree.

### Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set pin = job.CreatePinObject()
Set signal = job.CreateSignalObject()
Set testPoint = job.CreateTestpointObject()

Const DEVICE_PIN = 1
Const CONNECTOR_PIN = 2

pinCount = job.GetTreeSelectedPinIds( pinIds )      'get selected pins in the project tree
If pinCount > 0 Then
    For pinIndex = 1 To pinCount

        pinId = pin.SetId( pinIds( pinIndex ) )
        pinType = pin.GetTypeId()
        If pinType = DEVICE_PIN Or pinType = CONNECTOR_PIN Then

            pinName = pin.GetName()

            result = testPoint.Create( pinId )
            If result = 0 Then
                message = "Error creating test point item for pin " & pinName & " ( " & pinIndex & " ) "
            Else
                testPointName = testPoint.GetName()
                message = "Test point " & testPointName & " ( " & result & " ) created for pin " & pinName
            End If
            e3Application.PutInfo 0, message          'output result of operation

        End If
    Next
End If

signalCount = job.GetSelectedSignalIds( signalIds )      'get selected signal items
If signalCount > 0 Then
    For signalIndex = 1 To signalCount                  'loop through all signal items
        signalId = signal.SetId( signalIds( signalIndex ) )
        signalName = signal.GetName()

        result = testPoint.Create( signalId )
        If result = 0 Then
            message = "Error creating test point item for signal " & signalName & " ( " & signalIndex & " ) "
        Else
            testPointName = testPoint.GetName()
            message = "Test point " & testPointName & " ( " & result & " ) created for signal " & signalName
        End If
        e3Application.PutInfo 0, message
    Next
End If
```

## SetAsMaster - e3Symbol

```
End If
e3Application.PutInfo 0, message      'output result of operation

Next
End If

Set testPoint = Nothing
Set signal = Nothing
Set pin = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3TestPoint - Overview](#)
  - [Delete\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3TestPoint.Delete()**

## Syntax

**Integer** Delete()

## Description

Deletes the current test point item from the project.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
0	Inconclusive	Always returned

## Remarks

This function is relevant for use with **E<sup>3</sup>.logic**.

The current structure node item is set to 0.

The deletion of the test point item can be verified by assigning the interface object with the deleted test point item's identifier using SetId(). If the return value of SetId() is 0, the test point item does not exist and therefore can be assumed to have been deleted.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project containing test points.

Visual Basic Script

Examples

## SetAsMaster - e3Symbol

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set testPoint = job.CreateTestpointObject()

Dim testPointName : testPointName = "1"

foundTestPointId = testPoint.Search( testPointName )
If foundTestPointId > 0 Then

    result = testPoint.Delete()
    If result = 0 Then
        testPointId = testPoint.SetId( foundTestPointId )
        If testPointId > 0 Then
            message = "Error deleting test point " & testPointName & " ( " & foundTestPointId & " )"
        Else
            message = "Test point " & testPointName & " ( " & foundTestPointId & " ) deleted"
        End If
        e3Application.PutInfo 0, message          'output result of operation
    End If
End If

Set testPoint = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3TestPoint - Overview](#)
  - [Create\(\)](#)
  - [SetId\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3TestPoint.DeleteAttribute( *name* )**

## Syntax

**Integer** DeleteAttribute( [in]**String** *name* )

## Description

Removes an attribute from the test point item.

Function has been deprecated.

## Parameters

Type	Parameter	Description
[in] <b>String</b>	<i>name</i>	Name of the attribute

## Return Values

Value	Status	Description
> 0	Success	Attribute was removed
0	Failure	Error occurred

## Remarks

This function is relevant for use with **E<sup>3</sup>.logic**.

*name* must be the name of an existing attribute.

## Examples

The best results from the example can be achieved by:

- Ensuring the attribute exists or creating it with **E<sup>3</sup>.series** Database Editor if necessary
- Opening an **E<sup>3</sup>.series** project containing test points

## Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set testPoint = job.CreateTestPointObject()

Dim attributeName : attributeName = "Example"           'attribute with this name should exist

testPointCount = job.GetTestPointIds( testPointIds )    'get test points
If testPointCount > 0 Then
    For testPointIndex = 1 To testPointCount

        testPointId = testPoint.SetId( testPointIds( testPointIndex ) )
        testPointName = testPoint.GetName()

        result = testPoint.DeleteAttribute( attributeName )
        If result = 0 Then
            message = "Error deleting attribute " & attributeName & " from test point item " & testPointName
        Else
            message = "Attribute " & attributeName & " deleted from test point item " & testPointName
        End If
        e3Application.PutInfo 0, message                'output result of operation

    Next
End If

Set testPoint = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

Deprecated in v2009-8.52.

---

## See Also

- [e3TestPoint - Overview](#)
  - [AddAttributeValue\(\)](#)
  - [GetAttributeValue\(\)](#)
  - [HasAttribute\(\)](#)
-

## SetAsMaster - e3Symbol

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).





v2022-23.00

**e3TestPoint.DisplayAttributeValueAt( *name*, *sheetid*, *x*, *y* )**

## Syntax

**Integer** DisplayAttributeValueAt( [in]**String** *name*, [in]**Integer** *sheetid*, [in]**Double** *x*, [in]**Double** *y* )

## Description

Displays an attribute value from the test point item at a specific position.

Function has been deprecated.

## Parameters

Type	Parameter	Description
[in] <b>String</b>	<i>name</i>	Name of the attribute
[in] <b>Integer</b>	<i>sheetid</i>	Sheet identifier to display the attribute value on
[in] <b>Double</b>	<i>x</i>	Placement position on the x-axis
[in] <b>Double</b>	<i>y</i>	Placement position on the y-axis

## Return Values

Value	Status	Description
> 0	Success	Text item identifier displaying the attribute value
0	Failure	Error occurred

## Remarks

This function is relevant for use with **E<sup>3</sup>.logic**.

*name* must be the name of an existing attribute.

*x* and *y* should be in project measurement units.

If successful, the attribute will be displayed even if it is already displayed elsewhere.

Remarks

# Examples

The best results from the example can be achieved by:

- Ensuring the attribute exists or creating it with **E<sup>3</sup>.series** Database Editor if necessary
- Opening an **E<sup>3</sup>.series** project containing test points

## Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set sheet = job.CreateSheetObject()
Set testPoint = job.CreateTestPointObject()

POSITION_OFFSET = 25

Dim attributeName : attributeName = "Example1"           'attribute should exist
Dim xPosition : xPosition = 50
Dim yPosition : yPosition = 50

sheetId = job.GetActiveSheetId()
If sheetId > 0 Then
    sheet.SetId sheetId

    testPointCount = job.GetTestPointIds( testPointIds )      'get test points
    If testPointCount > 0 Then

        testPointId = testPoint.SetId( testPointIds( 1 ) )
        testPointName = testPoint.GetName()

        result = testPoint.DisplayAttributeValueAt( attributeName, sheetId, xPosition, yPosition )
        If result = 0 Then
            message = "Error displaying attribute value of test point item " & testPointName
        Else
            message = "Attribute " & attributeName & " of test point item " & testPointName
        End If
        e3Application.PutInfo 0, message          'output result of operation

    End If
End If

Set testPoint = Nothing
Set sheet = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

Deprecated in v2009-8.52.

---

## See Also

- [e3TestPoint - Overview](#)
  - [DisplayAttributeValueAt\(\)](#)
  - [GetAttributeValue\(\)](#)
  - [HasAttribute\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3TestPoint.Dump()**

## Syntax

*Integer* Dump()

## Description

Outputs information about the test point item to the Messages window.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
> 0	Inconclusive	Identifier of the current test point item is always returned

## Remarks

This function is relevant for use with **E<sup>3</sup>.logic**.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project containing test points.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set testPoint = job.CreateTestpointObject()

testPointCount = job.GetTestpointIds( testPointIds )           'get test points
```

## SetAsMaster - e3Symbol

```
If testPointCount > 0 Then
    e3Application.PutInfo 0, "Project has " & result & " test points:"
    For testPointIndex = 1 To testPointCount          'loop through all test point items

        testPointId = testPoint.SetId( testpointIds( testPointIndex ) )
        testPointName = testPoint.GetName()

        testPoint.Dump
        e3Application.PutInfo 0, "Information output for test point " & testPointName & "

    Next
End If

Set testPoint = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3TestPoint - Overview](#)
  - [GetGID\(\)](#)
  - [GetName\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3TestPoint.GetAttributeCount()**

## Syntax

**Integer** GetAttributeCount()

## Description

Gets the number of the test point item's attributes.

Function has been deprecated.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
> 0	Success	Number of attributes found
0	Inconclusive	No attribute found or an error occurred

## Remarks

This function is relevant for use with **E<sup>3</sup>.logic**.

Due caution is recommended on relying on the return value of 0 meaning no attributes found since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project.

Visual Basic Script

## SetAsMaster - e3Symbol

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set testPoint = job.CreateTestPointObject()

testPointCount = job.GetTestPointIds( testPointIds )      'get test points
If testPointCount > 0 Then
    For testPointIndex = 1 To testPointCount

        testPointId = testPoint.SetId( testPointIds( 1 ) )
        testPointName = testPoint.GetName()

        result = testPoint.GetAttributeCount()
        If result = 0 Then
            message = "No attributes found for test point " & testPointName & " ( " & testPointId & " ) "
        Else
            message = result & " attributes found for test point " & testPointName & " ( " & testPointId & " ) "
        End If
        e3Application.PutInfo 0, message                  'output result of operation

    Next
End If

Set testPoint = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

Deprecated in v2009-8.52.

---

## See Also

- [e3TestPoint - Overview](#)
  - [GetAttributeValue\(\)](#)
  - [HasAttribute\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3TestPoint.GetAttributeIds( *ids*, *attnam* )**

## Syntax

**Integer** GetAttributeIds( [out]**Integer Array** *ids*, [in][optional]**String** *attnam* )

## Description

Gets identifiers of the attributes assigned to the test point item.

Function has been deprecated.

## Parameters

Type	Parameter	Description
[out] <b>Integer Array</b>	<i>ids</i>	Array of identifiers of attributes
		Attribute name filter
		Only attribute identifiers with this attribute name will be supplied
[in][optional] <b>String</b>	<i>attnam</i>	All attribute names are supplied if <i>attnam</i> is an empty string
		The default string value is empty

## Return Values

Value	Status	Description
> 0	Success	Number of items in <i>ids</i>
0	Inconclusive	No assigned attribute identifiers are found or an error has occurred



## Remarks

*ids* is a 1-based array.

This function is relevant for use with **E<sup>3</sup>.logic**.

The attribute identifiers can be used by e3Attribute objects to handle them.

Due caution is recommended on relying on the return value of 0 meaning there are no items in *ids* since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project containing test points.

Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set testPoint = job.CreateTestPointObject()
Set attribute = job.CreateAttributeObject()

testPointCount = job.GetTestPointIds( testPointIds )           'get test points
If testPointCount > 0 Then
    For testPointIndex = 1 To testPointCount

        testPointId = testPoint.SetId( testPointIds( 1 ) )
        testPointName = testPoint.GetName()

        result = testPoint.GetAttributeIds( attributeIds )
        If result = 0 Then
            e3Application.PutInfo 0, "No attribute ids found for test point " & testPointName
        Else
            e3Application.PutInfo 0, result & " attribute ids found for test point " & testPointName
            For attributeIndex = 1 To result
                attributeId = attribute.SetId( attributeIds( attributeIndex ) )
                attributeName = attribute.GetName()
                e3Application.PutInfo 0, "    Attribute " & attributeName & " ( " & attributeIndex
            Next
        End If
    Next
End If

Next
End If

Set attribute = Nothing
Set testPoint = Nothing
Set job = Nothing

```

Set e3Application = Nothing

## Version Information

Introduced in v2009-8.50.

Deprecated in v2009-8.52.

---

## See Also

- [e3TestPoint - Overview](#)
  - [GetAttributeValue\(\)](#)
  - [SetAttributeValue\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3TestPoint.GetAttributeValue( *name* )**

## Syntax

*String* GetAttributeValue( [in]*String name* )

## Description

Gets the test point item's specified attribute value.

Function has been deprecated.

## Parameters

Type	Parameter	Description
[in] <i>String</i>	<i>name</i>	Name of the attribute

## Return Values

Value	Status	Description
"<Text>"	Success	Attribute value supplied
"<Empty>"	Inconclusive	Attribute value could not be found or an error occurred

## Remarks

This function is relevant for use with **E<sup>3</sup>.logic**.

*name* must be the name of an existing attribute.

Due caution is recommended on relying on the return value of "<Empty>" meaning there is no value set since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting symbols on a sheet.

## Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set testPoint = job.CreateTestPointObject()

Dim attributeName :    attributeName = "Example"           'attribute should exist

testPointCount = job.GetTestPointIds( testPointIds )      'get test points
If testPointCount > 0 Then
    For testPointIndex = 1 To testPointCount

        testPointId = testPoint.SetId( testPointIds( testPointIndex ) )
        testPointName = testPoint.GetName()

        result = testPoint.GetAttributeValue( attributeName )
        If Len( "" & result ) = 0 Then
            message = "Attribute " & attributeName & " value of test point " & testPointName
        Else
            message = "Attribute " & attributeName & " value of test point " & testPointName
        End If
        e3Application.PutInfo 0, message                  'output result of operation

    Next
End If

Set testPoint = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

Deprecated in v2009-8.52.

---

## See Also

- [e3TestPoint - Overview](#)
  - [HasAttribute\(\)](#)
  - [SetAttributeValue\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



**v2022-23.00**

**e3Testpoint.GetGID()**

## Syntax

*String* GetGID()

## Description

Gets the global identifier of the current test point item.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
"<GID>"	Success	Global identifier of the current test point item
"<Empty>"	Failure	No test point item

## Remarks

This function is relevant for use with **E<sup>3</sup>.logic**.

Global identifiers (GIDs) are used for item identification in multiuser projects.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting symbols on a sheet.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set testPoint = job.CreateTestpointObject()
```

```

testPointCount = job.GetTestpointIds( testPointIds )
If testPointCount > 0 Then
    For testPointIndex = 1 To testPointCount

        testPointId = testPoint.SetId( testPointIds( testPointIndex ) )
        result = testPoint.GetGID()
        If Len( "" & result ) = 0 Then
            message = "No test point item is set"
        Else
            testPointName = testPoint.GetName()
            message = "GID of test point item " & testPointName & " ( " & testPointId & "
        End If
        e3Application.PutInfo 0, message          'output result of operation

    Next
End If

Set testPoint = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2022-23.00.

---

## See Also

- [e3Testpoint - Overview](#)
  - [GetId\(\)](#)
  - [SetGID\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Testpoint.GetId()**

## Syntax

Integer GetId()

## Description

Gets the identifier of the current test point item.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
> 0	Success	Current test point item identifier
0	Failure	No test point item

## Remarks

This function is relevant for use with **E<sup>3</sup>.logic**.

The function returns the identifier value set by SetId() unless the item no longer exists.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
```

## SetAsMaster - e3Symbol

```
Set testPoint = job.CreateTestpointObject()

testPointCount = job.GetTestpointIds( testPointIds )
If testPointCount > 0 Then
    For testPointIndex = 1 To testPointCount

        testPoint.SetId( testPointIds( testPointIndex ) )
        result = testPoint.GetId()
        If result = 0 Then
            message = "No test point item is set"
        Else
            testPointName = testPoint.GetName()
            message = "Test point item " & testPointName & " ( " & result & " ) has been s
        End If
        e3Application.PutInfo 0, message          'output result of operation

    Next
End If

Set testPoint = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3TestPoint - Overview](#)
  - [SetId\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).





v2022-23.00

**e3TestPoint.GetName()**

## Syntax

*String* GetName()

## Description

Gets the test point item's name.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
"<Text>"	Success	Test point name
"<Empty>"	Failure	Error occurred

## Remarks

This function is relevant for use with **E<sup>3</sup>.logic**.

The test point name can be modified using [SetName\(\)](#).

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set testPoint = job.CreateTestpointObject()
```

## SetAsMaster - e3Symbol

```
testPointCount = job.GetTestpointIds( testPointIds )           'get test points
If testPointCount > 0 Then
    e3Application.PutInfo 0, "Project has " & result & " test points:"
    For testPointIndex = 1 To testPointCount                   'loop through all test point items

        testPointId = testPoint.SetId( testpointIds( testPointIndex ) )
        result = testPoint.GetName()
        If Len( "" & result ) = 0 Then
            message = "    Error getting name for test point " & testPointId
        Else
            message = "    Name of test point " & testPointId & " is " & result
        End If
        e3Application.PutInfo 0, message                       'output result of operation

    Next
End If

Set testPoint = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3TestPoint - Overview](#)
  - [SetName\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3TestPoint.HasAttribute( *name* )**

## Syntax

**Integer** HasAttribute( [in]**String** *name* )

## Description

Gets the number of the test point item's specified attributes.

Function has been deprecated.

## Parameters

Type	Parameter	Description
[in] <b>String</b>	<i>name</i>	Name of the attribute

## Return Values

Value	Status	Description
> 0	Success	Number of attributes found
0	Inconclusive	No attribute found or an error occurred

## Remarks

This function is relevant for use with **E<sup>3</sup>.logic**.

*name* should be the name of an existing attribute.

Due caution is recommended on relying on the return value of 0 meaning no attributes found since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project containing test points.

## Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set testPoint = job.CreateTestPointObject()

Dim attributeName :    attributeName = "Example"           'attribute should exist

testPointCount = job.GetTestPointIds( testPointIds )       'get test points
If testPointCount > 0 Then
    For testPointIndex = 1 To testPointCount

        testPointId = testPoint.SetId( testPointIds( testPointIndex ) )
        testPointName = testPoint.GetName()

        result = testPoint.HasAttribute( attributeName )
        If result = 0 Then
            message = "Attribute: " & attributeName & " of test point " & testPointName &
        Else
            message = "Attribute: " & attributeName & " of test point " & testPointName &
        End If
        e3Application.PutInfo 0, message                   'output result of operation

    Next
End If

Set testPoint = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

Deprecated in v2009-8.52.

---

## See Also

- [e3TestPoint- Overview](#)
  - [GetAttributeCount\(\)](#)
  - [GetAttributeValue\(\)](#)
  - [SetAttributeValue\(\)](#)
-

## SetAsMaster - e3Symbol

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

## e3TestPoint.Highlight()

### Syntax

**Integer** Highlight()

### Description

Highlights the current test point item.

Function has been deprecated.

### Parameters

No parameters defined.

### Return Values

Value	Status	Description
0	Inconclusive	Always returned

### Remarks

This function is relevant for use with **E<sup>3</sup>.logic**.

If the function is used, a warning message "CTestpointInterface::Highlight() not yet implemented" will appear in the Messages window.

### Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
```

## SetAsMaster - e3Symbol

```
Set testPoint = job.CreateTestpointObject()

testPointCount = job.GetTestpointIds( testPointIds )           'get test points
If testPointCount > 0 Then
    e3Application.PutInfo 0, "Project has " & result & " test points:"
    For testPointIndex = 1 To testPointCount                   'loop through all test point items

        testPointId = testPoint.SetId( testpointIds( testPointIndex ) )
        testPointName = testPoint.GetName()

        result = testPoint.Highlight()
        If result = 0 Then
            message = "Highlight() used for test point " & testPointName & " ( " & testPointId & " )"
        End If
        e3Application.PutInfo 0, message                       'output result of operation

    Next
End If

Set testPoint = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

Deprecated in v2009-8.52.

---

## See Also

- [e3TestPoint - Overview](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3TestPoint.Search( *name* )**

## Syntax

**Integer** Search( [in]**String** *name* )

## Description

Searches for a test point item matching the name.

## Parameters

Type	Parameter	Description
[in] <b>String</b>	name	Name of the test point

## Return Values

Value	Status	Description
> 0	Success	Found test point identifier
0	Inconclusive	No test point was found or an error has occurred

## Remarks

This function is relevant for use with **E<sup>3</sup>.logic**.

Due caution is recommended on relying on the return value of 0 meaning the test point was not found since this also could mean an error has occurred.

If successful, the test point will be set as the current item.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project containing test points.

Visual Basic Script

Examples



## SetAsMaster - e3Symbol

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set testPoint = job.CreateTestpointObject()

Dim testPointName : testPointName = "1"

result = testPoint.Search( testPointName )
If result = 0 Then
    message = "Test point " & testPointName & " could not be found"
Else
    message = "Test point " & testPointName & " ( " & result & " ) found"
End If
e3Application.PutInfo 0, message          'output result of operation

Set testPoint = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3TestPoint - Overview](#)
  - [GetName\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3TestPoint.SetAttributeValue( *name*, *value* )**

## Syntax

**Integer** SetAttributeValue( [in]**String** *name*, [in]**String** *value* )

## Description

Sets the test point item's specified attribute value.

Function has been deprecated.

## Parameters

Type	Parameter	Description
[in] <b>String</b>	<i>name</i>	Name of the attribute
[in] <b>String</b>	<i>value</i>	Value of the attribute

## Return Values

Value	Status	Description
> 0	Success	Identifier of attribute
0	Failure	Error occurred

## Remarks

This function is relevant for use with **E<sup>3</sup>.logic**.

*name* must be the name of an existing attribute.

*value* maximum length is 252 characters.

A valid symbol item identifier value must be assigned using SetId(), otherwise 0 is returned.

Adding a new attribute to the test point item cannot be done using this function.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project containing test points.

Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set testPoint = job.CreateTestPointObject()

Dim attributeName : attributeName = "Example"           'attribute should exist
Dim attributeValue : attributeValue = "Legatus nec violatur, nec laeditur"

testPointCount = job.GetTestPointIds( testPointIds )    'get test points
If testPointCount > 0 Then
    For testPointIndex = 1 To testPointCount

        testPointId = testPoint.SetId( testPointIds( testPointIndex ) )
        testPointName = testPoint.GetName()

        result = testPoint.SetAttributeValue( attributeName, attributeValue )
        If result = 0 Then
            message = "Attribute " & attributeName & " value not set for test point " & testPointName
        Else
            message = "Attribute " & attributeName & " value set for test point " & testPointName
        End If
        e3Application.PutInfo 0, message                'output result of operation

    Next
End If

Set testPoint = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

Deprecated in v2009-8.52.

---

## See Also

- [e3TestPoint - Overview](#)
- [GetAttributeValue\(\)](#)

- HasAttribute()
  - SetId()
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3TestPoint.SetAttributeVisibility( *name*, *onoff* )**

## Syntax

**Integer** SetAttributeVisibility( [in]**String** *name*, [in]**Integer** *onoff* )

## Description

Sets the visibility status of all text items representing the test point item's specified attribute value.

Function has been deprecated.

## Parameters

Type	Parameter	Description
[in] <b>String</b>	<i>name</i>	Name of the attribute
		Indicates whether the text items' visibility status should be shown or hidden
[in] <b>Integer</b>	<i>onoff</i>	If >0, the text items are shown
		If 0, the text items are hidden

## Return Values

Value	Status	Description
> 0	Success	Number of text items changed
0	Failure	No text items found or an error occurred

## Remarks

This function is relevant for use with **E<sup>3</sup>.logic**.

*name* must be the name of an existing attribute.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>**.series project containing test points.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set testPoint = job.CreateTestpointObject()

Const HIDE_ATTRIBUTE_TEXTS = 0

Dim attributeName :      attributeName = "Function"           'attribute should exist
Dim attributeTextVisibilityStatus :      attributeTextVisibilityStatus = HIDE_ATTRIBUTE_TEXTS

testPointCount = job.GetTestpointIds( testPointIds )          'get test points
If testPointCount > 0 Then
    For testPointIndex = 1 To testPointCount                  'loop through all test point items

        testPointId = testPoint.SetId( testpointIds( testPointIndex ) )
        testPointName = testPoint.GetName()

        result = testPoint.SetAttributeVisibility( attributeName, attributeTextVisibilityStatus )
        If result = 0 Then
            message = "No text items found for attribute " & attributeName & " of test point " & testPointName
        Else
            message = result & " text items found for attribute " & attributeName & " of test point " & testPointName
        End If
        e3Application.PutInfo 0, message                      'output result of operation

    Next
End If

Set testPoint = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

Deprecated in v2009-8.52.

---

## See Also

- [e3TestPoint - Overview](#)

- GetAttributeValue()
  - HasAttribute()
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Testpoint.SetGID( *gid* )**

## Syntax

*String* SetGID( [in]*String* *gid* )

## Description

Sets a test point item as the current item.

## Parameters

Type	Parameter	Description
[in] <i>String</i>	<i>gid</i>	Global identifier value of a test point item

## Return Values

Value	Status	Description
"<GID>"	Success	Global identifier of the current test point item
"<Empty>"	Failure	No test point item

## Remarks

This function is relevant for use with **E<sup>3</sup>.logic**.

Global identifiers (GIDs) are used for item identification in multiuser projects.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set testPoint = job.CreateTestpointObject()
```



## SetAsMaster - e3Symbol

```
Set gidList = CreateObject( "System.Collections.ArrayList" )

testPointCount = job.GetTestpointIds( testPointIds )
If testPointCount > 0 Then
    For testPointIndex = 1 To testPointCount

        testPoint.SetId testPointIds( testPointIndex )
        gidId = testPoint.GetGID()
        gidList.Add gidId

    Next
End If

For Each gidId in gidList
    result = testPoint.SetGID( gidId )
    If Len( "" & result ) = 0 Then
        message = "No test point item is set"
    Else
        testPointId = testPoint.GetId()
        testPointName = testPoint.GetName()
        message = "Test point " & testPointName & " ( " & testPointId & " ) has been set u
    End If
    e3Application.PutInfo 0, message          'output result of operation
Next

Set gidList = Nothing
Set testPoint = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2022-23.00.

---

## See Also

- [e3Testpoint - Overview](#)
  - [GetGID\(\)](#)
  - [SetId\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Testpoint.SetId( *id* )**

## Syntax

*Integer* SetId( [in]*Integer id* )

## Description

Sets a test point item as the current item.

## Parameters

Type	Parameter	Description
[in] <i>Integer</i>	<i>id</i>	Unique value identifying a test point item

## Return Values

Value	Status	Description
> 0	Success	Current test point item identifier
0	Failure	Error occurred

## Remarks

This function is relevant for use with **E<sup>3</sup>.logic**.

*id* will remain the current test point item until it is deleted or replaced.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set testPoint = job.CreateTestpointObject()
```

```

testPointCount = job.GetTestpointIds( testPointIds )
If testPointCount > 0 Then
    For testPointIndex = 1 To testPointCount

        result = testPoint.SetId( testPointIds( testPointIndex ) )
        If result = 0 Then
            message = "No test point item is set"
        Else
            testPointName = testPoint.GetName()
            message = "Test point item " & testPointName & " ( " & result & " ) has been s
        End If
        e3Application.PutInfo 0, message          'output result of operation

    Next
End If

Set testPoint = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3Testpoint - Overview](#)
  - [GetId\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3TestPoint.SetName( *name* )**

## Syntax

**Integer** SetName( [in]**String** *name* )

## Description

Sets the test point item's name.

Function has been deprecated.

## Parameters

Type	Parameter	Description
[in] <b>String</b>	name	New name value

## Return Values

Value	Status	Description
0	Failure	Error occurred

## Remarks

If the function is used, a warning message "CTestpointInterface::SetName(n) not yet implemented" will appear in the Messages window.

GetName() can be used to retrieve the point item's name value.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project containing test point items.

Visual Basic Script

## SetAsMaster - e3Symbol

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set testPoint = job.CreateTestpointObject()

testPointCount = job.GetTestpointIds( testPointIds )      'get test points
If testPointCount > 0 Then

    testPointId = testPoint.SetId( testpointIds( 1 ) )
    testPointName = testPoint.GetName()

    result = testPoint.SetName( 100 )
    If result = 0 Then
        e3Application.PutInfo 0, "SetName() is not supported. Name of test point " & testPointName
    End If

Next
End If

Set testPoint = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

Deprecated in v2009-8.52.

---

## See Also

- [e3TestPoint - Overview](#)
  - [GetName\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

## e3Text

# Description

Encapsulates the functionality for retrieving and modifying information for text items.

## e3Text Construction Functions

Function	Description
<u><a href="#">e3Job.CreateTextObject()</a></u>	Creates an instance of e3Text

## Item Creation/Destruction Functions

Function	Description
<u><a href="#">e3Graph.CreateText()</a></u>	e3Text does not have its own function for creating text items
<u><a href="#">Delete()</a></u>	Deletes the text from the project

## Retrieval Functions

Function	Description
<u><a href="#">GetAlignment()</a></u>	Gets the text's current alignment value
<u><a href="#">GetAllowedLength()</a></u>	Gets the maximum number of characters allowed for the text
<u><a href="#">GetBallooning()</a></u>	Gets the text's ballooning value
<u><a href="#">GetBox()</a></u>	Gets the text item's box width and height
<u><a href="#">GetColour()</a></u>	Gets the text item's text color
<u><a href="#">GetFontName()</a></u>	Gets the text item's font name
<u><a href="#">GetGID()</a></u>	Gets the global identifier of the current text item
<u><a href="#">GetHeight()</a></u>	Gets the text item's font height
<u><a href="#">GetHyperlinkAddress()</a></u>	Gets the text item's hyperlink address
<u><a href="#">GetId()</a></u>	Gets the identifier of the current text item
<u><a href="#">GetInternalText()</a></u>	Gets the text item's text value without text token translations

<u>GetLanguageId()</u>	Gets the text item's translation identifier value
<u>GetLeftJustifiedSchemaLocation()</u>	Gets the text item's box's lower right corner position in text reading direction
<u>GetLevel()</u>	Gets the text item's display level value
<u>GetLinearMeasureWithoutUnit()</u>	Gets the text item's flag value determining if the length measurement unit is displayed along with the value
<u>GetLocking()</u>	Gets the text item's flag value determining if the text position is locked
<u>GetMode()</u>	Gets the text item's text ratio
<u>GetPictogram()</u>	Gets the text item's flag value determining if the text is displayed in the pictogram language
<u>GetRightJustifiedSchemaLocation()</u>	Gets the text item's box's lower right corner position in text reading direction
<u>GetRotation()</u>	Gets the text item's rotation value
<u>GetSchemaLocation()</u>	Gets the text item's position within the project
<u>GetSingleLine()</u>	Gets the text item's flag value determining if multi-line text is displayed on a single line
<u>GetStyle()</u>	Gets the text item's font style
<u>GetText()</u>	Gets the text item's text value including text token translations
<u>GetTextExtent()</u>	Gets the text item's dimension as a geometric shape
<u>GetTextExtentSingleLine()</u>	Gets the text item's dimension as a geometric shape for each line
<u>GetType()</u>	Gets the text item's text type
<u>GetTypeId()</u>	Gets the text item's text type
<u>GetVisibility()</u>	Gets the text item's visibility status
<u>GetWidth()</u>	Gets the text item's box width
<u>IsOffline()</u>	<b>Deprecated</b> Gets the text's accessibility status by an <b>E<sup>3</sup>.multiuser</b> client
<u>IsRedlined()</u>	Gets the text's redlined status

## Modification Functions

Function	Description
<u>DeleteBox()</u>	Deletes the text item's text box
<u>SendToBackground()</u>	Moves the placed symbol item to the sheet background
<u>SendToForeground()</u>	Moves the placed symbol item to the sheet foreground

<u>SetAlignment()</u>	Sets the text's alignment value
<u>SetBallooning()</u>	Sets the text's ballooning value
<u>SetBox()</u>	Sets the text item's box width and height
<u>SetColour()</u>	Sets the text item's text color
<u>SetFontName()</u>	Sets the text item's font
<u>SetGID()</u>	Sets a text item as the current item
<u>SetHeight()</u>	Sets the text item's font height
<u>SetHyperlinkAddress()</u>	Sets the text item's hyperlink address
<u>SetId()</u>	Sets a text as the current item
<u>SetLanguageId()</u>	Sets the text item's translation identifier value
<u>SetLevel()</u>	Sets the text item's display level value
<u>SetLinearMeasurementWithoutUnit()</u>	Sets the text item's flag value determining if the length measurement unit is displayed along with the value
<u>SetLocking()</u>	Sets the text item's flag value determining if the text position is locked
<u>SetMode()</u>	Sets the text item's text ratio
<u>SetPictogram()</u>	Sets the text item's flag value determining if the text is displayed in the pictogram language
<u>SetRedlined()</u>	Sets the text's redlined status
<u>SetRotation()</u>	Sets the text item's rotation value
<u>SetSchemaLocation()</u>	Sets the text item's position
<u>SetSingleLine()</u>	Sets the text item's flag value determining if multi-line text is displayed on a single line
<u>SetStyle()</u>	Sets the text item's font style
<u>SetText()</u>	Sets the text item's text value
<u>SetVisibility()</u>	Sets the text item's visibility status

## Attribute Functions

Function	Description
<u>AddAttributeValue()</u>	Adds an attribute to the text item
<u>DeleteAttribute()</u>	Removes an attribute from the text item
<u>GetAttributeCount()</u>	Gets the number of the text item's attributes
<u>GetAttributeIds()</u>	Gets identifiers of the attributes assigned to the text item
<u>GetAttributeValue()</u>	Gets the text item's specified attribute value
<u>HasAttribute()</u>	Gets the number of the text item's specified attributes
<u>SetAttributeValue()</u>	Sets the text item's specified attribute value



## Calculation Functions

Function	Description
<u>CalculateBoxAt()</u>	Calculates the placement position of a new text box
<u>CalculateBoxHeight()</u>	<b>Deprecated</b> Calculates the text box rectangle required to hold the given text
<u>CalculateBoxHeightEx()</u>	Calculates the text box rectangle required to hold the given text

## Variant/Option Functions

Function	Description
<u>GetAssignedOptionExpressions()</u>	Gets option names and boolean expressions (combinations of options) assigned to the text item
<u>GetAssignedOptionExpressionsEx()</u>	Gets option names and boolean expressions (combinations of options) assigned to the text item
<u>GetAssignedOptionIds()</u>	Gets identifiers of options assigned to the text item
<u>SetOptionExpressions()</u>	Replaces the text item's option expressions

## Remarks

Text items always belong to symbols or sheets and also show specific information about their owner item. Changing a text item's values may change the properties of the owner item.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting texts on a sheet. The example will make a copy of each text adjacent to the original text if possible.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()

textCount = job.GetSelectedTextIds( textIds )           'get selected texts

If textCount > 0 Then
    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )
    
```

## SetAsMaster - e3Symbol

```

sheetId = text.GetSchemaLocation( x, y, grid )      'get sheet and position of the c

xPosition = x + text.GetWidth() + 1
result = CopyText( textId, sheetId, xPosition, y )   'copy the text beside the or

Select Case result
Case -1
    message = "Failed to copy text: id - " & textId & " unknown"
Case -2
    message = "Failed to copy text " & textId & ": sheet " & sheetId & " unknown"
Case -3
    message = "Failed to copy text " & textId & " to sheet " & sheetId & " at " &
Case Else
    message = "Text " & textId & " copied to text " & result & " on sheet " & sheetId
End Select
e3Application.PutInfo 0, message                    'output result of operation

Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing

Function CopyText( ByVal sourceId, ByVal sheetId, ByVal xPosition, ByVal yPosition )

    returnValue = 0

    Dim graph : Set graph = job.CreateGraphObject()
    Dim sourceText : Set sourceText = job.CreateTextObject()
    Dim targetText : Set targetText = job.CreateTextObject()
    Dim targetSheet : Set targetSheet = job.CreateSheetObject()

    If sourceText.SetId( sourceId ) = 0 Then
        returnValue = -1      'sourceId not valid
    Else
        If targetSheet.SetId( sheetId ) = 0 Then
            returnValue = -2      'sheetId not valid
        Else
            targetTextId = graph.CreateText( sheetId, sourceText.GetInternalText, xPosition, yPosition )
            If targetText.SetId( targetTextId ) = 0 Then
                returnValue = -3      'unable to create a text item on the sheet at the position
            Else
                sourceText.GetBox width, height
                targetText.SetBox width, height

                ChangeFont targetTextId, sourceText.GetFontName, sourceText.GetHeight, sourceText.GetColor
                targetText.SetAlignment sourceText.GetAlignment()
            End If
        End If
    End If
End Function

```

## SetAsMaster - e3Symbol

```

        targetText.SetBallooning True, sourceText.GetBallooning()
        targetText.SetHyperlinkAddress sourceText.GetHyperlinkAddress()
        targetText.SetLevel sourceText.GetLevel()
        targetText.SetPictogram sourceText.GetPictogram()
        targetText.SetRotation sourceText.GetRotation()
        targetText.SetSingleLine sourceText.GetSingleLine()
        targetText.SetVisibility sourceText.GetVisibility()

        returnValue = targetTextId

    End If
End If

Set targetSheet = Nothing
Set targetText = Nothing
Set sourceText = Nothing
Set graph = Nothing

CopyText = returnValue

End Function

Function ChangeFont( ByVal textId, ByVal fontName, ByVal fontHeight, ByVal fontStyle, ByVal fontColour )

    returnValue = 1

    Const AUTO_COLOUR = -1
    Const MAX_COLOUR = 255

    Dim targetText : Set targetText = job.CreateTextObject()

    If targetText.SetId( textId ) = 0 Then
        returnValue = 0
    Else
        If textColour < AUTO_COLOUR Or textColour > MAX_COLOUR Then 'check for invalid colour
            returnValue = 0
        Else
            oldFontName = targetText.GetFontName()
            oldFontHeight = targetText.GetHeight()
            oldFontStyle = targetText.GetStyle()
            oldTextMode = targetText.GetMode()

            'if a font function call fails...
            If ( targetText.SetFontName( fontName ) = 0 Or targetText.SetHeight( fontHeight ) = 0 ) Then
                '...reset old values
                targetText.SetFontName oldFontName
                targetText.SetHeight oldFontHeight
                targetText.SetStyle oldFontStyle
            End If
        End If
    End If
End Function

```

## SetAsMaster - e3Symbol

```
targetText.SetMode oldTextMode

returnValue = 0          'font changed failed
Else
    targetText.SetColour textColour      'SetColour return value cannot be e
End If
End If
End If

Set targetText = Nothing
ChangeFont = returnValue

End Function
```

## Version Information

Introduced in v2010-9.10.

---

## See Also

- [Classes - Overview](#)
  - [e3Attribute - Overview](#)
  - [e3Graph - Overview](#)
  - [e3Sheet - Overview](#)
  - [e3Symbol - Overview](#)
  - [e3Variant - Overview](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.AddAttributeValue( *name*, *value* )**

## Syntax

**Integer** AddAttributeValue( [in]**String** *name*, [in]**String** *value* )

## Description

Adds an attribute to the text item.

## Parameters

Type	Parameter	Description
[in] <b>String</b>	<i>name</i>	Name of the new attribute
[in] <b>String</b>	<i>value</i>	Value of the new attribute

## Return Values

Value	Status	Description
> 0	Success	Identifier of the new attribute
0	Failure	Error occurred
-1	Failure	<i>value</i> string too long
		Available since v2019-20.00

## Remarks

*name* must be the name of an existing attribute.

*value* maximum length is 252 characters.

## Examples

The best results from the example can be achieved by ensuring the attribute exists or creating it with **E<sup>3</sup>**.series in DBE mode if necessary, opening an **E<sup>3</sup>**.series project and

selecting texts.

## Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()

Dim attributeName : attributeName = "Example"           'attribute with this name should exist
Dim attributeValue : attributeValue = "Legatus nec violatur, nec laeditur"

textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then
    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )
        result = text.AddAttributeValue( attributeName, attributeValue )

        If result = 0 Then
            message = "Text: " & textId & " error occurred"
        Else
            message = "Text: " & textId & "; Attribute " & result & ": " & attributeName & " "
        End If
        e3Application.PutInfo 0, message                 'output result of operation

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2012-11.00.

Modified in v2019-20.00.

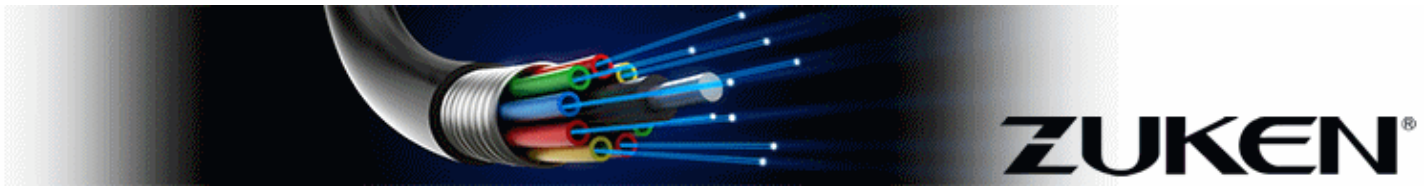
---

## See Also

- [e3Text - Overview](#)
  - [DeleteAttribute\(\)](#)
  - [GetAttributeValue\(\)](#)
  - [HasAttribute\(\)](#)
-

## SetAsMaster - e3Symbol

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.CalculateBoxAt( *shti*, *text*, *x*, *y*, *rotation*, *height*, *mode*, *style*, *fontname*, *just*, *balloon*, *bx*, *by* )**

## Syntax

**Integer** CalculateBoxAt( [in]**Integer** *shti*, [in]**String** *text*, [in]**Double** *x*, [in]**Double** *y*, [in]**Double** *rotation*, [in]**Double** *height*, [in]**Integer** *mode*, [in]**Integer** *style*, [in]**String** *fontname*, [in]**Integer** *just*, [in]**Integer** *balloon*, [out]**Double Array** *bx*, [out] **Double Array** *by* )

## Description

Calculates the placement position of a new text box.

## Parameters

Type	Parameter	Description
[in] <b>Integer</b>	<i>shti</i>	Currently not used
[in] <b>String</b>	<i>text</i>	Text of the text field
[in] <b>Double</b>	<i>x</i>	Placement position on the x-axis
[in] <b>Double</b>	<i>y</i>	Placement position on the y-axis
[in] <b>Double</b>	<i>rotation</i>	Rotation of the text field
[in] <b>Double</b>	<i>height</i>	Size of the text field font
[in] <b>Integer</b>	<i>mode</i>	Calculated height of the box
[in] <b>Integer</b>	<i>style</i>	Calculated width of the box
		Name of the text field font
[in] <b>String</b>	<i>fontname</i>	"Arial" is set as default if value is empty
[in] <b>Integer</b>	<i>just</i>	Alignment of the text field
		Balloon style of the text field
[in] <b>Integer</b>	<i>balloon</i>	See <u>Ballooning</u> for possible values
	<i>bx</i>	



[out]Double  
Array

An array of 4 elements representing the corner positions on the x-axis

[out]Double  
Array by

An array of 4 elements representing the corner positions on the y-axis

## Return Values

Value	Status	Description
1	Success	Calculation was successful
0	Failure	Error occurred

## Remarks

Localized variables contained in *text* are taken from the database and replace the values when they are found. There is no warning or error message if a text variable is not found.

*rotation* value rotates anticlockwise.

*mode* can be one of the following values:

Value	Description
1	Normal
2	Narrow
3	Wide

*style* can be a combination of the following values:

Value	Description
0	Regular
1	<b>Bold</b>
2	<i>Italics</i>
4	<u>Underline</u>
8	Strikethrough
16	Opaque

*just* can be one of the following values:

Value	Description
1	Left Align
2	Center
3	Right Align

*bx* and *by* are 1-based arrays.

## Examples

The best results from the example can be achieved by opening or creating an **E<sup>3</sup>.series** project.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()

sheetId = 0          'not used
textString = "Example text"
xPlacement = 50.0
yPlacement = 50.0
rotation = 0.0       'no rotation
fontHeight = 5.0
mode = 1             'normal
style = 0            'normal
fontName = "Verdana"
alignment = 1        'left
ballooning = 0       'No ballooning

result = text.CalculateBoxAt( sheetId, textString, xPlacement, yPlacement, rotation, fontHeight )
If result = 0 Then
    e3Application.PutInfo 0, "Text: No calculation was performed"
Else
    e3Application.PutInfo 0, "Text Box Calculation coordinates:"
    For index = 1 To 4
        e3Application.PutInfo 0, "    PosX: " & Xpositions( index ) & "; PosY: " & Ypositions( index )
    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2010-9.10.

---

## See Also

- [e3Text - Overview](#)
- [e3Sheet.GetWorkingArea\(\)](#)
- [Ballooning](#)

- CalculateBoxHeightEx()
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.CalculateBoxHeight( *width, text, fontName, fontSize, fontStyle, wordBreak, recHeight, recWidth, lines* )**

## Syntax

*Integer* CalculateBoxHeight( [in]*Double* width, [in]*String* text, [in]*String* fontName, [in]*Double* fontSize, [in]*Integer* fontStyle, [in]*Boolean* wordBreak, [out]*Double* recHeight, [out]*Double* recWidth, [out]*Integer* lines )

## Description

Calculates the text box rectangle required to hold the given text.

**Function has been deprecated.** Please use CalculateBoxHeightEx() instead.

## Parameters

No parameters defined.

Type	Parameter	Description
[in] <i>Double</i>	width	Width of the text field
[in] <i>String</i>	text	Text of the text field
[in] <i>String</i>	fontName	Name of the text field font
[in] <i>Double</i>	fontSize	Size of the text field font
[in] <i>Integer</i>	fontStyle	Style of the text field font
		Texts are broken within words if <i>wordBreak</i> is <i>True</i> .
[in] <i>Boolean</i>	wordBreak	Texts are always broken after a blank, a tab or one of the following characters: .,?!;-
[out] <i>Double</i>	recHeight	Calculated height of the box
[out] <i>Double</i>	recWidth	Calculated width of the box
[out] <i>Integer</i>	lines	Number of lines of text in the box

## Return Values

Value	Status	Description
3	Inconclusive	No calculation was performed
2	Inconclusive	<i>wordBreak</i> value is <b>False</b> and <i>width</i> is too small for the longest word
1	Inconclusive	Calculation was performed but an alternative default value was used For example in the case <i>fontName</i> value was not found
0	Success	Calculation was successful
-1	Failure	Error occurred due to invalid <i>width</i>
-2	Failure	Internal error

## Remarks

The calculation doesn't take into account if the text is rotated or within ballooning.

Localized variables contained in *text* are taken from the database and replace the values when they are found. There is no warning or error message if a text variable is not found.

*fontStyle* can be a combination of the following values:

Value	Description
0	Regular
1	<b>Bold</b>
2	<i>Italics</i>
4	<u>Underline</u>
8	Strikethrough
16	Opaque

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>**.series project and selecting text items on a sheet.

Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()

textCount = job.GetSelectedTextIds( textIds )      'get selected texts
If textCount > 0 Then

```

```

result = 0
maxWidth = 30.0

For textIndex = 1 To textCount

    textId = text.SetId( textIds( textIndex ) )
    text.GetBox width, height

    If width > maxWidth Then 'increase height and reduce width if text box is w

        result = text.CalculateBoxHeight( maxWidth, text.GetText, text.GetFontName, te
        Select Case result
        Case 3
            message = "Text: " & textId & "No calculation was performed"
        Case 2
            message = "Text: " & textId & "wordBreak value is False and width is too s
        Case 1
            message = "Text: " & textId & "Calculation was performed but an alternativ
        Case -1
            message = "Text: " & textId & "Error occurred due to invalid width"
        Case -2
            message = "Text: " & textId & "Internal error"
        Case Else
            text.SetBox newWidth, newHeight
            message = "Text: " & textId & ": New width = " & newWidth & "; New height
        End Select
        e3Application.PutInfo 0, message 'output result of operation

    End If
Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2016-17.31 and v2017-1810.

Deprecated in v2018-19.00.

---

## See Also

- [e3Text - Overview](#)
  - [CalculateBoxHeightEx\(\)](#)
-

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.CalculateBoxHeightEx( *width, text, fontName, fontSize, fontMode, fontStyle, recHeight, recWidth, lines* )**

## Syntax

**Integer** CalculateBoxHeightEx( [in]**Double** *width*, [in]**String** *text*, [in]**String** *fontName*, [in]**Double** *fontSize*, [in]**Integer** *fontMode*, [in]**Integer** *fontStyle*, [out]**Double** *recHeight*, [out]**Double** *recWidth*, [out]**Integer** *lines* )

## Description

Calculates the text box rectangle required to hold the given text.

## Parameters

No parameters defined.

Type	Parameter	Description
[in] <b>Double</b>	<i>width</i>	Width of the text field
[in] <b>String</b>	<i>text</i>	Text of the text field
[in] <b>String</b>	<i>fontName</i>	Name of the text field font
[in] <b>Double</b>	<i>fontSize</i>	Size of the text field font
[in] <b>Integer</b>	<i>fontMode</i>	Ratio of the text field font
[in] <b>Integer</b>	<i>fontStyle</i>	Style of the text field font
[out] <b>Double</b>	<i>recHeight</i>	Calculated height of the box
[out] <b>Double</b>	<i>recWidth</i>	Calculated width of the box
[out] <b>Integer</b>	<i>lines</i>	Number of lines of text in the box

## Return Values

Value	Status	Description
3	Inconclusive	No calculation was performed
1	Inconclusive	



Calculation was performed but an alternative default value was used

For example in the case *fontName* value was not found

0	Success	Calculation was successful
-1	Failure	Error occurred due to invalid <i>width</i>
-2	Failure	Internal error

## Remarks

The calculation doesn't take into account if the text is rotated or within ballooning.

Localized variables contained in *text* are taken from the database and replace the values when they are found. There is no warning or error message if a text variable is not found.

*fontMode* can be one of the following values:

Value Description

1	Normal
2	Narrow
3	Wide

*fontStyle* can be a combination of the following values:

Value Description

0	Regular
1	<b>Bold</b>
2	<i>Italics</i>
4	<u>Underline</u>
8	Strikethrough
16	Opaque

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>**.series project and selecting text items on a sheet.

Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()

textCount = job.GetSelectedTextIds( textIds )      'get selected texts
If textCount > 0 Then

```

```

result = 0
maxWidth = 30.0

For textIndex = 1 To textCount

    textId = text.SetId( textIds( textIndex ) )
    text.GetBox width, height

    If width > maxWidth Then 'increase height and reduce width if text box is w

        result = text.CalculateBoxHeightEx( maxWidth, text.GetText, text.GetFontName,
        Select Case result
        Case 3
            message = "Text: " & textId & "No calculation was performed"
        Case 1
            message = "Text: " & textId & "Calculation was performed but an alternativ
        Case -1
            message = "Text: " & textId & "Error occurred due to invalid width"
        Case -2
            message = "Text: " & textId & "Internal error"
        Case Else
            text.SetBox newWidth, newHeight
            message = "Text: " & textId & ": New width = " & newWidth & "; New height
        End Select
        e3Application.PutInfo 0, message 'output result of operation

    End If
Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2016-17.33 and v2017-1810.

## See Also

- [e3Text - Overview](#)
- [CalculateBoxHeight\(\) \(deprecated\)](#)

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.Delete()**

## Syntax

**Integer** Delete()

## Description

Deletes the text from the project.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
>0	Failure	Identifier of the text item
0	Inconclusive	Text item was deleted or no text item was set

## Remarks

The text item cannot be deleted under the following criteria:

- It is locked
- It is a symbol text defined in the database
- It is grouped

Due caution is recommended on relying on the return value of 0 meaning a text item has been deleted as this is the return value if no current item has been set.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting texts.

## Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()

textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then

    result = 0

    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )
        result = text.Delete()

        If result = 0 Then
            message = "Text: " & textId & " was deleted."
        Else
            message = "Text: " & textId & " was not deleted."
        End If
        e3Application.PutInfo 0, message                 'output result of operation

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2010-9.10.

---

## See Also

- [e3Text - Overview](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.DeleteAttribute( *name* )**

## Syntax

**Integer** DeleteAttribute( [in]**String** *name* )

## Description

Removes an attribute from the text item.

## Parameters

Type	Parameter	Description
[in] <b>String</b>	<i>name</i>	Name of the attribute

## Return Values

Value	Status	Description
> 0	Success	Attribute was removed
0	Failure	Error occurred

## Remarks

*name* must be the name of an existing attribute.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting texts using the given attribute.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()
```

## SetAsMaster - e3Symbol

```
textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then

    result = 0
    attributeName = "Example"           'attribute should exist.

    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )
        result = text.DeleteAttribute( attributeName )

        If result = 0 Then
            message = "Text: " & textId & " error occurred"
        Else
            message = "Text: " & textId & "; Attribute: " & attributeName & " was deleted."
        End If
        e3Application.PutInfo 0, message           'output result of operation

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2012-11.00.

---

## See Also

- [e3Text - Overview](#)
  - [AddAttributeValue\(\)](#)
  - [GetAttributeValue\(\)](#)
  - [HasAttribute\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.DeleteBox()**

## Syntax

**Integer** DeleteBox()

## Description

Deletes the text item's text box.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
1	Success	Text box was deleted
0	Failure	Delete text box operation failed

## Remarks

The text box cannot be deleted if the text item is locked.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting texts.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()
```

## SetAsMaster - e3Symbol

```
textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then

    result = 0

    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )
        result = text.DeleteBox()

        If result = 0 Then
            message = "Text: " & textId & " box was deleted."
        Else
            message = "Text: " & textId & " box was not deleted."
        End If
        e3Application.PutInfo 0, message                 'output result of operation

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2010-9.10.

---

## See Also

- [e3Text - Overview](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).





v2022-23.00

**e3Text.GetAlignment()**

## Syntax

**Integer** GetAlignment()

## Description

Gets the text's current alignment value.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
0	Failure	Alignment value could not be supplied
1	Success	Left aligned
2	Success	Centered
3	Success	Right aligned

## Remarks

A valid text id value must be assigned using SetId() otherwise 0 is returned.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting sheets in the sheet project tree.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
```

## SetAsMaster - e3Symbol

```
Set job = e3Application.CreateJobObject()
Set sheet = job.CreateSheetObject()
Set text = job.CreateTextObject()

sheetCount = job.GetTreeSelectedSheetIds( sheetIds)           'get all selected sheets from
If sheetCount > 0 Then

    sheet.SetId( sheetIds( 1 ) )           'use first selected sheet
    textCount = sheet.GetTextIds( textIds )           'get all texts on the sheet

    If textCount > 0 Then
        For index = 1 To textCount           'loop through all texts

            textId = text.SetId( textIds( index ) )
            result = text.GetAlignment()

            Select Case result
            Case 1
                message = "Left"
            Case 2
                message = "Center"
            Case 3
                message = "Right"
            Case Else
                message = "Undefined"
            End Select
            e3Application.PutInfo 0, "Text " & textId & ": alignment is " & message

        Next
    End If
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2010-9.10.

---

## See Also

- [e3Text - Overview](#)
  - [SetAlignment\(\)](#)
-

## SetAsMaster - e3Symbol

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.GetAllowedLength()**

## Syntax

**Integer** GetAllowedLength()

## Description

Gets the maximum number of characters allowed for the text.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
> 0	Success	Maximum text length value
0	Inconclusive	No maximum text length supplied

## Remarks

Due caution is recommended on relying on the return value of 0 because it can occur due to one of the following reasons:

- An error has occurred
- There is no maximum text length
- Maximum text length is 255

The maximum text length value is dependent on the text type. See [Text Types](#) for a list of predefined text type values and descriptions.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project

and then opening a sheet.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set sheet = job.CreateSheetObject()
Set text = job.CreateTextObject()

sheetId = job.GetActiveSheetId()           'get active sheet
sheet.SetId( sheetId )

textCount = sheet.GetTextIds( textIds )    'get the sheet's text ids
If textCount > 0 Then
    For textIndex = 1 To textCount          'loop through each sheet text

        textId = text.SetId( textIds( textIndex ) )
        result = text.GetAllowedLength()
        e3Application.PutInfo 0, "Sheet Text: " & textId & "; Type: " & text.GetType & " m

    Next
End If

Set text = Nothing
Set sheet = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2010-9.10.

## See Also

- [e3Text - Overview](#)
- [Text Types](#)

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.GetAssignedOptionExpressions( *expressions*, *Term* )**

## Syntax

**Integer** GetAssignedOptionExpressions( [out]**String Array** *expressions*,  
[in][optional]**Integer** *Term* )

## Description

Gets option names and boolean expressions (combinations of options) assigned to the text item.

## Parameters

Type	Parameter	Description
[out] <b>String Array</b>	<i>expressions</i>	Array of strings of all option names, boolean expressions or alias names of boolean expressions Indicates whether aliases should be resolved
[in][optional] <b>Integer</b>	<i>Term</i>	If the value is 1, alias names are resolved and the resulting expressions with option names are included in <i>expressions</i> The default value is 0

## Return Values

Value	Status	Description
> 0	Success	Number of items in <i>expressions</i> assigned
0	Inconclusive	No assigned option expressions are supplied or an error has occurred

## Remarks

*expressions* is a 1-based array.

Due caution is recommended on relying on the return value of 0 meaning there are no items in *expressions* since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>**.series project containing options and selecting texts.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()

textCount = job.GetSelectedTextIds ( textIds )           'get selected texts
If textCount > 0 Then
    For textIndex = 1 To textCount                       'loop through each text

        textId = text.SetId( textIds( textIndex ) )

        result = text.GetAssignedOptionExpressions( optionExpressions )           'get the te
        If result > 0 Then
            e3Application.PutInfo 0, "Text: " & textId & "; assigned option expressions:"
            For expressionIndex = 1 To result
                e3Application.PutInfo 0, "    " & optionExpressions( expressionIndex )
            Next
        Else
            e3Application.PutInfo 0, "Text: " & textId & "; No assigned option expressions"
        End If

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

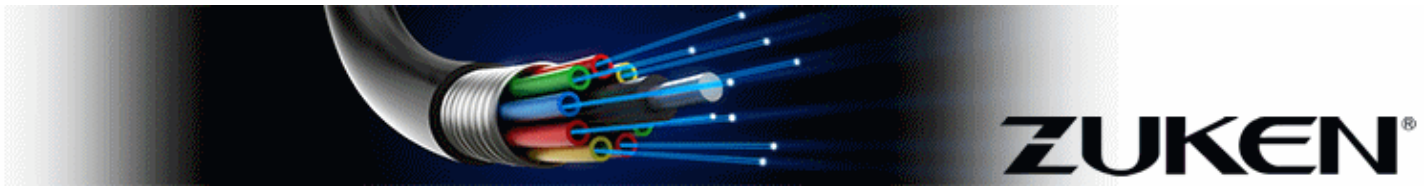
Introduced in v2010-9.10.

## See Also

- [e3Text - Overview](#)
- [GetAssignedOptionExpressionsEx\(\)](#)
- [SetOptionExpressions\(\)](#)

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).





v2022-23.00

**e3Text.GetAssignedOptionExpressionsEx( *expressions*, *Term* )**

## Syntax

**Integer** GetAssignedOptionExpressionsEx( [out]**String Array** *expressions*,  
[in][optional]**Integer** *Term* )

## Description

Gets option names and boolean expressions (combinations of options) assigned to the text item.

## Parameters

Type	Parameter	Description
[out] <b>String Array</b>	<i>expressions</i>	Array of strings of all option names, boolean expressions or alias names of boolean expressions Indicates whether aliases should be resolved
[in][optional] <b>Integer</b>	<i>Term</i>	If the value is 1, alias names are resolved and the resulting expressions with option names are included in <i>expressions</i> The default value is 0

## Return Values

Value	Status	Description
> 0	Success	Number of items in <i>expressions</i>
0	Inconclusive	No assigned option expressions are supplied or an error has occurred

## Remarks

Variant instances are currently available only for devices and wires.

*expressions* is a 1-based array.

Due caution is recommended on relying on the return value of 0 meaning there are no items in *expressions* since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>**.series project containing options and selecting texts.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()

textCount = job.GetSelectedTextIds ( textIds )           'get selected texts
If textCount > 0 Then
    For textIndex = 1 To textCount                         'loop through each text

        textId = text.SetId( textIds( textIndex ) )

        result = text.GetAssignedOptionExpressionsEx( optionExpressions )           'get the
        If result > 0 Then
            e3Application.PutInfo 0, "Text: " & textId & "; assigned option expressions:"
            For expressionIndex = 1 To result
                e3Application.PutInfo 0, "    " & optionExpressions( expressionIndex )
            Next
        Else
            e3Application.PutInfo 0, "Text: " & textId & "; No assigned option expressions"
        End If

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2010-9.12.

## See Also

- [e3Text - Overview](#)
- [GetAssignedOptionExpressions\(\)](#)

- SetOptionExpressions()
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.GetAssignedOptionIds( *ids* )**

## Syntax

**Integer** GetAssignedOptionIds( [out]**Integer Array** *ids* )

## Description

Gets identifiers of options assigned to the text item.

## Parameters

Type	Parameter	Description
[out] <b>Integer Array</b>	<i>ids</i>	Array of identifiers of assigned options

## Return Values

Value	Status	Description
> 0	Success	Number of items in <i>ids</i>
0	Inconclusive	No assigned option identifiers are supplied or an error has occurred
-1	Failure	An option contains a boolean expression or an alias

## Remarks

*ids* is a 1-based array.

The option identifiers can be used by e3Option objects to handle them.

Due caution is recommended on relying on the return value of 0 meaning there are no items in *ids* since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project

and selecting text items on a sheet.

## Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()

textCount = job.GetSelectedTextIds ( textIds )           'get selected texts
If textCount > 0 Then
    For textIndex = 1 To textCount                         'loop through each text

        textId = text.SetId( textIds( textIndex ) )

        result = text.GetAssignedOptionIds( optionIds )
        If result > 0 Then
            e3Application.PutInfo 0, "Text: " & textId & "; assigned option ids:"
            For optionIndex = 1 To result
                e3Application.PutInfo 0, "    " & optionIds( optionIndex )
            Next
        Else
            e3Application.PutInfo 0, "Text: " & textId & "; No assigned option ids found"
        End If
    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2010-9.10.

---

## See Also

- [e3Text - Overview](#)
  - [SetOptionExpressions\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.GetAttributeCount()**

## Syntax

**Integer** GetAttributeCount()

## Description

Gets the number of the text item's attributes.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
> 0	Success	Number of attributes found
0	Inconclusive	No attribute found or an error occurred

## Remarks

Due caution is recommended on relying on the return value of 0 meaning no attributes found since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting texts on a sheet.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()
```

```

textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then
    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )
        result = text.GetAttributeCount()

        If result = 0 Then
            message = "Text: " & textId & "no attributes found"
        Else
            message = "Text: " & textId & "; " & result & " attributes found"
        End If
        e3Application.PutInfo 0, message                'output result of operation

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

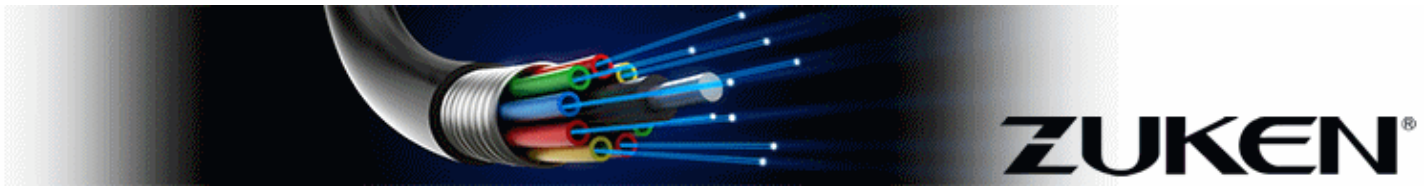
Introduced in v2012-11.00.

---

## See Also

- [e3Text - Overview](#)
  - [GetAttributeValue\(\)](#)
  - [HasAttribute\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.GetAttributeIds( *ids*, *attnam* )**

## Syntax

**Integer** GetAttributeIds( [out]**Integer Array** *ids*, [in][optional]**String** *attnam* )

## Description

Gets identifiers of the attributes assigned to the text item.

## Parameters

Type	Parameter	Description
[out] <b>Integer Array</b>	<i>ids</i>	Array of identifiers of attributes
		Attribute name filter
		Only attribute identifiers with this attribute name will be supplied
[in][optional] <b>String</b>	<i>attnam</i>	All attribute names are supplied if <i>attnam</i> is an empty string
		The default value is "<Empty>"

## Return Values

Value	Status	Description
> 0	Success	Number of items in <i>ids</i>
0	Inconclusive	No assigned attribute identifiers are found or an error has occurred

## Remarks

*ids* is a 1-based array.

The attribute identifiers can be used by e3Attribute objects to handle them.



Due caution is recommended on relying on the return value of 0 meaning there are no items in *ids* since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>**.series project and selecting texts.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()

textCount = job.GetSelectedTextIds ( textIds )           'get selected texts
If textCount > 0 Then
    For textIndex = 1 To textCount                       'loop through each text

        textId = text.SetId( textIds( textIndex ) )
        result = text.GetAttributeIds( attributeIds )
        If result = 0 Then
            e3Application.PutInfo 0, "Text: " & textId & "; No attribute ids found"
        Else
            e3Application.PutInfo 0, "Text: " & textId & "; Attribute ids:"
            For attributeIndex = 1 To result
                e3Application.PutInfo 0, "    " & attributeIds( attributeIndex )
            Next
        End If
    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2012-11.00.

## See Also

- [e3Text - Overview](#)
- [GetAttributeValue\(\)](#)
- [SetAttributeValue\(\)](#)

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.GetAttributeValue( *name* )**

## Syntax

*String* GetAttributeValue( [in]*String* *name* )

## Description

Gets the text item's specified attribute value.

## Parameters

Type	Parameter	Description
[in] <i>String</i>	<i>name</i>	Name of the attribute

## Return Values

Value	Status	Description
"<Text>"	Success	Attribute value supplied
"<Empty>"	Inconclusive	Attribute value could not be found or an error occurred

## Remarks

*name* must be the name of an existing attribute.

Due caution is recommended on relying on the return value of "<Empty>" meaning there is no value set since this also could mean an error has occurred.

Since v2019-20.00 the attribute value of the original text is returned if the text is in a view and the attribute *name* is undefined.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>**.series project and selecting texts using the given attribute.

## Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()

textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then

    result = 0
    attributeName = "Example"           'attribute should exist.

    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )
        result = text.GetAttributeValue( attributeName )

        If Len( "" & result ) = 0 Then
            message = "Text: " & textId & "; Attribute: " & attributeName & " = empty string"
        Else
            message = "Text: " & textId & "; Attribute: " & attributeName & " = " & result
        End If
        e3Application.PutInfo 0, message           'output result of operation

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2012-11.00.

Modified in v2019-20.00.

---

## See Also

- [e3Text - Overview](#)
  - [HasAttribute\(\)](#)
  - [SetAttributeValue\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.GetBallooning()**

## Syntax

**Integer** GetBallooning()

## Description

Gets the text's ballooning value.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
> 0	Success	Ballooning value applied See <a href="#">Ballooning</a> for possible values
0	Inconclusive	Ballooning value could not be supplied or there is no ballooning

## Remarks

Due caution is recommended on relying on the return value of 0 meaning no ballooning since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>**.series project and selecting texts on a sheet.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
```

## SetAsMaster - e3Symbol

```
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()

textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then
    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )
        result = text.GetBallooning()

        Select Case result
        Case 0
            message = "No ballooning or error occurred"
        Case 1
            message = "Circle"
        Case 2
            message = "Oval"
        Case 4
            message = "Rectangle"
        Case 8
            message = "Ellipse"
        Case 16
            message = "Line To owner"
        Case 17
            message = "Circle and line to owner"
        Case 18
            message = "Oval and line to owner"
        Case 20
            message = "Rectangle and line to owner"
        Case 24
            message = "Ellipse and line to owner"
        Case 32
            message = "Horizontal line on centre"
        Case 64
            message = "Horizontal line on bottom"
        Case 128
            message = "Horizontal line on top"
        End Select
        e3Application.PutInfo 0, "Text: " & textId & ": " & message      'output result

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2010-9.10.

---

## See Also

- [e3Text - Overview](#)
  - [Ballooning](#)
  - [SetBallooning\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.GetBox( *xsize*, *ysize* )**

## Syntax

**Integer** GetBox( [out]**Double** *xsize*, [out]**Double** *ysize* )

## Description

Gets the text item's box width and height.

## Parameters

Type	Parameter	Description
[out] <b>Double</b>	<i>xsize</i>	Width of box
[out] <b>Double</b>	<i>ysize</i>	Height of box

## Return Values

Value	Status	Description
1	Success	<i>xsize</i> and <i>ysize</i> values supplied
0	Failure	Error occurred

## Remarks

The box represents a rectangle containing the text.

Lines of text are broken at the defined box width preferably after a blank, a tab or one of the following characters: . ? ! ; : -

The line break is made in the middle of the word if none of these characters is found within the defined width.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting texts on a sheet.



## Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()

textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then
    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )
        result = text.GetBox( width, height )

        If result = 0 Then
            e3Application.PutInfo 0, "Text: " & textId & " error occurred getting box"
        Else
            e3Application.PutInfo 0, "Text: " & textId & "; width = " & width & "; height
        End If
        e3Application.PutInfo 0, message                'output result of operation

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing

```

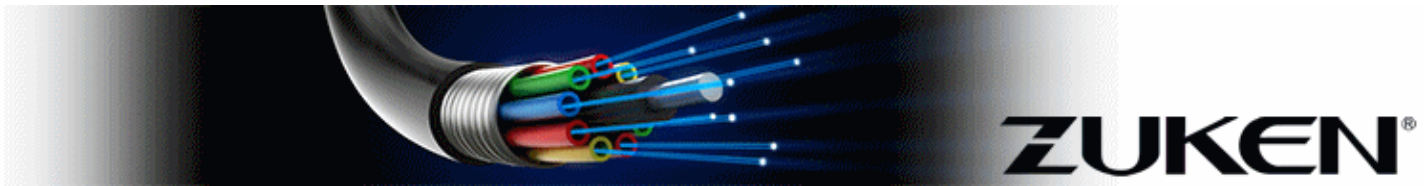
## Version Information

Introduced in v2010-9.10.

## See Also

- [e3Text - Overview](#)
- [SetBox\(\)](#)

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.GetColour()**

## Syntax

**Integer** GetColour()

## Description

Gets the text item's text color.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
1..255	Success	Color value See <a href="#">Colors</a> for possible values
0	Inconclusive	Black (RGB: 0, 0, 0 ) or the operation has failed
-1	Success	Automatic color

## Remarks

Due caution is recommended on relying on the return value of 0 meaning black since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting texts on a sheet.

Visual Basic Script

## SetAsMaster - e3Symbol

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()

textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then
    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )
        result = text.GetColour()

        If result = -1 Then
            message = "Text: " & textId & " color is automatic"
        Else
            message = "Text: " & textId & " color = " & result
        End If
        e3Application.PutInfo 0, message                 'output result of operation

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2010-9.10.

---

## See Also

- [e3Text - Overview](#)
  - [Colors](#)
  - [SetColour\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.GetFontName()**

## Syntax

*String* GetFontName()

## Description

Gets the text item's font name.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
"<Text>"	Success	Name of the text's font
"Arial"	Inconclusive	Name of the text's font is Arial or the font name could not be supplied

## Remarks

Due caution is recommended on relying on the return value of "Arial" meaning the Arial font since this also could mean an error has occurred.

The default font name for all free texts is provided by a global setting that can be accessed using [e3Job.GetGraphTextFontName\(\)](#) and [e3Job.SetGraphTextFontName\(\)](#). Once the font name of a text item is changed, it is no longer subject to the global setting.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>**.series project and selecting texts on a sheet.

## Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()

textCount = job.GetSelectedTextIds( textIds )      'get selected texts
If textCount > 0 Then
    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )
        result = text.GetFontName()
        e3Application.PutInfo 0, "Text: " & textId & " font name is " & result      'out

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2010-9.10.

---

## See Also

- [e3Text - Overview](#)
  - [e3Job.GetGraphTextFontName\(\)](#)
  - [e3Job.SetGraphTextFontName\(\)](#)
  - [SetFontName\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.GetGID()**

## Syntax

*String* GetGID()

## Description

Gets the global identifier of the current text item.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
"<GID>"	Success	Global identifier of the current text item
"<Empty>"	Failure	No text item

## Remarks

Global identifiers (GIDs) are used for item identification in multiuser projects.

## Examples

The best results from the example can be achieved by opening an *E<sup>3</sup>.series* project and selecting text items on a sheet.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()

textCount = job.GetSelectedTextIds( textIds )           'get selected texts
```

## SetAsMaster - e3Symbol

```
If textCount > 0 Then
  For textIndex = 1 To textCount

    textId = text.SetId( textIds( textIndex ) )

    result = text.GetGID()
    If Len( "" & result ) = 0 Then
      message = "No text item is set"
    Else
      message = "GID of text item " & textId & " is " & result
    End If
    e3Application.PutInfo 0, message          'output result of operation

  Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2022-23.00.

---

## See Also

- [e3Text - Overview](#)
  - [GetId\(\)](#)
  - [SetGID\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.GetHeight()**

## Syntax

Double GetHeight()

## Description

Gets the text item's font height.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
> 0.0	Success	Size of the text's font height
0.0	Failure	Font height could not be supplied

## Remarks

The text's font height is in project measurement units.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting texts on a sheet.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()
```



## SetAsMaster - e3Symbol

```
textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then
    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )
        result = text.GetHeight()
        If result = 0 Then
            message = "Text: " & textId & " font height could not be supplied"
        Else
            message = "Text: " & textId & " font height is " & result
        End If
        e3Application.PutInfo 0, message                 'output result of operation

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2010-9.10.

---

## See Also

- [e3Text - Overview](#)
  - [SetHeight\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.GetHyperlinkAddress()**

## Syntax

*String* GetHyperlinkAddress()

## Description

Gets the text item's hyperlink address.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
"<Text>"	Success	Hyperlink address supplied
"<Empty>"	Failure	An error occurred or the text does not have a hyperlink address

## Remarks

The function does not work for texts linked with the **Hyperlink** attribute.

The return value's syntax uses the following format:

Value	Description
"<empty>"	No hyperlink address
"http:<target>"	Internet link using Hypertext Transfer Protocol (HTTP)
"https:<target>"	Internet link using Hypertext Transfer Protocol Secured (HTTPS)
"ftp:<target>"	Internet link using File Transfer Protocol (FTP)
"ftps:<target>"	Internet link using File Transfer Protocol Secured (FTPS)
"sftp:<target>"	Internet link using SSH File Transfer Protocol (SFTP)

Remarks

"file:<target>" File link using a system path

"gid:<target>" Sheet, symbol or document using a Global Identifier (GID)

Due caution is recommended on relying on the return value of "<Empty>" meaning the text has no hyperlink address since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>**.series project and selecting texts containing hyperlinks on a sheet.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()

textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then
    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )
        result = text.GetHyperlinkAddress()
        If Len( "" & result ) = 0 Then
            message = "Text: " & textId & " hyperlink address is empty or could not be sup
        Else
            message = "Text: " & textId & " hyperlink address is " & result
        End If
        e3Application.PutInfo 0, message                 'output result of operation

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2010-9.10.

## See Also

- [e3Text - Overview](#)
- [SetHyperlinkAddress\(\)](#)

## SetAsMaster - e3Symbol

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.GetId()**

## Syntax

Integer GetId()

## Description

Gets the identifier of the current text item.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
> 0	Success	Current text item identifier
0	Failure	No text item

## Remarks

The function returns the identifier value set by SetId() or e3Graph.CreateText() unless the item no longer exists.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>**.series project and selecting texts on a sheet.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()
```

```

textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then
    For textIndex = 1 To textCount

        text.SetId textIds( textIndex )

        result = text.GetId()
        If result = 0 Then
            message = "No text item is set"
        Else
            message = "Text " & result & " has been set"
        End If
        e3Application.PutInfo 0, message                'output result of operation

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2010-9.10.

---

## See Also

- [e3Text - Overview](#)
  - [e3Graph.CreateText\(\)](#)
  - [SetId\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**GetInternalText()**

## Syntax

*String* GetInternalText()

## Description

Gets the text item's text value without text token translations.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
"<Text>"	Success	Text value is supplied
"<Empty>"	Failure	Text does not have a value or an error occurred

## Remarks

The return value includes text tokens (&#lt;num>;) within the text value and not their resulting translations. GetText() is available for getting the text value with translations instead of the text tokens.

A value of "<Empty>" will be returned for unassigned text items, typically represented with an "X" on sheets.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>**.series project and selecting texts on a sheet.

Visual Basic Script

## SetAsMaster - e3Symbol

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()

textCount = job.GetSelectedTextIds( textIds )      'get selected texts
If textCount > 0 Then
    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )
        result = text.GetInternalText()
        If Len( "" & result ) = 0 Then
            message = "Text " & textId & ": text is empty"
        Else
            message = "Text " & textId & ": text = " & result
        End If
        e3Application.PutInfo 0, message            'output result of operation

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2010-9.10.

---

## See Also

- [e3Text - Overview](#)
  - [GetText\(\)](#)
  - [SetText\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).





v2022-23.00

**e3Text.GetLanguageId()**

## Syntax

Integer GetLanguageId()

## Description

Gets the text item's translation identifier value.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
<TranslationId>	Success	The text item's translation identifier
0	Inconclusive	Text item has no translation identifier or an error has occurred

## Remarks

Due caution is recommended on relying on the return value of 0 meaning the text has no translation identifier assigned since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>**.series project and selecting texts using text translations on a sheet.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
```

## SetAsMaster - e3Symbol

```
Set text = job.CreateTextObject()

textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then
    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )

        result = text.GetLanguageId()
        If result = 0 Then
            message = "Text: " & textId & " no language found."
        Else
            message = "Text: " & textId & " language = " & result
        End If
        e3Application.PutInfo 0, message                'output result of operation

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2010-9.10.

---

## See Also

- [e3Text - Overview](#)
  - [SetLanguageId\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.GetLeftJustifiedSchemaLocation( *x*, *y*, *grid* )**

## Syntax

**Integer** GetLeftJustifiedSchemaLocation( [out]**Double** *x*, [out]**Double** *y*, [out]**String** *grid* )

## Description

Gets the text item's box's lower right corner position in text reading direction.

## Parameters

Type	Parameter	Description
[out] <b>Double</b>	<i>x</i>	Position on the x-axis
[out] <b>Double</b>	<i>y</i>	Position on the y-axis
[out] <b>String</b>	<i>grid</i>	Location in the format "/<sheet>.<grid>"

## Return Values

Value	Status	Description
> 0	Success	Sheet id of the text item
0	Failure	Error occurred

## Remarks

The function is meant for left aligned text. For right aligned text [GetRightJustifiedSchemaLocation\(\)](#) is available.

*x* and *y* are in project measurement units.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting texts on a sheet.

## Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()
Set sheet = job.CreateSheetObject()

textCount = job.GetSelectedTextIds( textIds )      'get selected texts
If textCount > 0 Then
    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )
        result = text.GetLeftJustifiedSchemaLocation( xPosition, yPosition, grid )

        If result = 0 Then
            e3Application.PutInfo 0, "Text: " & textId & " error finding left justified so
        Else
            sheet.SetId result
            e3Application.PutInfo 0, "Text: " & textId & ":"
            e3Application.PutInfo 0, "    Sheet: " & sheet.GetName
            e3Application.PutInfo 0, "    x: " & xPosition
            e3Application.PutInfo 0, "    y: " & yPosition
            e3Application.PutInfo 0, "    grid: " & grid
        End If

    Next
End If

Set sheet = Nothing
Set text = Nothing
Set job = Nothing
Set e3Application = Nothing

```

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting texts on a sheet.

## Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()
Set sheet = job.CreateSheetObject()
Set graphic = job.CreateGraphObject()

height = job.GetGraphTextHeight()
job.SetGraphTextColour( 13 )      'red
pi = 3.1415926535897932

textCount = job.GetSelectedTextIds( textIds )      'get selected texts
If textCount > 0 Then

```

## SetAsMaster - e3Symbol

```
For textIndex = 1 To textCount      'add a free graphic text string below the text i

    textId = text.SetId( textIds( textIndex ) )
    sheetId = text.GetLeftJustifiedSchemaLocation( x, y, grid )
    rotation = text.GetRotation()

    xe = x + cos( (rotation - 90.0 ) * pi / 180.0 ) * height
    ye = y + sin( (rotation - 90.0 ) * pi / 180.0 ) * height

    rotatedTextId = graphic.CreateRotatedText( sheetId, textId, xe, ye, rotation )
    text.SetId rotatedTextId
    text.SetText grid

Next
End If

Set graphic = Nothing
Set sheet = Nothing
Set text = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

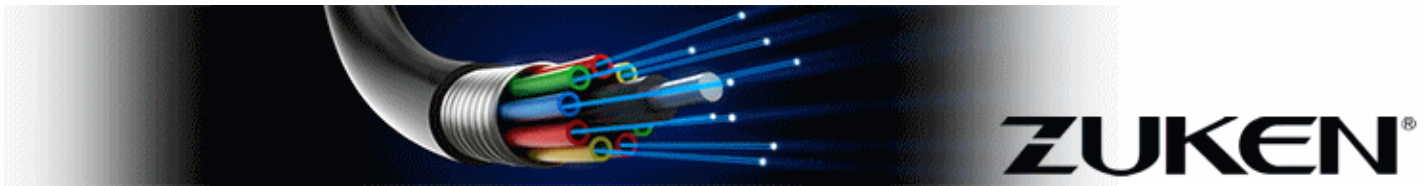
Introduced in v2010-9.10.

---

## See Also

- [e3Text - Overview](#)
  - [GetRightJustifiedSchemaLocation\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.GetLevel()**

## Syntax

**Integer** GetLevel()

## Description

Gets the text item's display level value.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
> 0	Success	Display level value supplied
0	Failure	Error occurred

## Remarks

GetVisibility() and SetVisibility() can be used to hide or show the text item independent of display level values.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting texts on a sheet.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()
```

```

textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then
    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )
        result = text.GetLevel()
        If result = 0 Then
            message = "Text: " & textId & " display level could not be supplied"
        Else
            message = "Text: " & textId & " display level is " & result
        End If
        e3Application.PutInfo 0, message                'output result of operation

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2010-9.10.

## See Also

- [e3Text - Overview](#)
- [GetVisibility\(\)](#)
- [SetLevel\(\)](#)
- [SetVisibility\(\)](#)

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.GetLinearMeasureWithoutUnit()**

## Syntax

**Boolean** GetLinearMeasureWithoutUnit()

## Description

Gets the text item's flag value determining if the length measurement unit is displayed along with the value.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
<b>True</b>	Success	Length measurement unit is not shown
<b>False</b>	Inconclusive	Length measurement unit is shown or an error occurred

## Remarks

Due caution is recommended on relying on the return value of **False** meaning the length measurement unit is shown since it also could mean an error has occurred.

**False** will be returned for any text items of text types not displaying length measurements.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>**.series project and selecting symbols on a sheet.

Visual Basic Script



## SetAsMaster - e3Symbol

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set symbol = job.CreateSymbolObject()
Set text = job.CreateTextObject()

symbolCount = job.GetSelectedSymbolIds( symbolIds )           'get selected symbols
If symbolCount > 0 Then
    For symbolIndex = 1 To symbolCount

        symbolId = symbol.SetId( symbolIds( symbolIndex ) )
        textCount = symbol.GetTextIds( textIds )             'get texts of symbol

        If textCount > 0 Then
            e3Application.PutInfo 0, "Symbol: " & symbolId & " ; Name: " & symbol.GetName

            For textIndex = 1 To textCount

                textId = text.SetId( textIds( textIndex ) )
                result = text.GetLinearMeasureWithoutUnit()

                If result = False Then
                    message = "    Text: " & textId & " linear measurement is shown with u
                Else
                    message = "    Text: " & textId & " linear measurement is shown without u
                End If
                e3Application.PutInfo 0, message               'output result of operation

            Next
        End If
    Next
End If

Set text = Nothing
Set symbol = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2017-17.70.

---

## See Also

- [e3Text - Overview](#)
  - [SetLinearMeasureWithoutUnit\(\)](#)
-

## SetAsMaster - e3Symbol

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.GetLocking()**

## Syntax

**Boolean** GetLocking()

## Description

Gets the text item's flag value determining if the text position is locked.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
<b>True</b>	Success	Text position is locked
<b>False</b>	Inconclusive	Text position is unlocked or an error occurred

## Remarks

Due caution is recommended on relying on the return value of **False** meaning the text position is locked since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting texts on a sheet.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()
```

```

textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then
    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )
        result = text.GetLocking()

        If result = 0 Then
            message = "Text: " & textId & " position is not locked"
        Else
            message = "Text: " & textId & " position is locked"
        End If
        e3Application.PutInfo 0, message                'output result of operation

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2010-9.10.

---

## See Also

- [e3Text - Overview](#)
  - [SetLocking\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.GetMode()**

## Syntax

**Integer** GetMode()

## Description

Gets the text item's text ratio.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
1..3	Success	Text ratio value supplied
0	Failure	Text ratio value could not be supplied

## Remarks

A successful return value can mean one of the following:

Value	Description
1	Normal
2	Narrow
3	Wide

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting texts on a sheet.

Visual Basic Script

## SetAsMaster - e3Symbol

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()

textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then
    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )
        result = text.GetMode()

        Select Case result
            Case 1
                message = "Normal"
            Case 2
                message = "Narrow"
            Case 3
                message = "Wide"
            Case Else
                message = "Undefined"
        End Select
        e3Application.PutInfo 0, "Text " & textId & ": ratio is " & message    'output

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2010-9.10.

---

## See Also

- [e3Text - Overview](#)
  - [SetMode\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.GetPictogram()**

## Syntax

Boolean GetPictogram()

## Description

Gets the text item's flag value determining if the text is displayed in the pictogram language.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
True	Success	Text is displayed in the pictogram language
False	Inconclusive	Text is not displayed in the pictogram language or an error occurred

## Remarks

Due caution is recommended on relying on the return value of **False** meaning the text is not displayed in the pictogram language since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project using pictograms and selecting texts on a sheet.

Visual Basic Script

## SetAsMaster - e3Symbol

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()

textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then
    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )
        result = text.GetPictogram()

        If result = False Then
            message = "Text: " & textId & " not using pictogram"
        Else
            message = "Text: " & textId & " using pictogram"
        End If
        e3Application.PutInfo 0, message                 'output result of operation

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2017-17.70.

---

## See Also

- [e3Text - Overview](#)
  - [SetPictogram\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).





v2022-23.00

**e3Text.GetRightJustifiedSchemaLocation( *x*, *y*, *grid* )**

## Syntax

**Integer** GetRightJustifiedSchemaLocation( [out]**Double** *x*, [out]**Double** *y*, [out]**String** *grid* )

## Description

Gets the text item's box's lower right corner position in text reading direction.

## Parameters

Type	Parameter	Description
[out] <b>Double</b>	<i>x</i>	Position on the x-axis
[out] <b>Double</b>	<i>y</i>	Position on the y-axis
[out] <b>String</b>	<i>grid</i>	Location in the format "/<sheet>.<grid>"

## Return Values

Value	Status	Description
> 0	Success	Sheet id of the text item
0	Failure	Error has occurred

## Remarks

The function is meant for right aligned text. For left aligned text [GetLeftJustifiedSchemaLocation\(\)](#) is available.

*x* and *y* are in project measurement units.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting texts on a sheet.

## Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()
Set sheet = job.CreateSheetObject()

textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then
    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )
        result = text.GetRightJustifiedSchemaLocation( xPosition, yPosition, grid )

        If result = 0 Then
            e3Application.PutInfo 0, "Text: " & textId & " error finding right justified s
        Else
            sheet.SetId result
            e3Application.PutInfo 0, "Text: " & textId & ":"
            e3Application.PutInfo 0, "    Sheet: " & sheet.GetName
            e3Application.PutInfo 0, "    x: " & xPosition
            e3Application.PutInfo 0, "    y: " & yPosition
            e3Application.PutInfo 0, "    grid: " & grid
        End If

    Next
End If

Set sheet = Nothing
Set text = Nothing
Set job = Nothing
Set e3Application = Nothing

```

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting texts on a sheet.

## Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()
Set sheet = job.CreateSheetObject()
Set graphic = job.CreateGraphObject()

height = job.GetGraphTextHeight()
job.SetGraphTextColour( 13 )           'red
pi = 3.1415926535897932

textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then

```

## SetAsMaster - e3Symbol

```
For textIndex = 1 To textCount      'add a free graphic text string below the text i

    textId = text.SetId( textIds( textIndex ) )
    sheetId = text.GetRightJustifiedSchemaLocation( x, y, grid )
    rotation = text.GetRotation()

    xe = x + cos( ( rotation - 90.0 ) * pi / 180.0 ) * height
    ye = y + sin( ( rotation - 90.0 ) * pi / 180.0 ) * height

    rotatedTextId = graphic.CreateRotatedText( sheetId, textId, xe, ye, rotation )
    text.SetId rotatedTextId
    text.SetText grid

Next
End If

Set graphic = Nothing
Set sheet = Nothing
Set text = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2010-9.10.

---

## See Also

- [e3Text - Overview](#)
  - [GetLeftJustifiedSchemaLocation\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.GetRotation()**

## Syntax

Double GetRotation()

## Description

Gets the text item's rotation value.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
> 0.0..< 360.0	Success	Text rotation value in degrees supplied
0.0	Inconclusive	Text rotation value supplied or an error occurred

## Remarks

The value expresses counterclockwise rotation.

Due caution is recommended on relying on the return value of 0.0 meaning the text is not rotated since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an *E<sup>3</sup>.series* project and selecting texts on a sheet.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
```

## SetAsMaster - e3Symbol

```
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()

textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then
    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )
        result = text.GetRotation()
        e3Application.PutInfo 0, "Text: " & textId & " rotation is " & result    'output

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2010-9.10.

---

## See Also

- [e3Text - Overview](#)
  - [SetRotation\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.GetSchemaLocation( *x*, *y*, *grid*, *column\_value*, *row\_value* )**

## Syntax

**Integer** GetSchemaLocation( [out]**Double** *x*, [out]**Double** *y*, [out]**String** *grid*, [out][optional]**String** *column\_value*, [out][optional]**String** *row\_value* )

## Description

Gets the text item's position within the project.

## Parameters

Type	Parameter	Description
[out] <b>Double</b>	<i>x</i>	Placement position on the x-axis
[out] <b>Double</b>	<i>y</i>	Placement position on the y-axis
[out] <b>String</b>	<i>grid</i>	Location in the format "/<sheet>.<grid>"
[out][optional] <b>String</b>	<i>column_value</i>	Sheet placement column of the text
[out][optional] <b>String</b>	<i>row_value</i>	Sheet placement row of the text

## Return Values

Value	Status	Description
> 0	Success	Identifier of the text item's sheet
0	Failure	Error occurred

## Remarks

*x* and *y* are in project measurement units.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting texts on a sheet.

## Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()
Set sheet = job.CreateSheetObject()

textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then
    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )
        result = text.GetSchemaLocation( x, y, grid, column, row )

        If result = 0 Then
            e3Application.PutInfo 0, "Text: " & textId & " error getting schema location"
        Else
            sheet.SetId result
            e3Application.PutInfo 0, "Text: " & textId & ":"
            e3Application.PutInfo 0, "    Sheet: " & sheet.GetName
            e3Application.PutInfo 0, "    x: " & x
            e3Application.PutInfo 0, "    y: " & y
            e3Application.PutInfo 0, "    grid: " & grid
            e3Application.PutInfo 0, "    column: " & column
            e3Application.PutInfo 0, "    row: " & row
        End If
    Next
End If

Set sheet = Nothing
Set text = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2010-9.10.

---

## See Also

- [e3Text - Overview](#)
  - [SetSchemaLocation\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.GetSingleLine()**

## Syntax

Integer GetSingleLine()

## Description

Gets the text item's flag value determining if multi-line text is displayed on a single line.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
1	Success	Text is displayed on a single line
0	Inconclusive	Text is not displayed on a single line or an error occurred

## Remarks

Due caution is recommended on relying on the return value of 0 meaning the text is not displayed on a single line since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>**.series project selecting multi-line texts on a sheet.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
```



## SetAsMaster - e3Symbol

```
Set text = job.CreateTextObject()

textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then
    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )
        result = text.GetSingleLine()

        If result = 0 Then
            message = "Text: " & textId & " is not flagged to be single line"
        Else
            message = "Text: " & textId & " is flagged to be single line"
        End If
        e3Application.PutInfo 0, message                 'output result of operation

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

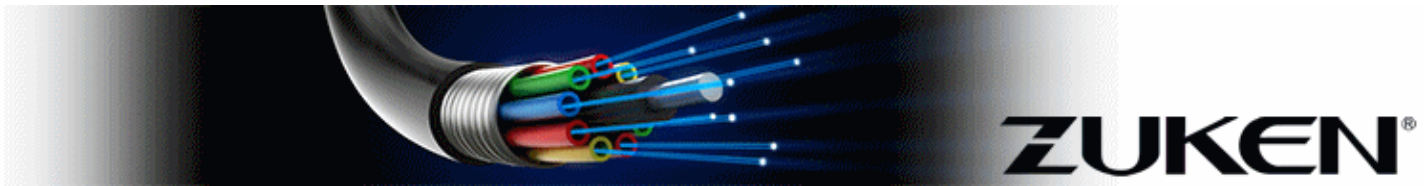
Introduced in v2010-9.10.

---

## See Also

- [e3Text - Overview](#)
  - [SetSingleLine\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.GetStyle()**

## Syntax

**Integer** GetStyle()

## Description

Gets the text item's font style.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
1..7	Success	Font style value supplied
0	Inconclusive	Regular font style value used or an error occurred

## Remarks

A successful return value can be a combination of the following values:

Value	Description
0	Regular
1	<b>Bold</b>
2	<i>Italics</i>
4	<u>Underline</u>
8	Strikethrough
16	Opaque

Due caution is recommended on relying on the return value of 0 meaning the font style is regular since this also could mean an error has occurred.

# Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>**.series project and selecting texts on a sheet.

## Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()

textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then
    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )
        result = text.GetStyle()

        If result = 0 Then
            e3Application.PutInfo 0, "Text " & textId & ": fontstyle is regular"
        Else
            e3Application.PutInfo 0, "Text " & textId & ": fontstyle is:"
            If result And 1 Then
                e3Application.PutInfo 0, "    Bold"
            End If
            If result And 2 Then
                e3Application.PutInfo 0, "    Italics"
            End If
            If result And 4 Then
                e3Application.PutInfo 0, "    Underline"
            End If
            If result And 8 Then
                e3Application.PutInfo 0, "    Strikethrough"
            End If
            If result And 16 Then
                e3Application.PutInfo 0, "    Opaque"
            End If
        End If
        e3Application.PutInfo 0, ""           'line break

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2010-9.10.

---

## See Also

- [e3Text - Overview](#)
  - [SetStyle\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.GetText()**

## Syntax

*String* GetText()

## Description

Gets the text item's text value including text token translations.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
"<Text>"	Success	Text value is supplied
"<Empty>"	Inconclusive	Text item does not have a value or an error occurred

## Remarks

Due caution is recommended on relying on the return value of "<Empty>" meaning the text item does not have a value since this also could mean an error has occurred.

The function will replace any text tokens (&#<num>;) with the translations. If the text exceeds the maximum number of characters, the return value will be automatically truncated. [GetInternalText\(\)](#) is available for getting the text value with text tokens instead of the translations.

A value of "<Empty>" will be returned for unassigned text items, typically represented with an "X" on sheets.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting texts on a sheet.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()

textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then
    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )
        result = text.GetText()

        If Len( "" & result ) = 0 Then
            message = "Text " & textId & ": Displayed text is empty"
        Else
            message = "Text " & textId & ": Displayed text = " & result
        End If
        e3Application.PutInfo 0, message                 'output result of operation

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2010-9.10.

---

## See Also

- [e3Text - Overview](#)
  - [GetInternalText\(\)](#)
  - [SetText\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.GetTextExtent( *xarr*, *yarr* )**

## Syntax

**Integer** GetTextExtent( [out]**Double Array** *xarr*, [out]**Double Array** *yarr* )

## Description

Gets the text item's dimension as a geometric shape.

## Parameters

Type	Parameter	Description
[out] <b>Double Array</b>	<i>xarr</i>	Array of dimension values on the x-axis
[out] <b>Double Array</b>	<i>yarr</i>	Array of dimension values on the y-axis

## Return Values

Value	Status	Description
1	Success	Text extent is supplied
0	Failure	Error occurred

## Remarks

*xarr* and *yarr* are 1-based indexed arrays containing 5 elements. The value of the first and last elements are identical.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting texts on a sheet.

Visual Basic Script

## SetAsMaster - e3Symbol

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()

textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then
    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )
        result = text.GetTextExtent( xPositions, yPositions )

        If result = 0 Then
            e3Application.PutInfo 0, "Text " & textId & ": unknown text extent"
        Else
            e3Application.PutInfo 0, "Text " & textId & ": "
            For positionIndex = 1 To 5
                e3Application.PutInfo 0, "Position " & positionIndex & ": x = " & xPositions(positionIndex)
            Next
        End If
        e3Application.PutInfo 0, ""                      'line break

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2011-10.40.

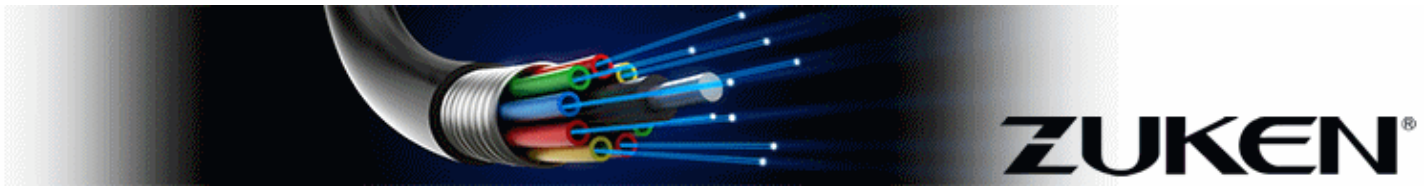
---

## See Also

- [e3Text - Overview](#)
  - [GetTextExtentSingleLine\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).





v2022-23.00

**e3Text.GetTextExtentSingleLine( *nlines*, *xarr*, *yarr* )**

## Syntax

**Integer** GetTextExtentSingleLine( [out]**Integer** *nlines*, [out]**Double 2D-Array** *xarr*, [out]**Double 2D-Array** *yarr* )

## Description

Gets the text item's dimension as a geometric shape for each line.

## Parameters

Type	Parameter	Description
[out] <b>Integer</b>	<i>nlines</i>	Number of lines
[out] <b>Double 2D-Array</b>	<i>xarr</i>	2-dimensional array of lines of dimension values on the x-axis
[out] <b>Double 2D-Array</b>	<i>yarr</i>	2-dimensional array of lines of dimension values on the y-axis

## Return Values

Value	Status	Description
1	Success	Text extent is supplied
0	Failure	Error occurred

## Remarks

*xarr* and *yarr* are **1-based indexed** and contain array elements each representing a line of text. Every inner array is **0-based indexed** and contains 4 elements holding dimension values. The value of the first and last elements of each inner array are identical.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting texts on a sheet.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()

textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then
    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )
        result = text.GetTextExtentSingleLine( lines, xPositions, yPositions )

        If result = 0 Then
            e3Application.PutInfo 0, "Text " & textId & ": unknown text extent"
        Else
            e3Application.PutInfo 0, "Text " & textId & ": "
            For lineIndex = 1 To lines                    'the outer lines array uses 1-based indexing
                e3Application.PutInfo 0, "    Line = " & lineIndex

                For positionIndex = 0 To 4                'the inner line dimension array uses 0-based indexing
                    e3Application.PutInfo 0, "        Position " & positionIndex & ": x = " & xPositions( lineIndex - 1, positionIndex )
                Next
            Next
        End If
        e3Application.PutInfo 0, ""                      'line break
    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2011-10.40.

---

## See Also

- [e3Text - Overview](#)
- [e3Symbol.GetReferenceTextExtentSingleLine\(\)](#)

- GetTextExtent()
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.GetType()**

## Syntax

**Integer** GetType()

## Description

Gets the text item's text type.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
> 0	Success	Text type supplied
0	Failure	Error occurred

## Remarks

See [Text Types](#) for a list of predefined text type values and descriptions.

This function is identical in behaviour to [GetTypeId\(\)](#)

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>**.series project and selecting texts on a sheet.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
```

## SetAsMaster - e3Symbol

```
Set text = job.CreateTextObject()

textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then
    For textIndex = 1 To textCount                       'loop through each text

        textId = text.SetId( textIds( textIndex ) )
        result = text.GetType()

        If result = 0 Then
            message = "Text " & textId & ": unknown text type"
        Else
            message = "Text " & textId & ": text type = " & result
        End If
        e3Application.PutInfo 0, message                 'output result of operation

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2010-9.10.

---

## See Also

- [e3Text - Overview](#)
  - [Text Types](#)
  - [GetTypeId\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.GetTypeId()**

## Syntax

**Integer** GetTypeId()

## Description

Gets the text item's text type.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
> 0	Success	Text type supplied
0	Failure	Error occurred

## Remarks

See [Text Types](#) for a list of predefined text type values and descriptions.

This function is identical in behaviour to [GetType\(\)](#)

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>**.series project and selecting texts on a sheet.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
```

## SetAsMaster - e3Symbol

```
Set text = job.CreateTextObject()

textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then
    For textIndex = 1 To textCount                       'loop through each text

        textId = text.SetId( textIds( textIndex ) )
        result = text.GetTypeId()

        If result = 0 Then
            message = "Text " & textId & ": unknown text type"
        Else
            message = "Text " & textId & ": text type = " & result
        End If
        e3Application.PutInfo 0, message                 'output result of operation

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2010-9.10.

---

## See Also

- [e3Text - Overview](#)
  - [Text Types](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.GetVisibility()**

## Syntax

**Integer** GetVisibility()

## Description

Gets the text item's visibility status.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
1	Success	Text visibility status is shown
0	Inconclusive	Text visibility status is hidden or an error occurred

## Remarks

Due caution is recommended on relying on the return value of 0 visibility status is hidden since this also could mean an error has occurred.

The function returns 1 if the display level is not visible but would have been visible if the display level was turned on.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting symbols on a sheet.

Visual Basic Script



## SetAsMaster - e3Symbol

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set symbol = job.CreateSymbolObject()
Set text = job.CreateTextObject()

symbolCount = job.GetSelectedSymbolIds( symbolIds )           'get selected symbols
If symbolCount > 0 Then
    For symbolIndex = 1 To symbolCount

        symbolId = symbol.SetId( symbolIds( symbolIndex ) )
        textCount = symbol.GetTextIds( textIds )             'get texts of symbol

        If textCount > 0 Then
            e3Application.PutInfo 0, "Symbol: " & symbolId & " ; Name: " & symbol.GetName

            For textIndex = 1 To textCount

                textId = text.SetId( textIds( textIndex ) )
                result = text.GetVisibility()
                If result = 0 Then
                    message = "    Text: " & textId & " is hidden"
                Else
                    message = "    Text: " & textId & " is shown"
                End If
                e3Application.PutInfo 0, message               'output result of operation

            Next
        End If
    Next
End If

Set text = Nothing
Set symbol = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2010-9.10.

---

## See Also

- [e3Text - Overview](#)
  - [SetVisibility\(\)](#)
-

## SetAsMaster - e3Symbol

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.GetWidth()**

## Syntax

Double GetWidth()

## Description

Gets the text item's box width.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
> 0.0	Success	Width of the text supplied
0.0	Failure	Error occurred

## Remarks

The return value is rounded to 2 decimal places and is in project measurement units.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting texts on a sheet.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()
```

## SetAsMaster - e3Symbol

```
textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then
    For textIndex = 1 To textCount                       'loop through each text

        textId = text.SetId( textIds( textIndex ) )
        result = text.GetWidth()

        If result = 0 Then
            message = "Text " & textId & ": error occurred getting text width"
        Else
            message = "Text " & textId & ": text width = " & result
        End If
        e3Application.PutInfo 0, message                 'output result of operation

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2010-9.10.

---

## See Also

- [e3Text - Overview](#)
  - [GetBox\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.HasAttribute( *name* )**

## Syntax

**Integer** HasAttribute( [in]**String** *name* )

## Description

Gets the number of the text item's specified attributes.

## Parameters

Type	Parameter	Description
[in] <b>String</b>	<i>name</i>	Name of the attribute

## Return Values

Value	Status	Description
> 0	Success	Number of attributes found
0	Inconclusive	No attribute found or an error occurred

## Remarks

*name* should be the name of an existing attribute.

Due caution is recommended on relying on the return value of 0 meaning no attributes found since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting texts using the given attribute.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
```

## SetAsMaster - e3Symbol

```
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()

textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then

    result = 0
    attributeName = "Example"           'attribute should exist.

    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )
        result = text.HasAttribute( attributeName )

        If result = 0 Then
            message = "Text: " & textId & "; Attribute: " & attributeName & " = could not
        Else
            message = "Text: " & textId & "; Attribute: " & attributeName & " = found " &
        End If
        e3Application.PutInfo 0, message           'output result of operation

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

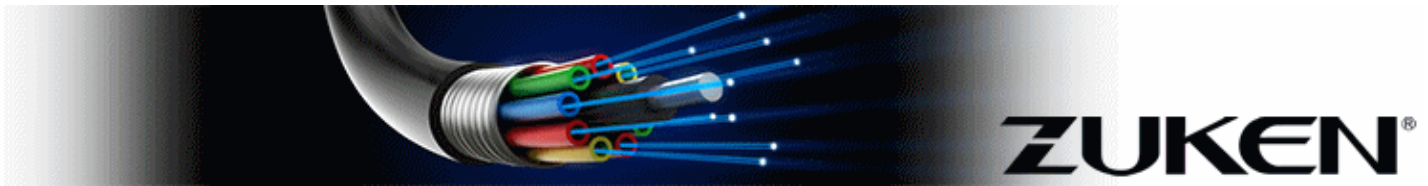
Introduced in v2012-11.00.

---

## See Also

- [e3Text - Overview](#)
  - [GetAttributeCount\(\)](#)
  - [GetAttributeValue\(\)](#)
  - [SetAttributeValue\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.IsOffline()**

## Syntax

**Integer** IsOffline()

## Description

Gets the text's accessibility status by an **E<sup>3</sup>.multiuser** client.

**Function has been deprecated.** It is no longer necessary to use this function.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
1	Success	Text is inaccessible from an <b>E<sup>3</sup>.multiuser</b> client
0	Inconclusive	Text is accessible from an <b>E<sup>3</sup>.multiuser</b> client or an error occurred

## Remarks

The functionality leading to the necessity to call this function no longer exists as of v2016-17.00. The function only remains for backward compatibility.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting text items on a sheet.

Visual Basic Script

## SetAsMaster - e3Symbol

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()

textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then
    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )
        result = text.IsOffline()

        If result = 0 Then
            message = "Text: " & textId & "; is accessible via multi-user"
        Else
            message = "Text: " & textId & "; is inaccessible via multi-user"
        End If
        e3Application.PutInfo 0, message                 'output result of operation

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2010-9.10.

Deprecated in v2016-17.00.

---

## See Also

- [e3Text - Overview](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).





v2022-23.00

**e3Text.IsRedlined()**

## Syntax

*Integer* IsRedlined()

## Description

Gets the text's redlined status.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
1	Success	redlined status is on
0	Success	redlined status is off

## Remarks

If the text redlined status is on, it is identified as an **E<sup>3</sup>.redliner** object.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting text items on a sheet.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()
```

## SetAsMaster - e3Symbol

```
textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then

    result = 0

    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )

        result = text.IsRedlined()
        If result = 1 Then           'invert the redliner status
            text.SetRedlined False
        Else
            text.SetRedlined True
        End If
        result = text.IsRedlined()
        e3Application.PutInfo 0, "Text: " & textId & ": " & text.GetText & ": redliner sta

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2010-9.10.

---

## See Also

- [e3Text - Overview](#)
  - [SetRedlined\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.SendToBackground()**

## Syntax

**Integer** SendToBackground()

## Description

Moves the text item to the background.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
0	Success	Text item moved to the background
-1	Failure	No project open
-2	Failure	No text item set
-3	Failure	Text item not placed
-4	Failure	Error occurred
-5	Failure	Error occurred

## Remarks

A valid text id value must be assigned using SetId() otherwise 0 is returned.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting text items on a sheet.

Visual Basic Script

Examples

## SetAsMaster - e3Symbol

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()

textCount= job.GetSelectedTextIds( textIds )
If textCount > 0 Then
    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )
        result = text.SendToBackground()

        Select Case result
        Case 0
            message = "Text " & textId & " moved to background"
        Case -1
            message = "No project open"
        Case -2
            message = "Text item not set"
        Case -3
            message = "Text " & textId & " not placed"
        Case -4
            message = "Error occurred moving text " & textId & " to background"
        Case -5
            message = "Error occurred moving text " & textId & " to background"
        End Select
        e3Application.PutInfo 0, message           'output result of operation

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2020-21.12.

---

## See Also

- [e3Text - Overview](#)
  - [SendToForeground\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.SendToForeground()**

## Syntax

**Integer** SendToForeground()

## Description

Moves the text item to the foreground.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
0	Success	Text item moved to the foreground
-1	Failure	No project open
-2	Failure	No text item set
-3	Failure	Text item not placed
-4	Failure	Error occurred
-5	Failure	Error occurred

## Remarks

A valid text id value must be assigned using SetId() otherwise 0 is returned.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting text items on a sheet.

Visual Basic Script

Examples

## SetAsMaster - e3Symbol

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()

textCount= job.GetSelectedTextIds( textIds )
If textCount > 0 Then
    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )
        result = text.SendToForeground()

        Select Case result
        Case 0
            message = "Text " & textId & " moved to foreground"
        Case -1
            message = "No project open"
        Case -2
            message = "Text item not set"
        Case -3
            message = "Text " & textId & " not placed"
        Case -4
            message = "Error occurred moving text " & textId & " to foreground"
        Case -5
            message = "Error occurred moving text " & textId & " to foreground"
        End Select
        e3Application.PutInfo 0, message           'output result of operation

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2020-21.12.

---

## See Also

- [e3Text - Overview](#)
  - [SendToBackground\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.SetAlignment( *newval* )**

## Syntax

**Integer** SetAlignment( [in]**Integer** *newval* )

## Description

Sets the text's alignment value.

## Parameters

Type	Parameter	Description
[in] <b>Integer</b>	<i>newval</i>	Alignment value to be set

## Return Values

Value	Status	Description
0	Success	Alignment value set
1	Failure	Error occurred

## Remarks

*newval* can be one of the following values:

Value	Description
1	Left align
2	Center
3	Right align

From v2021-22.00 the return values are exclusively used for alignment. SetRotation() can be used to rotate the text.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting sheets in the sheet project tree.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set sheet = job.CreateSheetObject()
Set text = job.CreateTextObject()

Const ALIGN_CENTER = 2

sheetCount = job.GetTreeSelectedSheetIds( sheetIds )           'get all selected sheets from
If sheetCount > 0 Then

    sheet.SetId( sheetIds( 1 ) )           'use first selected sheet
    textCount = sheet.GetTextIds( textIds )           'get all texts on the sheet

    If textCount > 0 Then
        For index = 1 To textCount           'loop through all texts

            textId = text.SetId( textIds( index ) )
            result = text.SetAlignment( ALIGN_CENTER )

            If result = 0 Then           'output result of operation
                message = "Text " & textId & ": aligned to center"
            Else
                message = "Text " & textId & ": alignment operation failed"
            End If
            e3Application.PutInfo 0, message           'output result of operation

        Next
    End If
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2010-9.10.

Modified in v2021-22.00.

---



## See Also

- [e3Text - Overview](#)
  - [GetAlignment\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.SetAttributeValue( *name*, *value* )**

## Syntax

**Integer** SetAttributeValue( [in]**String** *name*, [in]**String** *value* )

## Description

Sets the text item's specified attribute value.

## Parameters

Type	Parameter	Description
[in] <b>String</b>	<i>name</i>	Name of the attribute
[in] <b>String</b>	<i>value</i>	Value of the attribute

## Return Values

Value	Status	Description
> 0	Success	Identifier of attribute
0	Failure	Error occurred

## Remarks

*name* must be the name of an existing attribute.

*value* maximum length is 252 characters.

A valid text id value must be assigned using SetId() otherwise 0 is returned.

Since v2014-1400 the value of 0 is returned if the text is locked or the attribute is not defined as changeable.

Adding a new attribute to the text cannot be done using this function.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting texts using the given attribute.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()

textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then

    result = 0
    attributeName = "Example"           'attribute should exist.
    attributeValue = "Legatus nec violatur, nec laeditur"

    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )
        gotAttribute = text.HasAttribute( attributeName )
        If gotAttribute = 1 Then

            result = text.SetAttributeValue( attributeName, attributeValue )

            If result = 0 Then
                message = "Text: " & textId & "; Attribute: " & attributeName & " value no"
            Else
                message = "Text: " & textId & "; Attribute: " & attributeName & " value se"
            End If
            e3Application.PutInfo 0, message           'output result of operation

        End If
    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2012-11.00.

Modified in v2014-14.00.

---

## See Also

- [e3Text - Overview](#)
  - [GetAttributeValue\(\)](#)
  - [HasAttribute\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.SetBallooning( *onoff*, *type* )**

## Syntax

**Integer** SetBallooning( [in]**Boolean** *onoff*, [in]**Integer** *type* )

## Description

Sets the text's ballooning value.

## Parameters

Type	Parameter	Description
		Indicating if the value should be added or removed
[in] <b>Boolean</b>	<i>onoff</i>	If <b>True</b> , the value is added
		If <b>False</b> , the value is removed
[in] <b>Integer</b>	<i>type</i>	New ballooning value

## Return Values

Value	Status	Description
> 0	Success	Previous ballooning value
		See <a href="#">Ballooning</a> for possible values
0	Inconclusive	Previously no ballooning value set or an error occurred

## Remarks

Due caution is recommended on relying on the return value of 0 meaning previously no ballooning since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting texts on a sheet.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()

textCount = job.GetSelectedTextIds( textIds )      'get selected texts
If textCount > 0 Then

    ballooning = 24      ' Ellipse and line to owner

    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )
        result = text.SetBallooning( True, ballooning )

        Select Case result
        Case 0
            message = "No ballooning or error occurred"
        Case 1
            message = "Circle"
        Case 2
            message = "Oval"
        Case 4
            message = "Rectangle"
        Case 8
            message = "Ellipse"
        Case 16
            message = "Line To owner"
        Case 17
            message = "Circle and line to owner"
        Case 18
            message = "Oval and line to owner"
        Case 20
            message = "Rectangle and line to owner"
        Case 24
            message = "Ellipse and line to owner"
        Case 32
            message = "Horizontal line on centre"
        Case 64
            message = "Horizontal line on bottom"
        Case 128
            message = "Horizontal line on top"

```

## SetAsMaster - e3Symbol

```
End Select
e3Application.PutInfo 0, "Text: " & textId & ": Previously = " & message

Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2010-9.10.

---

## See Also

- [e3Text - Overview](#)
  - [Ballooning](#)
  - [GetBallooning\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.SetBox( *xsize*, *ysize* )**

## Syntax

**Integer** SetBox( [in]**Double** *xsize*, [in]**Double** *ysize* )

## Description

Sets the text item's box width and height.

## Parameters

Type	Parameter	Description
[in] <b>Double</b>	<i>xsize</i>	Width of box
[in] <b>Double</b>	<i>ysize</i>	Height of box

## Return Values

Value	Status	Description
1	Success	Box values applied
0	Failure	Error occurred

## Remarks

The box represents a rectangle containing the text.

Lines of text are broken at the defined box width preferably after a blank, a tab or one of the following characters: . ? ! ; : -

The line break is made in the middle of the word if none of these characters is found within the defined width.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting texts on a sheet.



## Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()

Dim width : width = 25
Dim height : height = 50

textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then
    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )
        result = text.SetBox( width, height )

        If result = 0 Then
            message = "Text: " & textId & " error occurred setting box"
        Else
            message = "Text: " & textId & "; new width = " & width & "; new height = " & height
        End If
        e3Application.PutInfo 0, message                 'output result of operation

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2010-9.10.

---

## See Also

- [e3Text - Overview](#)
  - [GetBox\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.SetColour( *newval* )**

## Syntax

**Integer** SetColour( [in]**Integer** *newval* )

## Description

Sets the text item's text color.

## Parameters

Type	Parameter	Description
		Color value to apply
[in] <b>integer</b>	<i>newval</i>	See <a href="#">Colors</a> for possible values

## Return Values

Value	Status	Description
1	Success	Color value applied
0	Failure	Error occurred

## Remarks

The text color can be retrieved using [GetColour\(\)](#).

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting texts on a sheet.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
```

## SetAsMaster - e3Symbol

```
Set text = job.CreateTextObject()

textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then

    color = 128           'new color to apply; in this case 0, 218, 85 (greenish)

    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )
        result = text.SetColour( color )

        If result = 0 Then
            message = "Text: " & textId & " color could not be set"
        Else
            message = "Text: " & textId & " color set to greenish hue"
        End If
        e3Application.PutInfo 0, message           'output result of operation

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2010-9.10.

---

## See Also

- [e3Text - Overview](#)
  - [Colors](#)
  - [GetColour\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.SetFontName( *newname* )**

## Syntax

**Integer** SetFontName( [in]**String** *newname* )

## Description

Sets the text item's font.

## Parameters

Type	Parameter	Description
[in] <b>String</b>	<i>newname</i>	Name of font to apply

## Return Values

Value	Status	Description
1	Success	Font applied
0	Failure	Error occurred

## Remarks

A valid text id value must be assigned using SetId() otherwise 0 is returned.

*newname* is case insensitive.

The default font name for all free texts is provided by a global setting that can be accessed using e3Job.GetGraphTextFontName() and e3Job.SetGraphTextFontName(). Once the font name of a text item is changed, it is no longer subject to the global setting.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting texts on a sheet.

## Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()

textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then

    fontName = "Parchment"                             'font should already be installed on the system

    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )
        result = text.SetFontName( fontName )
        If result = 0 Then
            message = "Text: " & textId & " font name: " & fontName & " could not be assigned"
        Else
            message = "Text: " & textId & " font name is now " & fontName
        End If
        e3Application.PutInfo 0, message                 'output result of operation

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2010-9.10.

## See Also

- [e3Text - Overview](#)
- [e3Job.GetGraphTextFontName\(\)](#)
- [e3Job.SetGraphTextFontName\(\)](#)
- [GetFontName\(\)](#)

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.SetGID( *gid* )**

## Syntax

*String* SetGID( [in]*String* *gid* )

## Description

Sets a text item as the current item.

## Parameters

Type	Parameter	Description
[in] <i>String</i>	<i>gid</i>	Global identifier value of a text item

## Return Values

Value	Status	Description
"<GID>"	Success	Global identifier of the current text item
"<Empty>"	Failure	No text item

## Remarks

Global identifiers (GIDs) are used for item identification in multiuser projects.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting text items on a sheet.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()
```

## SetAsMaster - e3Symbol

```
Set gidList = CreateObject( "System.Collections.ArrayList" )

textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then
    For textIndex = 1 To textCount

        text.SetId textIds( textIndex )
        gidId = text.GetGID()
        gidList.Add gidId

    Next
End If

For Each gidId in gidList
    result = text.SetGID( gidId )
    If Len( "" & result ) = 0 Then
        message = "No text item is set"
    Else
        textId = text.GetId()
        message = "Text " & textId & " has been set using GID " & gidId
    End If
    e3Application.PutInfo 0, message                    'output result of operation
Next

Set gidList = Nothing
Set text = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2022-23.00.

---

## See Also

- [e3Text - Overview](#)
  - [GetGID\(\)](#)
  - [SetId\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.SetHeight( *newval* )**

## Syntax

**Integer** SetHeight( [in]**Double** *newval* )

## Description

Sets the text item's font height.

## Parameters

Type	Parameter	Description
[in] <b>Double</b>	<i>newval</i>	Value of font height to apply

## Return Values

Value	Status	Description
1	Success	Font height applied
0	Failure	Error occurred

## Remarks

The text's font height should be in project measurement units.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting texts on a sheet.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()
```



## SetAsMaster - e3Symbol

```
textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then

    fontHeight = 8.0           'font size in project measurement units - millimetres are assumed

    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )
        result = text.SetHeight( fontHeight )
        If result = 0 Then
            message = "Text: " & textId & " font height could not be applied"
        Else
            message = "Text: " & textId & " new font height is " & fontHeight
        End If
        e3Application.PutInfo 0, message           'output result of operation

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2010-9.10.

---

## See Also

- [e3Text - Overview](#)
  - [GetHeight\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.SetHyperlinkAddress( *newtext* )**

## Syntax

**Integer** SetHyperlinkAddress( [in]**String** *newtext* )

## Description

Sets the text item's hyperlink address.

## Parameters

Type	Parameter	Description
[in] <b>String</b>	<i>newtext</i>	Hyperlink address to apply

## Return Values

Value	Status	Description
1	Success	Hyperlink address applied
0	Failure	Error occurred

## Remarks

The function does not work for texts linked with the **Hyperlink** attribute.

*newtext* value's syntax should adhere to the following format:

Value	Description
"<empty>"	No hyperlink address
"http:<target>"	Internet link using Hypertext Transfer Protocol (HTTP)
"https:<target>"	Internet link using Hypertext Transfer Protocol Secured (HTTPS)
"ftp:<target>"	Internet link using File Transfer Protocol (FTP)
"ftps:<target>"	Internet link using File Transfer Protocol Secured (FTPS)

"sftp:<target>"	Internet link using SSH File Transfer Protocol (SFTP)
"file:<target>"	File link using a system path
"gid:<target>"	Sheet, symbol or document using a Global Identifier (GID)

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>**.series project and selecting texts on a sheet.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()

Dim hyperlinkAddress : hyperlinkAddress = "https://www.zuken.com"

textCount = job.GetSelectedTextIds( textIds )      'get selected texts
If textCount > 0 Then
    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )
        result = text.SetHyperlinkAddress( hyperlinkAddress )
        If Len( "" & result ) = 0 Then
            message = "Text: " & textId & " hyperlink address is empty or could not be app
        Else
            message = "Text: " & textId & " new hyperlink address is " & result
        End If
        e3Application.PutInfo 0, message            'output result of operation

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2010-9.10.

## See Also

- [e3Text - Overview](#)
- [GetHyperlinkAddress\(\)](#)

## SetAsMaster - e3Symbol

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.SetId( *id* )**

## Syntax

*Integer* SetId( [in]*Integer id* )

## Description

Sets a text as the current item.

## Parameters

Type	Parameter	Description
[in] <i>Integer</i>	<i>id</i>	Unique value identifying a text item

## Return Values

Value	Status	Description
> 0	Success	Current text item identifier
0	Failure	Error occurred

## Remarks

*id* will remain the current text item until it is deleted or replaced.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting texts on a sheet.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()
```

## SetAsMaster - e3Symbol

```
textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then
    For textIndex = 1 To textCount                       'loop through all selected texts

        result = text.SetId( textIds( textIndex ) )
        If result = 0 Then
            message = "Error setting text " & textIds( textIndex )
        Else
            message = "Text item " & result & " has been set"
        End If
        e3Application.PutInfo 0, message                 'output result of operation

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2010-9.10.

---

## See Also

- [e3Text - Overview](#)
  - [e3Graph.CreateText\(\)](#)
  - [GetId\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.SetLanguageId( *newval* )**

## Syntax

*Integer* SetLanguageId( [in]*Integer newval* )

## Description

Sets the text item's translation identifier value.

## Parameters

Type	Parameter	Description
[in] <i>Integer</i>	<i>newval</i>	Translation identifier to apply

## Return Values

Value	Status	Description
0	Inconclusive	No other value is returned

## Remarks

It is recommended to rely on GetLanguageId() to verify *newval* is in use.

Valid translation identifiers are dependent on the project's language database. The identifier refers to the indexed position of the translation text within the table.

A *newval* value of 0 will set text translation not to be used.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>**.series project and selecting texts on a sheet.

Visual Basic Script

## SetAsMaster - e3Symbol

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()

textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then

    languageId = 1           'translation text identifier

    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )

        text.SetLanguageId languageId

        result = text.GetLanguageId()
        If result = 0 Then
            e3Application.PutInfo 0, "Text: " & textId & " no language found."
        Else
            e3Application.PutInfo 0, "Text: " & textId & "; language = " & result
        End If
    Next

End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2010-9.10.

---

## See Also

- [e3Text - Overview](#)
  - [GetLanguageId\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).





v2022-23.00

**e3Text.SetLevel( *newval* )**

## Syntax

**Integer** SetLevel( [in]**Integer** *newval* )

## Description

Sets the text item's display level value.

## Parameters

Type	Parameter	Description
[in] <b>Integer</b>	<i>newval</i>	Display level value to apply

## Return Values

Value	Status	Description
1	Success	Display level value applied
0	Failure	Error occurred

## Remarks

GetVisibility() and SetVisibility() can be used to hide or show the text item independent of display level values.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting texts on a sheet.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()
```

```

textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then

    level = 12

    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )
        result = text.SetLevel( level )
        If result = 0 Then
            message = "Text: " & textId & " display level could not be applied"
        Else
            message = "Text: " & textId & " display level is now " & level
        End If
        e3Application.PutInfo 0, message                'output result of operation

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2010-9.10.

---

## See Also

- [e3Text - Overview](#)
  - [GetLevel\(\)](#)
  - [GetVisibility\(\)](#)
  - [SetVisibility\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.SetLinearMeasureWithoutUnit( *newval* )**

## Syntax

**Boolean** SetLinearMeasureWithoutUnit( [in]**Boolean** *newval* )

## Description

Sets the text item's flag value determining if the length measurement unit is displayed along with the value.

## Parameters

Type	Parameter	Description
		Value for the hiding the length measure unit to apply
[in] <b>Boolean</b>	<i>newval</i>	<b>True</b> hides the measurement unit
		<b>False</b> shows the measurement unit

## Return Values

Value	Status	Description
<b>True</b>	Success	Value is applied
<b>False</b>	Failure	Error occurred

## Remarks

**False** will be returned for any text items of text types not displaying length measurements.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting symbols on a sheet.

## Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set symbol = job.CreateSymbolObject()
Set text = job.CreateTextObject()

symbolCount = job.GetSelectedSymbolIds( symbolIds )           'get selected symbols
If symbolCount > 0 Then
    For symbolIndex = 1 To symbolCount

        symbolId = symbol.SetId( symbolIds( symbolIndex ) )
        textCount = symbol.GetTextIds( textIds )             'get texts of symbol

        If textCount > 0 Then
            e3Application.PutInfo 0, "Symbol: " & symbolId & " ; Name: " & symbol.GetName

            For textIndex = 1 To textCount
                textId = text.SetId( textIds( textIndex ) )
                result = text.SetLinearMeasureWithoutUnit( True )

                If result = False Then
                    message = "    Text: " & textId & " linear measurement unit could not
                Else
                    message = "    Text: " & textId & " linear measurement unit hidden"
                End If
                e3Application.PutInfo 0, message                'output result of operation
            Next
        End If
    Next
End If

Set text = Nothing
Set symbol = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2017-17.70.

## See Also

- [e3Text - Overview](#)
- [GetLinearMeasureWithoutUnit\(\)](#)

## SetAsMaster - e3Symbol

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.SetLocking( *newval* )**

## Syntax

**Boolean** SetLocking( [in]**Boolean** *newval* )

## Description

Gets the text item's flag value determining if the text position is locked.

## Parameters

Type	Parameter	Description
		Position locking value to apply
[in] <b>Boolean</b>	<i>newval</i>	<b>True</b> will lock the text position <b>False</b> will unlock the text position

## Return Values

Value	Status	Description
<b>True</b>	Success	Text position locking value is applied
<b>False</b>	Failure	Error occurred

## Remarks

Locking the text position will stop the text from being moved or resized.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting texts on a sheet.

Visual Basic Script

## SetAsMaster - e3Symbol

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()

textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then
    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )
        result = text.SetLocking( True )

        If result = False Then
            message = "Text: " & textId & " unable to lock text position"
        Else
            message = "Text: " & textId & " position is now locked"
        End If
        e3Application.PutInfo 0, message                 'output result of operation

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2010-9.10.

---

## See Also

- [e3Text - Overview](#)
  - [GetLocking\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.SetMode( *newval* )**

## Syntax

**Integer** SetMode( [in]**Integer** *newval* )

## Description

Sets the text item's text ratio.

## Parameters

Type	Parameter	Description
[in] <b>Integer</b>	<i>newval</i>	Text ratio value to apply

## Return Values

Value	Status	Description
1	Success	Text ratio value applied
0	Failure	Error occurred

## Remarks

*newval* can be one of the following values:

Value	Description
1	Normal
2	Narrow
3	Wide

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>**.series project and selecting texts on a sheet.

Visual Basic Script

Examples



## SetAsMaster - e3Symbol

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()

textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then

    ratio = 3           'wide text ratio

    For textIndex = 1 To textCount
        textId = text.SetId( textIds( textIndex ) )
        result = text.SetMode( ratio )

        If result = 0 Then
            message = "    Text: " & textId & " unable to set text ratio"
        Else
            message = "    Text: " & textId & " ratio set to wide"
        End If
        e3Application.PutInfo 0, message           'output result of operation
    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2010-9.10.

---

## See Also

- [e3Text - Overview](#)
  - [GetMode\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.SetOptionExpressions( *expressions* )**

## Syntax

**Integer** SetOptionExpressions( [in]**String Array** *expressions* )

## Description

Replaces the text item's option expressions.

## Parameters

Type	Parameter	Description
[in] <b>String Array</b>	<i>expressions</i>	Array of strings of all option names, boolean expressions or alias names of boolean expressions

## Return Values

Value	Status	Description
> 0	Success	Number of items in <i>expressions</i> assigned
0	Inconclusive	No assigned option expressions are supplied or an error occurred

## Remarks

Variant instances are currently available only for devices and wires.

All expressions already assigned to the text item are replaced by those in the *expressions* array.

*expressions* is a 1-based array.

The option expressions in the *expressions* array must be contained in the project.

Due caution is recommended on relying on the return value of 0 meaning no items in *expressions* were assigned since this also could mean an error has occurred. This could be the case if the intention is to remove all existing assigned option expressions by using an empty *expressions* array in the function call.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project containing options and selecting texts.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()

Dim expressions(2)          'array with some option names
expressions(0) = "Option1"
expressions(1) = "Option2"

textCount = job.GetSelectedTextIds ( textIds )          'get selected texts
If textCount > 0 Then
    For textIndex = 1 To textCount          'loop through each text

        textId = text.SetId( textIds( textIndex ) )

        result = text.SetOptionExpressions( expressions )          'replace the text's options
        If result = 0 Then
            e3Application.PutInfo 0, "Text: " & textId & "; No assigned option expressions"
        Else
            expressionCount = text.GetAssignedOptionExpressionsEx( optionExpressions )
            If expressionCount > 0 Then

                e3Application.PutInfo 0, "Text: " & textId & "; assigned option expressions"
                For expressionIndex = 1 To expressionCount
                    e3Application.PutInfo 0, "    " & optionExpressions( expressionIndex )
                Next
            Else
                e3Application.PutInfo 0, "Text: " & textId & "; No assigned option expressions"
            End If
        End If

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2010-9.12.

---

## See Also

- [e3Text - Overview](#)
  - [GetAssignedOptionExpressionsEx\(\)](#)
  - [GetAssignedOptionIds\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.SetPictogram( *newval* )**

## Syntax

**Boolean** SetPictogram( [in]**Boolean** *newval* )

## Description

Sets the text item's flag value determining if the text is displayed in the pictogram language.

## Parameters

Type	Parameter	Description
		Text pictogram value to apply
[in] <b>Boolean</b>	<i>newval</i>	<b>True</b> allows the text to use pictograms <b>False</b> disallows the text to use pictograms

## Return Values

Value	Status	Description
<b>True</b>	Success	Text pictogram value applied
<b>False</b>	Failure	Error occurred

## Remarks

Changing the pictogram value will have no visual effect unless the text item is using a translated text employing a pictogram.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project using pictograms and selecting texts on a sheet.

## Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()

textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then
    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )
        result = text.SetPictogram( True )             'use pictogram

        If result = False Then
            message = "Text: " & textId & " pictogram useage could not be applied"
        Else
            message = "Text: " & textId & " pictogram usage applied"
        End If
        e3Application.PutInfo 0, message                'output result of operation

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2017-17.70.

## See Also

- [e3Text - Overview](#)
- [GetPictogram\(\)](#)

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

SetRedlined( *onoff* )

## Syntax

**Integer** SetRedlined( [in][optional]**Boolean** *onoff* )

## Description

Sets the text's redlined status.

## Parameters

Type	Parameter	Description
		Redlined status value to apply
		If <b>True</b> , the redlined status is set to on
[in][optional] <b>Boolean</b>	<i>onoff</i>	If <b>False</b> , the redlined status is set to off
		Default value is <b>True</b>

## Return Values

Value	Status	Description
1	Success	Previous redlined status was on
0	Inconclusive	Previous redlined status was off or an error occurred

## Remarks

If the text redlined status is set to on it is identified as an **E<sup>3</sup>.redliner** object.

Due caution is recommended on relying on the return value of 0 meaning the previous redlined status is off since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting text items on a sheet.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()

textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then

    result = 0

    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )

        result = text.IsRedlined()
        If result = 1 Then           'invert the redliner status
            text.SetRedlined False
        Else
            text.SetRedlined True
        End If
        result = text.IsRedlined()
        e3Application.PutInfo 0, "Text: " & textId & ": " & text.GetText & ": redliner sta

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in 2017-18.00.

## See Also

- [e3Text - Overview](#)
- [IsRedlined\(\)](#)



## SetAsMaster - e3Symbol

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.SetRotation( *rotation* )**

## Syntax

**Double** SetRotation( [in]**Double** *rotation* )

## Description

Sets the text item's rotation value.

## Parameters

Type	Parameter	Description
[in] <b>Double</b>	<i>rotation</i>	Rotation value in degrees to apply

## Return Values

Value	Status	Description
> 0.0..< 360.0	Success	Previous text rotation value in degrees supplied
0.0	Inconclusive	Previous text rotation value or an error occurred

## Remarks

The angle value of *rotation* expresses absolute counterclockwise rotation.

Due caution is recommended on relying on the return value of 0.0 meaning the previous text rotation value is 0.0 since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting texts on a sheet.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
```

## SetAsMaster - e3Symbol

```
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()

textCount = job.GetSelectedTextIds( textIds )      'get selected texts
If textCount > 0 Then

    rotation = 36.0      '36 degrees

    For textIndex = 1 To textCount
        textId = text.SetId( textIds( textIndex ) )
        result = text.SetRotation( rotation )

        e3Application.PutInfo 0, "Text: " & textId & " old rotation is " & result & "; new

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2010-9.10.

---

## See Also

- [e3Text - Overview](#)
  - [GetRotation\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.SetSchemaLocation( x, y )**

## Syntax

**Integer** SetSchemaLocation( [in]**Double** x, [in]**Double** y )

## Description

Sets the text item's position.

## Parameters

Type	Parameter	Description
[in] <b>Double</b>	x	Placement position on the x-axis
[in] <b>Double</b>	y	Placement position on the y-axis

## Return Values

Value	Status	Description
> 0	Success	Identifier of the text item's sheet
0	Failure	Error occurred

## Remarks

The display will not be updated until a refresh is executed on the sheet following successful execution of the function.

x and y are in project measurement units.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting texts on a sheet.

Visual Basic Script

## SetAsMaster - e3Symbol

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()
Set sheet = job.CreateSheetObject()

textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then
    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )

        sheetId = text.GetSchemaLocation( x, y, grid )
        If sheetId = 0 Then
            e3Application.PutInfo 0, "Text: " & textId & " error getting schema location"
        Else
            sheet.SetId sheetId
            e3Application.PutInfo 0, "Text: " & textId & ":"
            e3Application.PutInfo 0, "    Sheet: " & sheet.GetName
            e3Application.PutInfo 0, "    x: " & x
            e3Application.PutInfo 0, "    y: " & y
            e3Application.PutInfo 0, "    grid: " & grid
            e3Application.PutInfo 0, "    column: " & column
            e3Application.PutInfo 0, "    row: " & row

            x = x + 20
            y = y + 20
            result = text.SetSchemaLocation( x, y )       'move text on sheet

            If result = 0 Then
                e3Application.PutInfo 0, "Text: " & textId & " error setting position"
            Else
                e3Application.PutInfo 0, "Text: " & textId & ":"
                e3Application.PutInfo 0, "    New x: " & x
                e3Application.PutInfo 0, "    New y: " & y
            End If
        End If
    End If

Next
End If

Set sheet = Nothing
Set text = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2010-9.10.

---

## See Also

- [e3Text - Overview](#)
  - [GetSchemaLocation](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.SetSingleLine( *newval* )**

## Syntax

**Integer** SetSingleLine( [in]**Integer** *newval* )

## Description

Sets the text item's flag value determining if multi-line text is displayed on a single line.

## Parameters

Type	Parameter	Description
		Single line value to apply
[in] <b>Integer</b>	<i>newval</i>	If 1, mutli-line text is displayed on a single line
		If 0, mutli-line text is displayed on a multiple lines

## Return Values

Value	Status	Description
1	Success	Single line value applied
0	Failed	Error occurred

## Remarks

All line breaks in the text item are replaced by a blank space when displayed in single line mode.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project selecting multi-line texts on a sheet.

## Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()

textCount = job.GetSelectedTextIds( textIds )           'get selected texts
If textCount > 0 Then
    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )
        result = text.SetSingleLine( 1 )

        If result = 0 Then
            message = "Text: " & textId & " unable to set to single line"
        Else
            message = "Text: " & textId & " set to single line"
        End If
        e3Application.PutInfo 0, message                 'output result of operation
    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2010-9.10.

## See Also

- [e3Text - Overview](#)
- [GetSingleLine\(\)](#)

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).





v2022-23.00

**e3Text.SetStyle( *newval* )**

## Syntax

*Integer* SetStyle( [in]*Integer* *newval* )

## Description

Sets the text item's font style.

## Parameters

Type	Parameter	Description
[in] <i>Integer</i>	<i>newval</i>	Font style to apply

## Return Values

Value	Status	Description
1	Success	Font style value supplied
0	Failure	No font style value used or an error occurred

## Remarks

*newval* can be a combination of the following values:

Value	Description
0	Regular
1	<b>Bold</b>
2	<i>Italics</i>
4	<u>Underline</u>
8	<del>Strikethrough</del>
16	Opaque

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>**.series project and selecting texts on a sheet.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()

textCount = job.GetSelectedTextIds( textIds )      'get selected texts
If textCount > 0 Then

    fontStyle = 7      'Bold + Italic + Underline

    For textIndex = 1 To textCount
        textId = text.SetId( textIds( textIndex ) )
        result = text.SetStyle( fontStyle )

        If result = 0 Then
            message = "Text " & textId & ": new fontstyle could not be set"
        Else
            message = "Text " & textId & ": bold + italic + underline fontstyle set"
        End If
        e3Application.PutInfo 0, message      'output result of operation

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2010-9.10.

## See Also

- [e3Text - Overview](#)
- [GetStyle\(\)](#)

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.SetText( *newtext* )**

## Syntax

*Integer* SetText( [in]*String* *newtext* )

## Description

Sets the text item's text value.

## Parameters

Type	Parameter	Description
[in] <i>String</i>	<i>newtext</i>	Text value to apply

## Return Values

Value	Status	Description
1	Success	Text value is applied
0	Failure	Error occurred

## Remarks

Setting the text value of **designation of device** (text type 12) is dependent on the **CHANGE\_COMPLETE\_DEVICE** setting. This setting can be changed using e3job.SetSettingValue(). The symbol will be assigned to an existing device or a new device will be created if the **CHANGE\_COMPLETE\_DEVICE** setting value is 0.

## Examples

The best results from the example can be achieved by opening an *E<sup>3</sup>.series* project and selecting texts on a sheet.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
```

## SetAsMaster - e3Symbol

```
Set job = e3Application.CreateJobObject()
Set text = job.CreateTextObject()

Dim textValue : textValue = "&#18; : Legatus nec violatur, nec laeditur"

textCount = job.GetSelectedTextIds( textIds )      'get selected texts
If textCount > 0 Then
    For textIndex = 1 To textCount

        textId = text.SetId( textIds( textIndex ) )
        result = text.SetText( textValue )

        If result = 0 Then
            message = "Text " & textId & ": new text not applied"
        Else
            message = "Text " & textId & ": new text = " & text.GetText
        End If
        e3Application.PutInfo 0, message            'output result of operation

    Next
End If

Set text = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2010-9.10.

---

## See Also

- [e3Text - Overview](#)
  - [GetInternalText\(\)](#)
  - [GetText\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Text.SetVisibility( *newval* )**

## Syntax

**Integer** SetVisibility( [in]**Integer** *newval* )

## Description

Sets the text item's visibility status.

## Parameters

Type	Parameter	Description
		Visibility status value
[in] <b>Integer</b>	<i>newval</i>	If 1, the text is shown
		If 0, the text is hidden

## Return Values

Value	Status	Description
1	Success	Text visibility status value applied
0	Failure	Error occurred

## Remarks

Setting the visibility status to 1 does not guarantee the text will be shown as this is also dependent on the display level being turned on.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting symbols with some invisible texts on a sheet.

Visual Basic Script

Examples

## SetAsMaster - e3Symbol

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set symbol = job.CreateSymbolObject()
Set text = job.CreateTextObject()

symbolCount = job.GetSelectedSymbolIds( symbolIds )           'get selected symbols
If symbolCount > 0 Then
    For symbolIndex = 1 To symbolCount

        symbolId = symbol.SetId( symbolIds( symbolIndex ) )
        textCount = symbol.GetTextIds( textIds )             'get texts of symbol

        If textCount > 0 Then
            e3Application.PutInfo 0, "Symbol: " & symbolId & " ; Name: " & symbol.GetName

            For textIndex = 1 To textCount

                textId = text.SetId( textIds( textIndex ) )
                visible = text.GetVisibility()
                If visible = 0 Then
                    result = text.SetVisibility( True )
                    If result = 0 Then
                        message = "    Text: " & textId & " failed to be shown"
                    Else
                        message = "    Text: " & textId & " is now visible"
                    End If
                    e3Application.PutInfo 0, message           'output result of operation
                End If
            Next
        End If
    Next
End If

Next
End If

Set text = Nothing
Set symbol = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

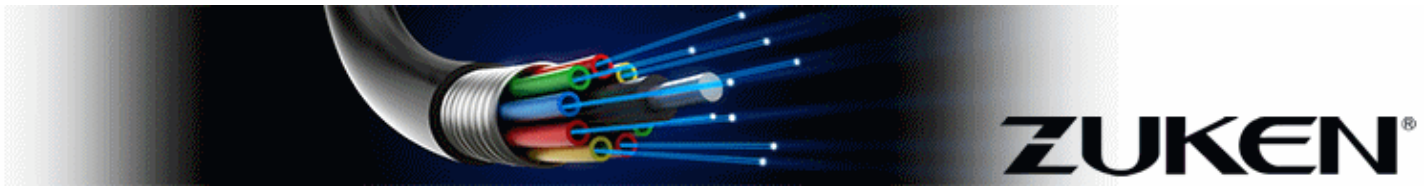
Introduced in v2010-9.10.

---

## See Also

- [e3Text - Overview](#)
  - [GetVisibility\(\)](#)
-

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

## e3Tree

# Description

Encapsulates the functionality for retrieving and modifying information for tree items.

## e3Tree Construction Functions

Function	Description
<u><a href="#">e3Job.CreateTreeObject()</a></u>	Creates an instance of e3Tree

## Item Creation/Destruction Functions

Function	Description
<u><a href="#">Create()</a></u>	Creates a new tree item and places it in the project tree structure
<u><a href="#">Delete()</a></u>	Deletes the current tree item from the project tree structure

## Retrieval Functions

Function	Description
<u><a href="#">GetId()</a></u>	Gets the identifier of the current tree item
<u><a href="#">GetName()</a></u>	Gets the tree item's name
<u><a href="#">GetNames()</a></u>	Gets the project tree item's names in all languages
<u><a href="#">GetSelectedAllDeviceIds()</a></u>	Gets the identifiers of devices of all types selected in the project tree item
<u><a href="#">GetSelectedAllDevicesIdsByFolder()</a></u>	Gets the identifiers of devices in selected folders in the project tree item
<u><a href="#">GetSelectedBlockIds()</a></u>	Gets the identifiers of blocks selected in the project tree item
<u><a href="#">GetSelectedBundleIds()</a></u>	Gets the identifiers of bundles selected in the project tree item
<u><a href="#">GetSelectedBusbarIds()</a></u>	



	Gets the identifiers of busbars selected in the project tree item
<u>GetSelectedCableIds()</u>	Gets the identifiers of cables selected in the project tree item
<u>GetSelectedConnectorIds()</u>	Gets the identifiers of connectors selected in the project tree item
<u>GetSelectedDeviceIds()</u>	Gets the identifiers of devices selected in the project tree item
<u>GetSelectedExternalDocumentIds()</u>	Gets the identifiers of external document items selected in the project tree item
<u>GetSelectedExternalDocumentIdsByFolder()</u>	Gets the identifiers of external documents in selected node structures in the project tree item
<u>GetSelectedPinIds()</u>	Gets the identifiers of pins selected in the project tree item
<u>GetSelectedSheetIds()</u>	Gets the identifiers of sheets selected in the project tree item
<u>GetSelectedSheetIdsByFolder()</u>	Gets the identifiers of sheets in selected node structures in the project tree item
<u>GetSelectedStructureNodeIds()</u>	Gets the identifiers of structure nodes selected in the project tree item
<u>GetSelectedSymbolIds()</u>	Gets the identifiers of symbols and gates selected in the project tree item
<u>GetSelectedTerminalIds()</u>	Gets the identifiers of terminals selected in the project tree item
<u>GetSortingMethod()</u>	Gets sorting method of elements in the tree item
<u>GetTreeType()</u>	Gets a value representing the tree item's subtype
<u>GetVisibleInfoTypes()</u>	<b>Deprecated</b> Gets lists of views and schematic types displayed in the project tree item
<u>GetVisibleInfoTypesEx()</u>	Gets arrays of views, schematic types and formboard sheets displayed in the project tree item
<u>GetVisibleObjectTypes()</u>	Gets an array of the project tree item's visible object types and their display information settings
<u>IsActive()</u>	Indicates whether the tree item is the currently active tree
<u>IsVisible()</u>	Indicates whether the tree item is displayed

## Modification Functions

Function	Description
<u>SetIcon()</u>	Sets the tree item's display icon
<u>SetId()</u>	Sets a tree item as the current item
<u>SetName()</u>	Sets the tree item's name
<u>SetNames()</u>	Sets the tree item's names in all languages
<u>SetSortingMethod()</u>	Sets sorting method of elements in the project tree
<u>SetVisibleInfoTypes()</u>	<b>Deprecated</b> Sets the views and schematic types displayed in the project tree item
<u>SetVisibleInfoTypesEx()</u>	Sets the views, schematic types and formboard sheets displayed in the project tree item
<u>SetVisibleObjectTypes()</u>	Sets the project tree item's visible object types and their display information settings

## Miscellaneous Functions

Function	Description
<u>ViewSignalTree()</u>	Gets a unique value identifying the signal tree item and shows or hides it

## Remarks

The project tree structure is displayed in **E<sup>3</sup>.series** as the Project Tree Window.

The project tree is displayed in **E<sup>3</sup>.series** as a tab in the Project Tree Window.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting a project tree item.

Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set tree = job.CreateTreeObject()

Const ATTRIBUTE_VALUE = "A"

treeId = job.GetActiveTreeID()
If treeId > 0 Then

    tree.SetId treeId
    treeName = tree.GetName()

```

## SetAsMaster - e3Symbol

```
result = tree.GetVisibleObjectTypes( treeObjectTypes )
If result = 0 Then
    e3Application.PutInfo 0, "Tree " & treeName & " ( " & treeId & " ) : error getting
Else
    e3Application.PutInfo 0, "Tree " & treeName & " ( " & treeId & " ) has " & result
    For treeObjectIndex = 1 To result

        objectType = treeObjectTypes( treeObjectIndex, 0 )
        infoType = treeObjectTypes( treeObjectIndex, 1 )
        infoValue = treeObjectTypes( treeObjectIndex, 2 )
        objectFlags = treeObjectTypes( treeObjectIndex, 3 )

        Select Case objectType
        Case 0
            treeTypeName = "Slot or contour"
        Case 1
            treeTypeName = "Device"
        Case 2
            Select Case objectFlags
            Case 1
                treeTypeName = "Placed symbol"
            Case 2
                treeTypeName = "Unplaced symbol"
            Case Else
                treeTypeName = "Symbol"
            End Select
        Case 3
            treeTypeName = "Symbol Pin"
        Case 4
            Select Case objectFlags
            Case 1
                treeTypeName = "Placed model"
            Case 2
                treeTypeName = "Unplaced model"
            Case Else
                treeTypeName = "Model"
            End Select
        Case 5
            treeTypeName = "Model Pin"
        Case 6
            treeTypeName = "Sheet"
        Case 7
            treeTypeName = "Field"
        Case 8
            treeTypeName = "Functional unit"
        Case 9
            treeTypeName = "Functional port"
        End Select
    Next treeObjectIndex
End For
```

```

If Len( "" & infoType ) = 0 Then
    infoTypeName = "No column information"
Else
    If StrComp( infoType, ATTRIBUTE_VALUE, 1 ) = 0 Then
        infoTypeName = "Attribute column information: " & infoValue
    Else
        infoTypeName = "Predefined value column information: " & infoValue
    End If
End If

e3Application.PutInfo 0, "    Object Type is " & treeTypeName
e3Application.PutInfo 0, "        " & infoTypeName

Next
End If
End If

Set tree = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

## See Also

- [Classes - Overview](#)
- [e3Bundle - Overview](#)
- [e3Device - Overview](#)
- [e3ExternalDocument - Overview](#)
- [e3Pin - Overview](#)
- [e3Sheet - Overview](#)
- [e3StructureNode - Overview](#)
- [e3Symbol - Overview](#)

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Tree.Create( *name*, *position*, *before* )**

## Syntax

**Integer** Create( [in]**String** *name*, [in][optional]**Integer** *position*, [in][optional]**Integer** *before* )

## Description

Creates a new tree item and places it in the project tree structure.

## Parameters

Type	Parameter	Description
[in] <b>String</b>	name	Name of the tree item
[in][optional] <b>Integer</b>	position	Identifier of an existing tree item to use as the insertion position of the new tree item in the project tree structure
		Default value is 0
		Indicates whether to insert the tree item before or after <i>position</i>
[in][optional] <b>Integer</b>	before	If 1, the tree item is inserted before <i>position</i>
		If 0, the tree item is inserted after <i>position</i>
		Default value is 0

## Return Values

Value	Status	Description
>0	Success	Identifier of the new tree item
0	Failure	Error occurred

## Remarks

If *position* is 0, the tree item is placed in the first position of the project tree structure.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>**.series project.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set tree = job.CreateTreeObject()

Dim newTreeName: newTreeName = "Sanjuro"
Dim placeBefore : placeBefore = 1

treeCount = job.GetTreeIds( treeIds )      'get all the trees in the project
If treeCount > 0 Then

    treeId = tree.SetId( treeIds( 1 ) )
    treeName = tree.GetName()

    result = tree.Create( newTreeName, treeId, placeBefore )
    If result = 0 Then
        message = "Error creating tree " & newTreeName & " placed before " & treeName
    Else
        message = "Tree " & newTreeName & " with an identifier of " & result & " created a"
    End If
    e3Application.PutInfo 0, message        'output result of operation

End If

Set tree = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

## See Also

- [e3Tree - Overview](#)
- [Delete\(\)](#)

## SetAsMaster - e3Symbol

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Tree.Delete()**

## Syntax

**Integer** Delete()

## Description

Deletes the current tree item from the project tree structure.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
0	Inconclusive	Tree item was deleted or an error occurred
-1	Failure	Tree item not found

## Remarks

Due caution is recommended on relying on the return value of 0 meaning the tree item was deleted since this also could mean an error has occurred. SetId() may be used to verify if the tree item has been successfully deleted.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting a tree.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
```



## SetAsMaster - e3Symbol

```
Set tree = job.CreateTreeObject()

treeId = job.GetActiveTreeID()
If treeId > 0 Then

    tree.SetId treeId
    treeName = tree.GetName()

    result = tree.Delete()
    If result = -1 Then
        message = "Error deleting tree " & treeName & " ( " & treeId & " )"
    Else
        currentId = tree.GetId()
        If currentId = treeId Then
            message = "Error deleting tree " & treeName & " ( " & treeId & " )"
        Else
            message = "Tree " & treeName & " deleted"
        End If
    End If
End If
e3Application.PutInfo 0, message          'output result of operation

End If

Set tree = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3Tree - Overview](#)
  - [Create\(\)](#)
  - [SetId\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Tree.GetId()**

## Syntax

Integer GetId()

## Description

Gets the identifier of the current tree item.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
> 0	Success	Current tree item identifier
0	Failure	No tree item

## Remarks

The function returns the identifier value set by SetId() unless the tree item no longer exists.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>**.series project and selecting a tree item.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set tree = job.CreateTreeObject()
```

```
treeId = job.GetActiveTreeID()  
If treeId > 0 Then  
  
    tree.SetId treeId  
  
    result = tree.GetId()  
    If result = 0 Then  
        message = "No tree item is set"  
    Else  
        message = "Tree item " & result & " has been set"  
    End If  
    e3Application.PutInfo 0, message          'output result of operation  
  
End If  
  
Set tree = Nothing  
Set job = Nothing  
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3Tree - Overview](#)
  - [SetId\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Tree.GetName()**

## Syntax

*String* GetName()

## Description

Gets the tree item's name.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
"<Text>"	Success	Tree item's name
"<Empty>"	Failure	Error occurred

## Remarks

The name provided is in the currently defined **E<sup>3</sup>**.series language.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>**.series project and selecting a tree item.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set tree = job.CreateTreeObject()
```

## SetAsMaster - e3Symbol

```
treeId = job.GetActiveTreeID()  
If treeId > 0 Then  
  
    tree.SetId treeId  
  
    result = tree.GetName()  
    If Len( "" & result ) = 0 Then  
        message = "Error getting name of tree " & treeId  
    Else  
        message = "Name of tree " & treeId & " is " & result  
    End If  
    e3Application.PutInfo 0, message          'output result of operation  
  
End If  
  
Set tree = Nothing  
Set job = Nothing  
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3Tree - Overview](#)
  - [GetNames\(\)](#)
  - [SetName\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Tree.GetNames( *names* )**

## Syntax

**Integer** GetNames( [out]**String Array** *names* )

## Description

Gets the tree item's names in all languages.

## Parameters

Type	Parameter	Description
[out] <b>String Array</b>	<i>names</i>	Array of tree item names by language

## Return Values

Value	Status	Description
> 0	Success	Number of name elements in <i>names</i>
0	Failure	Error occurred

## Remarks

*names* is a 1-based array.

*names* **Array** contains the following elements:

Index	Description
1	Chinese tree name
2	Dutch tree name
3	English tree name
4	French tree name
5	German tree name
6	Italian tree name

- 7 Japanese tree name
- 8 Portuguese tree name
- 9 Russian tree name
- 10 Spanish tree name
- 11 Turkish tree name
- Polish tree name
- 12 Available since v2018-19.00

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting a tree item.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set tree = job.CreateTreeObject()

treeId = job.GetActiveTreeID()
If treeId > 0 Then

    tree.SetId treeId

    result = tree.GetNames( names )
    If result = 0 Then
        e3Application.PutInfo 0, "Error getting names of tree " & treeId
    Else
        e3Application.PutInfo 0, "Names of tree " & treeId & " are:"
        For nameIndex = 1 To result
            e3Application.PutInfo 0, "    " & names( nameIndex )
        Next
    End If

End If

Set tree = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2010-9.00.

Modified in v2018-19.00.

## See Also

- [e3Tree - Overview](#)
  - [GetName\(\)](#)
  - [SetNames\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).





v2022-23.00

**e3Tree.GetSelectedAllDeviceIds( *ids* )**

## Syntax

**Integer** GetSelectedAllDeviceIds( [out]**Integer Array** *ids* )

## Description

Gets the identifiers of devices of all types selected in the project tree item.

## Parameters

Type	Parameter	Description
[out] <b>Integer Array</b>	<i>ids</i>	Array of identifiers of devices selected in the project tree item

## Return Values

Value	Status	Description
> 0	Success	Number of identifiers in <i>ids</i>
0	Inconclusive	No devices are selected or an error occurred

## Remarks

*ids* is a 1-based array.

*ids* may include block, cable, connector, device and terminal identifiers. The device identifiers can be used by [e3Device](#) objects to handle them.

Due caution is recommended on relying on the return value of 0 meaning no devices are selected in the project tree item since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting devices in a project tree item.

## Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set tree = job.CreateTreeObject()

treeId = job.GetActiveTreeID()
If treeId > 0 Then

    tree.SetId treeId
    treeName = tree.GetName()

    result = tree.GetSelectedAllDeviceIds( deviceIds )
    If result = 0 Then
        e3Application.PutInfo 0, "Project tree " & treeName & " ( " & treeId & " ) has no
    Else
        e3Application.PutInfo 0, "Project tree " & treeName & " ( " & treeId & " ) has " &
        For deviceIndex = 1 To result
            e3Application.PutInfo 0, "      " & deviceIds( deviceIndex )
        Next
    End If

End If

Set tree = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3Tree - Overview](#)
- [e3Device - Overview](#)
- [GetSelectedAllDeviceIdsByFolder\(\)](#)
- [GetSelectedBlockIds\(\)](#)
- [GetSelectedCableIds\(\)](#)
- [GetSelectedConnectorIds\(\)](#)
- [GetSelectedDeviceIds\(\)](#)
- [GetSelectedTerminalIds\(\)](#)

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Tree.GetSelectedAllDeviceIdsByFolder( *ids* )**

## Syntax

**Integer** GetSelectedAllDeviceIdsByFolder( [out]**Integer Array** *ids* )

## Description

Gets the identifiers of devices in selected folders in the project tree item.

## Parameters

Type	Parameter	Description
[out] <b>Integer Array</b>	<i>ids</i>	Array of identifiers of devices in the project tree item

## Return Values

Value	Status	Description
> 0	Success	Number of identifiers in <i>ids</i>
0	Inconclusive	No folders containing devices are selected or an error occurred

## Remarks

*ids* is a 1-based array.

*ids* may include block, cable, connector, device and terminal identifiers. The device identifiers can be used by [e3Device](#) objects to handle them.

Due caution is recommended on relying on the return value of 0 meaning no folders containing devices are selected in the project tree item since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting folders containing devices in a project tree item.

## Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set tree = job.CreateTreeObject()

treeId = job.GetActiveTreeID()
If treeId > 0 Then

    tree.SetId treeId
    treeName = tree.GetName()

    result = tree.GetSelectedAllDeviceIdsByFolder( deviceIds )
    If result = 0 Then
        e3Application.PutInfo 0, "Project tree " & treeName & " ( " & treeId & " ) has no
    Else
        e3Application.PutInfo 0, "Project tree " & treeName & " ( " & treeId & " ) has " &
        For deviceIndex = 1 To result
            e3Application.PutInfo 0, "      " & deviceIds( deviceIndex )
        Next
    End If

End If

Set tree = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2010-9.10.

---

## See Also

- [e3Tree - Overview](#)
- [e3Device - Overview](#)
- [GetSelectedAllDeviceIds\(\)](#)
- [GetSelectedBlockIds\(\)](#)
- [GetSelectedCableIds\(\)](#)
- [GetSelectedConnectorIds\(\)](#)
- [GetSelectedDeviceIds\(\)](#)
- [GetSelectedTerminalIds\(\)](#)

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Tree.GetSelectedBlockIds( *ids* )**

## Syntax

**Integer** GetSelectedBlockIds( [out]**Integer Array** *ids* )

## Description

Gets the identifiers of blocks selected in the project tree item.

## Parameters

Type	Parameter	Description
[out] <b>Integer Array</b>	<i>ids</i>	Array of identifiers of blocks selected in the project tree item

## Return Values

Value	Status	Description
> 0	Success	Number of identifiers in <i>ids</i>
0	Inconclusive	No blocks are selected or an error occurred

## Remarks

*ids* is a 1-based array.

The block identifiers can be used by e3Device objects to handle them.

Due caution is recommended on relying on the return value of 0 meaning no blocks are selected in the project tree item since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting blocks in a project tree item.

## Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set tree = job.CreateTreeObject()

treeId = job.GetActiveTreeID()
If treeId > 0 Then

    tree.SetId treeId
    treeName = tree.GetName()

    result = tree.GetSelectedBlockIds( blockIds )
    If result = 0 Then
        e3Application.PutInfo 0, "Project tree " & treeName & " ( " & treeId & " ) has no
    Else
        e3Application.PutInfo 0, "Project tree " & treeName & " ( " & treeId & " ) has " &
        For blockIndex = 1 To result
            e3Application.PutInfo 0, "      " & blockIds( blockIndex )
        Next
    End If

End If

Set tree = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3Tree - Overview](#)
  - [e3Device - Overview](#)
  - [GetSelectedAllDeviceIds\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).





v2022-23.00

**e3Tree.GetSelectedBundleIds( *ids* )**

## Syntax

**Integer** GetSelectedBundleIds( [out]**Integer Array** *ids* )

## Description

Gets the identifiers of bundles selected in the project tree item.

## Parameters

Type	Parameter	Description
[out] <b>Integer Array</b>	<i>ids</i>	Array of identifiers of bundles selected in the project tree item

## Return Values

Value	Status	Description
> 0	Success	Number of identifiers in <i>ids</i>
0	Inconclusive	No bundles are selected or an error occurred

## Remarks

*ids* is a 1-based array.

The bundle identifiers can be used by [e3Bundle](#) objects to handle them.

Due caution is recommended on relying on the return value of 0 meaning no bundles are selected in the project tree item since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting bundles in a project tree item.

## Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set tree = job.CreateTreeObject()

treeId = job.GetActiveTreeID()
If treeId > 0 Then

    tree.SetId treeId
    treeName = tree.GetName()

    result = tree.GetSelectedBundleIds( bundleIds )
    If result = 0 Then
        e3Application.PutInfo 0, "Project tree " & treeName & " ( " & treeId & " ) has no
    Else
        e3Application.PutInfo 0, "Project tree " & treeName & " ( " & treeId & " ) has " &
        For bundleIndex = 1 To result
            e3Application.PutInfo 0, "      " & bundleIds( bundleIndex )
        Next
    End If

End If

Set tree = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3Tree - Overview](#)
  - [e3Bundle - Overview](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Tree.GetSelectedBusbarIds( *ids* )**

## Syntax

**Integer** GetSelectedBusbarIds( [out]**Integer Array** *ids* )

## Description

Gets the identifiers of busbars selected in the project tree item.

## Parameters

Type	Parameter	Description
[out] <b>Integer Array</b>	<i>ids</i>	Array of identifiers of busbars selected in the project tree item.

## Return Values

Value	Status	Description
> 0	Success	Number of identifiers in <i>ids</i>
0	Inconclusive	No busbars or busbar sub-elements are selected or an error occurred

## Remarks

*ids* is a 1-based array.

The busbars identifiers can be used by e3Device objects to handle them.

Due caution is recommended on relying on the return value of 0 meaning no busbars or busbar sub-elements are selected in the project tree item since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting busbars in a project tree item.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set tree = job.CreateTreeObject()
Set device = job.CreateDeviceObject()

treeId = job.GetActiveTreeID()
If treeId > 0 Then

    tree.SetId treeId
    treeName = tree.GetName()

    result = tree.GetSelectedBusbarIds( busbarIds )
    If result = 0 Then
        e3Application.PutInfo 0, "Project tree " & treeName & " ( " & treeId & " ) has no
    Else
        e3Application.PutInfo 0, "Project tree " & treeName & " ( " & treeId & " ) has " &
        For busbarIndex = 1 To result
            deviceId = device.SetId( busbarIds( busbarIndex ) )
            deviceName = device.GetName()
            deviceAssignment = device.GetAssignment()
            deviceLocation = device.GetLocation()

            e3Application.PutInfo 0, "    Busbar " & deviceName & " " & deviceAssignment &
        Next
    End If

End If

Set device = Nothing
Set tree = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2022-23.00.

## See Also

- [e3Tree - Overview](#)
- [e3Device - Overview](#)

## SetAsMaster - e3Symbol

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Tree.GetSelectedCableIds( *ids* )**

## Syntax

**Integer** GetSelectedCableIds( [out]**Integer Array** *ids* )

## Description

Gets the identifiers of cables selected in the project tree item.

## Parameters

Type	Parameter	Description
[out] <b>Integer Array</b>	<i>ids</i>	Array of identifiers of cables selected in the project tree item.

## Return Values

Value	Status	Description
> 0	Success	Number of identifiers in <i>ids</i>
0	Inconclusive	No cables or cable sub-elements are selected or an error occurred

## Remarks

*ids* is a 1-based array.

The identifiers of cables of selected sub-elements such as conductors will be included in *ids*.

The cable identifiers can be used by e3Device objects to handle them.

Due caution is recommended on relying on the return value of 0 meaning no cables or cable sub-elements are selected in the project tree item since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting cables in a project tree item.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set tree = job.CreateTreeObject()

treeId = job.GetActiveTreeID()
If treeId > 0 Then

    tree.SetId treeId
    treeName = tree.GetName()

    result = tree.GetSelectedCableIds( cableIds )
    If result = 0 Then
        e3Application.PutInfo 0, "Project tree " & treeName & " ( " & treeId & " ) has no
    Else
        e3Application.PutInfo 0, "Project tree " & treeName & " ( " & treeId & " ) has " &
        For cableIndex = 1 To result
            e3Application.PutInfo 0, "      " & cableIds( cableIndex )
        Next
    End If

End If

Set tree = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3Tree - Overview](#)
  - [e3Device - Overview](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Tree.GetSelectedConnectorIds( *ids* )**

## Syntax

**Integer** GetSelectedConnectorIds( [out]**Integer Array** *ids* )

## Description

Gets the identifiers of connectors selected in the project tree item.

## Parameters

Type	Parameter	Description
[out] <b>Integer Array</b>	<i>ids</i>	Array of identifiers of connectors selected in the project tree item

## Return Values

Value	Status	Description
> 0	Success	Number of identifiers in <i>ids</i>
0	Inconclusive	No connectors are selected or an error occurred

## Remarks

*ids* is a 1-based array.

The connector identifiers can be used by [e3Device](#) objects to handle them.

Due caution is recommended on relying on the return value of 0 meaning no connectors are selected in the project tree item since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting connectors in a project tree item.



## Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set tree = job.CreateTreeObject()

treeId = job.GetActiveTreeID()
If treeId > 0 Then

    tree.SetId treeId
    treeName = tree.GetName()

    result = tree.GetSelectedConnectorIds( connectorIds )
    If result = 0 Then
        e3Application.PutInfo 0, "Project tree " & treeName & " ( " & treeId & " ) has no
    Else
        e3Application.PutInfo 0, "Project tree " & treeName & " ( " & treeId & " ) has " &
        For connectorIndex = 1 To result
            e3Application.PutInfo 0, "      " & connectorIds( connectorIndex )
        Next
    End If

End If

Set tree = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3Tree - Overview](#)
  - [e3Device - Overview](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Tree.GetSelectedDeviceIds( *ids* )**

## Syntax

**Integer** GetSelectedDeviceIds( [out]**Integer Array** *ids* )

## Description

Gets the identifiers of devices selected in the project tree item.

## Parameters

Type	Parameter	Description
[out] <b>Integer Array</b>	<i>ids</i>	Array of identifiers of devices selected in the project tree item

## Return Values

Value	Status	Description
> 0	Success	Number of identifiers in <i>ids</i>
0	Inconclusive	No devices are selected or an error occurred

## Remarks

*ids* is a 1-based array.

The identifiers of devices of selected sub-elements such as gates will be included in *ids*.

The device identifiers can be used by e3Device objects to handle them.

Due caution is recommended on relying on the return value of 0 meaning no devices are selected in the project tree item since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting devices in a project tree item.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set tree = job.CreateTreeObject()

treeId = job.GetActiveTreeID()
If treeId > 0 Then

    tree.SetId treeId
    treeName = tree.GetName()

    result = tree.GetSelectedDeviceIds( deviceIds )
    If result = 0 Then
        e3Application.PutInfo 0, "Project tree " & treeName & " ( " & treeId & " ) has no
    Else
        e3Application.PutInfo 0, "Project tree " & treeName & " ( " & treeId & " ) has " &
        For deviceIndex = 1 To result
            e3Application.PutInfo 0, "      " & deviceIds( deviceIndex )
        Next
    End If

End If

Set tree = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3Tree - Overview](#)
  - [e3Device - Overview](#)
  - [GetSelectedAllDeviceIds\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support

## SetAsMaster - e3Symbol

(ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Tree.GetSelectedExternalDocumentIds( *ids* )**

## Syntax

**Integer** GetSelectedExternalDocumentIds( [out]**Integer Array** *ids* )

## Description

Gets the identifiers of external document items selected in the project tree item.

## Parameters

Type	Parameter	Description
[out] <b>Integer Array</b>	<i>ids</i>	Array of identifiers of external document items selected in the project tree item

## Return Values

Value	Status	Description
> 0	Success	Number of identifiers in <i>ids</i>
0	Inconclusive	No external document items are selected or an error occurred

## Remarks

*ids* is a 1-based array.

The external documents identifiers can be used by [e3ExternalDocument](#) objects to handle them.

Due caution is recommended on relying on the return value of 0 meaning no external document items are selected in the project tree item since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting external document items in a project tree item.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set tree = job.CreateTreeObject()

treeId = job.GetActiveTreeID()
If treeId > 0 Then

    tree.SetId treeId
    treeName = tree.GetName()

    result = tree.GetSelectedExternalDocumentIds( externalDocumentIds )
    If result = 0 Then
        e3Application.PutInfo 0, "Project tree " & treeName & " ( " & treeId & " ) has no
    Else
        e3Application.PutInfo 0, "Project tree " & treeName & " ( " & treeId & " ) has " &
        For externalDocumentIndex = 1 To result
            e3Application.PutInfo 0, "      " & externalDocumentIds( externalDocumentIndex )
        Next
    End If

End If

Set tree = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

## See Also

- [e3Tree - Overview](#)
- [e3ExternalDocument - Overview](#)
- [GetSelectedExternalDocumentIdsByFolder\(\)](#)

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Tree.GetSelectedExternalDocumentIdsByFolder( *ids* )**

## Syntax

**Integer** GetSelectedExternalDocumentIdsByFolder( [out]**Integer Array** *ids* )

## Description

Gets the identifiers of external documents in selected folders in the project tree item.

## Parameters

Type	Parameter	Description
[out] <b>Integer Array</b>	<i>ids</i>	Array of identifiers of external documents in the project tree item

## Return Values

Value	Status	Description
> 0	Success	Number of identifiers in <i>ids</i>
0	Inconclusive	No folders containing external documents are selected or an error occurred

## Remarks

*ids* is a 1-based array.

The external document identifiers can be used by e3ExternalDocument objects to handle them.

Due caution is recommended on relying on the return value of 0 meaning no folders containing external documents are selected in the project tree item since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>**.series project and selecting folders containing external documents in a project tree item.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set tree = job.CreateTreeObject()

treeId = job.GetActiveTreeID()
If treeId > 0 Then

    tree.SetId treeId
    treeName = tree.GetName()

    result = tree.GetSelectedExternalDocumentIdsByFolder( externalDocumentIds )
    If result = 0 Then
        e3Application.PutInfo 0, "Project tree " & treeName & " ( " & treeId & " ) has no
    Else
        e3Application.PutInfo 0, "Project tree " & treeName & " ( " & treeId & " ) has " &
        For externalDocumentIndex = 1 To result
            e3Application.PutInfo 0, "    " & externalDocumentIds( externalDocumentIndex )
        Next
    End If

End If

Set tree = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2010-9.10.

---

## See Also

- [e3Tree - Overview](#)
  - [e3ExternalDocument - Overview](#)
  - [GetSelectedExternalDocumentIds\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support



## SetAsMaster - e3Symbol

(ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Tree.GetSelectedPinIds( *ids* )**

## Syntax

**Integer** GetSelectedPinIds( [out]**Integer Array** *ids* )

## Description

Gets the identifiers of pins selected in the project tree item.

## Parameters

Type	Parameter	Description
[out] <b>Integer Array</b>	<i>ids</i>	Array of identifiers of pins selected in the project tree item

## Return Values

Value	Status	Description
> 0	Success	Number of identifiers in <i>ids</i>
0	Inconclusive	No pins are selected or an error occurred

## Remarks

*ids* is a 1-based array.

The pin identifiers can be used by e3Pin objects to handle them.

Due caution is recommended on relying on the return value of 0 meaning no pins are selected in the project tree item since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting pins in a project tree item.

## Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set tree = job.CreateTreeObject()

treeId = job.GetActiveTreeID()
If treeId > 0 Then

    tree.SetId treeId
    treeName = tree.GetName()

    result = tree.GetSelectedPinIds( pinIds )
    If result = 0 Then
        e3Application.PutInfo 0, "Project tree " & treeName & " ( " & treeId & " ) has no
    Else
        e3Application.PutInfo 0, "Project tree " & treeName & " ( " & treeId & " ) has " &
        For pinIndex = 1 To result
            e3Application.PutInfo 0, "      " & pinIds( pinIndex )
        Next
    End If

End If

Set tree = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3Tree - Overview](#)
  - [e3Pin - Overview](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Tree.GetSelectedSheetIds( *ids* )**

## Syntax

**Integer** GetSelectedSheetIds( [out]**Integer Array** *ids* )

## Description

Gets the identifiers of sheets selected in the project tree item.

## Parameters

Type	Parameter	Description
[out] <b>Integer Array</b>	<i>ids</i>	Array of identifiers of sheets selected in the project tree item

## Return Values

Value	Status	Description
> 0	Success	Number of identifiers in <i>ids</i>
0	Inconclusive	No sheets are selected or an error occurred

## Remarks

*ids* is a 1-based array.

The sheet identifiers can be used by [e3Sheet](#) objects to handle them.

Due caution is recommended on relying on the return value of 0 meaning no sheets are selected in the project tree item since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting sheets in a project tree item.

## Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set tree = job.CreateTreeObject()

treeId = job.GetActiveTreeID()
If treeId > 0 Then

    tree.SetId treeId
    treeName = tree.GetName()

    result = tree.GetSelectedSheetIds( sheetIds )
    If result = 0 Then
        e3Application.PutInfo 0, "Project tree " & treeName & " ( " & treeId & " ) has no
    Else
        e3Application.PutInfo 0, "Project tree " & treeName & " ( " & treeId & " ) has " &
        For sheetIndex = 1 To result
            e3Application.PutInfo 0, "      " & sheetIds( sheetIndex )
        Next
    End If

End If

Set tree = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3Tree - Overview](#)
  - [e3Sheet - Overview](#)
  - [GetSelectedSheetIdsByFolder\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Tree.GetSelectedSheetIdsByFolder( *ids* )**

## Syntax

**Integer** GetSelectedSheetIdsByFolder( [out]**Integer Array** *ids* )

## Description

Gets the identifiers of sheets in selected folders in the project tree item.

## Parameters

Type	Parameter	Description
[out] <b>Integer Array</b>	<i>ids</i>	Array of identifiers of sheets in the project tree item

## Return Values

Value	Status	Description
> 0	Success	Number of identifiers in <i>ids</i>
0	Inconclusive	No folders containing sheets are selected or an error occurred

## Remarks

*ids* is a 1-based array.

The sheet identifiers can be used by [e3Sheet](#) objects to handle them.

Due caution is recommended on relying on the return value of 0 meaning no folders containing sheets are selected in the project tree item since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>**.series project and selecting folders containing sheets in a project tree item.

## Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set tree = job.CreateTreeObject()

treeId = job.GetActiveTreeID()
If treeId > 0 Then

    tree.SetId treeId
    treeName = tree.GetName()

    result = tree.GetSelectedSheetIdsByFolder( sheetIds )
    If result = 0 Then
        e3Application.PutInfo 0, "Project tree " & treeName & " ( " & treeId & " ) has no
    Else
        e3Application.PutInfo 0, "Project tree " & treeName & " ( " & treeId & " ) has " &
        For sheetIndex = 1 To result
            e3Application.PutInfo 0, "      " & sheetIds( sheetIndex )
        Next
    End If

End If

Set tree = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2010-9.10.

---

## See Also

- [e3Tree - Overview](#)
  - [e3Sheet - Overview](#)
  - [GetSelectedSheetIds\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Tree.GetSelectedStructureNodeIds( *ids* )**

## Syntax

**Integer** GetSelectedStructureNodeIds( [out]**Integer Array** *ids* )

## Description

Gets the identifiers of structure nodes selected in the project tree item.

## Parameters

Type	Parameter	Description
[out] <b>Integer Array</b>	<i>ids</i>	Array of identifiers of structure nodes selected in the project tree item

## Return Values

Value	Status	Description
> 0	Success	Number of identifiers in <i>ids</i>
0	Inconclusive	No structure nodes are selected or an error occurred

## Remarks

*ids* is a 1-based array.

The structure node identifiers can be used by e3StructureNode objects to handle them.

Due caution is recommended on relying on the return value of 0 meaning no structure nodes are selected in the project tree item since this also could mean an error has occurred.

## Examples



The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting structure nodes in a project tree item.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set tree = job.CreateTreeObject()

treeId = job.GetActiveTreeID()
If treeId > 0 Then

    tree.SetId treeId
    treeName = tree.GetName()

    result = tree.GetSelectedStructureNodeIds( structureNodeIds )
    If result = 0 Then
        e3Application.PutInfo 0, "Project tree " & treeName & " ( " & treeId & " ) has no
    Else
        e3Application.PutInfo 0, "Project tree " & treeName & " ( " & treeId & " ) has " &
        For structureNodeIndex = 1 To result
            e3Application.PutInfo 0, "      " & structureNodeIds( structureNodeIndex )
        Next
    End If

End If

Set tree = Nothing
Set job = Nothing
Set e3Application = Nothing

```

### Version Information

Introduced in v2009-8.50.

### See Also

- [e3Tree - Overview](#)
- [e3StructureNode - Overview](#)

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Tree.GetSelectedSymbolIds( *ids* )**

## Syntax

**Integer** GetSelectedSymbolIds( [out]**Integer Array** *ids* )

## Description

Gets the identifiers of symbols and gates selected in the project tree item.

## Parameters

Type	Parameter	Description
[out] <b>Integer Array</b>	<i>ids</i>	Array of identifiers of symbols and gates selected in the project tree item

## Return Values

Value	Status	Description
> 0	Success	Number of identifiers in <i>ids</i>
0	Inconclusive	No symbols or gates are selected or an error occurred

## Remarks

*ids* is a 1-based array.

The identifiers of symbols and gates of selected sub-elements such as pins will be included in *ids*.

The symbol and gate identifiers can be used by [e3Symbol](#) objects to handle them.

Due caution is recommended on relying on the return value of 0 meaning no symbols or gates are selected in the project tree item since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting symbols in a project tree item.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set tree = job.CreateTreeObject()

treeId = job.GetActiveTreeID()
If treeId > 0 Then

    tree.SetId treeId
    treeName = tree.GetName()

    result = tree.GetSelectedSymbolIds( symbolIds )
    If result = 0 Then
        e3Application.PutInfo 0, "Project tree " & treeName & " ( " & treeId & " ) has no
    Else
        e3Application.PutInfo 0, "Project tree " & treeName & " ( " & treeId & " ) has " &
        For symbolIndex = 1 To result
            e3Application.PutInfo 0, "      " & symbolIds( symbolIndex )
        Next
    End If

End If

Set tree = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3Tree - Overview](#)
  - [e3Symbol - Overview](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Tree.GetSelectedTerminalIds( *ids* )**

## Syntax

**Integer** GetSelectedTerminalIds( [out]**Integer Array** *ids* )

## Description

Gets the identifiers of terminals selected in the project tree item.

## Parameters

Type	Parameter	Description
[out] <b>Integer Array</b>	<i>ids</i>	Array of identifiers of terminals selected in the project tree item

## Return Values

Value	Status	Description
> 0	Success	Number of identifiers in <i>ids</i>
0	Inconclusive	No terminals are selected or an error occurred

## Remarks

*ids* is a 1-based array.

The identifiers of terminals of selected sub-elements such as pins will be included in *ids*.

The terminal identifiers can be used by e3Device objects to handle them.

Due caution is recommended on relying on the return value of 0 meaning no terminals are selected in the project tree item since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting terminals in a project tree item.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set tree = job.CreateTreeObject()

treeId = job.GetActiveTreeID()
If treeId > 0 Then

    tree.SetId treeId
    treeName = tree.GetName()

    result = tree.GetSelectedTerminalIds( terminalIds )
    If result = 0 Then
        e3Application.PutInfo 0, "Project tree " & treeName & " ( " & treeId & " ) has no
    Else
        e3Application.PutInfo 0, "Project tree " & treeName & " ( " & treeId & " ) has " &
        For terminalIndex = 1 To result
            e3Application.PutInfo 0, "      " & terminalIds( terminalIndex )
        Next
    End If

End If

Set tree = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

## See Also

- [e3Tree - Overview](#)
- [e3Device - Overview](#)

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Tree.GetSortingMethod( *flags*, *structure*, *freetab* )**

## Syntax

**Integer** GetSortingMethod( [out]**Integer** *flags*, [out]**2D-Array** *structure*, [out]**2D-Array** *freetab* )

## Description

Gets sorting method of elements in the tree item.

## Parameters

Type	Parameter	Description
[out] <b>Integer</b>	<i>flags</i>	Global setting values relevant to the sorting method
[out] <b>2D-Array</b>	<i>structure</i>	Array describing the levels structure used for sorting
[out] <b>2D-Array</b>	<i>freetab</i>	Array describing the defined keys used for sorting

## Return Values

Value	Status	Description
1	Success	Sorting method supplied
≠1	Failure	Error occurred

## Remarks

The set of values for *flags* are a combination of the following bit values:

Bit Value	Description
1	If used, the tree item is sorted according to the <i>structure</i> array values:  It is only specified whether a certain structure level is to be displayed at all and what is displayed in the info column of one level

If unused, the tree item is sorted according to the *freetab* array values:

All elements are sorted according to the defined keys in this array

Higher level assignment, location and device designation level are arranged in sub-levels, when these are hierarchically arranged by a separator

2

Compliant with **IEC 81346**

4

Panel models are sorted according to their placement on the sheet

8

Sheets are sorted hierarchically in the tree item

16

Elements of all types are displayed on one level sorted by name

32

Assembly parts are displayed beneath parents

*structure* and *freetab* are 1-based arrays.

*structure* is a **2D-Array** of structure types. Each structure type array contains the following values:

Index Description

Name of the structure level

Either fix strings or internal attribute names are possible

## Possible fix string values are:

0

- <Project>
- <Assignment>
- <Location>
- <Device>
- <Product>
- <Unique Identifier>
- <Sheet>

Flags field

1

If 1, corresponding elements of the structure type are displayed

If 0, corresponding elements of the structure type are not displayed

Value type

2

If "A", the value in column 0 is an attribute

If "P", the value in column 0 is a predefined value

3

Name of the internal attribute or predefined string from which the info field shall be derived

4

Hierarchical level in the project tree structure

*freetab* is a **2D-Array** of up to 5 structure types. Each structure type array contains the following values:

Index	Description
	Value type
0	<p>If "A", the value in column 0 is an attribute</p> <p>If "P", the value in column 0 is a predefined value</p> <p>Internal attribute name of the predefined value from which the sorting value shall be derived</p> <p>Possible fix string values are:</p>
1	<ul style="list-style-type: none"> <li>• &lt;Project&gt;</li> <li>• &lt;Assignment&gt;</li> <li>• &lt;Location&gt;</li> <li>• &lt;Device&gt;</li> <li>• &lt;Product&gt;</li> <li>• &lt;Unique Identifier&gt;</li> </ul> <p>Flags field</p>
2	<p>If 1, the elements are sorted in ascending order</p> <p>If 0, the elements are sorted in descending order</p>

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project with sorted tree items.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject
Set tree = job.CreateTreeObject

treeId = job.GetActiveSheetTreeID()           'get currently active project tree
tree.SetId treeId
result = tree.GetSortingMethod( displaySettingsFlags, structure, freeTab )           'get cur

If result <> 1 Then
    e3Application.PutInfo 0, "Error occurred getting the tree sorting method"
Else
    If displaySettingsFlags And 2 Then
        e3Application.PutInfo 0, "Tree " & tree.GetName & ": uses IEC 81346 compliancy"
    End If
    If displaySettingsFlags And 4 Then
        e3Application.PutInfo 0, "Tree " & tree.GetName & ": panel models sorted according"
    End If
    If displaySettingsFlags And 8 Then
        e3Application.PutInfo 0, "Tree " & tree.GetName & ": sheets sorted hierarchically"
    End If
End If

```



## SetAsMaster - e3Symbol

```
End If
If displaySettingsFlags And 16 Then
    e3Application.PutInfo 0, "Tree " & tree.GetName & ": elements sorted by name on on
End If
If displaySettingsFlags And 32 Then
    e3Application.PutInfo 0, "Tree " & tree.GetName & ": assembly parts are displa
End If

If displaySettingsFlags And 1 Then          'look into structure for information
    structureCount = UBound( structure )    'get number of items in array
    If structureCount > 0 Then
        e3Application.PutInfo 0, "Tree " & tree.GetName & " Sorting Method (structure)
        For structureIndex = 1 To structureCount
            e3Application.PutInfo 0, "    Level Name: " & structure( structureIndex, 0
        Next
    End If
Else          'look into freeTab for information
    freeTabCount = UBound( freeTab )        'get number of items in array
    If freeTabCount > 0 Then
        e3Application.PutInfo 0, "Tree " & tree.GetName & " Sorting Method (free tab):
        For freeTabIndex = 1 To freeTabCount
            e3Application.PutInfo 0, "    Type: " & freeTab( freeTabIndex, 0 ) & " At
        Next
    End If
End If
End If

Set tree = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3Tree - Overview](#)
  - [SetSortingMethod\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Tree.GetTreeType()**

## Syntax

**Integer** GetTreeType()

## Description

Gets a value representing the tree item's subtype.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
-1	Failure	Error occurred
0..2	Success	Tree item's subtype

## Remarks

A successful return value will be one of the following values:

Value	Subtype
0	Project tree
1	Signal tree
2	Variant tree

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting a tree item.

Visual Basic Script

## SetAsMaster - e3Symbol

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set tree = job.CreateTreeObject()

treeId = job.GetActiveTreeID()
If treeId > 0 Then

    tree.SetId treeId
    treeName = tree.GetName()

    result = tree.GetTreeType()
    Select Case result
    Case 2
        message = "Tree " & treeName & " ( " & treeId & " ) type is variant tree"
    Case 1
        message = "Tree " & treeName & " ( " & treeId & " ) type is signal tree"
    Case 0
        message = "Tree " & treeName & " ( " & treeId & " ) type is project tree"
    Case -1
        message = "Error getting the type for tree " & treeName & " ( " & treeId & " )"
    End Select
    e3Application.PutInfo 0, message          'output result of operation

End If

Set tree = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

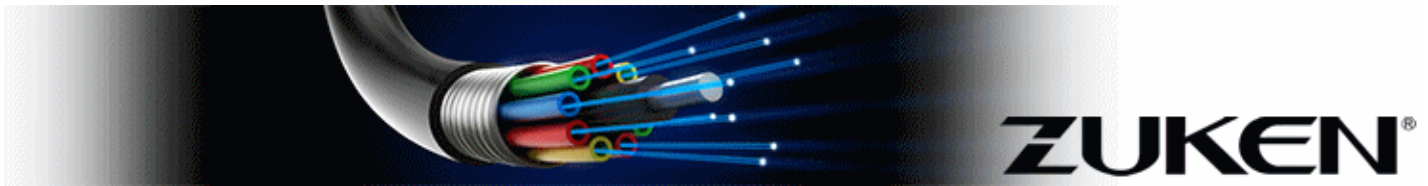
Introduced in v2016-17.03 and v2017-17.70.

---

## See Also

- [e3Tree - Overview](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Tree.GetVisibleInfoTypes( *views*, *schematic* )**

## Syntax

**Integer** GetVisibleInfoTypes( [out]**String** *views*, [out]**String** *schematic* )

## Description

Gets lists of views and schematic types displayed in the project tree item.

**Function has been deprecated.** Please use GetVisibleInfoTypesEx() instead.

## Parameters

Type	Parameter	Description
[out] <b>String</b>	<i>views</i>	Semi-colon (;) delimited list of displayed views
[out] <b>String</b>	<i>schematic</i>	Semi-colon (;) delimited list of displayed schematic types

## Return Values

Value	Status	Description
1	Success	Views and schematic type display information is supplied
0	Inconclusive	All views and schematics are displayed or an error occurred

## Remarks

If the *views* value is "<Empty>", all views are displayed. If *views* contains "0", the original view is displayed.

If the *schematic* value is "<Empty>", all schematic types are displayed.

*schematic* can contain the following characters:

Character	Schematic Type Description
"0"	Electric
"1"	Hydraulic
"2"	Pneumatic
"3"	Process, measurement and control
"4"	Tubes + instruments
"5"	Single-line diagram
"6"	Panel symbol
	Available since v2016-17.00

Due caution is recommended on relying on the return value of 0 meaning all views and schematic types are displayed in the project tree item since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>**.series project and selecting a project tree item.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set tree = job.CreateTreeObject()

treeId = job.GetActiveTreeID()
If treeId > 0 Then

    tree.SetId treeId
    treeName = tree.GetName()

    result = tree.GetVisibleInfoTypes( views, schematics )
    If result = 0 Then
        e3Application.PutInfo 0, "All views and schematic types are displayed for tree " & treeId
    Else
        e3Application.PutInfo 0, "Tree " & treeName & " ( " & treeId & " ) has visible info types: " & result
        e3Application.PutInfo 0, "    Displayed Views = " & views
        e3Application.PutInfo 0, "    Displayed Schematic Types = " & schematics
    End If
End If

Set tree = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

Modified in v2016-17.00.

Deprecated in v2016-17.00.

---

## See Also

- [e3Tree - Overview](#)
  - [GetVisibleInfoTypesEx\(\)](#)
  - [SetVisibleInfoTypes\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Tree.GetVisibleInfoTypesEx( *views*, *schematicTypes*, *formboardSheetIds* )**

## Syntax

**Integer** GetVisibleInfoTypesEx( [out]**Integer Array** *views*, [out]**integer Array** *schematicTypes*, [out]**Integer Array** *formboardSheetIds* )

## Description

Gets arrays of views, schematic types and formboard sheets displayed in the project tree item.

## Parameters

Type	Parameter	Description
[out] <b>Integer Array</b>	<i>views</i>	Array of displayed view numbers
[out] <b>Integer Array</b>	<i>schematicTypes</i>	Array of displayed schematic types
		See <a href="#">Schematic</a> for possible values
[out] <b>Integer Array</b>	<i>formboardSheetIds</i>	Array of identifiers of formboard sheets

## Return Values

Value	Status	Description
1	Success	Views, schematic type and formboard display information is supplied
0	Failure	Error occurred

## Remarks

*views*, *schematicTypes* and *formboardSheetIds* are 1-based arrays.

If *views* contains an element with the value 0, the original view is displayed.

If *formboardSheetIds* contains an element with the value -2, unused formboard views are set.

# Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting a project tree item.

## Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set tree = job.CreateTreeObject()

treeId = job.GetActiveTreeID()
If treeId > 0 Then

    tree.SetId treeId
    treeName = tree.GetName()

    result = tree.GetVisibleInfoTypesEx( views, schematics, formboardSheetIds )
    If result = 0 Then
        e3Application.PutInfo 0, "Error getting visible information types for tree " & treeName
    Else
        e3Application.PutInfo 0, "Tree " & treeName & " ( " & treeId & " ) visible information types: " & result

        viewCount = UBound( views )
        If viewCount = 0 Then
            e3Application.PutInfo 0, "Tree " & treeName & " ( " & treeId & " ) has no visible information types"
        Else
            e3Application.PutInfo 0, "Tree " & treeName & " ( " & treeId & " ) has " & result & " visible information types"
            For viewIndex = 1 To viewCount
                e3Application.PutInfo 0, "    " & views( viewIndex )
            Next
        End If

        schematicCount = UBound( schematics )
        e3Application.PutInfo 0, "Tree " & treeName & " ( " & treeId & " ) has " & schematicCount & " schematics"
        For schematicIndex = 1 To schematicCount
            e3Application.PutInfo 0, "    " & schematics( schematicIndex )
        Next

        formboardSheetCount = UBound( formboardSheetIds )
        If formboardSheetCount = 0 Then
            e3Application.PutInfo 0, "Tree " & treeName & " ( " & treeId & " ) has no formboard sheets"
        Else
            e3Application.PutInfo 0, "Tree " & treeName & " ( " & treeId & " ) has " & formboardSheetCount & " formboard sheets"
            For formboardSheetIndex = 1 To formboardSheetCount
                e3Application.PutInfo 0, "    " & formboardSheetIds( formboardSheetIndex )
            Next
        End If
    End If
End If

```



```
End If
End If

Set tree = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2016-17.00.

---

## See Also

- [e3Tree - Overview](#)
  - [Schematic](#)
  - [GetVisibleInfoTypes\(\)](#)
  - [SetVisibleInfoTypesEx\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Tree.GetVisibleObjectTypes( *type\_array* )**

## Syntax

**Integer** GetVisibleObjectTypes( [out]**2D-Array** *type\_array* )

## Description

Gets an array of the project tree item's visible object types and their display information settings.

## Parameters

Type	Parameter	Description
[out] <b>2D-Array</b>	<i>type_array</i>	Array containing the object type information

## Return Values

Value	Status	Description
> 0	Success	Number of object types in <i>type_array</i>
0	Failure	Error occurred

## Remarks

The tree item must have at least one visible object type.

*type\_array* is a 1-based array.

Each *type\_array* **2D-Array** item represents a visible object type and contains the following elements:

Index	<b>Type</b>	Description
-------	-------------	-------------

Indicates the object type

The following values are possible:

Value		Description
0	Integer	1 Device
		2 Symbol
		3 Symbol pin
		4 Model
		5 Model pin
		6 Sheet
		7 Field
		8 Functional unit
		9 Functional port
		Slot
		10 Available since v2020-21.12
1	String	Contour
		11 Available since v2020-21.12
		Indicates if the column information consists of a predefined value or an attribute value
1	String	If "P", the information column contains a predefined value
		If "A", the information column contains an attribute value
		If "<Empty>", there is no information column
2	String	Predefined value name or attribute name
3	Integer	Display flag values may be a combination of the following values:
Flag		Description
		Placed objects
1		Valid for symbol and model objects
		Unplaced objects
2		Valid for symbol and model objects
		Dynamic objects
4		Valid for contour objects

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting a project tree item.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set tree = job.CreateTreeObject()

Const ATTRIBUTE_VALUE = "A"

treeId = job.GetActiveTreeID()
If treeId > 0 Then

    tree.SetId treeId
    treeName = tree.GetName()

    result = tree.GetVisibleObjectTypes( treeObjectTypes )
    If result = 0 Then
        e3Application.PutInfo 0, "Tree " & treeName & " ( " & treeId & " ) : error getting
    Else
        e3Application.PutInfo 0, "Tree " & treeName & " ( " & treeId & " ) has " & result
        For treeObjectIndex = 1 To result

            objectType = treeObjectTypes( treeObjectIndex, 0 )
            infoType = treeObjectTypes( treeObjectIndex, 1 )
            infoValue = treeObjectTypes( treeObjectIndex, 2 )
            objectFlags = treeObjectTypes( treeObjectIndex, 3 )

            Select Case objectType
            Case 1
                treeTypeName = "Device"
            Case 2
                Select Case objectFlags
                Case 1
                    treeTypeName = "Placed symbol"
                Case 2
                    treeTypeName = "Unplaced symbol"
                Case Else
                    treeTypeName = "Symbol"
                End Select
            Case 3
                treeTypeName = "Symbol Pin"
            Case 4
                Select Case objectFlags

```

## SetAsMaster - e3Symbol

```
Case 1
    treeTypeName = "Placed model"
Case 2
    treeTypeName = "Unplaced model"
Case Else
    treeTypeName = "Model"
End Select

Case 5
    treeTypeName = "Model Pin"
Case 6
    treeTypeName = "Sheet"
Case 7
    treeTypeName = "Field"
Case 8
    treeTypeName = "Functional unit"
Case 9
    treeTypeName = "Functional port"
Case 10
    treeTypeName = "Slot"
Case 11
    treeTypeName = "Contour"
End Select

If Len( "" & infoType ) = 0 Then
    infoTypeName = "No column information"
Else
    If StrComp( infoType, ATTRIBUTE_VALUE, 1 ) = 0 Then
        infoTypeName = "Attribute column information: " & infoValue
    Else
        infoTypeName = "Predefined value column information: " & infoValue
    End If
End If

e3Application.PutInfo 0, "    Object Type is " & treeTypeName
e3Application.PutInfo 0, "        " & infoTypeName

Next
End If
End If

Set tree = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

Modified in v2020-21.12.

---

## See Also

- [e3Tree - Overview](#)
  - [SetVisibleObjectTypes\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Tree.IsActive()**

## Syntax

**Integer** IsActive()

## Description

Indicates whether the tree item is the currently active tree.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
1	Success	Tree item is active
0	Inconclusive	Tree item is inactive or an error occurred

## Remarks

Due caution is recommended on relying on the return value of 0 meaning the tree item is inactive since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting a tree item.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set tree = job.CreateTreeObject()
```

```
treeCount = job.GetTreeIds( treeIds )
If treeCount > 0 Then
    For treeIndex = 1 To treeCount

        treeId = tree.SetId( treeIds( treeIndex ) )
        treeName = tree.GetName()

        result = tree.IsActive()
        If result = 0 Then
            message = "Tree " & treeName & " ( " & treeId & " ) is inactive"
        Else
            message = "Tree " & treeName & " ( " & treeId & " ) is active"
        End If
        e3Application.PutInfo 0, message          'output result of operation

    Next
End If

Set tree = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

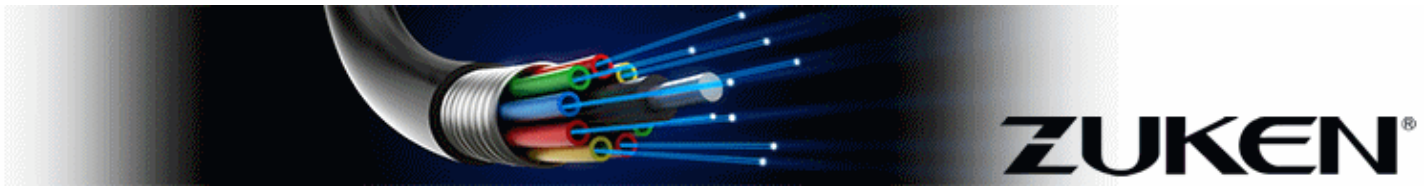
---

## See Also

- [e3Tree - Overview](#)
  - [IsVisible\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).





v2022-23.00

**e3Tree.IsVisible()**

## Syntax

**Integer** IsVisible()

## Description

Indicates whether the tree item is displayed.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
1	Success	Tree item is displayed
0	Inconclusive	Tree item is not displayed or an error occurred

## Remarks

Due caution is recommended on relying on the return value of 0 meaning the tree item is not displayed since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting a tree item.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set tree = job.CreateTreeObject()
```

```

treeCount = job.GetTreeIds( treeIds )
If treeCount > 0 Then
    For treeIndex = 1 To treeCount

        treeId = tree.SetId( treeIds( treeIndex ) )
        treeName = tree.GetName()

        result = tree.IsVisible()
        If result = 0 Then
            message = "Tree " & treeName & " ( " & treeId & " ) is not displayed"
        Else
            message = "Tree " & treeName & " ( " & treeId & " ) is displayed"
        End If
        e3Application.PutInfo 0, message           'output result of operation

    Next
End If

Set tree = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3Tree - Overview](#)
  - [IsActive\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Tree.SetIcon( *filename*, *index* )**

## Syntax

**Integer** SetIcon( [in]**String** *filename*, [in][optional]**Integer** *index* )

## Description

Sets the tree item's display icon.

## Parameters

Type	Parameter	Description
[in] <b>String</b>	filename	Path and name of an icon file
[in][optional] <b>Integer</b>	index	Index of an icon within <i>filename</i> if it is a file containing icon resources
		Default value is 0

## Return Values

Value	Status	Description
1	Success	New icon is applied
0	Failure	Tree item cannot be set with a new icon
-1	Failure	No tree item set
-2	Failure	<i>filename</i> is invalid
-3	Failure	Icon cannot be loaded

## Remarks

*filename* can refer to an icon file, typically having a file extension of ".ico ", or a file containing icon resources.

If *filename* refers to a file containing icon resources, *index* is used to specify which icon within is to be used.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting a tree item.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set tree = job.CreateTreeObject()

Dim iconFile : iconFile = "c:\temp\IconFile.ico"           'this file path and name should re

treeId = job.GetActiveTreeID()
If treeId > 0 Then

    tree.SetId treeId
    treeName = tree.GetName()

    result = tree.SetIcon( iconFile )
    Select Case result
    Case 1
        message = "Tree " & treeName & " ( " & treeId & " ) icon set using " & iconFile
    Case 0
        message = "Icon could not be set for tree " & treeName & " ( " & treeId & " )"
    Case -1
        message = "Error settting icon: No tree item set"
    Case -2
        message = "Error setting icon to tree " & treeName & " ( " & treeId & " ): Icon fi
    Case -3
        message = "Error setting icon to tree " & treeName & " ( " & treeId & " ): Icon fi
    End Select
    e3Application.PutInfo 0, message           'output result of operation

End If

Set tree = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

## See Also

- [e3Tree - Overview](#)

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Tree.SetId( *id* )**

## Syntax

*Integer* SetId( [in]*Integer id* )

## Description

Sets a tree item as the current item.

## Parameters

Type	Parameter	Description
[in] <i>Integer</i>	<i>id</i>	Unique value identifying a tree item

## Return Values

Value	Status	Description
> 0	Success	Current tree item identifier
0	Failure	Error occurred

## Remarks

*id* will remain the current tree item until it is deleted or replaced.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting a tree item.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set tree = job.CreateTreeObject()
```

## SetAsMaster - e3Symbol

```
treeId = job.GetActiveTreeID()  
If treeId > 0 Then  
  
    result = tree.SetId( treeId )  
    If result = 0 Then  
        message = "No tree item has been set"  
    Else  
        message = "Tree item " & result & " has been set"  
    End If  
    e3Application.PutInfo 0, message           'output result of operation  
  
End If  
  
Set tree = Nothing  
Set job = Nothing  
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3Tree - Overview](#)
  - [GetId\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Tree.SetName( *name* )**

## Syntax

**Integer** SetName( [in]**String** *name* )

## Description

Sets the tree item's name.

## Parameters

Type	Parameter	Description
[in] <b>String</b>	name	New tree item name

## Return Values

Value	Status	Description
1	Success	New tree item name
0	Failure	Error occurred

## Remarks

A valid tree item identifier value must be assigned using SetId() otherwise 0 is returned.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting a tree item.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set tree = job.CreateTreeObject()
```



```

Dim treeName : treeName = "Sanjuro"

treeId = job.GetActiveTreeID()
If treeId > 0 Then

    tree.SetId treeId
    oldName = tree.GetName()

    result = tree.SetName( treeName )
    If result = 0 Then
        message = "Error setting name of tree " & treeId & " to " & treeName
    Else
        message = "Tree " & treeId & " set from " & oldName & " to " & treeName
    End If
    e3Application.PutInfo 0, message          'output result of operation

End If

Set tree = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3Tree - Overview](#)
  - [GetName\(\)](#)
  - [SetNames\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Tree.SetNames( *names* )**

## Syntax

**Integer** SetNames( [in]**String Array** *names* )

## Description

Sets the tree item's names in all languages.

## Parameters

Type	Parameter	Description
[in] <b>String Array</b>	<i>names</i>	Array of tree item names by language

## Return Values

Value	Status	Description
1	Success	Number of name elements in <i>names</i>
0	Failure	Error occurred

## Remarks

*names* is a 1-based array.

*names* **Array** should contain the following elements:

Index	Description
1	Chinese tree name
2	Dutch tree name
3	English tree name
4	French tree name
5	German tree name
6	Italian tree name

- 7 Japanese tree name
- 8 Portuguese tree name
- 9 Russian tree name
- 10 Spanish tree name
- 11 Turkish tree name
- Polish tree name
- 12 Available since v2018-19.00

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>**.series project and selecting a tree item.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set tree = job.CreateTreeObject()

Const CHINESE_NAME = 1
Const DUTCH_NAME = 2
Const ENGLISH_NAME = 3
Const FRENCH_NAME = 4
Const GERMAN_NAME = 5
Const ITALIAN_NAME = 6
Const JAPANESE_NAME = 7
Const PORTUGUESE_NAME = 8
Const RUSSIAN_NAME = 9
Const SPANISH_NAME = 10
Const TURKISH_NAME = 11
Const POLISH_NAME = 12

Dim treeName : treeName = "Sanjuro"

treeId = job.GetActiveTreeID()
If treeId > 0 Then

    tree.SetId treeId
    hasNames = tree.GetNames( names )
    If hasNames > 0 Then

        names( ENGLISH_NAME ) = treeName
        names( JAPANESE_NAME ) = treeName

        result = tree.SetNames( names )
        If result = 0 Then

```

## SetAsMaster - e3Symbol

```
message = "Error setting names of tree " & treeId
Else
    message = "Names of tree " & treeId & " are set"
End If
e3Application.PutInfo 0, message           'output result of operation

End If
End If

Set tree = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2010-9.00.

Modified in v2018-19.00.

---

## See Also

- [e3Tree - Overview](#)
  - [GetNames\(\)](#)
  - [SetName\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Tree.SetSortingMethod( *flags*, *structure*, *freetab* )**

## Syntax

**Integer** SetSortingMethod( [in]**Integer** *flags*, [in]**2D-Array** *structure*, [in]**2D-Array** *freetab* )

## Description

Sets sorting method of elements in the project tree.

## Parameters

Type	Parameter	Description
[in] <b>Integer</b>	<i>flags</i>	Global settings values relevant to the sorting method
[in] <b>2D-Array</b>	<i>structure</i>	Array describing the levels structure used for sorting
[in] <b>2D-Array</b>	<i>freetab</i>	Array describing the defined keys used for sorting

## Return Values

Value	Status	Description
1	Success	Sorting method is applied
≠1	Failure	Error occurred

## Remarks

The set of values for *flags* are a combination of the following bit values:

Bit Value	Description
1	If used, project tree is sorted according to the <i>structure</i> array values:  It is only specified whether a certain structure level is to be displayed at all and what is displayed in the info column of one level

If unused, project tree is sorted according to the *freetab* array values:

All elements are sorted according to the defined keys in this array

Higher level assignment, location and device designation level are arranged in sub-levels, when these are hierarchically arranged by a separator

2

Compliant with **IEC 81346**

4

Panel models are sorted according to their placement on the sheet

8

Sheets are sorted hierarchically in the project tree

16

Elements of all types are displayed on one level sorted by name

32

Assembly parts are displayed beneath parents

*structure* and *freetab* are 1-based arrays.

*structure* is a [2D-Array](#) of structure types. Each structure type array contains the following values:

Index Description

Name of the structure level

Either predefined strings or internal attribute names are possible

## Possible predefined string values are:

0

- <Project>
- <Assignment>
- <Location>
- <Device>
- <Product>
- <Unique Identifier>
- <Sheet>

A structural type is ignored if its name does not exist

Flags field

1

If 1, corresponding elements of the structure type are displayed

If 0, corresponding elements of the structure type are not displayed

Value type

2

If "A", the value in column 0 is an attribute

If "P", the value in column 0 is a predefined value

3

Name of the internal attribute or predefined string from which the info field shall be derived

4

Hierarchical level in the project tree structure

*freetab* is a **2D-Array** of up to 5 structure types. Each structure type array contains the following values:

Index    Description

Value type

0        If "A", the value in column 0 is an attribute

         If "P", the value in column 0 is a predefined value

         Internal attribute name of the predefined value from which the sorting value shall be derived

Possible predefined string values are:

- 1        • <Project>  
          • <Assignment>  
          • <Location>  
          • <Device>  
          • <Product>  
          • <Unique Identifier>

Flags field

2        If 1, the elements are sorted in ascending order

         If 0, the elements are sorted in descending order

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project with items in project trees.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject
Set tree = job.CreateTreeObject

Dim flags : flags = 0
ReDim freeTab(3, 2)          'define free tab structure and data to set

freeTab(1, 0) = "A"
freeTab(1, 1) = ".DOCUMENT_TYPE"
freeTab(1, 2) = 1

freeTab(2, 0) = "A"
freeTab(2, 1) = "Class"
freeTab(2, 2) = 1

freeTab(3, 0) = "P"
```

## SetAsMaster - e3Symbol

```
freeTab(3, 1) = "<Assignment>"
freeTab(3, 2) = 1

treeId = job.GetActiveSheetTreeID()           'get currently active project tree
tree.SetId treeId
result = tree.SetSortingMethod( flags, structure, freeTab )      'set current sorting method

If result <> 1 Then
    message = "Project tree " & tree.GetName & ": Error occurred apply the tree sorting method"
Else
    message = "Project tree " & tree.GetName & ": Tree sorting method applied"
End If
e3Application.PutInfo 0, message              'output result of operation

Set tree = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3Tree - Overview](#)
  - [GetSortingMethod\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).





v2022-23.00

**e3Tree.SetVisibleInfoTypes( *views*, *schematic* )**

## Syntax

*Integer* SetVisibleInfoTypes( [in]*String* *views*, [in]*String* *schematic* )

## Description

Sets the views and schematic types displayed in the project tree item.

**Function has been deprecated.** Please use SetVisibleInfoTypesEx() instead.

## Parameters

Type	Parameter	Description
[in] <i>String</i>	<i>views</i>	Semi-colon (;) delimited list of displayed views
[in] <i>String</i>	<i>schematic</i>	Semi-colon (;) delimited list of displayed schematic types

## Return Values

Value	Status	Description
1	Success	Views and schematic type display information is applied
0	Failure	Error occurred
-1	Failure	Tree not found

## Remarks

If the *views* value is "<Empty>", all views are displayed. If *views* contains "0", the original view is displayed.

If the *schematic* value is "<Empty>", all schematic types are displayed.

*schematic* can contain the following characters:

Character	Schematic Type Description
"0"	Electric
"1"	Hydraulic
"2"	Pneumatic
"3"	Process, measurement and control
"4"	Tubes + instruments
"5"	Single-line diagram
"6"	Panel symbol
Available since v2016-17.00	

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting a project tree item.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set tree = job.CreateTreeObject()

Dim views : views = ""           'all views
Dim schematics : schematics = "0;1;3;4;5;6" 'everything except for pneumatic

treeId = job.GetActiveTreeID()
If treeId > 0 Then

    tree.SetId treeId
    treeName = tree.GetName()

    result = tree.SetVisibleInfoTypes( views, schematics )
    Select Case result
    Case 1
        message = "Visible information types of project tree " & treeName & " ( " & treeId
    Case 0
        message = "Error setting visible information types for project tree " & treeName & " ( " & treeId
    Case -1
        message = "Error finding tree for setting visible information types"
    End Select
    e3Application.PutInfo 0, message 'output result of operation

End If

Set tree = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

Modified in v2016-17.00.

Deprecated in v2016-17.00.

---

## See Also

- [e3Tree - Overview](#)
  - [GetVisibleInfoTypes\(\)](#)
  - [SetVisibleInfoTypesEx\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Tree.SetVisibleInfoTypesEx( *views*, *schematicTypes*, *formboardSheetIds* )**

## Syntax

**Integer** SetVisibleInfoTypesEx( [in]**Integer Array** *views*, [in]**integer Array** *schematicTypes*, [in]**Integer Array** *formboardSheetIds* )

## Description

Sets the views, schematic types and formboard sheets displayed in the project tree item.

## Parameters

Type	Parameter	Description
[in] <b>Integer Array</b>	<i>views</i>	Array of displayed view numbers
[in] <b>Integer Array</b>	<i>schematicTypes</i>	Array of displayed schematic types
		See <a href="#">Schematic</a> for possible values
[in] <b>Integer Array</b>	<i>formboardSheetIds</i>	Array of identifiers of formboard sheets

## Return Values

Value	Status	Description
1	Success	Views, schematic type and formboard display information is applied
0	Failure	Error occurred

## Remarks

*views*, *schematicTypes* and *formboardSheetIds* are 1-based arrays.

If *views* contains an element with the value -1, all existing views are set.

If *views* contains no elements, no view is set.

If *schematicTypes* contains no elements, all schematic types are set.

If *formboardSheetIds* contains an element with the value -1, all formboard sheets are set.

If *formboardSheetIds* contains an element with the value -2, unused formboard views are set.

If *formboardSheetIds* contains no elements, no formboard views are set.

*formboardSheetIds* can contain formboard region sheet identifiers and formboard parent sheet identifiers. Both sheet type identifiers function in the same way.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting a project tree item.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set formboardSheetIds = CreateObject( "System.Collections.ArrayList" )
Set job = e3Application.CreateJobObject()
Set sheet = job.CreateSheetObject()
Set tree = job.CreateTreeObject()

sheetCount = job.GetAllSheetIds( sheetIds )           'get all formboard sheet Ids to add to
If sheetCount > 0 Then
    For sheetIndex = 1 To sheetCount

        sheetId = sheet.SetId( sheetIds( sheetIndex ) )
        isFormboard = sheet.IsFormboard()
        If isFormboard = 1 Then
            formboardSheetIds.Add sheetId
        End If

    Next
End If

treeId = job.GetActiveTreeID()
If treeId > 0 Then

    tree.SetId treeId
    treeName = tree.GetName()
    hasOriginalSettings = tree.GetVisibleInfoTypesEx( views, schematics, formboardIds )

```

## SetAsMaster - e3Symbol

```
If hasOriginalSettings = 1 Then

    result = tree.SetVisibleInfoTypesEx( views, schematics, formboardSheetIds.ToArray()
    If result = 0 Then
        message = "Error setting visible information types for tree " & treeName & " ( " & treeId & " )
    Else
        message = "Visible information types for tree " & treeName & " ( " & treeId & " )
    End If
    e3Application.PutInfo 0, message           'output result of operation

End If

End If

Set tree = Nothing
Set sheet = Nothing
Set job = Nothing
Set formboardSheetIds = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2016-17.00.

---

## See Also

- [e3Tree - Overview](#)
  - [Schematic](#)
  - [GetVisibleInfoTypesEx\(\)](#)
  - [SetVisibleInfoTypes\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Tree.SetVisibleObjectTypes( *type\_array* )**

## Syntax

**Integer** SetVisibleObjectTypes( [in]**2D-Array** *type\_array* )

## Description

Sets the project tree item's visible object types and their display information settings.

## Parameters

Type	Parameter	Description
[in] <b>2D-Array</b>	<i>type_array</i>	Array containing the object type information

## Return Values

Value	Status	Description
1	Success	Tree item's visible object types and their display information are applied
0	Failure	Error occurred
-1	Failure	Tree not found
-4	Failure	<i>type_array</i> is invalid

## Remarks

The tree item must have at least one visible object type.

*type\_array* is a 1-based array.

Each *type\_array* **2D-Array** item represents a visible object type and should contain the following elements:

Index	Type	Description
-------	------	-------------

Indicates the object type

The following values are possible:

		Value	Description
0	Integer	1	Device
		2	Symbol
		3	Symbol pin
		4	Model
		5	Model pin
		6	Sheet
		7	Field
		8	Functional unit
		9	Functional port
			Slot
		10	Available since v2020-21.12
			Contour
11	Available since v2020-21.12		
	Indicates if the column information consists of a predefined value or an attribute value		
1	String	If "P", the information column contains a predefined value	
		If "A", the information column contains an attribute value	
		If "<Empty>", there is no information column	
2	String	Predefined value name or attribute name	
3	Integer	Display flag values should be a combination of the following values:	
		Flag	Description
			Placed objects
		1	Valid for symbol and model objects
			Unplaced objects
		2	Valid for symbol and model objects
		4	Dynamic objects
			Valid for contour objects



## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project and selecting a project tree item.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set tree = job.CreateTreeObject()

Const DEVICE_TYPE = 1
Const SYMBOL_TYPE = 2
Const MODEL_TYPE = 4
Const PLACED = 1

Dim treeObjectTypes( 4, 3 )

treeObjectTypes( 1, 0 ) = DEVICE_TYPE
treeObjectTypes( 1, 1 ) = "P"
treeObjectTypes( 1, 2 ) = "<Component name>"
treeObjectTypes( 1, 3 ) = 0

treeObjectTypes( 2, 0 ) = SYMBOL_TYPE
treeObjectTypes( 2, 1 ) = "A"
treeObjectTypes( 2, 2 ) = "Description"
treeObjectTypes( 2, 3 ) = PLACED

treeObjectTypes( 3, 0 ) = MODEL_TYPE
treeObjectTypes( 3, 1 ) = "P"
treeObjectTypes( 3, 2 ) = "<Model name>"
treeObjectTypes( 3, 3 ) = PLACED

treeId = job.GetActiveTreeID()
If treeId > 0 Then

    tree.SetId treeId
    treeName = tree.GetName()

    result = tree.SetVisibleObjectTypes( treeObjectTypes )
    Select Case result
    Case 1
        message = "Visible object types for tree " & treeName & " ( " & treeId & " ) set"
    Case 0
        message = "Error setting visible object types for tree " & treeName & " ( " & treeId & " )"
    Case -1
        message = "Error setting visible object types: tree not found"
    End Select
End If

```

## SetAsMaster - e3Symbol

```
Case -4
    message = "Error setting visible object types for tree " & treeName & " ( " & treeName & " ) "
End Select
e3Application.PutInfo 0, message 'output result of operation

End If

Set tree = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

Modified in v2020-21.12.

---

## See Also

- [e3Tree - Overview](#)
  - [GetVisibleObjectTypes\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Tree.ViewSignalTree( *bShowTree* )**

## Syntax

**Integer** ViewSignalTree( [in]**Boolean** *bShowTree* )

## Description

Gets a unique value identifying the signal tree item and shows or hides it.

## Parameters

Type	Parameter	Description
		Indicates if the signal tree should be displayed
[in] <b>Boolean</b>	<i>bShowTree</i>	If <b>True</b> , the signal tree is shown
		If <b>False</b> , the signal tree is hidden

## Return Values

Value	Status	Description
> 0	Success	Signal tree identifier
0	Failure	Error occurred

## Remarks

If successful, the signal tree item identifier will be set as the current tree item.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
```

## SetAsMaster - e3Symbol

```
Set job = e3Application.CreateJobObject()
Set tree = job.CreateTreeObject()

Const SHOW_SIGNAL_TREE = 1

result = tree.ViewSignalTree( SHOW_SIGNAL_TREE )
If result > 0 Then
    treeName = tree.GetName()
    message = "Signal tree " & treeName & " ( " & result & " ) is displayed"
Else
    message = "Error displaying the signal tree"
End If
e3Application.PutInfo 0, message          'output result of operation

Set tree = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

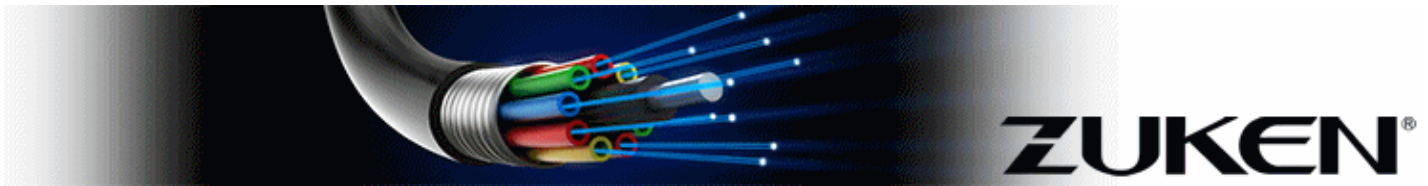
Introduced in v2011-10.20.

---

## See Also

- [e3Tree - Overview](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

## e3UserMenuItem

# Description

Encapsulates the functionality for retrieving and modifying information for the **E<sup>3</sup>.series** user menu items.

## e3UserMenuItem Construction Functions

Function	Description
<u><a href="#">e3Application.CreateMenuItemObject()</a></u>	Creates an instance of e3UserMenuItem
<u><a href="#">e3Clone.CreateMenuItemObject()</a></u>	Creates an instance of e3UserMenuItem

## Item Creation/Destruction Functions

Function	Description
<u><a href="#">Create()</a></u>	Creates a new user menu item
<u><a href="#">CreateContextSeparator()</a></u>	Creates a new separator menu item for the context menu
<u><a href="#">CreateContextUserTool()</a></u>	Creates a new user tool menu item for the context menu
<u><a href="#">CreateSeparator()</a></u>	Creates a new separator menu item
<u><a href="#">CreateUserTool()</a></u>	Creates a new user tool menu item
<u><a href="#">Delete()</a></u>	Deletes the current menu item
<u><a href="#">DeleteContext()</a></u>	Deletes the current context menu item
<u><a href="#">DeleteUserTool()</a></u>	Deletes the current menu item and associated user tool

## Retrieval Functions

Function	Description
<u><a href="#">GetCommand()</a></u>	Gets the path and name of the menu item's user tool
<u><a href="#">GetDescription()</a></u>	Gets the description of the menu item
<u><a href="#">GetEnable()</a></u>	Gets the menu item active status
<u><a href="#">GetFolder()</a></u>	Gets the menu item user tool's working folder
<u><a href="#">GetId()</a></u>	Gets the identifier of the current menu item

<u>GetImage()</u>	Gets the menu item's icon path and file name
<u>GetParameters()</u>	Gets the menu item's argument values to be passed into the user tool
<u>GetShortCut()</u>	Gets the menu item's short cut key combination
<u>GetText()</u>	Gets the menu item's display text
<u>GetType()</u>	Gets a value representing the type of the menu item
<u>GetVisible()</u>	Gets the executable window state of the menu item's user tool when run in console execution mode
<u>GetWaitForEndOfCommand()</u>	<b>Deprecated</b> Indicates whether the <b>E<sup>3</sup>.series</b> process is paused during the execution life time of the user tool
<u>IsDeleted()</u>	Indicates if the system menu item has been deleted

## Modification Functions

Function	Description
<u>SetCommand()</u>	Sets the path and name of the menu item's user tool
<u>SetDescription()</u>	Sets the description of the menu item
<u>SetEnable()</u>	Sets the menu item active status
<u>SetFolder()</u>	Sets the menu item user tool's working folder
<u>SetId()</u>	Sets a menu item as the current item
<u>SetImage()</u>	Sets the menu item's icon path and file name
<u>SetParameters()</u>	Sets the menu item's argument values to be passed into the user tool
<u>SetShortCut()</u>	Sets the menu item's short cut key combination
<u>SetText()</u>	Sets the menu item's display text
<u>SetVisible()</u>	Sets the executable window state of the menu item's user tool when run in console execution mode
<u>SetWaitForEndOfCommand()</u>	<b>Deprecated</b> Sets whether the <b>E<sup>3</sup>.series</b> process is paused during the execution life time of the user tool

## Process Functions

Function	Description
<u>UnDelete()</u>	<b>Deprecated</b> Restores the deleted menu item
<u>UpdateUserInterface()</u>	Updates the <b>E<sup>3</sup>.series</b> menus and short cuts

## Remarks

User tools can be scripts or executables.

## Examples

The best results from the example can be achieved by opening **E<sup>3</sup>.series**.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )

Dim userMenuItemIds : userMenuItemIds = Array( 1 )

userMenuItemCount = GetContextToolIds( userMenuItemIds )
If userMenuItemCount = 0 Then
    e3Application.PutInfo 0, "No context user menu items defined"
Else
    e3Application.PutInfo 0, userMenuItemCount & " context user menu items found:"
    For menuItemIndex = 1 To userMenuItemCount
        e3Application.PutInfo 0, userMenuItemIds( menuItemIndex )
    Next
End If

freeMenuItemId = GetUnusedUserToolId()
If result = 0 Then
    message = "No unused user tool identifier found"
Else
    message = "Next unused user tool menu item identifier is " & freeMenuItemId
End If
e3Application.PutInfo 0, message

result = GetUnusedSeparatorId()
If result = 0 Then
    message = "No unused separator identifier found"
Else
    message = "Next unused separator menu item identifier is " & result
End If
e3Application.PutInfo 0, message          'output result of operation

'gets an array of context tool menu item identifiers
Function GetContextToolIds( ByRef ids )

    Const CONTEXT_TOOL = 3
    Dim menuItemCount : menuItemCount = 0

    userMenuItemCount = e3Application.GetUserMenuItemIds( userMenuItemIds )
    If userMenuItemCount > 0 Then

        Dim menuItem : Set menuItem = e3Application.CreateMenuItemObject()
        ReDim ids( userMenuItemCount )          'set array capacity to hold maximum number of

```

## SetAsMaster - e3Symbol

```

    For menuItemIndex= 1 To userMenuItemCount

        menuItemId = menuItem.SetId( userMenuItemIds( menuItemIndex ) )
        menuItemType = menuItem.GetType()
        If menuItemType = CONTEXT_TOOL Then
            menuItemCount = menuItemCount + 1           'increment count
            ids( menuItemCount ) = menuItemId           'assign id
        End If
    Next
End If

Set menuItem = Nothing
ReDim Preserve ids( menuItemCount )           'remove unpopulated elements

GetContextToolIds = menuItemCount

End Function

'gets the next user tool menu item identifier not in used
Function GetUnusedUserToolId()

    menuItemId = 0

    Const MINIMUM_USER_TOOL_ID = 1
    Const MAXIMUM_USER_TOOL_ID = 100

    Dim menuItem : Set menuItem = e3Application.CreateMenuItemObject()

    For menuItemId = MINIMUM_USER_TOOL_ID To MAXIMUM_USER_TOOL_ID           'user tool ids

        menuItem.SetId menuItemId
        menuItemType = menuItem.GetType()
        If menuItemType = 0 Then
            Exit For
        End If
    Next

    Set menuItem = Nothing

    GetUnusedUserToolId = menuItemId

End Function

'gets the next separator menu item identifier not in used
Function GetUnusedSeparatorId()

    menuItemId = 0

```



```
Const MINIMUM_SEPARATOR_TOOL_ID = -50
Const MAXIMUM_SEPARATOR_TOOL_ID = -1

Dim menuItem : Set menuItem = e3Application.CreateMenuItemObject()

For menuItemId = MAXIMUM_SEPARATOR_TOOL_ID To MINIMUM_SEPARATOR_TOOL_ID Step -1

    menuItem.SetId menuItemId
    menuItemType = menuItem.GetType()
    If menuItemType = 0 Then
        Exit For
    End If
Next

Set menuItem = Nothing

GetUnusedSeparatorId = menuItemId

End Function

Set e3Application = Nothing
```

## Version Information

Introduced in v2011-10.00.

---

## See Also

- [Classes - Overview](#)
  - [e3Application - Overview](#)
  - [e3Clone - Overview](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3UserMenuItem.Create( *id, text, command, parameters, folder, image, shortcut, Visible, wait, flags* )**

## Syntax

**Integer** Create( [in]**Integer** *id*, [in]**String** *text*, [in]**String** *command*, [in]**String** *parameters*, [in]**String** *folder*, [in]**String** *image*, [in]**String** *shortcut*, [in]**Integer** *Visible*, [in]**Integer** *wait*, [in][optional]**Integer** *flags* )

## Description

Creates a new user menu item.

## Parameters

Type	Parameter	Description
[in] <b>Integer</b>	id	Identifier of the user menu item
[in] <b>String</b>	text	Display text and menu position See <a href="#">Menu Item Text</a> for possible values
[in] <b>String</b>	command	Path and name of the user tool
[in] <b>String</b>	parameters	Argument values to be passed into the user tool
[in] <b>String</b>	folder	Path of the working folder of the user tool
[in] <b>String</b>	image	Path and name of a Windows Bitmap (.bmp) file for the display icon
[in] <b>String</b>	shortcut	Short cut key combination See <a href="#">Short Cut</a> for possible values
[in] <b>Integer</b>	Visible	Executable window state when run in console execution mode  If 0, the executable window is hidden  If 1, the executable window is shown

## SetAsMaster - e3Symbol

If 2, the executable window is minimized

If 3, the executable window is maximized

Indicates whether the **E<sup>3</sup>.series** process should pause until the user tool execution is finished

[in]Integer wait

Since v2011-10.00 this functionality is no longer available

User menu item option flags

[in][optional]Integer flags

Default value is 1

## Return Values

Value	Status	Description
-------	--------	-------------

>0	Success	Identifier of the new user menu item
----	---------	--------------------------------------

0	Failure	Error occurred
---	---------	----------------

## Remarks

*id* is user-defined and should adhere to the following criteria:

- The value is between 1 and 100 if the user menu item is a user tool
- The value is between -1 and -50 if the user menu item is a separator
- The value should not already be in use
- The value should be the next available identifier in sequence

If the *parameters* value is "<Empty>", no arguments are passed into the user tool.

If the *folder* value is "<Empty>", the working folder is the current folder of **E<sup>3</sup>.series** or in the project folder if a project is open.

*image* should refer to a Windows Bitmap (.bmp) file. If the *image* value is "<Empty>", the icon from the user tool defined in *command* is used.

If the *shortcut* value is "<Empty>", no short cut key combination is created.

e3Application.SetModalWindow() or e3Clone.SetModalWindow() can be used to ensure the **E<sup>3</sup>.series** process is paused during the user tool's execution.

*flags* should be one of the following values:

Value	Description
-------	-------------

1	Menu item is created in the menu
---	----------------------------------

2	Menu item is created in the context menu
---	--

4	Menu item is created in the database context menu
---	---

Only valid for user tools

Remarks

Available since v2012-11.02

If successful, the new menu item is assigned as the current user menu item.

## Examples

The best results from the example can be achieved by opening **E<sup>3</sup>.series**.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set menuItem = e3Application.CreateMenuItemObject()

Const NO_MENU_ITEM_TYPE = 0
Const SHOW_CONSOLE = 1
Const MENU_ITEM = 1

Dim menuItemId : menuItemId = GetUnusedUserToolId()           'value should not already be in
Dim menuItemText : menuItemText = "Add-ons\Kambei Tool"       'menu names are specific to
Dim userToolFile : userToolFile = "c:\temp\script.vbs"       'should be a valid path to t
Dim userToolArguments : userToolArguments = ""
Dim workingFolder : workingFolder = ""
Dim menuItemIcon : menuItemIcon = ""
Dim menuItemShortCut : menuItemShortCut = ""
Dim consoleWindowState : consoleWindowState = SHOW_CONSOLE
Dim displayFlags : displayFlags = MENU_ITEM

result = menuItem.Create( menuItemId, menuItemText, userToolFile, userToolArguments, workingFolder, menuItemIcon, menuItemShortCut, consoleWindowState, displayFlags )
If result = 0 Then
    message = "Error creating new user menu item"
Else
    message = "New user menu item " & menuItemText & " created: " & result
End If
e3Application.PutInfo 0, message           'output result of operation

Set menuItem = Nothing

'gets the next user tool menu item identifier not in used
Function GetUnusedUserToolId()

    menuItemId = 0

    Const MINIMUM_USER_TOOL_ID = 1
    Const MAXIMUM_USER_TOOL_ID = 100

    Dim menuItem : Set menuItem = e3Application.CreateMenuItemObject()

    For menuItemId = MINIMUM_USER_TOOL_ID To MAXIMUM_USER_TOOL_ID           'user tool ids

```

```
menuItem.SetId menuItemId
menuItem.Type = menuItem.GetType()
If menuItem.Type = 0 Then
    Exit For
End If
Next

Set menuItem = Nothing

GetUnusedUserToolId = menuItemId

End Function
```

```
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

Modified in v2011-10.00.

Modified in v2012-11.02.

---

## See Also

- [e3UserMenuItem - Overview](#)
  - [Menu Item Text](#)
  - [Short Cut](#)
  - [e3Application.SetModalWindow\(\)](#)
  - [e3Close.SetModalWindow\(\)](#)
  - [CreateContextSeparator\(\)](#)
  - [CreateContextUserTool\(\)](#)
  - [CreateSeparator\(\)](#)
  - [CreateUserTool\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3UserMenuItem.CreateContextSeparator( *text* )**

## Syntax

**Integer** CreateContextSeparator( [in]**String** *text* )

## Description

Creates a new separator menu item for the context menu.

## Parameters

Type	Parameter	Description
		Display text and menu position
[in] <b>String</b>	<i>text</i>	See <u>Menu Item Text</u> for possible values

## Return Values

Value	Status	Description
<0	Inconclusive	Identifier of the new separator context menu item or an error occurred
0	Failure	Error occurred

## Remarks

Separator context menu items should be placed between existing normal user context menu items.

Due caution is recommended on relying on the return value of less than 0 meaning the separator context menu item has been created since this also could mean an error has occurred.

If successful, the new separator context menu item is assigned as the current user menu item.

## Examples

The best results from the example can be achieved by opening **E<sup>3</sup>.series**.

Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set menuItem = e3Application.CreateMenuItemObject()

Dim menuItemPosition : menuItemPosition = "@1"           'menu names are specific to the Appl

result = menuItem.CreateContextSeparator( menuItemPosition )
If result = 0 Then
    message = "Error creating new separator context menu item"
Else
    message = "New separator context menu item " & result
End If
e3Application.PutInfo 0, message           'output result of operation

Set menuItem = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3UserMenuItem - Overview](#)
  - [Menu Item Text](#)
  - [Create\(\)](#)
  - [CreateContextUserTool\(\)](#)
  - [CreateSeparator\(\)](#)
  - [CreateUserTool\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3UserMenuItem.CreateContextUserTool( *text*, *command* )**

## Syntax

**Integer** CreateContextUserTool( [in]**String** *text*, [in]**String** *command* )

## Description

Creates a new user tool menu item for the context menu.

## Parameters

Type	Parameter	Description
		Display text and context menu position
[in] <b>String</b>	<i>text</i>	See <u>Menu Item Text</u> for possible values
[in] <b>String</b>	<i>command</i>	Path and name of the user tool

## Return Values

Value	Status	Description
>0	Success	Identifier of the new user tool context menu item
0	Failure	Error occurred

## Remarks

If successful, the new user tool context menu item is assigned as the current user menu item.

## Examples

The best results from the example can be achieved by opening **E<sup>3</sup>.series**.

Visual Basic Script



## SetAsMaster - e3Symbol

```
Set e3Application = CreateObject( "CT.Application" )
Set menuItem = e3Application.CreateMenuItemObject()

Dim menuItemText : menuItemText = "Kambei Tool"           'menu names are specific to the App
Dim userToolFile : userToolFile = "c:\temp\script.vbs"     'should be a valid path to t

result = menuItem.CreateContextUserTool( menuItemText, userToolFile )
If result = 0 Then
    message = "Error creating new user tool menu item"
Else
    message = "New user tool menu item " & result & " created"
End If
e3Application.PutInfo 0, message           'output result of operation

Set menuItem = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3UserMenuItem - Overview](#)
  - [Menu Item Text](#)
  - [Create\(\)](#)
  - [CreateContextSeparator\(\)](#)
  - [CreateSeparator\(\)](#)
  - [CreateUserTool\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3UserMenuItem.CreateSeparator( *text* )**

## Syntax

**Integer** CreateSeparator( [in]**String** *text* )

## Description

Creates a new separator menu item.

## Parameters

Type	Parameter	Description
		Display text and menu position
[in] <b>String</b>	<i>text</i>	See <u>Menu Item Text</u> for possible values

## Return Values

Value	Status	Description
<0	Inconclusive	Identifier of the new separator menu item or an error occurred
0	Failure	Error occurred

## Remarks

Separator menu items should be placed between existing normal user menu items.

Due caution is recommended on relying on the return value of less than 0 meaning the separator menu item has been created since this also could mean an error has occurred.

If successful, the new separator menu item is assigned as the current user menu item.

## Examples

The best results from the example can be achieved by opening **E<sup>3</sup>.series**.

## Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set menuItem = e3Application.CreateMenuItemObject()

Dim menuItemPosition : menuItemPosition = "Add-ons\@1"           'menu names are specific to

result = menuItem.CreateSeparator( menuItemPosition )
If result = 0 Then
    message = "Error creating new separator menu item"
Else
    message = "New separator menu item " & result
End If
e3Application.PutInfo 0, message           'output result of operation

Set menuItem = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3UserMenuItem - Overview](#)
  - [Menu Item Text](#)
  - [Create\(\)](#)
  - [CreateContextSeparator\(\)](#)
  - [CreateContextUserTool\(\)](#)
  - [CreateUserTool\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3UserMenuItem.CreateUserTool( *text*, *command* )**

## Syntax

**Integer** CreateUserTool( [in]**String** *text*, [in]**String** *command* )

## Description

Creates a new user tool menu item.

## Parameters

Type	Parameter	Description
		Display text and menu position
[in] <b>String</b>	text	See <u>Menu Item Text</u> for possible values
[in] <b>String</b>	command	Path and name of the user tool

## Return Values

Value	Status	Description
>0	Success	Identifier of the new user tool menu item
0	Failure	Error occurred

## Remarks

If successful, the new user tool menu item is assigned as the current user menu item.

## Examples

The best results from the example can be achieved by opening **E<sup>3</sup>.series**.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
```

## SetAsMaster - e3Symbol

```
Set menuItem = e3Application.CreateMenuItemObject()

Dim menuItemText : menuItemText = "Add-ons\Kambei Tool"      'menu names are specific to
Dim userToolFile : userToolFile = "c:\temp\script.vbs"      'should be a valid path to t

result = menuItem.CreateUserTool( menuItemText, userToolFile )
If result = 0 Then
    message = "Error creating new user tool menu item"
Else
    message = "New user tool menu item " & result & " created"
End If
e3Application.PutInfo 0, message      'output result of operation

Set menuItem = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3UserMenuItem - Overview](#)
  - [Menu Item Text](#)
  - [Create\(\)](#)
  - [CreateContextSeparator\(\)](#)
  - [CreateContextUserTool\(\)](#)
  - [CreateSeparator\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3UserMenuItem.Delete()**

## Syntax

**Integer** Delete()

## Description

Deletes the current menu item.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
1	Success	Menu item was deleted
0	Failure	Error occurred

## Remarks

If the current menu item is a user-defined menu item, it is deleted.

If the menu item is associated with a user tool, the user tool is not removed.

If the current menu item is a standard **E<sup>3</sup>.series** menu item, it is deactivated in any tool bars and is no longer displayed on menus. It can be reactivated using the **Customize** dialog.

## Examples

The best results from the example can be achieved by opening **E<sup>3</sup>.series**.

Visual Basic Script

## SetAsMaster - e3Symbol

```
Set e3Application = CreateObject( "CT.Application" )
Set menuItem = e3Application.CreateMenuItemObject()

userMenuItemCount = e3Application.GetUserMenuItemIds( userMenuItemIds )
If userMenuItemCount > 0 Then

    menuItemId = menuItem.SetId( userMenuItemIds( userMenuItemCount ) )      'set the la
    result = menuItem.Delete()
    If result = 0 Then
        message = "Error deleting menu item " & menuItemId
    Else
        message = "Menu item " & menuItemId & " deleted"
    End If
    e3Application.PutInfo 0, message      'output result of operation

End If

Set menuItem = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3UserMenuItem - Overview](#)
  - [DeleteContext\(\)](#)
  - [DeleteUserTool\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

## e3UserMenuItem.DeleteContext()

### Syntax

**Integer** DeleteContext()

### Description

Deletes the current context menu item.

### Parameters

No parameters defined.

### Return Values

Value	Status	Description
1	Inconclusive	Context menu item was deleted or an error occurred
0	Failure	Error occurred

### Remarks

If the current menu item is a user-defined context menu item, it is deleted.

Due caution is recommended on relying on the return value 1 meaning the context menu item has been deleted since this also could mean an error has occurred. This can occur if the current menu item is not a context menu item.

If the menu item is associated with a user tool, the user tool is not removed.

### Examples

The best results from the example can be achieved by opening *E<sup>3</sup>.series*.

Visual Basic Script



## SetAsMaster - e3Symbol

```

Set e3Application = CreateObject( "CT.Application" )
Set menuItem = e3Application.CreateMenuItemObject()

Dim userMenuItemIds : userMenuItemIds = Array( 1 )

userMenuItemCount = GetContextToolIds( userMenuItemIds )
If userMenuItemCount > 0 Then

    menuItemId = menuItem.SetId( userMenuItemIds( userMenuItemCount ) )      'set the la
    result = menuItem.DeleteContext()
    If result = 0 Then
        message = "Error deleting context menu item " & menuItemId
    Else
        message = "Context menu item " & menuItemId & " deleted"
    End If
    e3Application.PutInfo 0, message      'output result of operation

End If

Set menuItem = Nothing

'gets an array of context tool menu item identifiers
Function GetContextToolIds( ByRef ids )

    Const MINIMUM_USER_TOOL_ID = 1
    Const MAXIMUM_USER_TOOL_ID = 100
    Const CONTEXT_TOOL = 3

    Dim menuItemCount : menuItemCount = 0
    Dim menuItem : Set menuItem = e3Application.CreateMenuItemObject()

    ReDim ids( MAXIMUM_USER_TOOL_ID )      'set array capacity to hold maximum number of
    For menuItemId = MINIMUM_USER_TOOL_ID To MAXIMUM_USER_TOOL_ID      'user tool ids

        menuItem.SetId menuItemId
        menuItemType = menuItem.GetType()
        If menuItemType = CONTEXT_TOOL Then
            menuItemCount = menuItemCount + 1      'increment count
            ids( menuItemCount ) = menuItemId      'assign id
        End If
    Next

    Set menuItem = Nothing
    ReDim Preserve ids( menuItemCount )      'remove unpopulated elements

    GetContextToolIds = menuItemCount

End Function

```

```
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3MenuItem - Overview](#)
  - [Delete\(\)](#)
  - [DeleteUserTool\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

## e3UserMenuItem.DeleteUserTool()

### Syntax

**Integer** DeleteUserTool()

### Description

Deletes the current menu item and associated user tool.

### Parameters

No parameters defined.

### Return Values

Value	Status	Description
1	Success	Menu item and user tool were deleted
0	Failure	Error occurred

### Remarks

If successful, the current menu item is set to 0.

### Examples

The best results from the example can be achieved by opening **E<sup>3</sup>.series**.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set menuItem = e3Application.CreateMenuItemObject()

menuItem.DeleteUserTool()

userMenuItemCount = e3Application.GetUserMenuItemIds( userMenuItemIds )
If userMenuItemCount > 0 Then
```

```
menuItemId = menuItem.SetId( userMenuItemIds( userMenuItemCount ) )      'set the la
result = menuItem.DeleteUserTool()
If result = 0 Then
    message = "Error deleting menu item " & menuItemId
Else
    message = "Menu item " & menuItemId & " deleted"
End If
e3Application.PutInfo 0, message      'output result of operation

End If

Set menuItem = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2011-10.10.

---

## See Also

- [e3UserMenuItem - Overview](#)
  - [Delete\(\)](#)
  - [DeleteContext\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3UserMenuItem.GetCommand()**

## Syntax

*String* GetCommand()

## Description

Gets the path and name of the menu item's user tool.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
"<User Tool>"	Success	Path and name of the menu item's user tool
"<Empty>"	Inconclusive	No user tool path and name or an error occurred

## Remarks

Due caution is recommended on relying on the return value of "<Empty>" meaning no user tool path or name since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening *E<sup>3</sup>.series*.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set menuItem = e3Application.CreateMenuItemObject()

userMenuItemCount = e3Application.GetUserMenuItemIds( userMenuItemIds )
```

## SetAsMaster - e3Symbol

```
If userMenuItemCount > 0 Then

    For menuItemIndex = 1 To userMenuItemCount
        menuItemId = menuItem.SetId( userMenuItemIds( menuItemIndex ) )
        menuItemText = menuItem.GetText()

        result = menuItem.GetCommand()
        If Len( "" & result ) = 0 Then
            message = "Menu item " & menuItemText & " ( " & menuItemId & " ) has no command"
        Else
            message = "Menu item " & menuItemText & " ( " & menuItemId & " ) command is " & result
        End If
        e3Application.PutInfo 0, message          'output result of operation

    Next
End If

Set menuItem = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3UserMenuItem - Overview](#)
  - [SetCommand\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

## e3UserMenuItem.GetDescription()

### Syntax

*String* GetDescription()

### Description

Gets the description of the menu item.

### Parameters

No parameters defined.

### Return Values

Value	Status	Description
"<User Tool>"	Success	Menu item description
"<Empty>"	Inconclusive	No menu item description or an error occurred

### Remarks

Due caution is recommended on relying on the return value of "<Empty>" meaning no menu item description since this also could mean an error has occurred.

### Examples

The best results from the example can be achieved by opening *E<sup>3</sup>.series*.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set menuItem = e3Application.CreateMenuItemObject()

userMenuItemCount = e3Application.GetUserMenuItemIds( userMenuItemIds )
```

## SetAsMaster - e3Symbol

```
If userMenuItemCount > 0 Then

    For menuItemIndex = 1 To userMenuItemCount
        menuItemId = menuItem.SetId( userMenuItemIds( menuItemIndex ) )
        menuItemText = menuItem.GetText()

        result = menuItem.GetDescription()
        If Len( "" & result ) = 0 Then
            message = "Menu item " & menuItemText & " ( " & menuItemId & " ) has no description"
        Else
            message = "Menu item " & menuItemText & " ( " & menuItemId & " ) description is " & result
        End If
        e3Application.PutInfo 0, message          'output result of operation

    Next
End If

Set menuItem = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3UserMenuItem - Overview](#)
  - [SetDescription\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).





v2022-23.00

**e3UserMenuItem.GetEnable()**

## Syntax

Integer GetEnable()

## Description

Gets the menu item active status.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
1	Success	Menu item is active
0	Success	Menu item is inactive
-1	Failure	Error occurred

## Remarks

If the current menu item is a system menu or a system separator, the menu item is hidden when the return value is 0 and shown when the return value is 1.

If the current menu item is a type other than system menu or system separator, the menu item is deactivated when the return value is 0 and activated when the return value is 1.

## Examples

The best results from the example can be achieved by opening **E<sup>3</sup>.series**.

Visual Basic Script

## SetAsMaster - e3Symbol

```
Set e3Application = CreateObject( "CT.Application" )
Set menuItem = e3Application.CreateMenuItemObject()

userMenuItemCount = e3Application.GetUserMenuItemIds( userMenuItemIds )
If userMenuItemCount > 0 Then

    For menuItemIndex = 1 To userMenuItemCount
        menuItemId = menuItem.SetId( userMenuItemIds( menuItemIndex ) )
        menuItemText = menuItem.GetText()

        result = menuItem.GetEnable()
        Select Case result
            Case 1
                message = "Menu item " & menuItemText & " ( " & menuItemId & " ) is active"
            Case 0
                message = "Menu item " & menuItemText & " ( " & menuItemId & " ) is inactive"
            Case -1
                message = "Error getting enable status for menu item " & menuItemText & " ( " & menuItemId & " )"
        End Select
        e3Application.PutInfo 0, message          'output result of operation

    Next
End If

Set menuItem = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3UserMenuItem - Overview](#)
  - [SetEnable\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3UserMenuItem.GetFolder()**

## Syntax

*String* GetFolder()

## Description

Gets the menu item user tool's working folder.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
"<Path>"	Success	Path of the user tool's working folder
"<Empty>"	Inconclusive	No working folder or an error occurred

## Remarks

Due caution is recommended on relying on the return value of "<Empty>" meaning no working folder since this also could mean an error has occurred.

If the return value is "<Empty>" and no error has occurred, the working folder is the current folder of *E<sup>3</sup>.series* or in the project folder when a project is open.

## Examples

The best results from the example can be achieved by opening *E<sup>3</sup>.series*.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
```

## SetAsMaster - e3Symbol

```
Set menuItem = e3Application.CreateMenuItemObject()

userMenuItemCount = e3Application.GetUserMenuItemIds( userMenuItemIds )
If userMenuItemCount > 0 Then

    For menuItemIndex = 1 To userMenuItemCount
        menuItemId = menuItem.SetId( userMenuItemIds( menuItemIndex ) )
        menuItemText = menuItem.GetText()

        result = menuItem.GetFolder()
        If Len( "" & result ) = 0 Then
            message = "Menu item " & menuItemText & " ( " & menuItemId & " ) has no working folder"
        Else
            message = "Menu item " & menuItemText & " ( " & menuItemId & " ) working folder is " & result
        End If
        e3Application.PutInfo 0, message           'output result of operation

    Next
End If

Set menuItem = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3UserMenuItem - Overview](#)
  - [SetFolder\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3UserMenuItem.GetId()**

## Syntax

**Integer** GetId()

## Description

Gets the identifier of the current menu item.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
> 0	Success	Current menu item identifier
0	Failure	No menu item

## Remarks

The function returns the identifier value successfully set by SetId(), Create(), CreateContextSeparator(), CreateContextUserTool(), CreateSeparator() or CreateUserTool() unless the item no longer exists.

## Examples

The best results from the example can be achieved by opening **E<sup>3</sup>.series**.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set menuItem = e3Application.CreateMenuItemObject()
```

## SetAsMaster - e3Symbol

```
userMenuItemCount = e3Application.GetUserMenuItemIds( userMenuItemIds )
If userMenuItemCount > 0 Then

    For menuItemIndex = 1 To userMenuItemCount
        menuItem.SetId( userMenuItemIds( menuItemIndex ) )

        result = menuItem.GetId()
        If result = 0 Then
            message = result & " not assigned as a menu item"
        Else
            menuType = menuItem.GetType()
            If menuType <> 0 Then
                message = result & " assigned as a menu item of type " & menuType
            Else
                message = result & " assigned as a menu item"
            End If
        End If
        e3Application.PutInfo 0, message           'output result of operation
    Next
End If

Set menuItem = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3UserMenuItem - Overview](#)
  - [Create\(\)](#)
  - [CreateContextSeparator\(\)](#)
  - [CreateContextUserTool\(\)](#)
  - [CreateSeparator\(\)](#)
  - [CreateUserTool\(\)](#)
  - [SetId\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3UserMenuItem.GetImage()**

## Syntax

*String* GetImage()

## Description

Gets the menu item's icon path and file name.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
"<File Path>"	Success	Path and file name of the icon image
"<Empty>"	Inconclusive	No icon image path and file name or an error occurred

## Remarks

Due caution is recommended on relying on the return value of "<Empty>" meaning no icon image path and file name since this also could mean an error has occurred.

If the return value is "<Empty>" and no error has occurred, the icon from the user tool defined as the command is in use.

## Examples

The best results from the example can be achieved by opening *E<sup>3</sup>.series*.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
```

## SetAsMaster - e3Symbol

```
Set menuItem = e3Application.CreateMenuItemObject()

userMenuItemCount = e3Application.GetUserMenuItemIds( userMenuItemIds )
If userMenuItemCount > 0 Then

    For menuItemIndex = 1 To userMenuItemCount
        menuItemId = menuItem.SetId( userMenuItemIds( menuItemIndex ) )
        menuItemText = menuItem.GetText()

        result = menuItem.GetImage()
        If Len( "" & result ) = 0 Then
            message = "Menu item " & menuItemText & " ( " & menuItemId & " ) has no icon image"
        Else
            message = "Menu item " & menuItemText & " ( " & menuItemId & " ) icon image file is"
        End If
        e3Application.PutInfo 0, message           'output result of operation

    Next
End If

Set menuItem = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3UserMenuItem - Overview](#)
  - [SetImage\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).





v2022-23.00

## e3UserMenuItem.GetParameters()

### Syntax

*String* GetParameters()

### Description

Gets the menu item's argument values to be passed into the user tool.

### Parameters

No parameters defined.

### Return Values

Value	Status	Description
"<Parameters>"	Success	Argument values
"<Empty>"	Inconclusive	No argument values or an error occurred

### Remarks

Due caution is recommended on relying on the return value of "<Empty>" meaning no argument values since this also could mean an error has occurred.

### Examples

The best results from the example can be achieved by opening *E<sup>3</sup>.series*.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set menuItem = e3Application.CreateMenuItemObject()

userMenuItemCount = e3Application.GetUserMenuItemIds( userMenuItemIds )
```

## SetAsMaster - e3Symbol

```
If userMenuItemCount > 0 Then

    For menuItemIndex = 1 To userMenuItemCount
        menuItemId = menuItem.SetId( userMenuItemIds( menuItemIndex ) )
        menuItemText = menuItem.GetText()

        result = menuItem.GetParameters()
        If Len( "" & result ) = 0 Then
            message = "Menu item " & menuItemText & " ( " & menuItemId & " ) has no user tool
        Else
            message = "Menu item " & menuItemText & " ( " & menuItemId & " ) user tool argument
        End If
        e3Application.PutInfo 0, message          'output result of operation

    Next
End If

Set menuItem = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3UserMenuItem - Overview](#)
  - [SetParameters\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3UserMenuItem.GetShortCut()**

## Syntax

*String* GetShortCut()

## Description

Gets the menu item's short cut key combination.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
"<Short Cut>"	Success	Short cut key combination
"<Empty>"	Inconclusive	No short cut key combination or an error occurred

## Remarks

See [Short Cut](#) for possible values.

Due caution is recommended on relying on the return value of "<Empty>" meaning no short cut key combination since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening *E<sup>3</sup>.series*.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set menuItem = e3Application.CreateMenuItemObject()
```

```

userMenuItemCount = e3Application.GetUserMenuItemIds( userMenuItemIds )
If userMenuItemCount > 0 Then

    For menuItemIndex = 1 To userMenuItemCount
        menuItemId = menuItem.SetId( userMenuItemIds( menuItemIndex ) )
        menuItemText = menuItem.GetText()

        result = menuItem.GetShortCut()
        If Len( "" & result ) = 0 Then
            message = "Menu item " & menuItemText & " ( " & menuItemId & " ) has no short cut"
        Else
            message = "Menu item " & menuItemText & " ( " & menuItemId & " ) short cut key com"
        End If
        e3Application.PutInfo 0, message          'output result of operation

    Next
End If

Set menuItem = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3UserMenuItem - Overview](#)
  - [Short Cut](#)
  - [SetShortCut\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3UserMenuItem.GetText()**

## Syntax

*String* GetText()

## Description

Gets the menu item's display text.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
"<Text>"	Success	Display text
"<Empty>"	Inconclusive	No display text or an error occurred

## Remarks

Due caution is recommended on relying on the return value of "<Empty>" meaning no display text since this also could mean an error has occurred.

The return value will include the menu and sub-menu names of the menu item.

Since v2011-10.00 only the first display name is returned if there are several user commands.

## Examples

The best results from the example can be achieved by opening *E<sup>3</sup>.series*.

Visual Basic Script

## SetAsMaster - e3Symbol

```
Set e3Application = CreateObject( "CT.Application" )
Set menuItem = e3Application.CreateMenuItemObject()

userMenuItemCount = e3Application.GetUserMenuItemIds( userMenuItemIds )
If userMenuItemCount > 0 Then

    For menuItemIndex = 1 To userMenuItemCount
        menuItemId = menuItem.SetId( userMenuItemIds( menuItemIndex ) )
        menuItemText = menuItem.GetText()

        result = menuItem.GetText()
        If Len( "" & result ) = 0 Then
            message = "Menu item ( " & menuItemId & " ) has no display name"
        Else
            message = "Menu item ( " & menuItemId & " ) display name is " & result
        End If
        e3Application.PutInfo 0, message           'output result of operation

    Next
End If

Set menuItem = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

Modified in v2011-10.00.

---

## See Also

- [e3UserMenuItem - Overview](#)
  - [SetText\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



**v2022-23.00**

**e3UserMenuItem.GetType()**

## Syntax

**Integer** GetType()

## Description

Gets a value representing the type of the menu item.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
> 0	Success	Type value of the menu item
0	Failure	Error occurred

## Remarks

A successful return value will be one of the following values:

Value	Type
1	User tool
2	User separator
3	Context tool
4	Context separator
5	System menu
6	System separator
7	Database context tool

Available since v2012-11.02

Database context separator

8

Available since v2012-11.02

## Examples

The best results from the example can be achieved by opening **E<sup>3</sup>.series**.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set menuItem = e3Application.CreateMenuItemObject()

userMenuItemCount = e3Application.GetUserMenuItemIds( userMenuItemIds )
If userMenuItemCount > 0 Then

    For menuItemIndex = 1 To userMenuItemCount
        menuItemId = menuItem.SetId( userMenuItemIds( menuItemIndex ) )
        menuItemText = menuItem.GetText()

        result = menuItem.GetType()
        Select Case result
            Case 0
                message = "Error getting type for menu item " & menuItemText & " ( " & menuItemId
            Case 1
                message = "Menu item " & menuItemText & " ( " & menuItemId & " ) type is user tool
            Case 2
                message = "Menu item " & menuItemText & " ( " & menuItemId & " ) type is user sepa
            Case 3
                message = "Menu item " & menuItemText & " ( " & menuItemId & " ) type is context t
            Case 4
                message = "Menu item " & menuItemText & " ( " & menuItemId & " ) type is context s
            Case 5
                message = "Menu item " & menuItemText & " ( " & menuItemId & " ) type is system me
            Case 6
                message = "Menu item " & menuItemText & " ( " & menuItemId & " ) type is system se
            Case 7
                message = "Menu item " & menuItemText & " ( " & menuItemId & " ) type is database
            Case 8
                message = "Menu item " & menuItemText & " ( " & menuItemId & " ) type is database
        End Select
        e3Application.PutInfo 0, message          'output result of operation

    Next
End If

Set menuItem = Nothing
Set e3Application = Nothing

```



## Version Information

Introduced in v2009-8.50.

Modified in v2012-11.02.

---

## See Also

- [e3MenuItem - Overview](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3UserMenuItem.GetVisible()**

## Syntax

Integer GetVisible()

## Description

Gets the executable window state of the menu item's user tool when run in console execution mode.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
3	Success	Window state is maximized
2	Success	Window state is minimized
1	Success	Window state is shown
0	Inconclusive	Window state is hidden or an error occurred

## Remarks

If the user tool is not run in console execution mode, this setting has no effect.

## Examples

The best results from the example can be achieved by opening **E<sup>3</sup>.series**.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
```

## SetAsMaster - e3Symbol

```
Set menuItem = e3Application.CreateMenuItemObject()

userMenuItemCount = e3Application.GetUserMenuItemIds( userMenuItemIds )
If userMenuItemCount > 0 Then

    For menuItemIndex = 1 To userMenuItemCount
        menuItemId = menuItem.SetId( userMenuItemIds( menuItemIndex ) )
        menuItemText = menuItem.GetText()

        result = menuItem.GetVisible()
        Select Case result
            Case 3
                message = "Menu item " & menuItemText & " ( " & menuItemId & " ) is maximized in c
            Case 2
                message = "Menu item " & menuItemText & " ( " & menuItemId & " ) is minimized in c
            Case 1
                message = "Menu item " & menuItemText & " ( " & menuItemId & " ) is displayed in c
            Case 0
                message = "Menu item " & menuItemText & " ( " & menuItemId & " ) is hidden in cons
        End Select
        e3Application.PutInfo 0, message          'output result of operation

    Next
End If

Set menuItem = Nothing
Set e3Application = Nothing
```

## Version Information

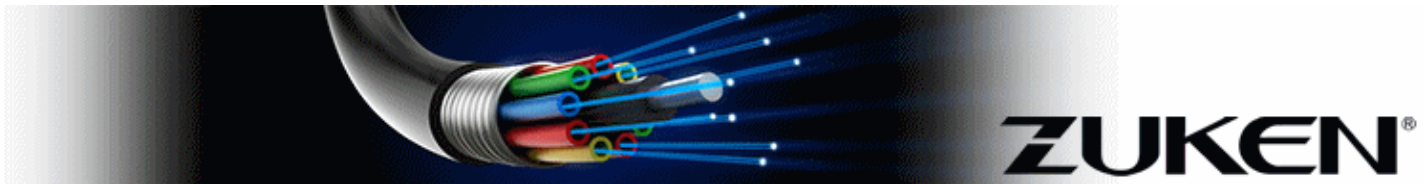
Introduced in v2009-8.50.

---

## See Also

- [e3UserMenuItem - Overview](#)
  - [SetVisible\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3UserMenuItem.WaitForEndOfCommand()**

## Syntax

Integer WaitForEndOfCommand()

## Description

Indicates whether the **E<sup>3</sup>.series** process is paused during the execution life time of the user tool.

Function has been deprecated.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
1	Success	<b>E<sup>3</sup>.series</b> process is paused during the execution life time of the user tool
0	Inconclusive	<b>E<sup>3</sup>.series</b> process runs during the execution life time of the user tool, the functionality is no longer supported or an error has occurred

## Remarks

Since v2011-10.00 this functionality is no longer supported and 0 is always returned.

The functionality can be utilized using [e3Application.GetModalWindow\(\)](#) and [e3Application.SetModalWindow\(\)](#).

## Examples

The best results from the example can be achieved by opening **E<sup>3</sup>.series**.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set menuItem = e3Application.CreateMenuItemObject()

Const NOT_SUPPORTED = 2011      'GetWaitForEndOfCommand() is not supported since this version
e3ApplicationVersion = CInt( e3Application.GetVersion() )      'get e3Application version

userMenuItemCount = e3Application.GetUserMenuItemIds( userMenuItemIds )
If userMenuItemCount > 0 Then

    For menuItemIndex = 1 To userMenuItemCount
        menuItemId = menuItem.SetId( userMenuItemIds( menuItemIndex ) )
        menuItemText = menuItem.GetText()

        result = menuItem.GetWaitForEndOfCommand()
        If result = 0 Then
            If e3ApplicationVersion < NOT_SUPPORTED Then
                message = "Menu item " & menuItemText & " ( " & menuItemId & " ) E3.series runs"
            Else
                message = "Menu item " & menuItemText & " ( " & menuItemId & " ): GetWaitForEndOfCommand() returns 0"
            End If
        Else
            message = "Menu item " & menuItemText & " ( " & menuItemId & " ) E3.series pauses"
        End If
        e3Application.PutInfo 0, message      'output result of operation

    Next
End If

Set menuItem = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

Deprecated in v2011-10.00.

---

## See Also

- [e3UserMenuItem - Overview](#)
- [e3Application.GetModalWindow\(\)](#)

- e3Application.SetModalWindow()
  - SetWaitForEndOfCommand()
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

## e3UserMenuItem.IsDeleted()

### Syntax

**Integer** IsDeleted()

### Description

Indicates if the system menu item has been deleted.

### Parameters

No parameters defined.

### Return Values

Value	Status	Description
1	Success	System menu item has been deleted
0	Success	System menu item has not been deleted
-1	Failure	Error occurred

### Remarks

-1 is returned if the menu item is not a system menu item or a system separator menu item.

### Examples

The best results from the example can be achieved by opening **E<sup>3</sup>.series**.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set menuItem = e3Application.CreateMenuItemObject()
```

## SetAsMaster - e3Symbol

```
userMenuItemCount = e3Application.GetUserMenuItemIds( userMenuItemIds )
If userMenuItemCount > 0 Then

    For menuItemIndex = 1 To userMenuItemCount
        menuItemId = menuItem.SetId( userMenuItemIds( menuItemIndex ) )
        menuItemText = menuItem.GetText()

        result = menuItem.IsDeleted()
        Select Case result
            Case 1
                message = "Menu item " & menuItemText & " ( " & menuItemId & " ) is deleted"
            Case 0
                message = "Menu item " & menuItemText & " ( " & menuItemId & " ) is not deleted"
            Case -1
                message = "Error getting deleted status for menu item " & menuItemText & " ( " & m
        End Select
        e3Application.PutInfo 0, message           'output result of operation

    Next
End If

Set menuItem = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3UserMenuItem - Overview](#)
  - [Delete\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).





v2022-23.00

**SetCommand( *newval* )**

## Syntax

**Integer** SetCommand( [in]**String** *newval* )

## Description

Sets the path and name of the menu item's user tool.

## Parameters

Type	Parameter	Description
[in] <b>String</b>	<i>newval</i>	Path and name of the user tool

## Return Values

Value	Status	Description
1	Success	Path and name of the user tool applied
0	Failure	Error occurred

## Remarks

If *newval* value is "<Empty>", the current command value is removed.

## Examples

The best results from the example can be achieved by opening **E<sup>3</sup>.series**.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set menuItem = e3Application.CreateMenuItemObject()
```

```
Dim userToolFile : userToolFile = "c:\temp\script.vbs"
```

'should be a valid path to t

## SetAsMaster - e3Symbol

```
userMenuItemCount = e3Application.GetUserMenuItemIds( userMenuItemIds )
If userMenuItemCount > 0 Then

    menuItemId = menuItem.SetId( userMenuItemIds( userMenuItemCount ) )      'set the la
    menuItemText = menuItem.GetText()

    result = menuItem.SetCommand( userToolFile )
    If result = 0 Then
        message = "Error setting command of menu item " & menuItemText & " ( " & menuItemId &
    Else
        message = "Menu item " & menuItemText & " ( " & menuItemId & " ) command set to " & us
    End If
    e3Application.PutInfo 0, message      'output result of operation

End If

Set menuItem = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3UserMenuItem - Overview](#)
  - [GetCommand\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3UserMenuItem.SetDescription( *newval* )**

## Syntax

*Integer* SetDescription( [in]*String* *newval* )

## Description

Sets the description of the menu item.

## Parameters

Type	Parameter	Description
[in] <i>String</i>	<i>newval</i>	Description of the menu item

## Return Values

Value	Status	Description
1	Success	Description applied
0	Failure	Error occurred

## Remarks

If *newval* value is "<Empty>", the current description value is removed.

## Examples

The best results from the example can be achieved by opening *E<sup>3</sup>.series*.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set menuItem = e3Application.CreateMenuItemObject()

Dim description : description = "Legatus nec violatur, nec laeditur"
```

## SetAsMaster - e3Symbol

```
userMenuItemCount = e3Application.GetUserMenuItemIds( userMenuItemIds )
If userMenuItemCount > 0 Then

    menuItemId = menuItem.SetId( userMenuItemIds( userMenuItemCount ) )      'set the la
    menuItemText = menuItem.GetText()

    result = menuItem.SetDescription( description )
    If result = 0 Then
        message = "Error setting description of menu item " & menuItemText & " ( " & menuItemI
    Else
        message = "Menu item " & menuItemText & " ( " & menuItemId & " ) description set to "
    End If
    e3Application.PutInfo 0, message      'output result of operation

End If

Set menuItem = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3UserMenuItem - Overview](#)
  - [GetDescription\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3UserMenuItem.SetEnable( *newval* )**

## Syntax

**Integer** SetEnable( [in]**Integer** *newval* )

## Description

Sets the menu item active status.

## Parameters

Type	Parameter	Description
		Indicates the active status to apply
[in] <b>Integer</b>	<i>newval</i>	If 0, the menu item is inactive
		If 1, the menu item is active

## Return Values

Value	Status	Description
1	Success	Active status is applied
0	Failure	Error occurred

## Remarks

If the current menu item is a system menu or a system separator, the menu item is hidden if *newval* is 0 and shown if *newval* is 1.

If the current menu item is a type other than system menu or system separator, the menu item is deactivated if *newval* is 0 and activated if *newval* is 1.

## Examples

The best results from the example can be achieved by opening **E<sup>3</sup>.series**.

## Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set menuItem = e3Application.CreateMenuItemObject()

Const INACTIVE = 0
Const ACTIVE = 1

Dim enableStatus : enableStatus = INACTIVE

userMenuItemCount = e3Application.GetUserMenuItemIds( userMenuItemIds )
If userMenuItemCount > 0 Then

    menuItemId = menuItem.SetId( userMenuItemIds( userMenuItemCount ) )      'set the la
    menuItem.GetText()

    result = menuItem.SetEnable( enableStatus )
    If result = 0 Then
        message = "Error setting the active status of menu item " & menuItem.GetText & " ( " & menuItemId & " ) active status set to 0"
    Else
        message = "Menu item " & menuItem.GetText & " ( " & menuItemId & " ) active status set to 1"
    End If
    e3Application.PutInfo 0, message      'output result of operation

End If

Set menuItem = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

## See Also

- [e3UserMenuItem - Overview](#)
- [GetEnable\(\)](#)

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3UserMenuItem.SetFolder( *newval* )**

## Syntax

**Integer** SetFolder( [in]**String** *newval* )

## Description

Sets the menu item user tool's working folder

## Parameters

Type	Parameter	Description
[in] <b>String</b>	<i>newval</i>	Working folder of the user tool to apply

## Return Values

Value	Status	Description
1	Success	Working folder value applied
0	Failure	Error occurred

## Remarks

If *newval* value is "<Empty>", the current working folder value is removed.

## Examples

The best results from the example can be achieved by opening **E<sup>3</sup>.series**.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set menuItem = e3Application.CreateMenuItemObject()

Dim workingFolder : workingFolder = "c:\temp" 'should be a valid path
```

## SetAsMaster - e3Symbol

```
userMenuItemCount = e3Application.GetUserMenuItemIds( userMenuItemIds )
If userMenuItemCount > 0 Then

    menuItemId = menuItem.SetId( userMenuItemIds( userMenuItemCount ) )      'set the la
    menuItemText = menuItem.GetText()

    result = menuItem.SetFolder( workingFolder )
    If result = 0 Then
        message = "Error setting the working folder of menu item " & menuItemText & " ( " & me
    Else
        message = "Menu item " & menuItemText & " ( " & menuItemId & " ) working folder set to
    End If
    e3Application.PutInfo 0, message      'output result of operation

End If

Set menuItem = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3UserMenuItem - Overview](#)
  - [GetFolder\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).





v2022-23.00

**e3UserMenuItem.SetId( *id* )**

## Syntax

**Integer** SetId( [in]**Integer** *id* )

## Description

Sets a menu item as the current item.

## Parameters

Type	Parameter	Description
[in] <b>Integer</b>	<i>id</i>	Unique value identifying a menu item

## Return Values

Value	Status	Description
> 0	Success	Current menu item identifier
0	Failure	Error occurred

## Remarks

*id* will remain the current menu item until it is deleted or replaced.

The *id* value will be accepted as the current menu item even if a menu item does not exist. GetType() can be utilized to ensure a valid menu item is associated with the identifier.

## Examples

The best results from the example can be achieved by opening **E<sup>3</sup>.series**.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
```

## SetAsMaster - e3Symbol

```
Set menuItem = e3Application.CreateMenuItemObject()

Dim workingFolder : workingFolder = "c:\temp"           'should be a valid path

userMenuItemCount = e3Application.GetUserMenuItemIds( userMenuItemIds )
If userMenuItemCount > 0 Then
    For menuItemIndex = 1 To userMenuItemCount

        result = menuItem.SetId( userMenuItemIds( menuItemIndex ) )
        If result = 0 Then
            message = "Error setting " & userMenuItemIds( menuItemIndex ) & " as the current menu item"
        Else
            menuItemType = menuItem.GetType()
            If menuItemType = 0 Then
                message = result & " set but is not a valid current menu item"
            Else
                message = result & " set as the current menu item"
            End If
        End If
        e3Application.PutInfo 0, message                'output result of operation

    Next
End If

Set menuItem = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3UserMenuItem - Overview](#)
  - [GetId\(\)](#)
  - [GetType\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3UserMenuItem.SetImage( *newval* )**

## Syntax

**Integer** SetImage( [in]**String** *newval* )

## Description

Sets the menu item's icon path and file name.

## Parameters

Type	Parameter	Description
[in] <b>String</b>	<i>newval</i>	Path and name of a Windows Bitmap (.bmp) file for the display icon

## Return Values

Value	Status	Description
1	Success	Icon image file is applied
0	Failure	Error occurred

## Remarks

*newval* should refer to a Windows Bitmap (.bmp) file. If the *newval* value is "<Empty>", the icon from the user tool is used.

## Examples

The best results from the example can be achieved by opening **E<sup>3</sup>.series**.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set menuItem = e3Application.CreateMenuItemObject()
```

## SetAsMaster - e3Symbol

```
Dim menuItemIcon : menuItemIcon = "c:\temp\Image.bmp"           'should be a valid path

userMenuItemCount = e3Application.GetUserMenuItemIds( userMenuItemIds )
If userMenuItemCount > 0 Then

    menuItemId = menuItem.SetId( userMenuItemIds( userMenuItemCount ) )           'set the la
    menuItemText = menuItem.GetText()

    result = menuItem.SetImage( menuItemIcon )
    If result = 0 Then
        message = "Error setting the icon image of menu item " & menuItemText & " ( " & menuItemId & " )
    Else
        message = "Menu item " & menuItemText & " ( " & menuItemId & " ) image icon set to " & menuItemIcon
    End If
    e3Application.PutInfo 0, message           'output result of operation

End If

Set menuItem = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3UserMenuItem - Overview](#)
  - [GetImage\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3UserMenuItem.SetParameters( *newval* )**

## Syntax

*Integer* SetParameters( [in]*String* *newval* )

## Description

Sets the menu item's argument values to be passed into the user tool.

## Parameters

Type	Parameter	Description
[in] <i>String</i>	<i>newval</i>	Argument values to be passed into the user tool

## Return Values

Value	Status	Description
1	Success	Argument values applied
0	Failure	Error occurred

## Remarks

If *newval* value is "<Empty>", the current argument values is removed.

## Examples

The best results from the example can be achieved by opening **E<sup>3</sup>.series**.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set menuItem = e3Application.CreateMenuItemObject()

Dim userToolArguments : userToolArguments = "/s /v"
```

## SetAsMaster - e3Symbol

```
userMenuItemCount = e3Application.GetUserMenuItemIds( userMenuItemIds )
If userMenuItemCount > 0 Then

    menuItemId = menuItem.SetId( userMenuItemIds( userMenuItemCount ) )      'set the la
    menuItemText = menuItem.GetText()

    result = menuItem.SetParameters( userToolArguments )
    If result = 0 Then
        message = "Error setting the user tool arguments of menu item " & menuItemText & " ( "
    Else
        message = "Menu item " & menuItemText & " ( " & menuItemId & " ) user tool arguments s
    End If
    e3Application.PutInfo 0, message      'output result of operation

End If

Set menuItem = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3UserMenuItem - Overview](#)
  - [GetParameters\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3UserMenuItem.SetShortCut( *newval* )**

## Syntax

**Integer** SetShortCut( [in]**String** *newval* )

## Description

Sets the menu item's short cut key combination.

## Parameters

Type	Parameter	Description
[in] <b>String</b>	<i>newval</i>	Short cut key combination

## Return Values

Value	Status	Description
1	Success	Short cut key combination value applied
0	Failure	Error occurred

## Remarks

See [Short Cut](#) for possible values.

If *newval* value is "<Empty>", the current short cut key combination is removed.

## Examples

The best results from the example can be achieved by opening **E<sup>3</sup>.series**.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set menuItem = e3Application.CreateMenuItemObject()
```

## SetAsMaster - e3Symbol

```
Dim menuItemShortCut : menuItemShortCut = "^G"

userMenuItemCount = e3Application.GetUserMenuItemIds( userMenuItemIds )
If userMenuItemCount > 0 Then

    menuItemId = menuItem.SetId( userMenuItemIds( userMenuItemCount ) )      'set the la
    menuItemText = menuItem.GetText()

    result = menuItem.SetShortcut( menuItemShortCut )
    If result = 0 Then
        message = "Error setting the short cut of menu item " & menuItemText & " ( " & menuItemId & " ) short cut set to " & menuItemShortCut
    Else
        message = "Menu item " & menuItemText & " ( " & menuItemId & " ) short cut set to " & menuItemShortCut
    End If
    e3Application.PutInfo 0, message      'output result of operation

End If

Set menuItem = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3UserMenuItem - Overview](#)
  - [Short Cut](#)
  - [GetShortcut\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).





v2022-23.00

**e3UserMenuItem.SetText( *newval* )**

## Syntax

*Integer* SetText( [in]*String* *newval* )

## Description

Sets the menu item's display text.

## Parameters

Type	Parameter	Description
[in] <i>String</i>	<i>newval</i>	Display text

## Return Values

Value	Status	Description
1	Success	Display text applied
0	Failure	Error occurred

## Remarks

The position cannot be changed.

## Examples

The best results from the example can be achieved by opening *E<sup>3</sup>.series*.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set menuItem = e3Application.CreateMenuItemObject()

Dim menuItemText : menuItemText = "Sanjuro"
```

```

userMenuItemCount = e3Application.GetUserMenuItemIds( userMenuItemIds )
If userMenuItemCount > 0 Then

    menuItemId = menuItem.SetId( userMenuItemIds( userMenuItemCount ) )      'set the la
    menuItemText = menuItem.GetText()

    result = menuItem.SetText( menuItemText )
    If result = 0 Then
        message = "Error setting the text of menu item " & menuItemText & " ( " & menuItemId &
    Else
        message = "Menu item " & menuItemText & " ( " & menuItemId & " ) text set to " & menuItemText
    End If
    e3Application.PutInfo 0, message      'output result of operation

    menuItem.UpdateUserInterface

End If

Set menuItem = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

Modified in v2011-10.00.

---

## See Also

- [e3UserMenuItem - Overview](#)
  - [SetText\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3UserMenuItem.SetVisible( *newval* )**

## Syntax

**Integer** SetVisible( [in]**Integer** *newval* )

## Description

Sets the executable window state of the menu item's user tool when run in console execution mode.

## Parameters

Type	Parameter	Description
[in] <b>Integer</b>	<i>newval</i>	User tool window state

## Return Values

Value	Status	Description
1	Success	User tool window state value applied
0	Failure	Error occurred

## Remarks

If the user tool is not run in console execution mode, this setting has no effect.

A valid value for *newval* can be one of the following values:

Value	Description
0	Window state is hidden
1	Window state is shown
2	Window state is minimized
3	Window state is maximized

## Examples

The best results from the example can be achieved by opening **E<sup>3</sup>.series**.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set menuItem = e3Application.CreateMenuItemObject()

Const HIDDEN = 0
Const NORMAL = 1
Const MINIMIZE = 2
Const MAXIMIZE = 3

Dim windowState : windowState = MINIMIZE

userMenuItemCount = e3Application.GetUserMenuItemIds( userMenuItemIds )
If userMenuItemCount > 0 Then

    menuItemId = menuItem.SetId( userMenuItemIds( userMenuItemCount ) )      'set the la
    menuItemText = menuItem.GetText()

    result = menuItem.SetVisible( windowState )
    If result = 0 Then
        message = "Error setting the console mode window state of menu item " & menuItemText &
    Else
        message = "Menu item " & menuItemText & " ( " & menuItemId & " ) console mode window s
    End If
    e3Application.PutInfo 0, message      'output result of operation

    menuItem.UpdateUserInterface

End If

Set menuItem = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

## See Also

- [e3UserMenuItem - Overview](#)
- [GetVisible\(\)](#)

## SetAsMaster - e3Symbol

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3UserMenuItem.SetWaitForEndOfCommand( *newval* )**

## Syntax

**Integer** SetWaitForEndOfCommand( [in]**Integer** *newval* )

## Description

Sets whether the **E<sup>3</sup>.series** process is paused during the execution life time of the user tool.

Function has been deprecated.

## Parameters

Type	Parameter	Description
		Value indicating whether the <b>E<sup>3</sup>.series</b> process is paused during the execution life time of the user tool
[in] <b>Integer</b>	<i>newval</i>	If 1, the <b>E<sup>3</sup>.series</b> process is paused during the execution life time of the user tool
		If 0, the <b>E<sup>3</sup>.series</b> process runs during the execution life time of the user tool

## Return Values

Value	Status	Description
1	Success	Value applied
0	Inconclusive	Functionality is no longer supported or an error has occurred

## Remarks

Since v2011-10.00 this functionality is no longer supported and 0 is always returned.

The functionality can be utilized using [e3Application.GetModalWindow\(\)](#) and [e3Application.SetModalWindow\(\)](#).

## Examples

The best results from the example can be achieved by opening **E<sup>3</sup>.series**.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set menuItem = e3Application.CreateMenuItemObject()

Const NOT_SUPPORTED = 2011      'GetWaitForEndOfCommand() is not supported since this version
e3ApplicationVersion = CInt( e3Application.GetVersion() )      'get e3Application version

Dim pauseApplication : pauseApplication = 1

userMenuItemCount = e3Application.GetUserMenuItemIds( userMenuItemIds )
If userMenuItemCount > 0 Then

    menuItemId = menuItem.SetId( userMenuItemIds( userMenuItemCount ) )      'set the last item
    menuItemText = menuItem.GetText()

    result = menuItem.SetWaitForEndOfCommand( pauseApplication )
    If result = 0 Then
        If e3ApplicationVersion < NOT_SUPPORTED Then
            message = "Error setting wait for end of command for menu item " & menuItemText & "
        Else
            message = "Menu item " & menuItemText & " ( " & menuItemId & " ): GetWaitForEndOfCommand()
        End If
    Else
        message = "Menu item " & menuItemText & " ( " & menuItemId & " ) set to pause E3.series
    End If
    e3Application.PutInfo 0, message      'output result of operation

End If

Set menuItem = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

Deprecated in v2011-10.00.

---

## See Also

- [e3UserMenuItem - Overview](#)
- [e3Application.GetModalWindow\(\)](#)

- e3Application.SetModalWindow()
  - GetWaitForEndOfCommand()
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).





v2022-23.00

## e3UserMenuItem.UnDelete()

## Syntax

**Integer** UnDelete()

## Description

Restores the deleted menu item.

Function has been deprecated.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
0	Inconclusive	This value has been returned since v2011-10.00

## Remarks

This function is no longer supported.

## Examples

The best results from the example can be achieved by opening **E<sup>3</sup>.series**.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set menuItem = e3Application.CreateMenuItemObject()

userMenuItemCount = e3Application.GetUserMenuItemIds( userMenuItemIds )
```

## SetAsMaster - e3Symbol

```
If userMenuItemCount > 0 Then

    menuItemId = menuItem.SetId( userMenuItemIds( userMenuItemCount ) )      'set the la
    menuItem.GetText()

    result = menuItem.UnDelete()
    If result = 0 Then
        message = "Menu item " & menuItem.GetText() & " ( " & menuItemId & " ): UnDelete() not supp
    Else
        message = "Undeleted() function used with menu item " & menuItem.GetText() & " ( " & menuItemId & " )
    End If
    e3Application.PutInfo 0, message      'output result of operation

End If

Set menuItem = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

Deprecated in v2011-10.00.

---

## See Also

- [e3UserMenuItem - Overview](#)
  - [Delete\(\)](#)
  - [IsDeleted\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

## e3UserMenuItem.UpdateUserInterface()

### Syntax

**Integer** UpdateUserInterface()

### Description

Updates the **E<sup>3</sup>.series** menus and short cuts.

### Parameters

No parameters defined.

### Return Values

Value	Status	Description
1	Success	<b>E<sup>3</sup>.series</b> menus and short cuts updated
0	Failure	Error occurred

### Remarks

This function can be used without being assigned a current menu item.

### Examples

The best results from the example can be achieved by opening **E<sup>3</sup>.series**.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set menuItem = e3Application.CreateMenuItemObject()

result = menuItem.UpdateUserInterface()
If result = 0 Then
```

## SetAsMaster - e3Symbol

```
message = "E³.series menu and short cuts failed to update"
Else
    message = "E³.series menu and short cuts updated"
End If
e3Application.PutInfo 0, message          'output result of operation

Set menuItem = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3UserMenuItem - Overview](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

## e3Variant

# Description

Encapsulates the functionality for retrieving and modifying information for variant items.

## e3Variant Construction Functions

Function	Description
<u>e3Job.CreateVariantObject()</u>	Creates an instance of e3Variant

## Item Creation/Destruction Functions

Function	Description
<u>Create()</u>	Creates a new variant item
<u>Delete()</u>	Deletes the variant item from the project

## Retrieval Functions

Function	Description
<u>GetGID()</u>	Gets the global identifier of the variant item
<u>GetGUID()</u>	Gets the globally unique identifier of the current variant item
<u>GetId()</u>	Gets the identifier of the current variant item
<u>GetName()</u>	Gets the variant item's name
<u>IsActive()</u>	Get the active state of the current variant item

## Modification Functions

Function	Description
<u>SetGID()</u>	Sets a variant as the current item
<u>SetGUID()</u>	Sets a variant as the current item
<u>SetId()</u>	Sets a variant as the current item
<u>SetName()</u>	Sets the variant item's name

# Process Functions

Function Description

Search() Searches for a variant item matching the name

## Remarks

Variants describe schematic parts that are to be activated exclusively. Only one variant in a group can be active. Project items belong either to no specific variant or specifically to one of them.

e3Option should be used instead of e3Variant since it offers a super-set of the e3Variant functionality and seamlessly handles both options and variants.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>**.series project.

Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set e3Variant = job.CreateVariantObject()

variantCount = job.GetVariantIds( variantIds )           'get variants
If variantCount > 0 Then
    For variantIndex = 1 To variantCount                 'loop through all variant items

        variantId = e3Variant.SetId( variantIds( variantIndex ) )
        result = e3Variant.GetName()
        If Len( "" & result ) = 0 Then
            message = "Error getting name for variant " & variantId
        Else
            message = "Name of variant " & variantId & " is " & result
        End If
        e3Application.PutInfo 0, message                 'output result of operation

    Next
End If

Set e3Variant = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

## See Also

- [Classes - Overview](#)
  - [e3Option - Overview](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Variant.Create( *name* )**

## Syntax

**Integer** Create( [in]**String** *name* )

## Description

Creates a new variant item.

## Parameters

Type	Parameter	Description
[in] <b>String</b>	<i>name</i>	Path and name of the variant

## Return Values

Value	Status	Description
> 0	Success	Identifier of the created variant
0	Failure	Error occurred

## Remarks

*name* should include the parent option names using " / " as a separator to form a path to the name of the variant.

The successfully created variant item is set as the current variant item.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series**.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
```



## SetAsMaster - e3Symbol

```
Set e3Variant = job.CreateVariantObject()

Dim variantName : variantName = "Option1 / Variant1"

result = e3Variant.Create( variantName )
If result = 0 Then
    message = "Error creating variant " & variantName
Else
    message = "Variant " & variantName & " ( " & result & " ) created"
End If
e3Application.PutInfo 0, message           'output result of operation

Set e3Variant = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3Variant - Overview](#)
  - [Delete\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Variant.Delete( *del* )**

## Syntax

**Integer** Delete( [in]**Integer** *del* )

## Description

Deletes the variant item from the project.

## Parameters

Type	Parameter	Description
		Indicates whether items the variant is exclusively assigned to are also removed from the project.
[in] <b>Integer</b>	<i>del</i>	If 0, the items are not deleted
		If greater than 0, the items are deleted

## Return Values

Value	Status	Description
> 0	Failure	Current variant item identifier
0	Inconclusive	Variant item was deleted or an error occurred

## Remarks

If successful, the current variant item is set to 0. GetId() can be used to evaluate the current variant item value.

Due caution is recommended on relying on the return value of 0 meaning the variant item was deleted since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project.

### Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set e3Variant = job.CreateVariantObject()

Dim deleteItems : deleteItems = 0

variantCount = job.GetVariantIds( variantIds )           'get variants
If variantCount > 0 Then
    For variantIndex = 1 To variantCount                 'loop through all variant items

        variantId = e3Variant.SetId( variantIds( variantIndex ) )
        variantName = e3Variant.GetName()

        result = e3Variant.Delete( deleteItems )
        If result = 0 Then
            currentId = e3Variant.GetId()
            If currentId = 0 Then
                message = "Variant " & variantName & " ( " & variantId & " ) deleted"
            Else
                message = "Error deleting variant"
            End If
        Else
            message = "Error deleting variant " & variantName & " ( " & variantId & " )"
        End If
        e3Application.PutInfo 0, message                 'output result of operation

    Next
End If

Set e3Variant = Nothing
Set job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2009-8.50.

## See Also

- [e3Variant - Overview](#)

## SetAsMaster - e3Symbol

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Variant.GetGID()**

## Syntax

*String* GetGID()

## Description

Gets the global identifier of the variant item.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
"<GID>"	Success	Global identifier of the variant item
"<Empty>"	Failure	No variant item

## Remarks

Global identifiers (GIDs) are used for item identification in multiuser projects.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>**.series project.

Visual Basic Script

```

Set e3Application = CreateObject( "CT.Application" )
Set e3Job = e3Application.CreateJobObject()
Set e3Variant = e3Job.CreateVariantObject()

variantCount = e3Job.GetVariantIds( variantIds )      'get selected variants
If variantCount > 0 Then

```

## SetAsMaster - e3Symbol

```
For variantIndex = 1 To variantCount

    variantId = e3Variant.SetId( variantIds( variantIndex ) )
    result = e3Variant.GetGID()
    If Len( "" & result ) = 0 Then
        message = "No variant item is set"
    Else
        variantName = e3Variant.GetName()
        message = "GID of variant " & variantName & " ( " & variantId & " ) is " & result
    End If
    e3Application.PutInfo 0, message           'output result of operation

Next
End If

Set e3Variant = Nothing
Set e3Job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2022-23.00.

---

## See Also

- [e3Variant - Overview](#)
  - [GetId\(\)](#)
  - [GetGUID\(\)](#)
  - [SetGID\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Variant.GetGUID()**

## Syntax

*String* GetGUID()

## Description

Gets the globally unique identifier of the current variant item.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
"<GUID>"	Success	Globally unique identifier of the current variant item
"<Empty>"	Failure	Error occurred

## Remarks

Globally unique identifiers (GUIDs) are used for item identification across projects.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>**.series project.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set e3Job = e3Application.CreateJobObject()
Set e3Variant = e3Job.CreateVariantObject()

variantCount = e3Job.GetVariantIds( variantIds )      'get selected variants
If variantCount > 0 Then
```

## SetAsMaster - e3Symbol

```
For variantIndex = 1 To variantCount

    variantId = e3Variant.SetId( variantIds( variantIndex ) )
    result = e3Variant.GetGUID()
    If Len( "" & result ) = 0 Then
        message = "No variant item is set"
    Else
        variantName = e3Variant.GetName()
        message = "GUID of variant item " & variantName & " ( " & variantId & " ) is "
    End If
    e3Application.PutInfo 0, message           'output result of operation

Next
End If

Set e3Variant = Nothing
Set e3Job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2022-23.00.

---

## See Also

- [e3Variant - Overview](#)
  - [GetGID\(\)](#)
  - [GetId\(\)](#)
  - [SetGUID\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).





v2022-23.00

**e3Variant.GetId()**

## Syntax

Integer GetId()

## Description

Gets the identifier of the current variant item.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
> 0	Success	Current variant item identifier
0	Failure	No variant item

## Remarks

The function returns the identifier value set by SetId() unless the item no longer exists.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>**.series project.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set e3Job = e3Application.CreateJobObject()
Set e3Variant = e3Job.CreateVariantObject()
```

## SetAsMaster - e3Symbol

```
variantCount = e3Job.GetVariantIds( variantIds )           'get selected variants
If variantCount > 0 Then
    For variantIndex = 1 To variantCount

        e3Variant.SetId variantIds( variantIndex )
        result = e3Variant.GetId()
        If result = 0 Then
            message = "No variant item is set"
        Else
            variantName = e3Variant.GetName()
            message = "Variant " & variantName & " ( " & result & " ) has been set"
        End If
        e3Application.PutInfo 0, message                   'output result of operation

    Next
End If

Set e3Variant = Nothing
Set e3Job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3Variant - Overview](#)
  - [SetId\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Variant.GetName()**

## Syntax

`String` GetName()

## Description

Gets the variant item's name.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
"<Text>"	Success	Variant name
"<Empty>"	Failure	Error occurred

## Remarks

The variant name can be modified using [SetName\(\)](#).

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>**.series project.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set e3Variant = job.CreateVariantObject()

variantCount = job.GetVariantIds( variantIds )           'get variants
```

## SetAsMaster - e3Symbol

```
If variantCount > 0 Then
    For variantIndex = 1 To variantCount          'loop through all variant items

        variantId = e3Variant.SetId( variantIds( variantIndex ) )
        result = e3Variant.GetName()
        If Len( "" & result ) = 0 Then
            message = "Error getting name for variant " & variantId
        Else
            message = "Name of variant " & variantId & " is " & result
        End If
        e3Application.PutInfo 0, message          'output result of operation

    Next
End If

Set e3Variant = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3Variant - Overview](#)
  - [SetName\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Variant.IsActive()**

## Syntax

**Integer** IsActive()

## Description

Get the active state of the current variant item.

## Parameters

No parameters defined.

## Return Values

Value	Status	Description
1	Success	Variant is active
0	Inconclusive	Variant is inactive or an error occurred

## Remarks

The variant item can be activated and deactivated using [e3Option.Activate\(\)](#) and [e3Option.Deactivate\(\)](#) respectively.

Due caution is recommended on relying on the return value of 0 meaning the variant is inactive since this also could mean an error has occurred.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>**.series project.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
```

## SetAsMaster - e3Symbol

```
Set job = e3Application.CreateJobObject()
Set e3Variant = job.CreateVariantObject()

variantCount = job.GetVariantIds( variantIds )           'get variants
If variantCount > 0 Then
    For variantIndex = 1 To variantCount                 'loop through all variant items

        variantId = e3Variant.SetId( variantIds( variantIndex ) )
        variantName = e3Variant.GetName()

        result = e3Variant.IsActive()
        If result = 0 Then
            message = "Variant " & variantName & " ( " & variantId & " ) is inactive"
        Else
            message = "Variant " & variantName & " ( " & variantId & " ) is active"
        End If
        e3Application.PutInfo 0, message                'output result of operation

    Next
End If

Set e3Variant = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3Variant - Overview](#)
  - [e3Option.Activate\(\)](#)
  - [e3Option.Deactivate\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Variant.Search( *name* )**

## Syntax

**Integer** Search( [in]**String** *name* )

## Description

Searches for a variant item matching the name.

## Parameters

Type	Parameter	Description
[in] <b>String</b>	name	Name of the variant

## Return Values

Value	Status	Description
> 0	Success	Found variant identifier
0	Inconclusive	No variant was found or an error has occurred

## Remarks

Due caution is recommended on relying on the return value of 0 meaning the variant was not found since this also could mean an error has occurred.

If successful, the variant will be set as the current item.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
```

## SetAsMaster - e3Symbol

```
Set e3Variant = job.CreateVariantObject()

Dim variantName : variantName = "Standard"

result = e3Variant.Search( variantName )
If result = 0 Then
    message = "Variant named " & variantName & " not found"
Else
    message = "Variant named " & variantName & " found ( " & result & " )"
End If
e3Application.PutInfo 0, message           'output result of operation

Set e3Variant = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2015-16.00.

---

## See Also

- [e3Variant - Overview](#)
  - [GetName\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).





v2022-23.00

**e3Variant.SetGID( *gid* )**

## Syntax

*String* SetGID( [in]*String* *gid* )

## Description

Sets a variant as the current item.

## Parameters

Type	Parameter	Description
[in] <i>String</i>	<i>gid</i>	Global identifier value of a variant item

## Return Values

Value	Status	Description
"<GID>"	Success	Global identifier of the current variant item
"<Empty>"	Failure	No variant item

## Remarks

Global identifiers (GIDs) are used for item identification in multiuser projects.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>**.series project.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set e3Job = e3Application.CreateJobObject()
Set e3Variant = e3Job.CreateVariantObject()

Set gidList = CreateObject( "System.Collections.ArrayList" )
```

## SetAsMaster - e3Symbol

```
variantCount = e3Job.GetVariantIds( variantIds )           'get selected variants
If variantCount > 0 Then
    For variantIndex = 1 To variantCount

        e3Variant.SetId variantIds( variantIndex )
        gidId = e3Variant.GetGID()
        gidList.Add gidId

    Next
End If

For Each gidId in gidList
    result = e3Variant.SetGID( gidId )
    If Len( "" & result ) = 0 Then
        message = "No variant item is set"
    Else
        variantId = e3Variant.GetId()
        variantName = e3Variant.GetName()
        message = "Variant " & variantName & " ( " & variantId & " ) has been set using GID"
    End If
    e3Application.PutInfo 0, message           'output result of operation
Next

Set gidList = Nothing
Set e3Variant = Nothing
Set e3Job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2022-23.00.

---

## See Also

- [e3Variant - Overview](#)
  - [GetGID\(\)](#)
  - [SetGUID\(\)](#)
  - [SetId\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Variant.SetGUID( *guid* )**

## Syntax

*String* SetGUID( [in]*String* *guid* )

## Description

Sets a variant as the current item.

## Parameters

Type	Parameter	Description
[in] <i>String</i>	<i>guid</i>	Globally unique identifier value of a variant item

## Return Values

Value	Status	Description
"<GUID>"	Success	Globally unique identifier of the current variant item
"<Empty>"	Failure	No variant item

## Remarks

Globally unique identifiers (GUIDs) are used for item identification across projects.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set e3Job = e3Application.CreateJobObject()
Set e3Variant = e3Job.CreateVariantObject()

Set guidList = CreateObject( "System.Collections.ArrayList" )
```

```

variantCount = e3Job.GetVariantIds( variantIds )           'get selected variants
If variantCount > 0 Then
    For variantIndex = 1 To variantCount

        e3Variant.SetId variantIds( variantIndex )
        guidId = e3Variant.GetGUID()
        guidList.Add guidId

    Next
End If

For Each guidId in guidList
    result = e3Variant.SetGUID( guidId )
    If Len( "" & result ) = 0 Then
        message = "No variant item is set"
    Else
        variantId = e3Variant.GetId()
        variantName = e3Variant.GetName()
        message = "Variant " & variantName & " ( " & variantId & " ) has been set using GI
    End If
    e3Application.PutInfo 0, message           'output result of operation
Next

Set gidList = Nothing
Set e3Variant = Nothing
Set e3Job = Nothing
Set e3Application = Nothing

```

## Version Information

Introduced in v2022-23.00.

## See Also

- [e3Variant - Overview](#)
- [GetGUID\(\)](#)
- [SetGID\(\)](#)
- [SetId\(\)](#)

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Variant.SetId( *id* )**

## Syntax

*Integer* SetId( [in]*Integer id* )

## Description

Sets a variant as the current item.

## Parameters

Type	Parameter	Description
[in] <i>Integer</i>	<i>id</i>	Unique value identifying a variant item

## Return Values

Value	Status	Description
> 0	Success	Current variant item identifier
0	Failure	Error occurred

## Remarks

*id* will remain the current variant item until it is deleted or replaced.

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set e3Job = e3Application.CreateJobObject()
Set e3Variant = e3Job.CreateVariantObject()
```

## SetAsMaster - e3Symbol

```
variantCount = e3Job.GetVariantIds( variantIds )           'get selected variants
If variantCount > 0 Then
    For variantIndex = 1 To variantCount

        result = e3Variant.SetId( variantIds( variantIndex ) )
        If result = 0 Then
            message = "No variant item is set"
        Else
            variantName = e3Variant.GetName()
            message = "Variant " & variantName & " ( " & result & " ) has been set"
        End If
        e3Application.PutInfo 0, message                   'output result of operation

    Next
End If

Set e3Variant = Nothing
Set e3Job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3Variant - Overview](#)
  - [GetId\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

**e3Variant.SetName( *name* )**

## Syntax

**Integer** SetName( [in]**String** *name* )

## Description

Sets the variant item's name.

## Parameters

Type	Parameter	Description
[in] <b>String</b>	<i>name</i>	New variant name value

## Return Values

Value	Status	Description
> 0	Success	Variant item identifier
0	Failure	Error occurred

## Remarks

The variant name can be retrieved using [GetName\(\)](#).

## Examples

The best results from the example can be achieved by opening an **E<sup>3</sup>.series** project.

Visual Basic Script

```
Set e3Application = CreateObject( "CT.Application" )
Set job = e3Application.CreateJobObject()
Set e3Variant = job.CreateVariantObject()
```

## SetAsMaster - e3Symbol

```
Dim standardString : standardString = "Standard"
Dim replacementString : replacementString = "Replacement"

variantCount = job.GetVariantIds( variantIds )           'get variants
If variantCount > 0 Then
    For variantIndex = 1 To variantCount                 'loop through all variant items

        variantId = e3Variant.SetId( variantIds( variantIndex ) )
        variantName = e3Variant.GetName()
        containsStandard = InStr( variantName, standardString )

        If containsStandard > 0 Then
            newName = Replace( variantName, standardString, replacementString )

            result = e3Variant.SetName( newName )
            If result = 0 Then
                message = "Error setting name for variant " & variantId & " from " & variantName
            Else
                message = "Name of variant " & variantId & " changed from " & variantName
            End If
            e3Application.PutInfo 0, message              'output result of operation

        End If

    Next
End If

Set e3Variant = Nothing
Set job = Nothing
Set e3Application = Nothing
```

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3Variant - Overview](#)
  - [GetName\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).





v2022-23.00

## Common Methods

Common methods are used for many objects. This chapter describes these methods, which are grouped by functional themes.

### Themes

Handling Internal  
Identifiers

Internal Identifier Handling

Parameters

Handling common  
parameters

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

## Internal Identifier Handling - Introduction

All **E<sup>3</sup>** COM objects must be initialized with an internal identifier before they can be used.

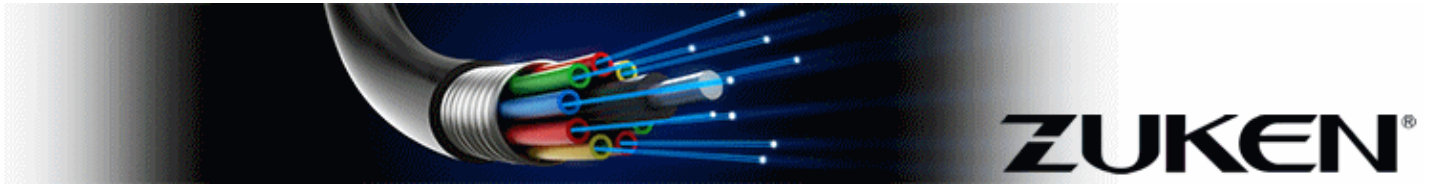
Such internal identifiers are always referred to as "Id"s. They are represented by positive integers that are generated by the **E<sup>3</sup>** kernel.

- See all dependencies between **E<sup>3</sup>** COM objects in Overview and Dependencies.
- Use any.SetId to initialize a COM object,
- any.GetId to read the internal id of an already initialized object and
- any.GetXxxIds methods to retrieve an array of Ids.
- Use Local Objects to write clear code.

The following chapters are available:

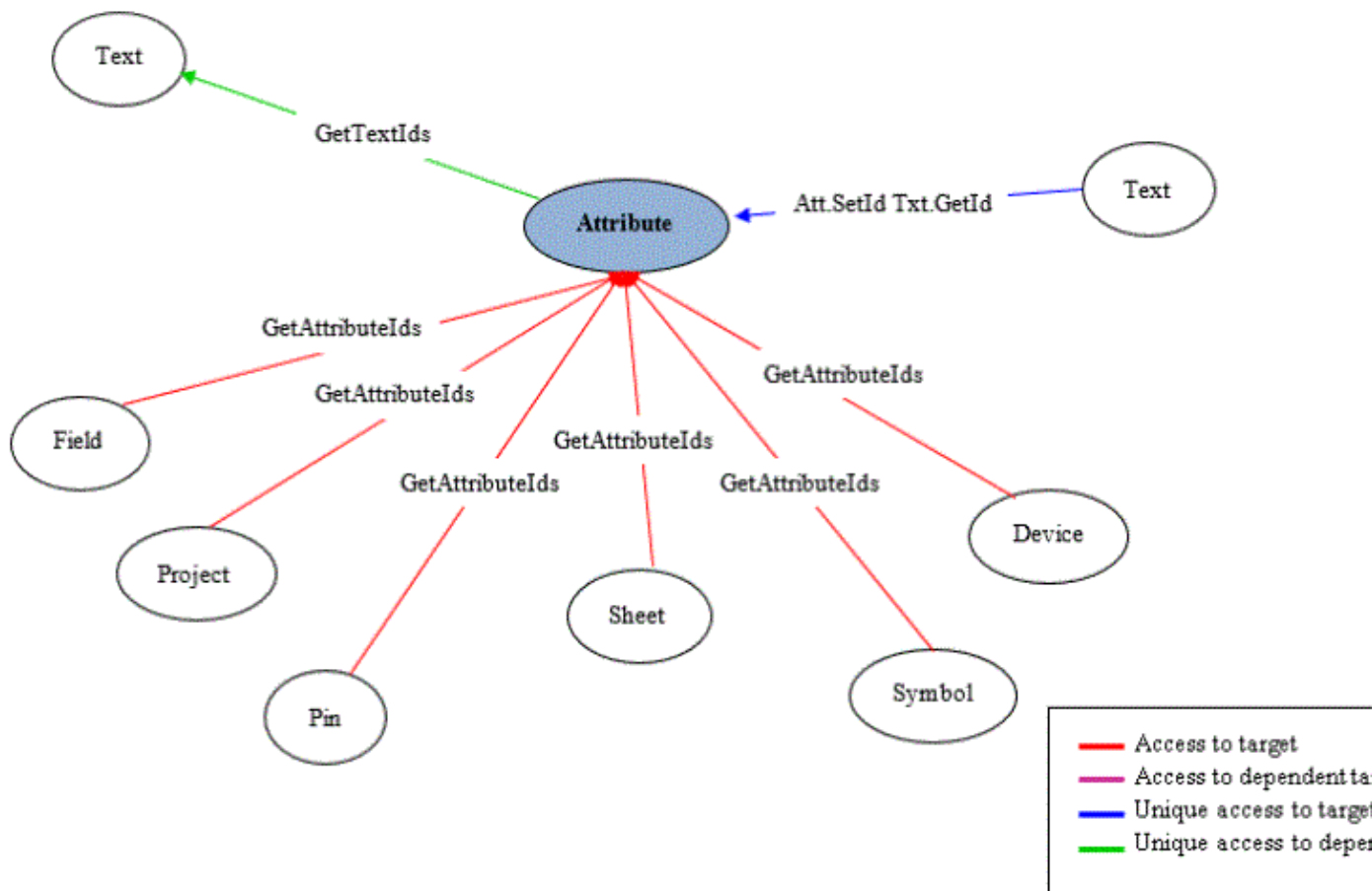
- Overview and Dependencies
- SetId
- GetId
- GetXxxIds
- Local Objects

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

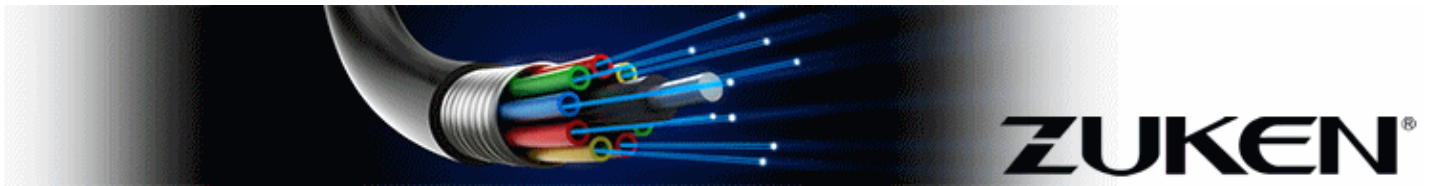
## Attribute



See also:

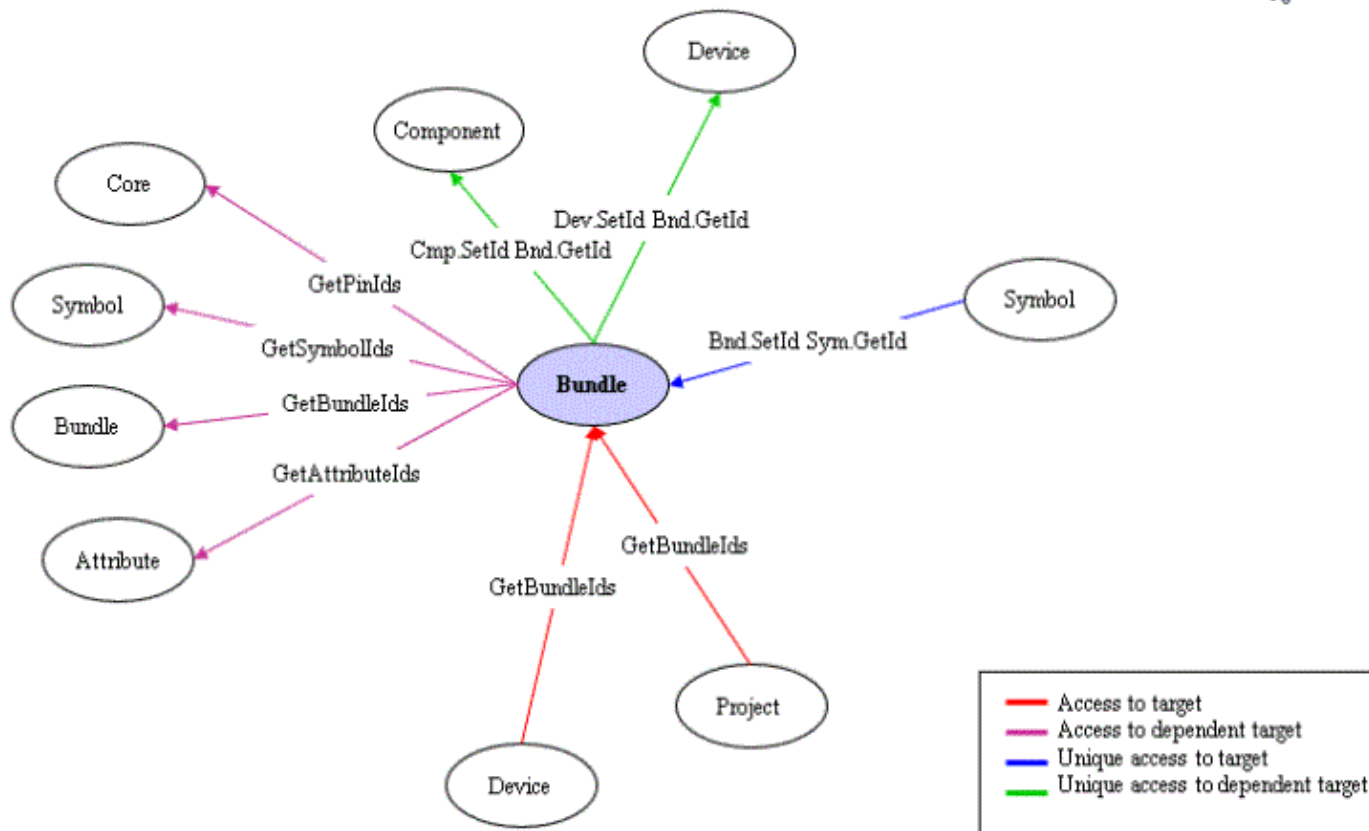
- **Handling Internal Identifiers - Overview-Dependencies**

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

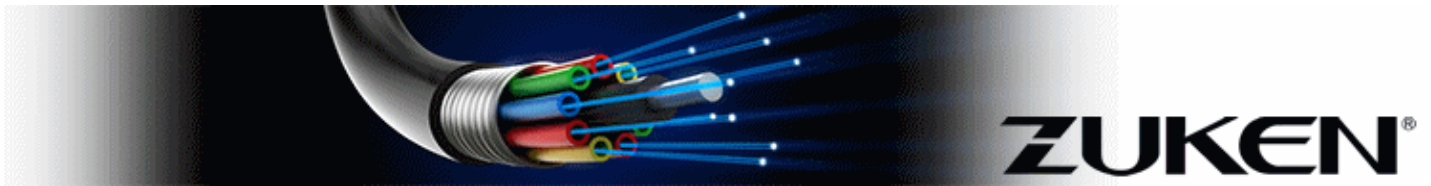
## Bundle



See also:

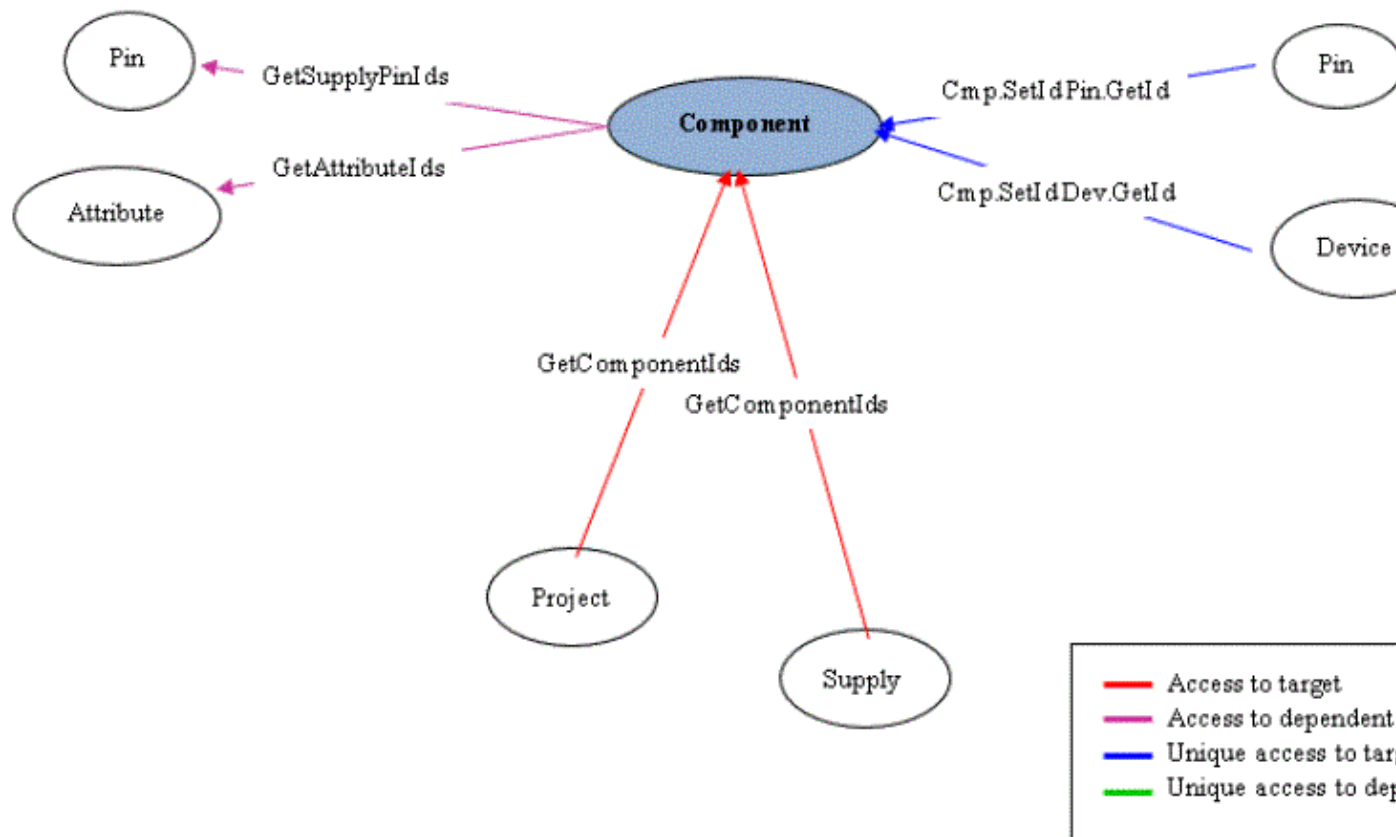
- **Handling Internal Identifiers - Overview-Dependencies**

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

## Component

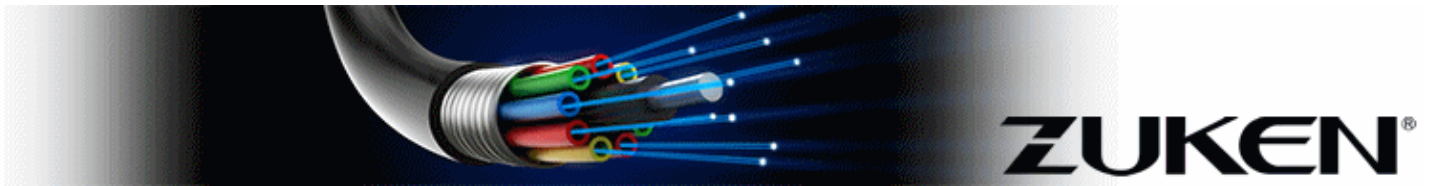


See also:

- **Handling Internal Identifiers - Overview-Dependencies**

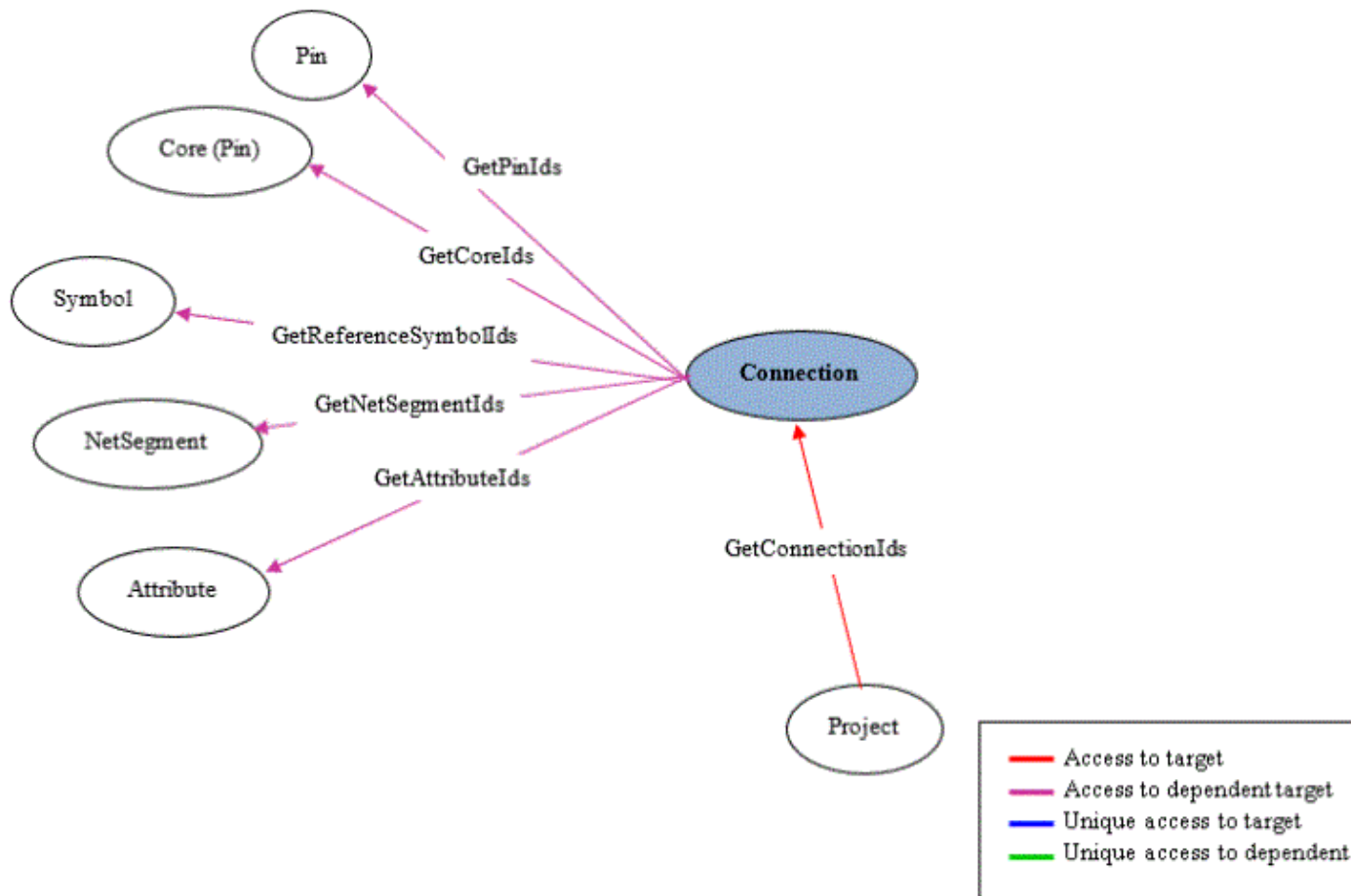
More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).





v2022-23.00

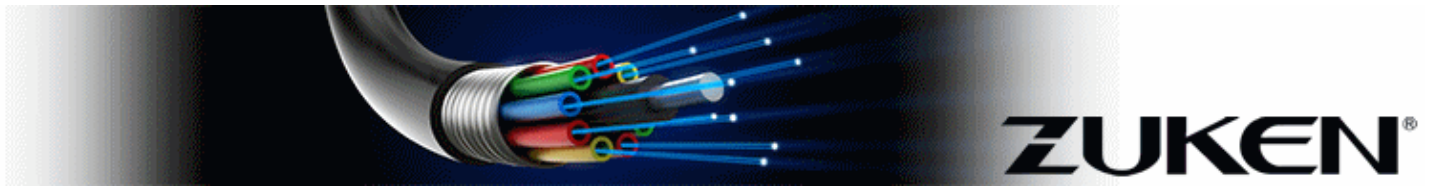
## Connection



See also:

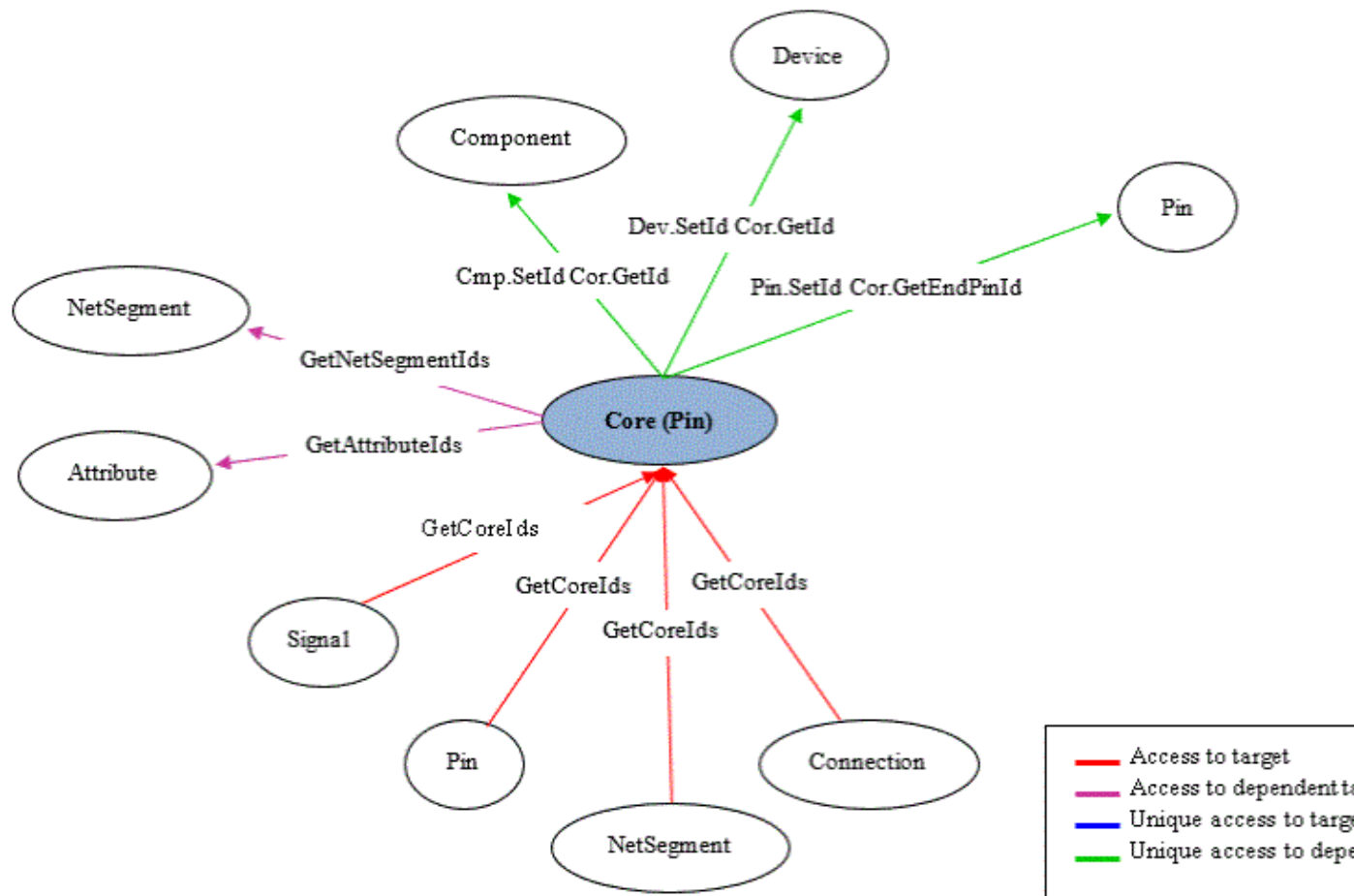
- **Handling Internal Identifiers - Overview-Dependencies**

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

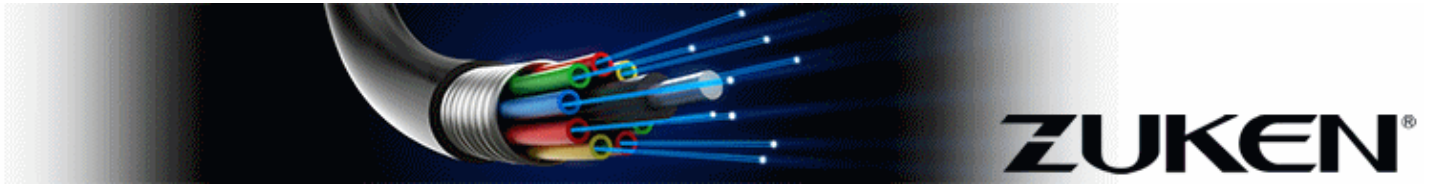
## Core



See also:

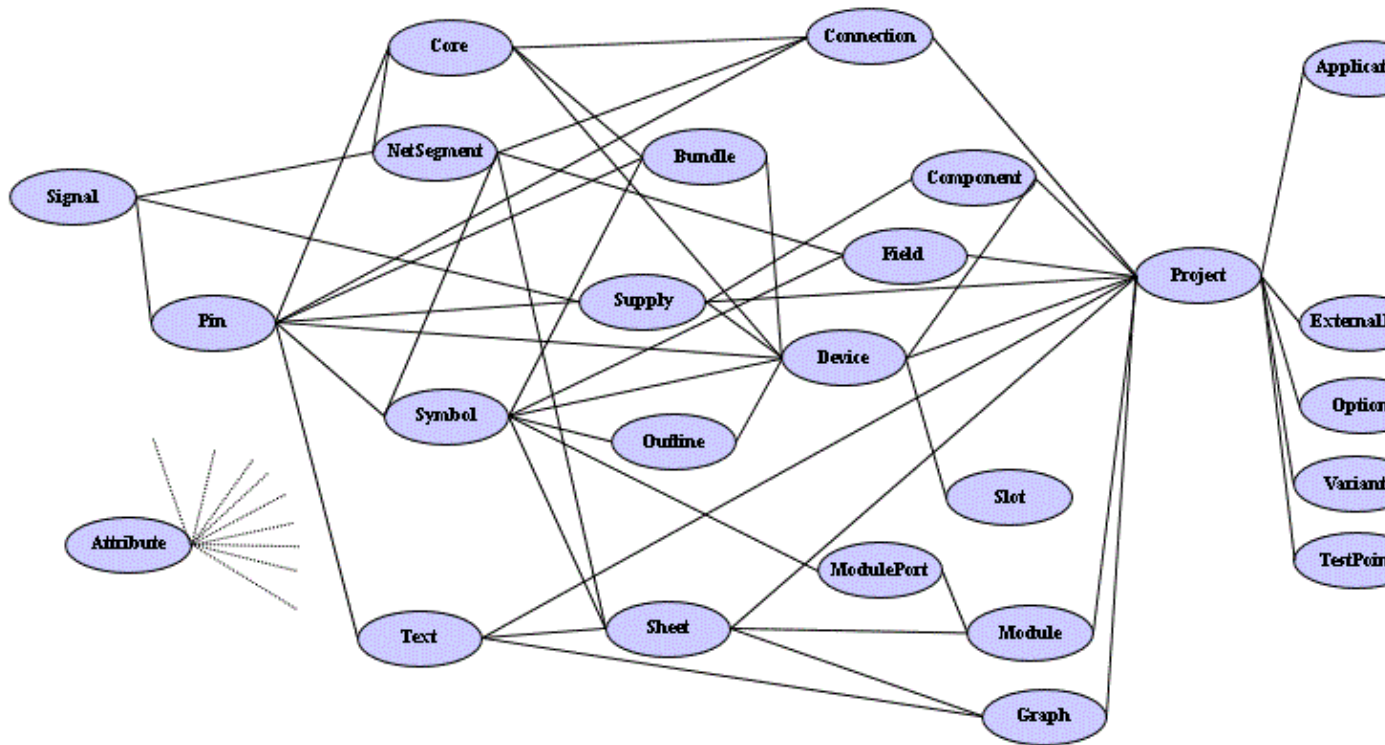
- **Handling Internal Identifiers - Overview-Dependencies**

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

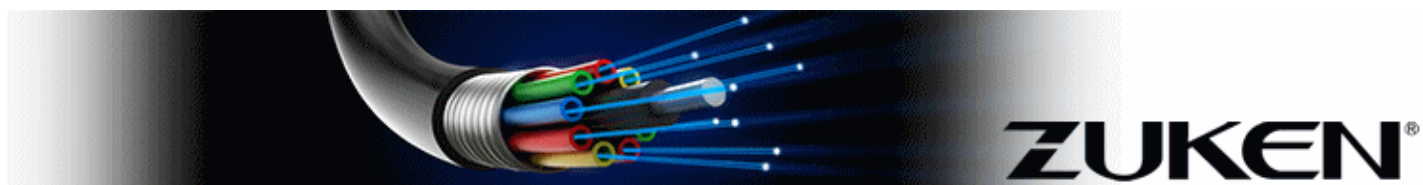
## Dependencies



See also:

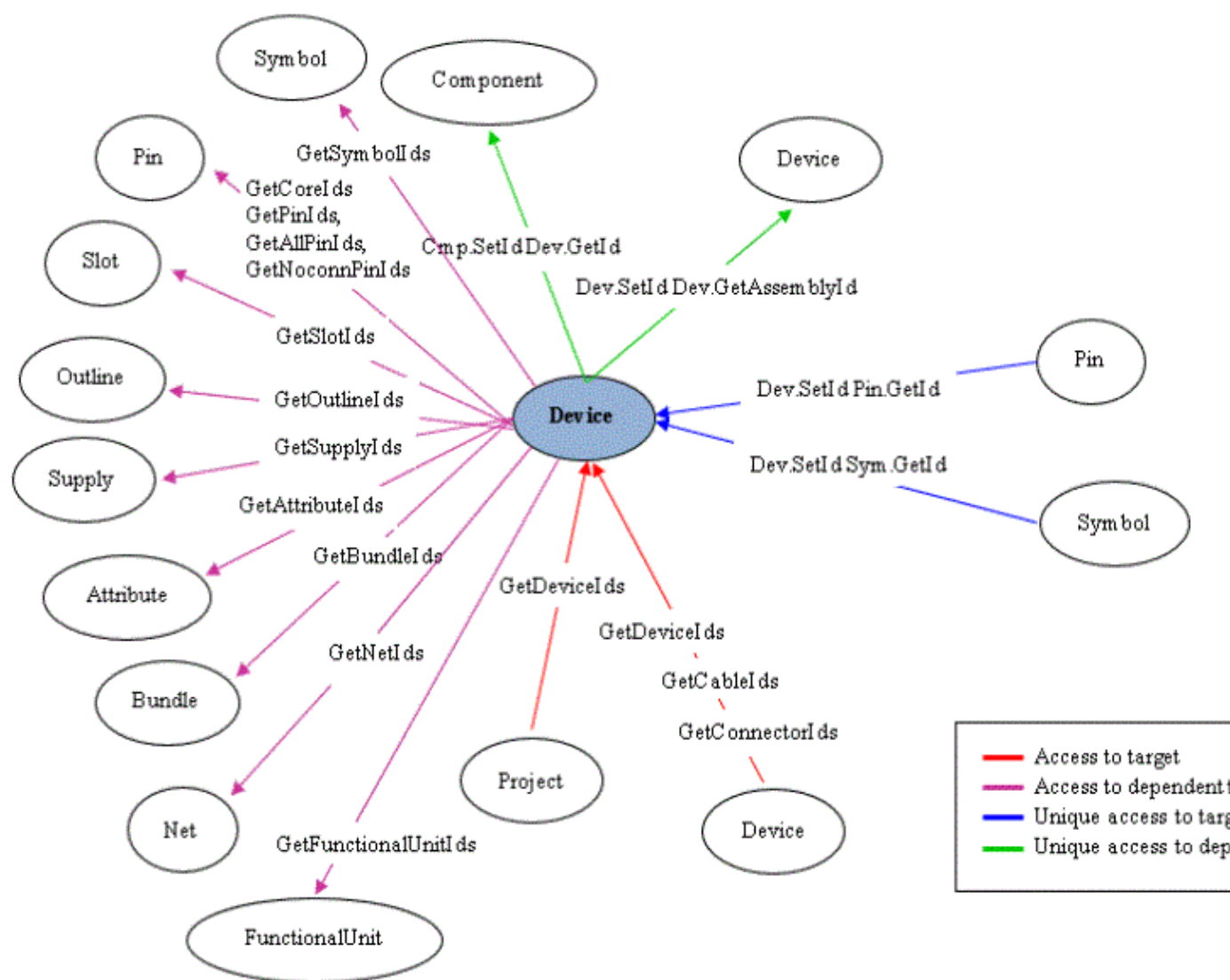
- **Handling Internal Identifiers - Overview-Dependencies**

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

## Device

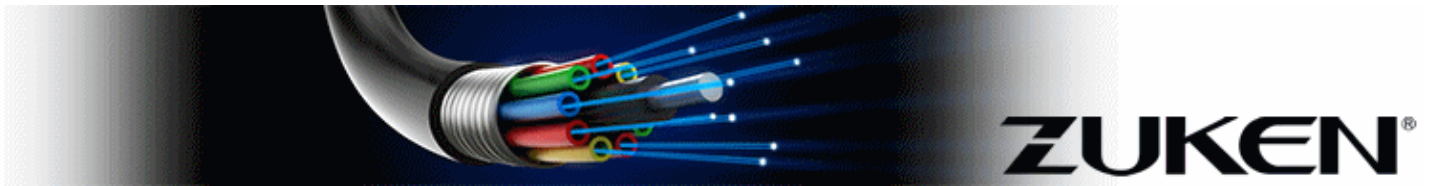


See also:

- **Handling Internal Identifiers - Overview-Dependencies**

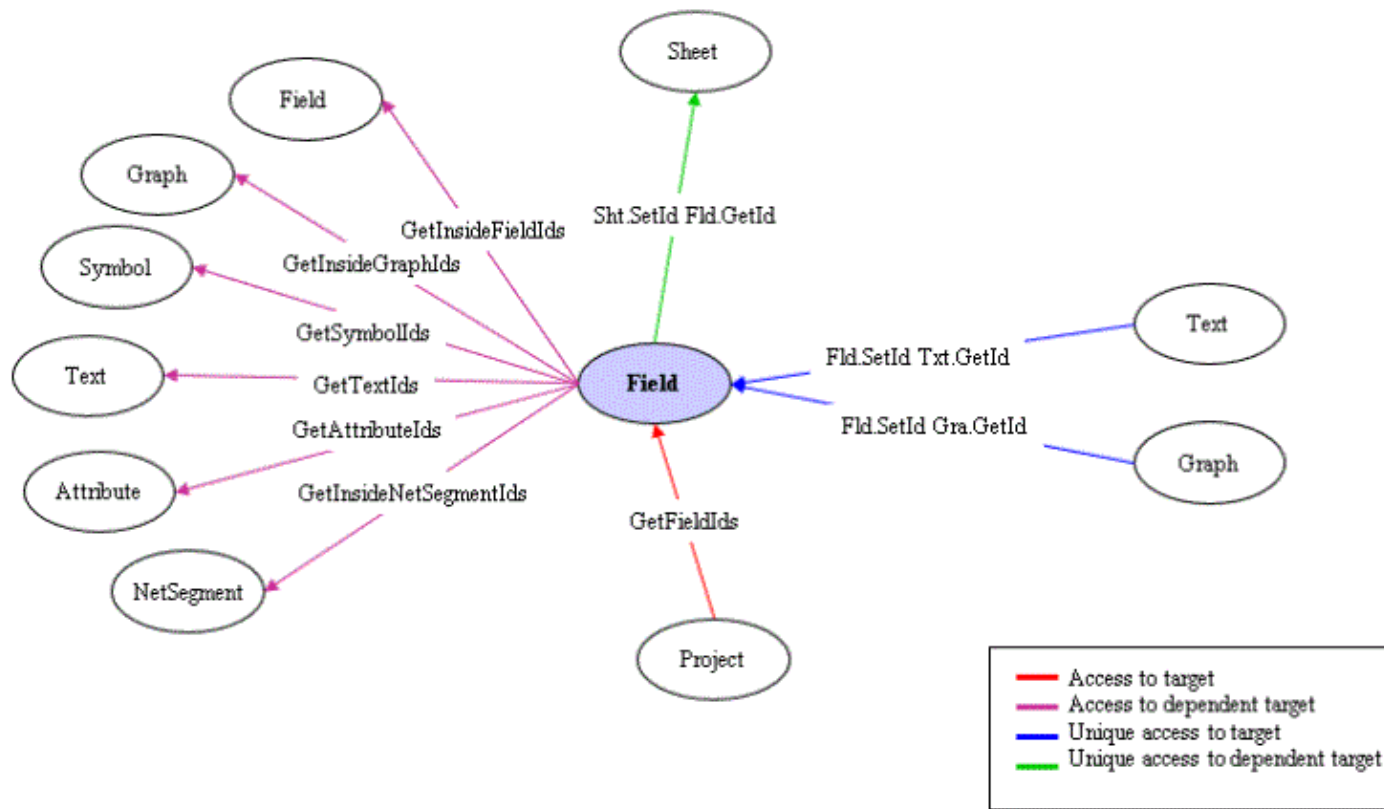
More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).





v2022-23.00

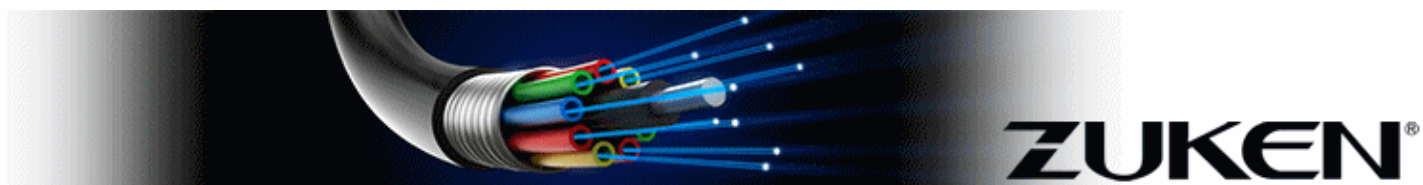
## Field



See also:

- **Handling Internal Identifiers - Overview-Dependencies**

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

## GetXxxIds idarr

**GetXxxIds** returns internal identifiers of requested type (Xxx) in given array **idarr**. The return value of these methods is the count of ids in **idarr**.

The following example shows how to display pin names within a given device:

```
Set dev = prj.CreateDeviceObject
Set pin = prj.CreatePinObject
...
pincnt = dev.GetPinIds( pinids )
For p = 1 To pincnt
    pin.SetId pinids(p)
    MsgBox "Pin " & pin.GetName
Next
```

See also:

- [Handling Internal Identifiers - Introduction](#)

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

## Local Objects

It is possible to create objects local to a subroutine or function. Please keep in mind though, that these local objects have to be initialized by **any.SetId** before they can be used.

Local objects are automatically destroyed when leaving the subroutine or the function.

The following example shows a function, that returns the fully qualified name of a device, given by an Id:

```

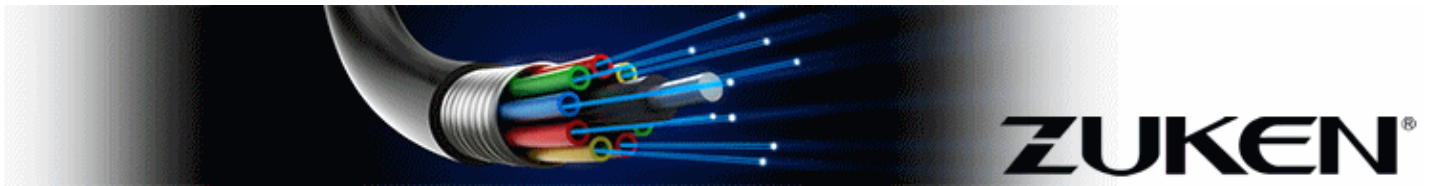
...
devnam = FullName( devid )
MsgBox "Device " & devnam & ": "
...
Function FullName( devid )
Dim dev
Set dev = prj.CreateDeviceObject
dev.SetId devid
FullName = dev.GetAssignment _
& dev.GetLocation _
& dev.GetName
End Function

```

See also:

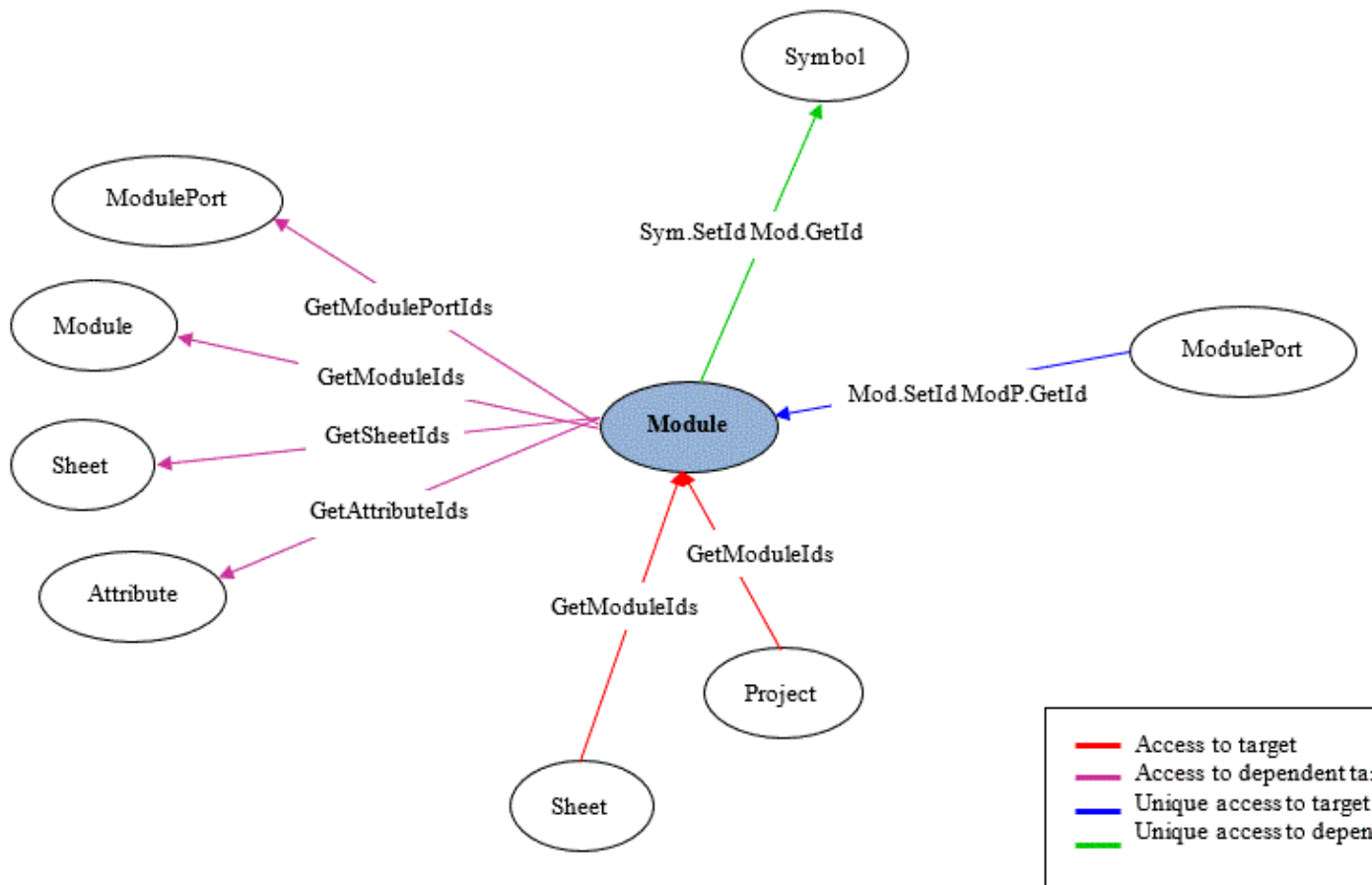
- [Handling Internal Identifiers - Introduction](#)

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

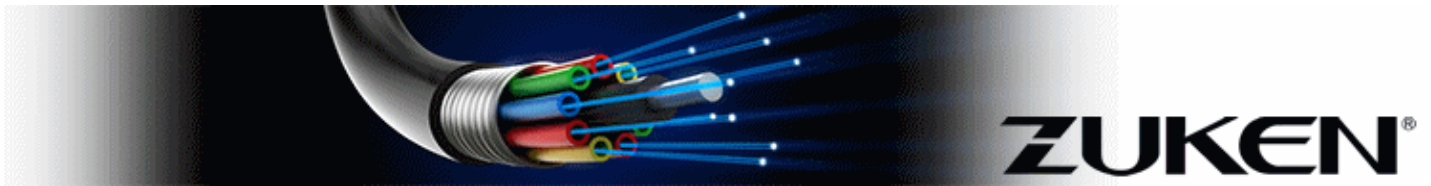
## Module



See also:

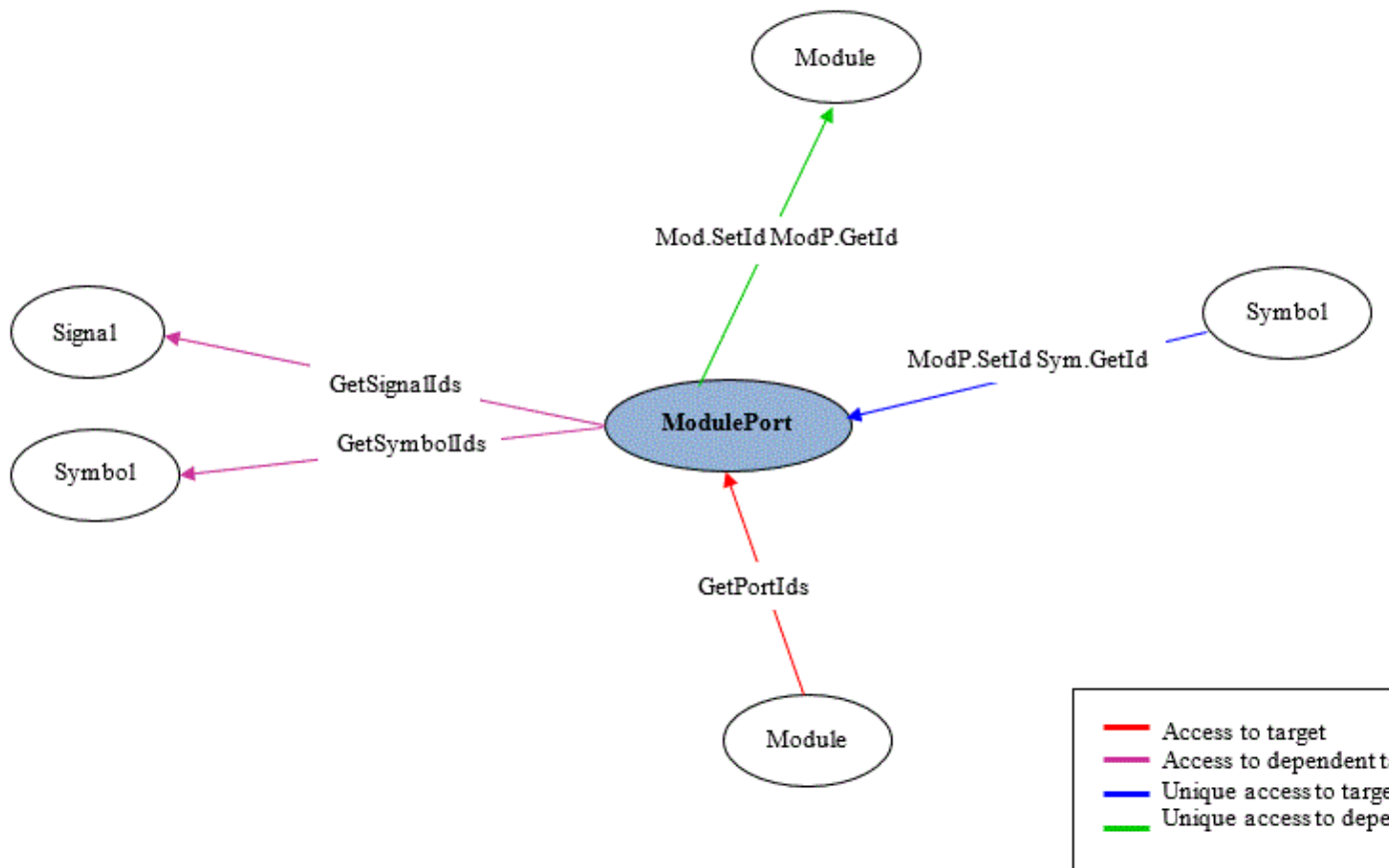
- **Handling Internal Identifiers - Overview-Dependencies**

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

## ModulePort

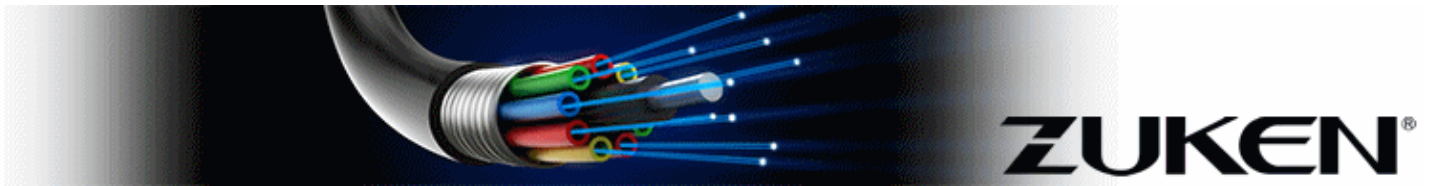


See also:

- **Handling Internal Identifiers - Overview-Dependencies**

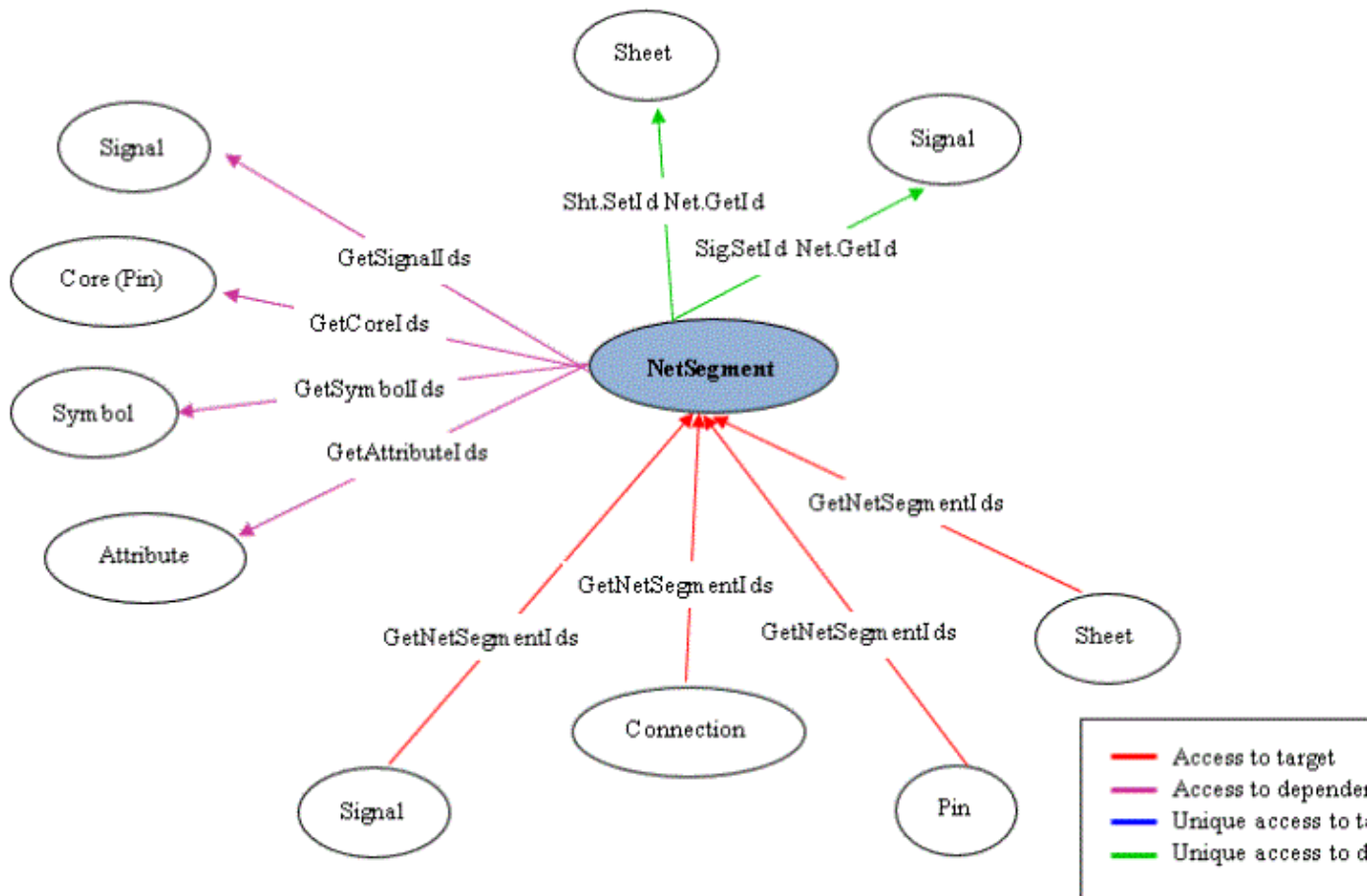
More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).





v2022-23.00

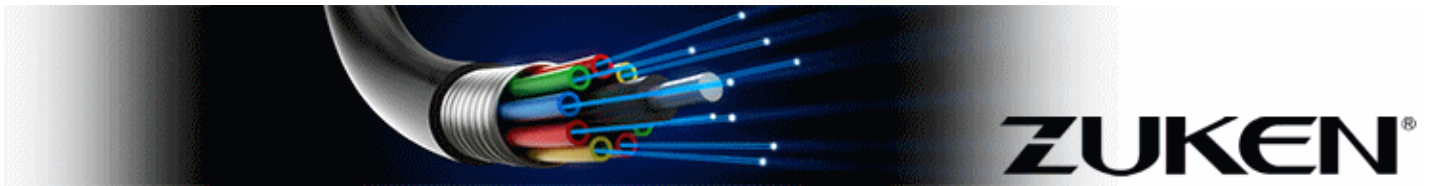
## NetSegment



See also:

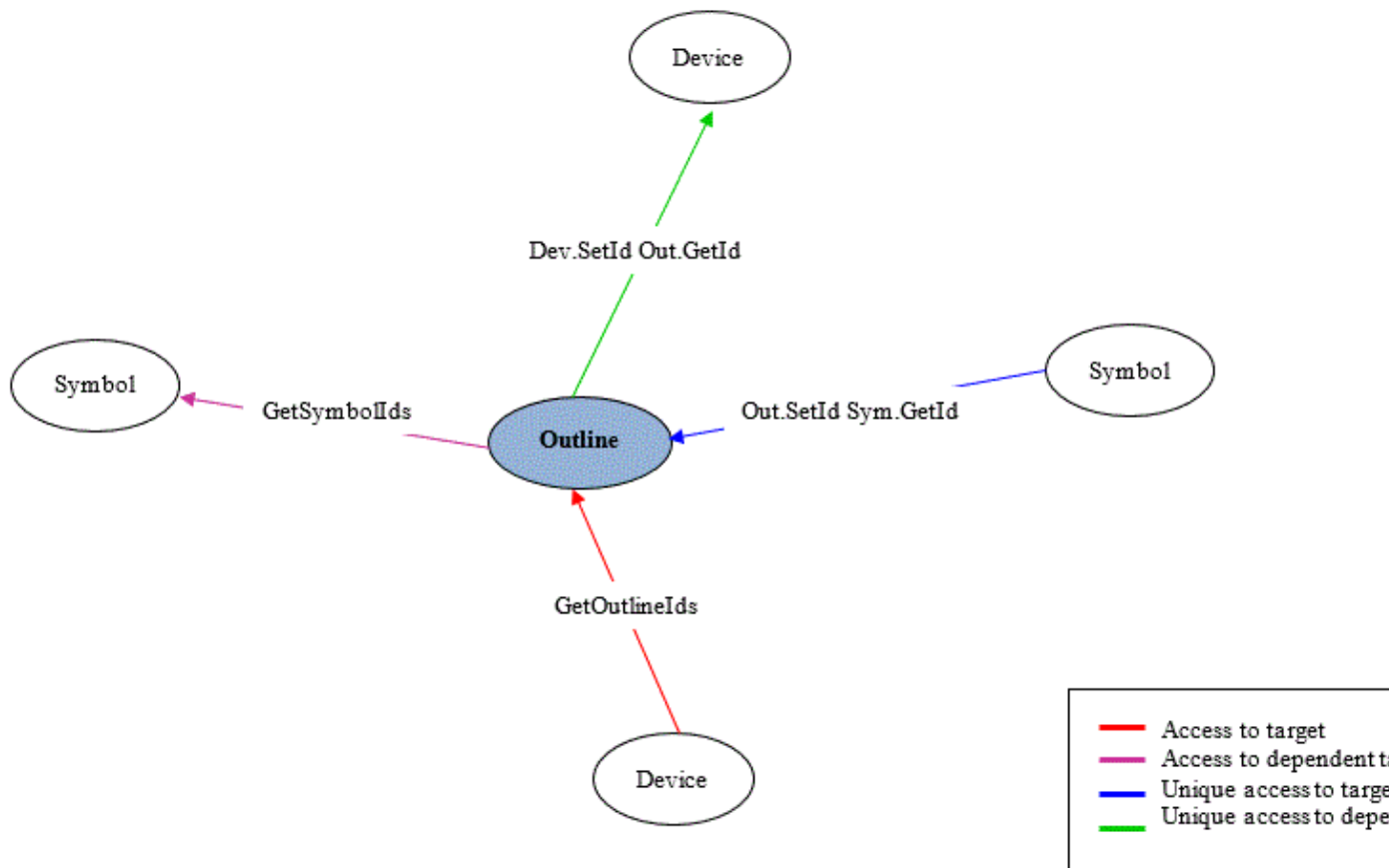
- **Handling Internal Identifiers - Overview-Dependencies**

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

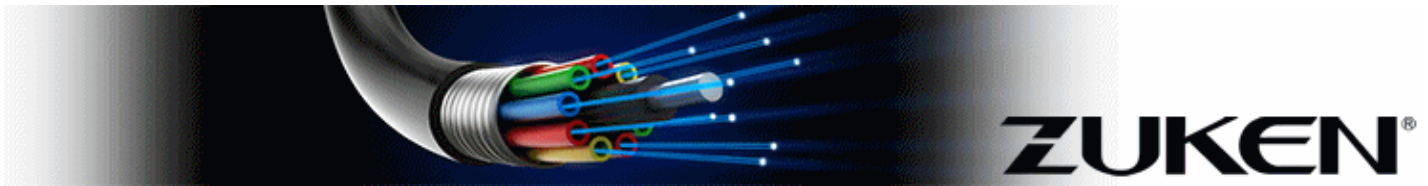
## Outline



See also:

- **Handling Internal Identifiers - Overview-Dependencies**

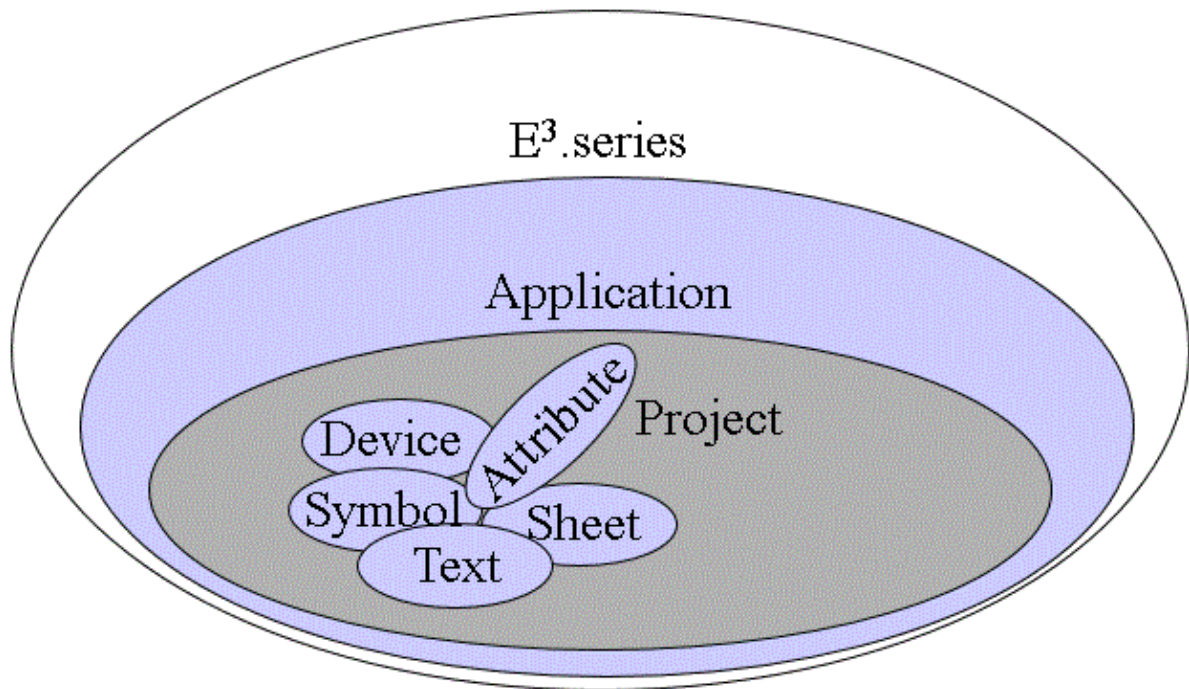
More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

## Overview

All objects in **E<sup>3</sup>** have dependencies between their classes.



See also:

- **Handling Internal Identifiers - Overview-Dependencies**

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

## Overview - Dependencies

All objects in **E<sup>3</sup>** have dependencies between their classes. See following chapters:

- Overview
- Dependencies

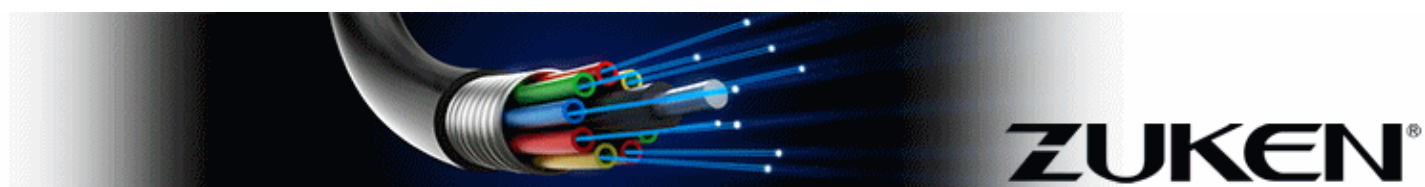
The following objects are available:

- Attribute
- Bundle
- Component
- Connection
- Core
- Device
- Field
- Module
- ModulePort
- NetSegment
- Outline
- Pin
- Sheet
- Signal
- Supply
- Symbol
- Text

See also:

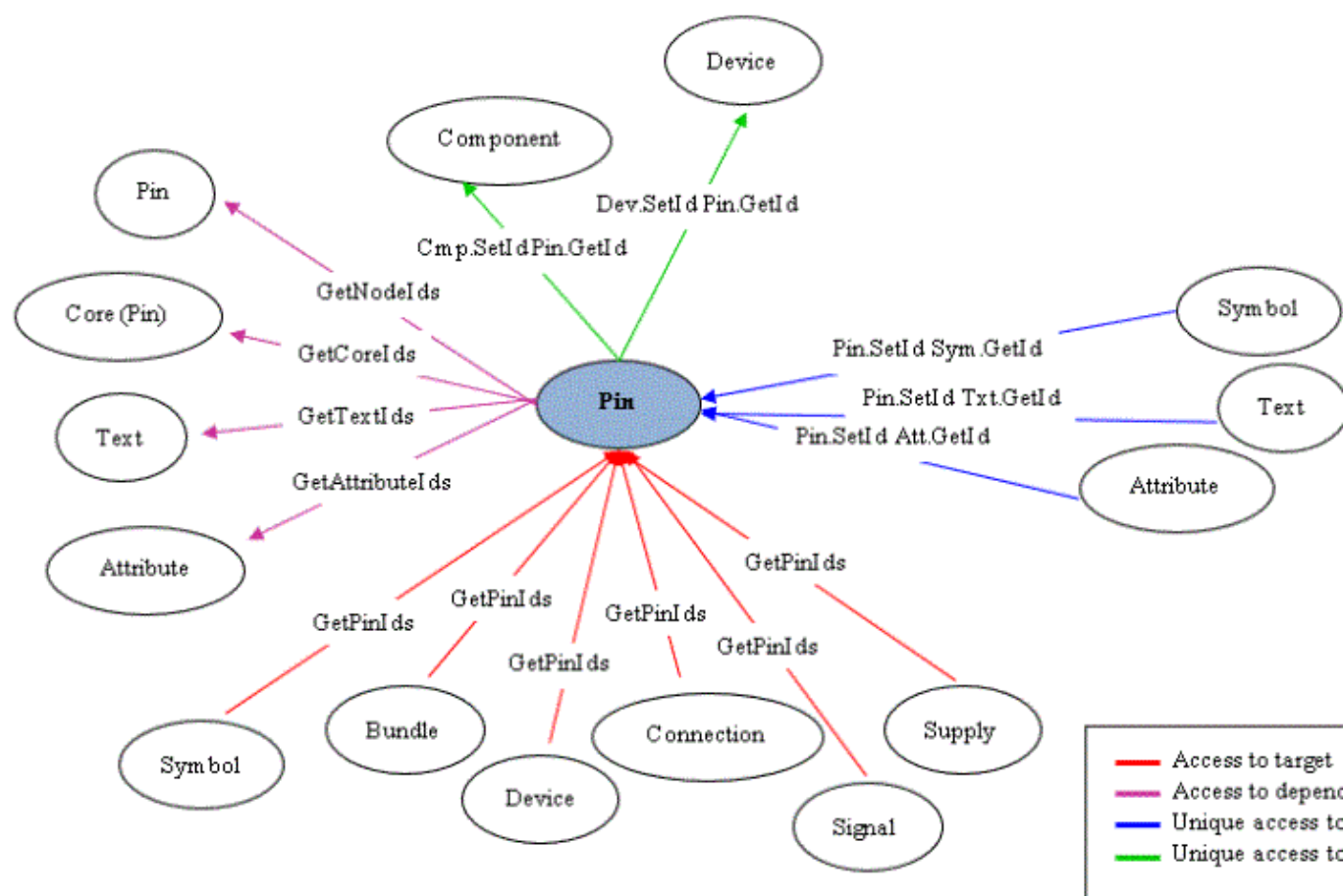
- **Handling Internal Identifiers - Introduction**

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

## Pin

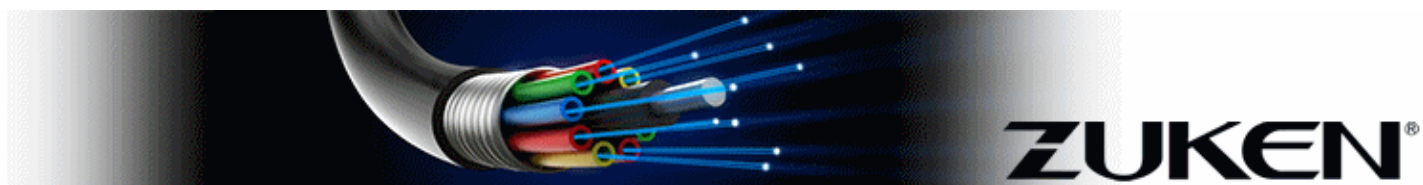




See also:

- **Handling Internal Identifiers - Overview-Dependencies**

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

## SetId new GetId

**SetId** is an intelligent method that allows to initialize high level objects by giving lower level identifiers that lead to a specific object.

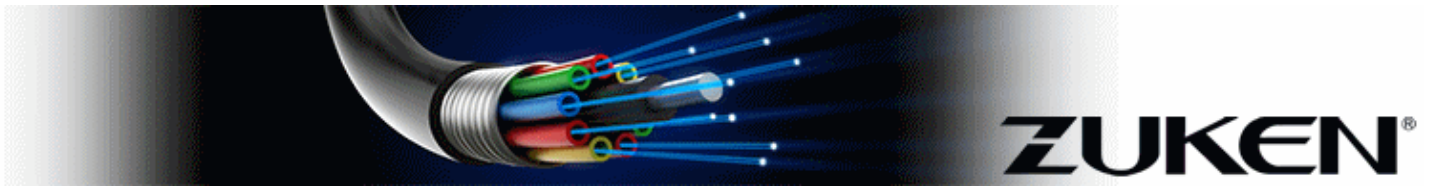
The following example illustrates how to initialize a device object by giving a pin Id to a device object.

```
Set dev = prj.CreateDeviceObject
Set pin = prj.CreatePinObject
...
dev.SetId pin.GetId
MsgBox "Device " & dev.GetName _
    & " contains pin " & pin.GetName
```

See also:

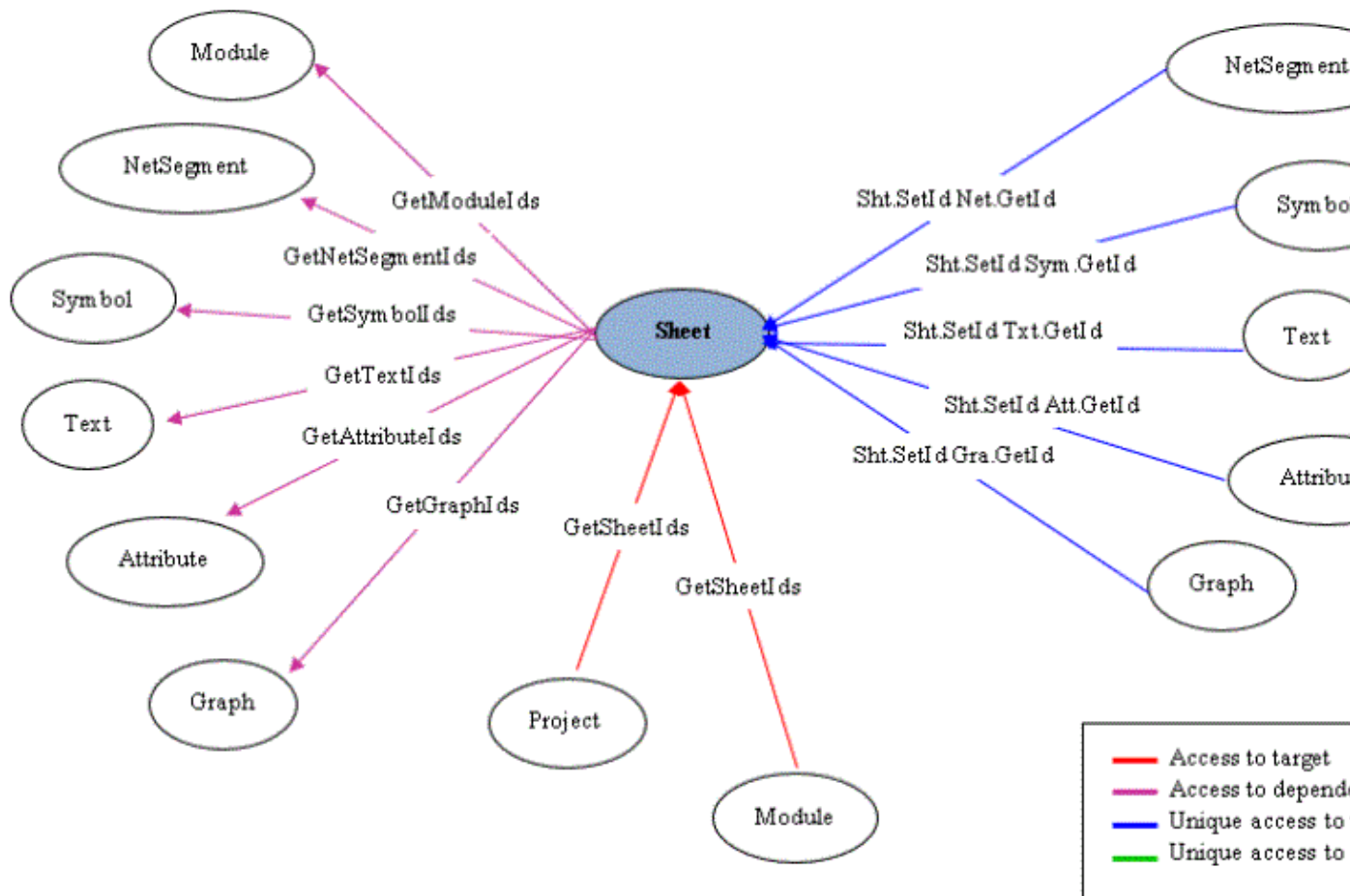
- [Handling Internal Identifiers - Introduction](#)

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

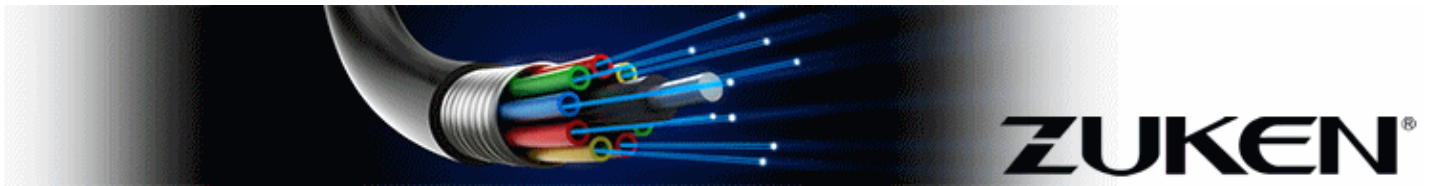
## Sheet



See also:

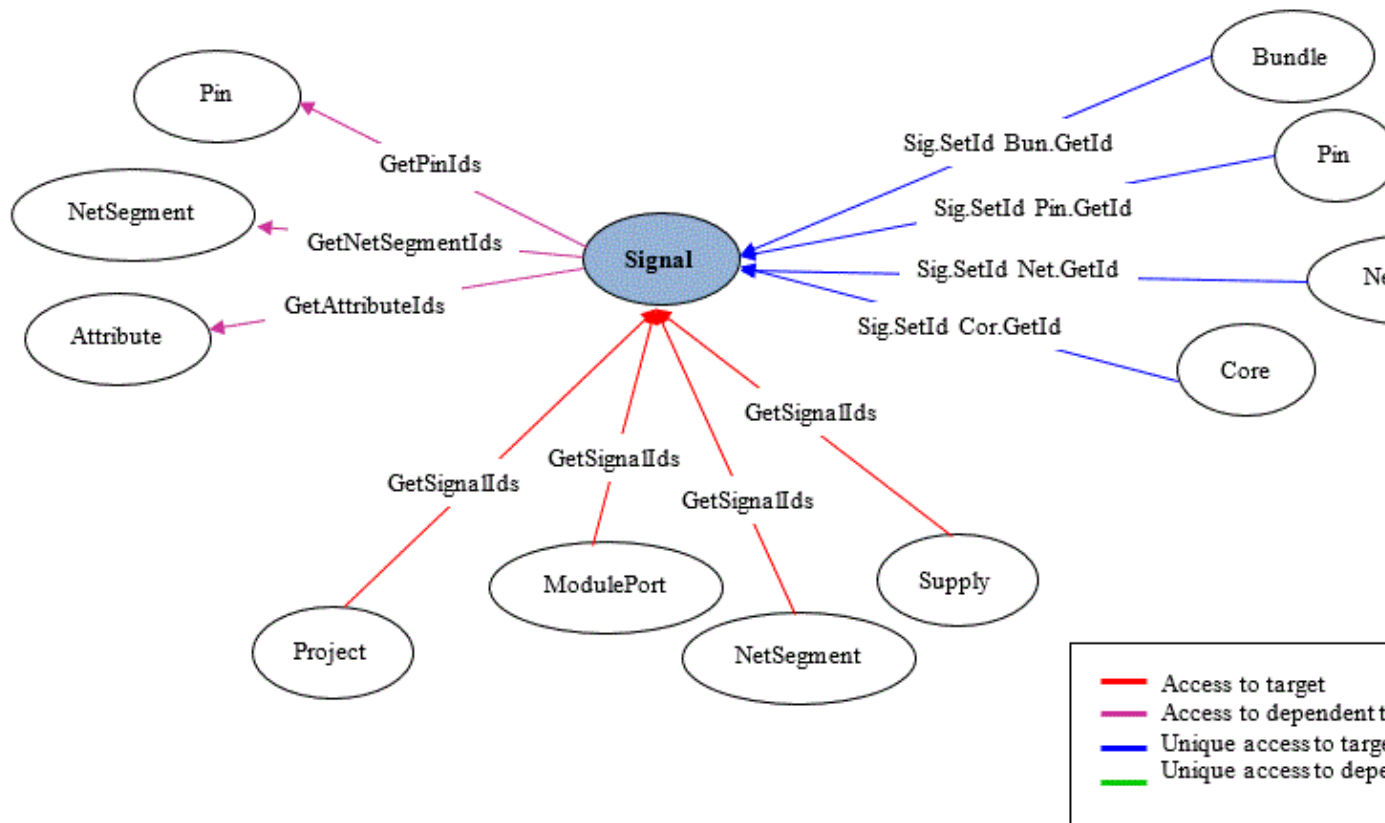
- **Handling Internal Identifiers - Overview-Dependencies**

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

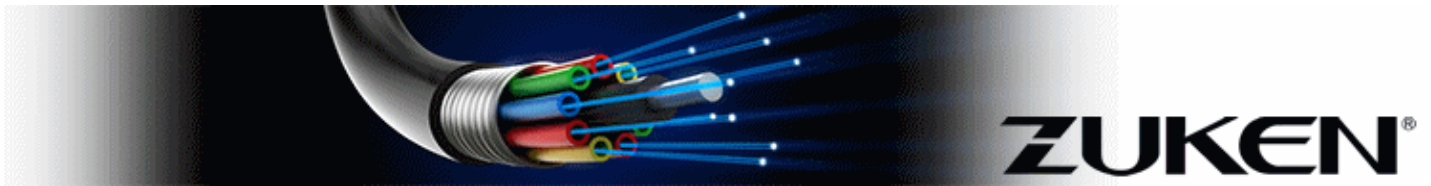
## Signal



See also:

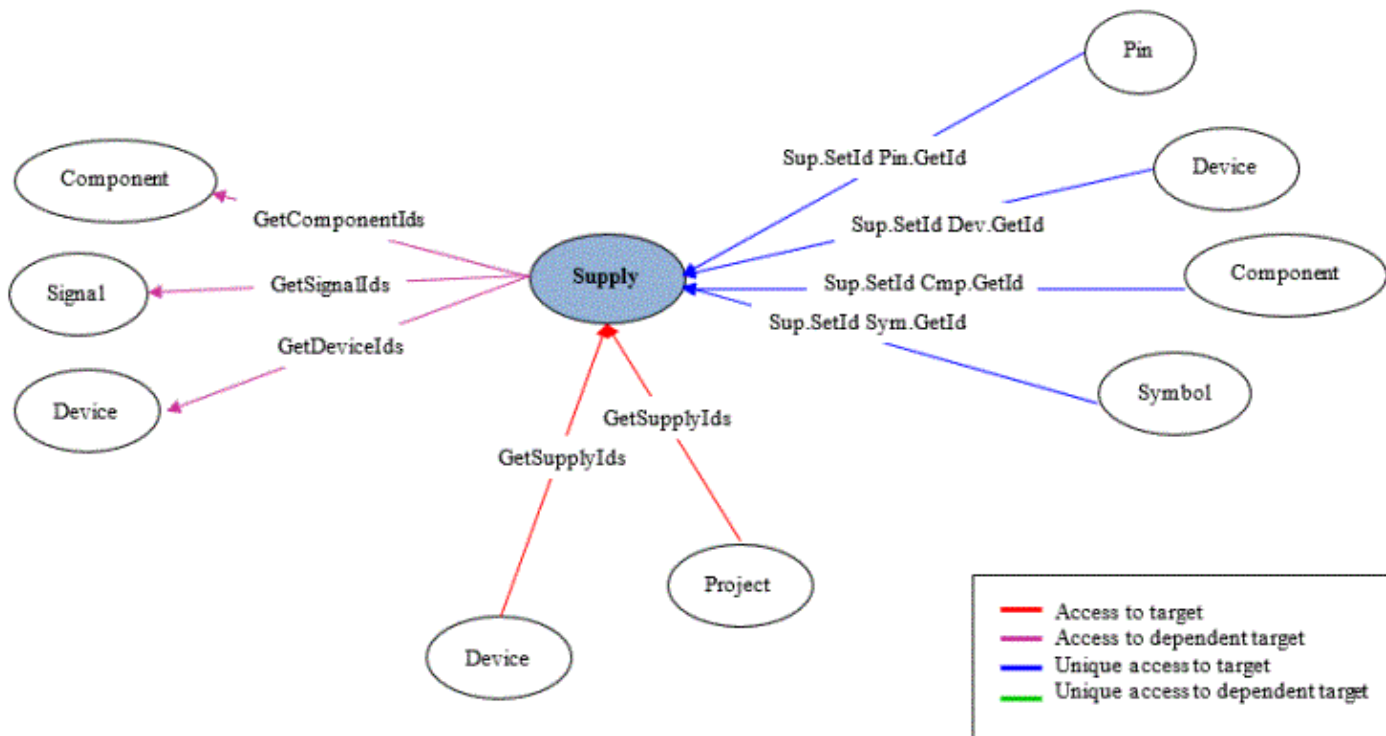
- **Handling Internal Identifiers - Overview-Dependencies**

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

## Supply

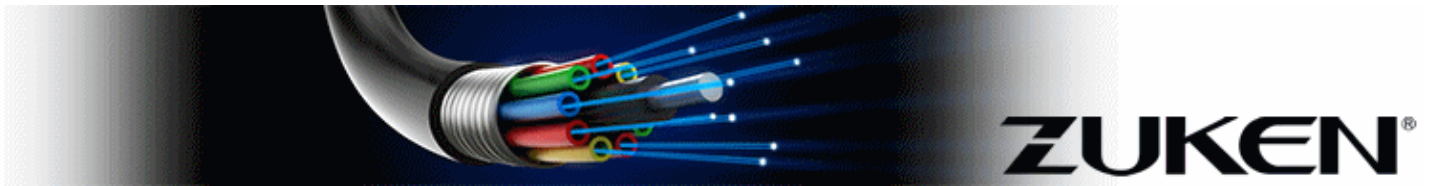


See also:

- **Handling Internal Identifiers - Overview-Dependencies**

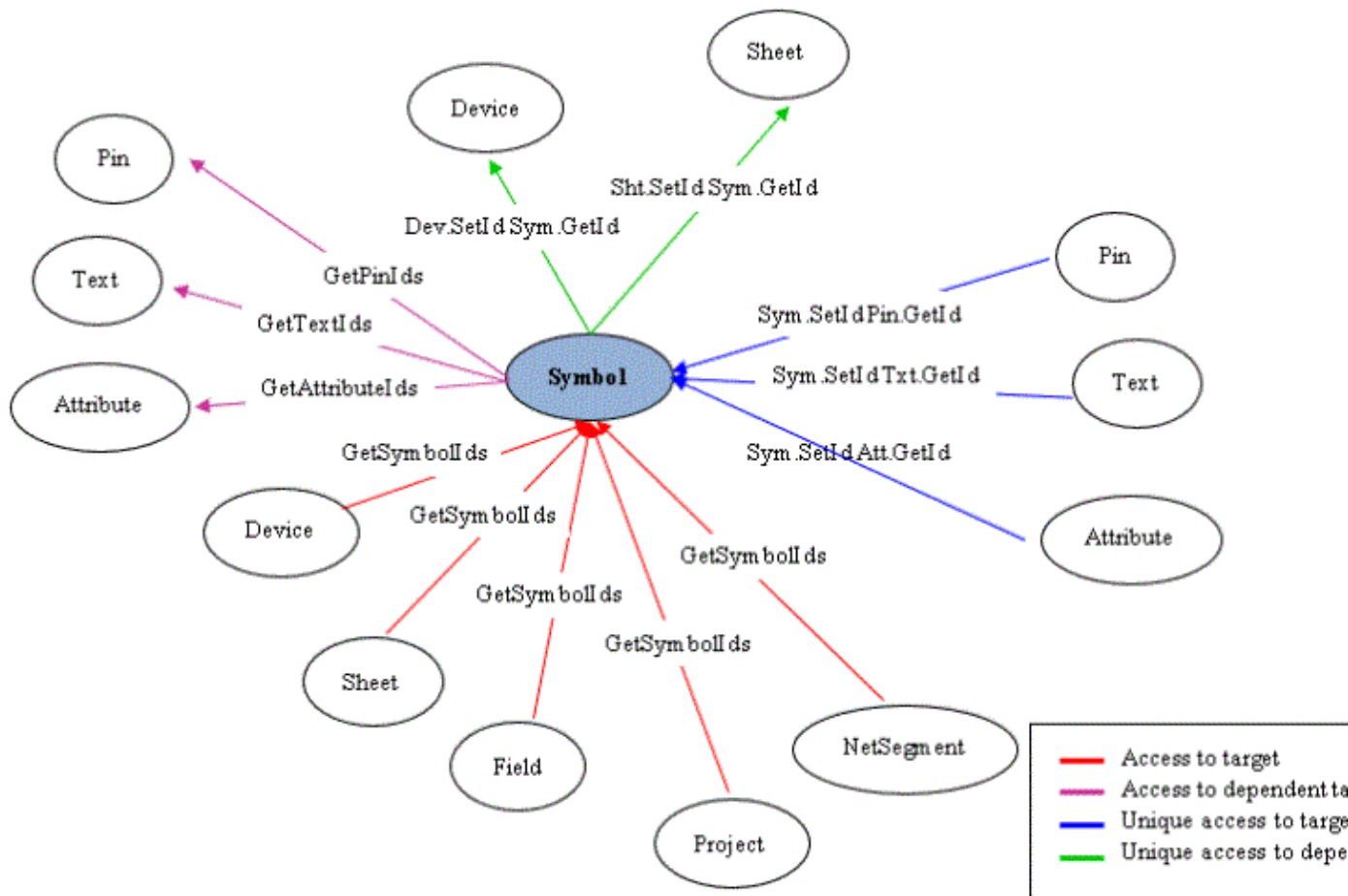
More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).





v2022-23.00

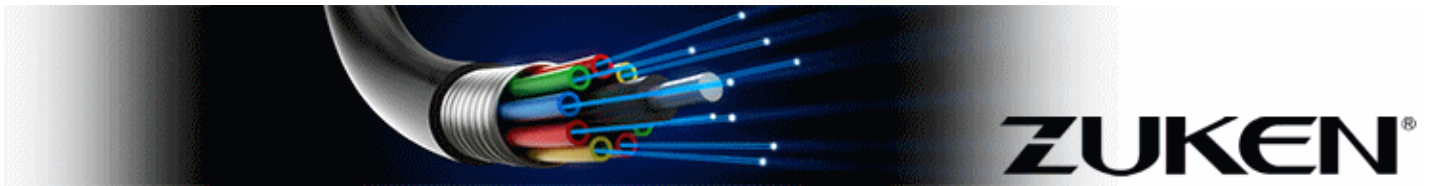
## Symbol



See also:

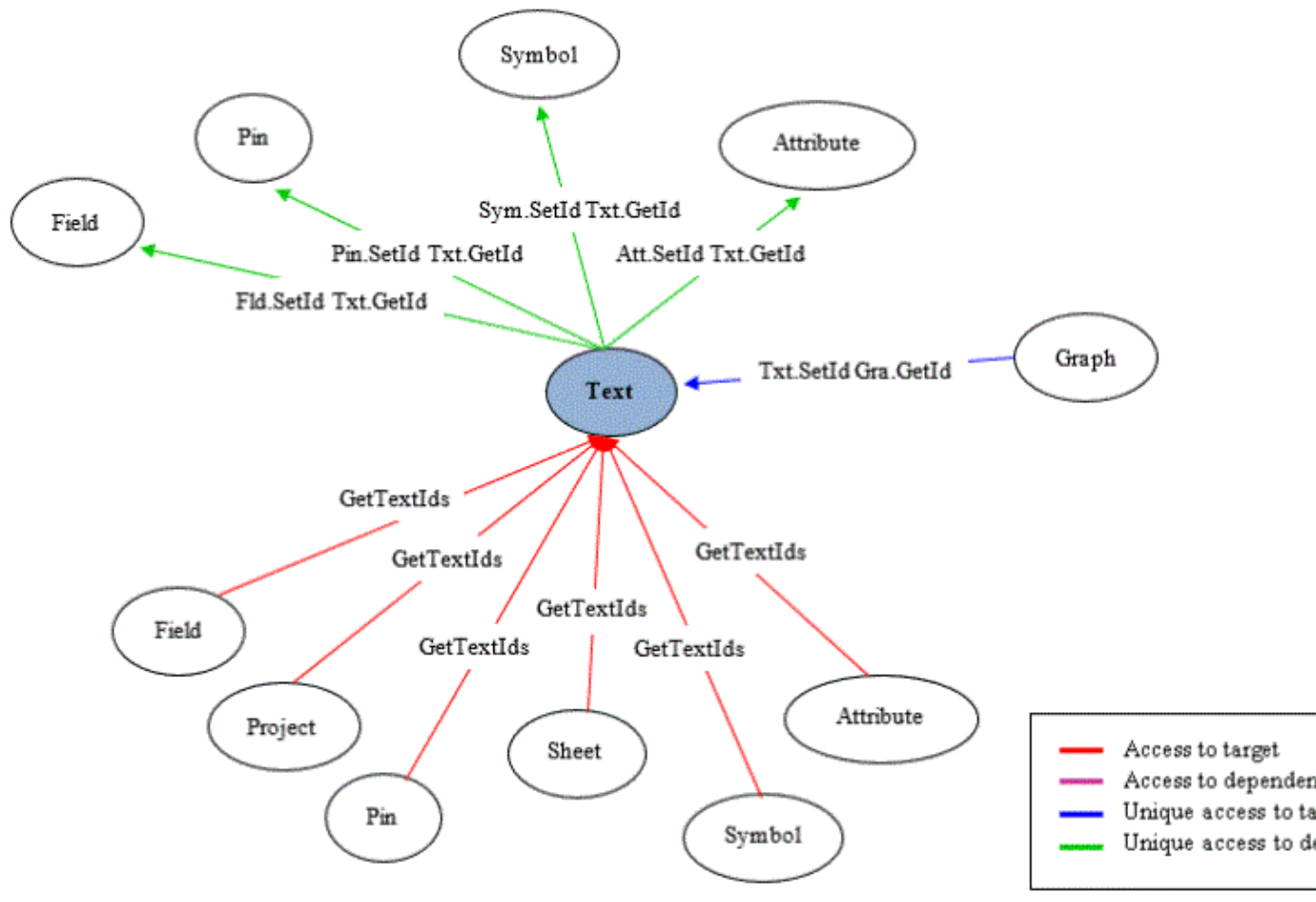
- **Handling Internal Identifiers - Overview-Dependencies**

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

## Text



See also:

- **Handling Internal Identifiers - Overview-Dependencies**

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

## Arguments

## Syntax

*String Arguments*

## Description

Standard **E<sup>3</sup>.series** start-up parameters.

## Possible Values

Value	Description
<filename>	Upon starting, the specified file will be opened
/cable	<b>E<sup>3</sup>.cable</b> module will be started (default)
/compareconfigfile	Indicates the path where the <b>.ini</b> configuration file is located for comparing projects
/comparenewfile	Indicates the path where the new <b>.e3s E<sup>3</sup>.series</b> project is located for comparing projects
/compareoldfile	Indicates the path where the old <b>.e3s E<sup>3</sup>.series</b> project is located for comparing projects
/dbe	<b>E<sup>3</sup>.series</b> will be started in Database Editor (DBE) mode
/demo	<b>E<sup>3</sup>.series</b> demonstration version will be started
/distdesign	<b>E<sup>3</sup>.series</b> will be started with distributed design functionality enabled
/economy	<b>E<sup>3</sup>.series</b> economy version will be started
/fluid	<b>E<sup>3</sup>.fluidPlus</b> module will be started
/formboard	<b>E<sup>3</sup>.formboard</b> module will be started
/functionaldesign	<b>E<sup>3</sup>.funtionaldesign</b> module will be started
/InprocRegOnly	Writes the COM interface to the Registry
/level	Allows the Levels dialog to be opened in <b>E<sup>3</sup>.view</b> if

## SetAsMaster - e3Symbol

used in combination with /view

/logic	<b>E<sup>3</sup>.logic</b> module will be started
/mucreate	Upon starting, a "Create a new multi-user project" dialog will appear if used in combination with /multiuser
/multiuser	Allows multi-user functionality when used in combination with /cable or /schema and not /sp or /sb
/muopen	Upon starting, an "Open a multi-user project" dialog will appear if used in combination with /multiuser
/muopen <project-name>	Upon starting the specified multi-user project will appear if used in combination with /multiuser
/new	Upon starting, a new project will be created and opened
/nocgm	<b>E<sup>3</sup>.series</b> will be started without CGM export functionality even if a licence for the functionality is available
/nodbe	<b>E<sup>3</sup>.series</b> will be started without the Database Editor
/noexportexf	<b>E<sup>3</sup>.series</b> will be started without EXF export functionality even if a licence for the functionality is available
/noimportruplan	<b>E<sup>3</sup>.series</b> will be started without Ruplan import functionality even if a licence for the functionality is available
/noimportstep	<b>E<sup>3</sup>.series</b> will be started without STEP import functionality even if a licence for the functionality is available
/nomilstandard	<b>E<sup>3</sup>.series</b> will be started without MIL-Standard functionality
/nopanel	<b>E<sup>3</sup>.series</b> will be started without Panel placement functionality even if a licence for the functionality is available
/nopdf	<b>E<sup>3</sup>.series</b> will be started without PDF export functionality regardless of licence
/nosplash	Start-up splash screen will not be displayed
/nowire	<b>E<sup>3</sup>.series</b> will be started without Panel wiring functionality regardless of licence
/noxvl	<b>E<sup>3</sup>.series</b> will be started without XVL export functionality regardless of licence
/plus	<b>E<sup>3</sup>.series</b> will be started with viewPlus functionality if used in combination with /view
/redliner	<b>E<sup>3</sup>.redliner</b> will be started
/register	<b>E<sup>3</sup>.series</b> will be started and the COM Interface will be registered

SetAsMaster - e3Symbol

Administrator rights are required

Module will be started as **E<sup>3</sup>.SmallProject** version when used in combination with module arguments

/sb [/sheet *nn*]

**E<sup>3</sup>.SmallProject** version limits the number of sheets to 30 sheets per project

/sp may be followed by /sheet and a number to specify a new limitation - For example: "/sp /sheet 20"

/schema

**E<sup>3</sup>.schema** module will be started

Module will be started as **E<sup>3</sup>.SmallProject** version when used in combination with module arguments

/sp [/sheet *nn*]

**E<sup>3</sup>.SmallProject** version limits the number of sheets to 30 sheets per project

/sp may be followed by /sheet and a number to specify a new limitation - For example: "/sp /sheet 20"

/startup="<*script*>.vbs"

Upon starting, the specified script is executed

/student

**E<sup>3</sup>.series** student version will be started

/topology

**E<sup>3</sup>.topology** module will be started

COM Interface will be unregistered

/unregister

Administrator rights are required

/view

**E<sup>3</sup>.series** will be started with viewer functionality

/wireLINK

**E<sup>3</sup>.series** wire-link version will be started

/Workspace=<configuration> **E<sup>3</sup>.series** will be started using the specified workspace configuration

## Remarks

Unless stated otherwise arguments can be compounded in any order so more than one can be used simultaneously. For example "/topology /formboard /multiuser".

Argument elements can be prefixed with "-" instead of "/".

Argument text is case insensitive.

---

## See Also

- [E3Starter.Start\(\)](#)
  - [E3Starter.StartSameImage\(\)](#)
-

## SetAsMaster - e3Symbol

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).





v2022-23.00

## Arrow Style



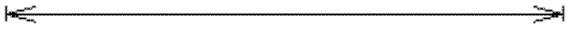



# Syntax

*Integer* *arrowstyles*

## Description

Parameter represents a arrow style value as an integer.

## Possible Values

Value	Arrow Style	Description
0		Filled
1		Closed
2		Open
3		Diagonal
4		Line
5		None

## Remarks

Arrow styles are used by dimension items for displaying the scope of the dimension.

## Version Information

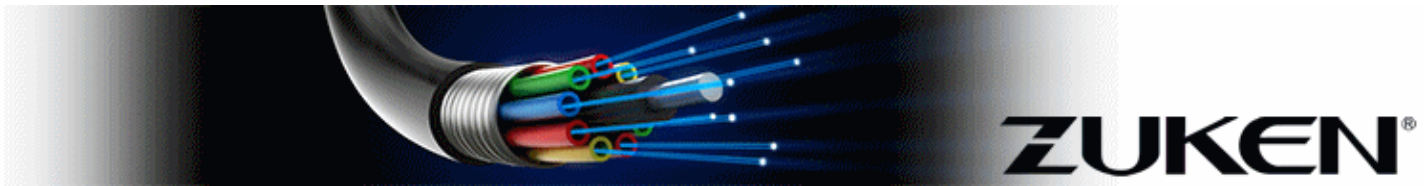
Introduced in v2009-8.50.

## See Also

- [e3Dimension.GetArrowMode\(\)](#)
- [e3Dimension.SetArrowMode\(\)](#)

## SetAsMaster - e3Symbol

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

## Attribute Name

## Syntax

*String* *AttributeName*

## Description

Parameter represents an **E<sup>3</sup>** attribute name as a string.

## Possible Predefined Values

Name	Display Name	Type
AdditionalPart	Additional Part	String
AddLength	Rest Length (mm)	Linear measure
Approval	Approval	String
ArticleType	Article Type	String
AttachedDocument	Data sheet	String
Attr4UseFromACPT	Attribute for use from active pin terminal	String
Attr4ViewSymbol	Attribute for use from View Symbol	String
AttrCurrent	Attr Current	String
AttrFile5	Attr File 5 (Circuit diagram)	String
AttrFile6	Attr File 6 (Product photo)	String
AttrFileArticleRepresentation	Attr File Article Representation	String
AttrFileBMP	Attr File BMP	String
AttrFileDXF	Attr File DXF	String

AttrFileJPG	Attr File JPG (Thumbnail)	String
AttrFileSTP	Attr File STP	String
AttrFileTIF	Attr File TIF	String
Attribute_Dimension	Example attribute owner Dimension	String
Attribute_Graphic	Example attribute owner Graphic	String
Attribute_Text	Example attribute owner Text	String
AttrVoltage	Attr Voltage	String
BendRadius	Bend radius	String
BOMpos	BOM Position	Integer
BundleType	Bundle Type	String
Car_body	Car_body	String
Car_code	Car_code	String
ChangeDate	Modification date	String
ChangeHistory	Modification history	String
ChangeReleased	Modification released	Boolean
ChangeShortText	Modification short text	String
ChangeText	Modification text	String
ChangeUser	Modification user	String
ChangeVersion	Modification index	String
Class_castellano	Database Class Spanish	String
Class_czech	Database Class Czech	String
Class_chinese	Database Class Chinese	String
Class_danish	Database Class Danish	String
Class_deutsch	Database Class German	String
Class_english	Database Class English	String
Class_finnish	Database Class Finnish	String
Class_francais	Database Class French	String
Class_greek	Database Class Greek	String
Class_hungarian	Database Class Hungarian	String
Class_italiano	Database Class Italian	String
Class_japanese		String

	Database Class Japanese	
Class_korean	Database Class Korean	String
Class_nederlands	Database Class Dutch	String
Class_norwegian	Database Class Norwegian	String
Class_polish	Database Class Polish	String
Class_portugues	Database Class Portuguese	String
Class_rumanian	Database Class Romanian	String
Class_russian	Database Class Russian	String
Class_slovenian	Database Class Slovene	String
Class_swedish	Database Class Swedish	String
Class_turkce	Database Class Turkish	String
cmNoComparison	rmNoComparision	Boolean
CompClass_Characteristic	(Class) Characteristic	String
CompClass_ContactsA	(Class) Auxillary Contacts	String
CompClass_ContactsM	(Class) Main Contacts	String
CompClass_CoreIdent	(Class) Core Identification	String
CompClass_Current	(Class) Current	String
CompClass_Diameter	(Class) Diameter	String
CompClass_NumOfCores	(Class) Number of cores	String
CompClass_NumOfPoles	(Class) Number of poles	String
CompClass_RPM	(Class) rpm	String
CompClass_Shield	(Class) Shield	String
CompClass_Ub	(Class) Operation voltage	String
ComponentGroup	Component Group	String
ConfigSubtractDescription	Configurator description	String
Copper	Copper Number	String
Country	Country	String

CSA	Permission CSA	String
Cut	Cut	Integer
CuttingLength	Cutting Length	String
DB_User	Database User	String
DemoFeature	New Feature	String
Description_castellano	Database Description Spanish	String
Description_chinese	Database Description Chinese	String
Description_czech	Database Description Czech	String
Description_danish	Database Description Danish	String
Description_deutsch	Database Description German	String
Description_english	Database Description English	String
Description_finnish	Database Description Finnish	String
Description_francais	Database Description French	String
Description_greek	Database Description Greek	String
Description_hungarian	Database Description Hungarian	String
Description_italiano	Database Description Italian	String
Description_japanese	Database Description Japanese	String
Description_korean	Database Description Korean	String
Descripton_nederlands	Database Description Dutch	String
Description_norwegian	Database Description Norwegian	String
Description_polish	Database Description Polish	String
Description_portugues	Database Description Portuguese	String
Description_rumanian	Database Description Romanian	String

Description_russian	Database Description Russian	String
Description_slovenian	Database Description Slovene	String
Description_swedish	Database Description Swedish	String
Description_text	Pin Description	String
Description_turkce	Database Description Turkish	String
Desina	Permission Desina	String
DevDes-EL-Customer	Customer device designation (electric)	String
DevDes-EL-Internal	Internal device designation (electric)	String
Device_Text_instances	Device designation (text instance)	String
DeviceName_Electrical	Device designation (electric)	String
DeviceName_Fluid	Device designation (fluid)	String
DeviceName_Pneumatic	Device designation (pneumatic)	String
DeviceSubFunction	Device sub-function	String
Diameter	Diameter for cables and cores (conductors)	String
Dismanteling	Amount of insulation to be cut back/stripped (mm)	Linear measure
Dismanteling_Shield	Amount of insulation to be cut back/stripped Shield (mm)	Linear measure
Displacement	Engine displacement	String
Display_in_ConnectionList	Display in Connection List	Boolean
Door	Door	String
DrillHoleDefinition	Drillhole Definition	String
E3.ePLM_Description	E3.ePLM Description	String
E3.ePLM_Function	E3.ePLM Function	String
E3.ePLM_FunctionName	E3.ePLM Function Name	String
EAN_Num	EAN Number	String

## SetAsMaster - e3Symbol

eCl@ss	eCl@ss class	String
eCl_Level0	eCl@ss Level0	String
eCl_Level1	eCl@ss Level1	String
eCl_Level2	eCl@ss Level2	String
eCl_Level3	eCl@ss Level3	String
Engine	Engine	String
ExcludeDevice	Exclude Device	Boolean
First_Pin_number	First pin number	String
FixDeviceName	Fix item designation	Boolean
Function	Function	String
FunctionalDescription	Functional Description	String
Group	Group	Integer
HighestChangeCount	Highest Change Number	Integer
InsulationType	Type of insulation	String
internalArticleNumber	Internal Article Number	String
InternalLength	Internal length	String
IsJIC	JIC Sheet	Boolean
KomaxID	KomaxID	String
Length	Length (mm)	Linear measure
MarkerArticleNumber	Marker Article Number	String
Material	Material	String
Max_CrossSection	Max. wire cross-section (mm <sup>2</sup> )	Real
Max_Wires	Max. number of wires	Integer
Min_CrossSection	Min. wire cross-section (mm <sup>2</sup> )	Real
MinBendRadius	Minimum Bend Radius	String
ModelName	M-CAD model name	String
NetSegCircumference	Net segment circumference	String
NeetSegCrossSection	Net segment cross-section	String
NominalWidth	Nominal Width	Integer
Option	Option	String
OuterDiameter	Thickness	String



Panel_Arrangement_FrontView_Closed	Panel arrangement, front view, closed	String
Panel_Arrangement_TopView	Panel arrangement, top view	String
PanelFrontViewSymbol	Panel front view symbol	String
PanelWireClass	Connection Class (Panel)	String
PID_EMSR-Function	PID EMSR-Points-Function	String
PID_EMSR-ItemDesignation	PID EMSR-Points-Item Designation	String
PID_MassFlow	PID Mass Flow	String
PID_OperatingPressure	PID Operating Pressure	String
PID_OperatingTemperature	PID Operating Temperature	String
PlcArticleNumber	PLC Article Number	String
PlcChannelDescription	PLC Channel Description	String
PlcComment	PLC Comment 1	String
PlcComment2	PLC Comment 2	String
PlcDevice	PLC Device	String
PlcDirection	PLC Direction	String
PlcFunctionCategory	PLC Function Category	String
PlcFunctionId	PLC Function ID	String
PlcFunctionType	PLC Function Type	String
PlcGroup	PLC Group	String
PlcIOType	PLC IO Type	String
PlcNetType	PLC Network Type	String
PlcObjectId	PLC Object ID	String
PlcPositionNumber	PLC Position Number	String
PlcSymbAdr	PLC Symbolic Address	String
PlcTopology	PLC Topology Device	String
PlcTopologyNet	PLC Topology Network	String
PlcType	PLC Type	String
PlcTypeIdentifier	PLC Type Identifier	String
PneumaticPart	Pneumatic part	Boolean
Power	Power	String

# SetAsMaster - e3Symbol

PowerLoss	Power loss W	String
PreferredDevice	Preferred Device	Boolean
Price	Price	Real
ProjectID	Project id	String
ProjectVersion	Project Version	String
Rack	Rack	String
ReleaseDate	Release Date	String
ReleaseShortText	Release Short Text	String
ReleaseText	Release Text	String
ReleaseUser	Release User	String
ReleaseVersion	Release Index	String
Revisor	Revisor	String
rmChangeDate	rmChangeDate	String
rmChangeHistory	rmChangeHistory	String
rmChangeIndex	rmChangeIndex	String
rmChangeReason	rmChangeReason	String
rmChangeReleased	rmChangeReleased	Boolean
rmChangeRevision	rmChangeRevision	String
rmChangeShortText	rmChangeShortText	String
rmChangeText	rmChangeText	String
rmChangeUser	rmChangeUser	String
rmChangeVersion	rmChangeVersion	String
rmLastUsedLanguage	rmLastUsedLanguage	String
rmMaxChangeNumber	rmMaxChangeNumber	Integer
rmReleaseDate	mReleaseDate	String
rmReleaseRevision	mReleaseRevision	String
rmRevisionShortText	mReleaseShortText	String
mReleaseText	mReleaseText	String
mReleaseUser	mReleaseUser	String
mReleaseVersion	mReleaseVersion	String
mVISFile	mVISFile	String
S7_Comment	S7 Comment	String
saberConnectorCurrent	Saber Connector Current	String
saberConnectorPin	Saber Connector Pin	String
saberConnectorVoltage	Saber Connector Voltage	String

saberCurrent	Saber Current	String
saberLib	Saber Library	String
saberModel	Saber Model Name	String
saberParam1	Saber Parameter1	String
saberParam2	Saber Parameter2	String
saberParam3	Saber Parameter3	String
saberParam4	Saber Parameter4	String
saberParam5	Saber Parameter5	String
saberParam6	Saber Parameter6	String
saberParam7	Saber Parameter7	String
saberParam8	Saber Parameter8	String
saberParam9	Saber Parameter9	String
saberParam10	Saber Parameter10	String
saberParam11	Saber Parameter11	String
saberParam12	Saber Parameter12	String
saberParam13	Saber Parameter13	String
saberParam14	Saber Parameter14	String
saberParam15	Saber Parameter15	String
saberParam16	Saber Parameter16	String
saberParam17	Saber Parameter17	String
saberParam18	Saber Parameter18	String
saberParam19	Saber Parameter19	String
saberParam20	Saber Parameter20	String
saberParam21	Saber Parameter21	String
saberParam22	Saber Parameter22	String
saberParam23	Saber Parameter23	String
saberParam24	Saber Parameter24	String
saberParam25	Saber Parameter25	String
saberParam26	Saber Parameter26	String
saberParam27	Saber Parameter27	String
saberParam28	Saber Parameter28	String
saberParam29	Saber Parameter29	String
saberParam30	Saber Parameter30	String
saberParam31	Saber Parameter31	String
saberParam32	Saber Parameter32	String
saberPin	Saber Pin	String

# SetAsMaster - e3Symbol

saberSimulationEnable	Saber Simulation Enable	Boolean
saberVoltage	Saber Voltage	String
Scope	Scope	String
SheetDate	Date	String
SheetMUPermission	Multi-User sheet permission	String
SheetName1	Name (1)	String
SheetName2	Name (2)	String
SheetType	Sheet Type	String
SheetUser	User	String
Slot	Slot	String
Source_DevDes	source item designation	String
Standard	Standard	String
Steering	Steering	String
Step	Increment	String
StorageDate	Project Storage date	String
Structure_Node_Attribute	Structure Node Attribute	String
SubProj	Sub-project	String
SUBTYP	Sub-type	Integer
Symbol_DatabaseEditor	Symbol generator	String
Symbolpintext-extended	Symbol pin text extended	String
Techdat1	Technical description 1	String
Techdat2	Technical description 2	String
Techdat3	Technical description 3	String
Techdat4	Technical description 4	String
Techdat5	Technical description 5	String
TerminalGroupCode	Terminal Group Code	String
TermPlanAssignedSymbol	Symbol for cable plan	String
Total_CrossSection	Max. wire cross-section (mm <sup>2</sup> ), total	Real
TransferDate	Project Transfer date	String
Transmission	Transmission	String
UL	Permission UL	String
Voltage	Voltage Uo/U	String

# SetAsMaster - e3Symbol

Weight	Weight	String
widgenDestinationInfo	WIDGEN Destination Info	String
widgenHarnessConfiguration	WIDGEN Harness Configuration	String
widgenPindescription	WIDGEN Pin Description	String
widgenPinDestinationRecord	WIDGEN Pin Destination Record	String
widgenSignalInfo	WIDGEN Signal Info	String
widgenVariantFamily	WIDGEN Variant Family	String
widgenVariantOption	WIDGEN Variant Option	String
widgenWireType	WIDGEN Wire Type	String
WireNumber	Wire Number	String
WireNumber_Add	Additional Wire number	String
WireNumber_Wire	Wire Number (Core)	String
.ACTIVE_COMPOSITE564	Active mating connector	String
.ARRANGEMENT_SYMBOL	Active mating connector	String
.ASSIGNED_END_BRACKET	Assigned end bracket	String
.ASSIGNED_END_COVER	Assigned end cover	String
.ASSIGNED_SEPARATION_PLATE	Assigned separating plate	String
.BACKPLANE_WIRE	Backplane wiring (grid)	Linear measure
.BLOCK_PIN_SYMBOL	Symbol for block connector	String
.BRIDGE	Jumper for terminal	Integer
.CABLE_DUCT_LENGTH	Length of cable duct	Linear measure
.CABLENAME	Cable Device designation	String
.CABLESHAPE	Cable Shape	String
.CABLETYPE	Cable Component code	String
.CALC_PARAM_AMBIENT_TEMP	Ambient temperature [degrees C]	Real
.CALC_PARAM_CURRENT	Current [A]	Real

.CALC_PARAM_CURRENT_SWITCH_EXPRESSION	Current switch expression	String
.CALC_PARAM_FUSE_SIZE	Fuse rating [A]	Real
.CALC_PARAM_LOAD_FACTOR	Load factor	Real
.CALC_PARAM_NOMINAL_TEMP	Nominal temperature [degrees C]	Real
.CALC_PARAM_OVER_CURRENT_RATIO	Over current ratio	Real
.CALC_PARAM_PERMISSION_CURRENT	Permission current [A]	Real
.CALC_PARAM_PIN_PROTECTION	Pin protection	Boolean
.CALC_PARAM_POWER_GROUND	Ground	Boolean
.CALC_PARAM_RESISTANCE	Resistance (for Calculation) [Ohm]	Real
.CALC_PARAM_VALID_SWITCH_EXPRESSION	Valid switch expression	String
.CALC_PARAM_VOLTAGE	Voltage supply [V]	Real
.CALC_PARAM_VOLTAGE_MAX	Maximum voltage value [V]	Real
.CALC_PARAM_VOLTAGE_MIN	Minimum voltage value [V]	Real
.CALC_PARAM_WIRE_KIND_ID	Wire kind	Integer
.CALC_RESULT_ACTUAL_RESISTANCE	Actual resistance [Ohm]	Real
.CALC_RESULT_CONDUCTOR_TEMP	Calculated: Conductor temperature [degrees C]	Real
.CALC_RESULT_CURRENT	Calculated: Current [A]	Real
.CALC_RESULT_FUSE_SIZE	Calculated: Fuse rating [A]	Real
.CALC_RESULT_LOAD_FACTOR	Calculated: Load factor	Real
.CALC_RESULT_OPTIMUM_CURRENT	Calculated: Optimum current [A]	Real
.CALC_RESULT_PERMISSION_CURRENT	Calculated: Permission current [A]	Real
.CALC_RESULT_VOLTAGE	Calculated: Supply voltage [V]	Real
.CALC_RESULT_VOLTAGE_DROP	Voltage drop [V]	Real
.CALC_RESULT_WIRE_IGNITION_POINT_TIME	Calculated: Wire Ignition point time [sec]	Real
.cmChangeInformation	rmChangeInformation	String
.cmData	rmData_1	String

.CONNECTOR_NAME	Internal device designation	String
.CORE_MANUFACTURING_LENGTH	Core (conductor) manufacturing length	Linear measure
.COST	Cost factor	String
.DEVICE_MAPPING_CODE	Device Mapping	String
.DOCUMENT_TYPE	Document Type	String
.DRILL_HOLE_GRID	Drill Hole Grid	String
.E3.ePLM.ActionField	E3.ePLM Action Field	String
.E3.ePLM_ID	E3.ePLM ID	String
.ECHECK_ACTUAL_CURRENT	Actual current (result) [A]	Real
.ECHECK_ACTUAL_TEMPERATURE	Actual temperature (result) [°C]	Real
.ECHECK_ACTUAL_VOLTAGE	Actual voltage (result) [V]	Real
.ECHECK_AMBIENT_TEMPERATURE	Ambient temperature [°C]	Real
.ECHECK_COLOUR	Wire colour	Integer
.ECHECK_CROSSEC	Wire cross-section	Real
.ECHECK_FUSE_OPERATING_TIME	Fuse operating time at given percentage of amperage rating [%][s]	String
.ECHECK_FUSE_RERATING_TEMPERATURE	Fuse rerating temperature	Real
.ECHECK_FUSE_RERATING_TEMPERATURE_RATING	Fuse current rerating factor at given temperature [°C][%]	String
.ECHECK_INTERNAL_PIN_RESISTANCE	Internal pin resistance [Ohm]	Real
.ECHECK_INTERNAL_RESISTANCE	Internal resistance [Ohm]	Real
.ECHECK_IN_ACTIVE	eCheck is active	Boolean
.ECHECK_MAXIMUM_CURRENT	Maximum current [A]	Real
.ECHECK_MAXIMUM_TEMPERATURE	Maximum temperature [°C]	Real
.ECHECK_MAXIMUM_VOLTAGE	Maximum voltage [V]	Real
.ECHECK_MINIMUM_CURRENT	Minimum current [A]	Real
.ECHECK_MINIMUM_VOLTAGE	Minimum voltage [V]	Real
.ECHECK_NOMINAL_CURRENT	Nominal current [A]	Real

.ECHECK_NOMINAL_VOLTAGE	Nominal voltage [V]	Real
.ECHECK_POWER	Power [Watt]	Real
.ECHECK_RATING	Rating (max. current fuse)	Real
.ECHECK_RESISTANCE	Specific resistance [Ohm x mm <sup>2</sup> /m]	Real
.ECHECK_TEST	Check passed	Boolean
.ECHECK_VOLTAGE_DROP	Voltage drop (Diode) [V]	Real
.ECHECK_WEIGHT	Wire weight [kg/km]	Real
.ENTRY_CABLEDUCT_FOR_AUTOCONNECT	Cable duct entry for autoconnect	String
.ENTRY_PORT	Terminal port entry	String
.EXTERN	External connection	String
.FILEINFO_CATEGORY	File information category	String
.FILEINFO_COMMENTS	File information comments	String
.FILEINFO_KEYWORDS	File information keywords	String
.FILEINFO_SUBJECT	File information subject	String
.FILEINFO_TITLE	File information title	String
.FUNIT_SYMBOL	Functional unit symbol	String
.HARNESS_FOLDER_PATH	Harness Drawings Folder Path	String
.HD_DRAW_FILL_COLOR	HarnessDesigner Drawing Fill Color	Integer
.HD_DRAW_FILL_TYPE	HarnessDesigner Drawing Fill Type	Integer
.HD_DRAW_LINE_COLOR	HarnessDesigner Drawing Line Color	Integer
.HD_DRAW_LINE_TYPE	HarnessDesigner Drawing Line Type	Integer
.HD_DRAW_LINE_WIDTH	HarnessDesigner Drawing Line Width	Real
.HYPERLINK	Hyperlink	String
.INFINITE_MODEL_TYPE	Infinite Model Type	String
.INFINITE_ORIGINAL_WIRE_BOOLEAN_EXPRESSION	E3.infinite Original Wire boolean expression	String



.INFINITE_ORIGINAL_WIRE_ID	E3.infinite Original Wire id	String
.INFINITE_ORIGINAL_WIRE_NAME	E3.infinite Original Wire name	String
.INTERN	Internal connection	String
.IS_FORMBOARD_SYMBOL	Used for formboard symbol	Integer
.ISOLATE	Insulation	String
.LENGTH	Net segment length	Linear measure
.MU_PROJPERM	Multi-user permission	String
.NET_CLASS	Net class	String
.NET_ROUTE_CODE	Net route code	String
.NET_SEGMENT_NAME	Net segment name	String
.NET_SPACING	Net spacing	String
.NUMBER_OF_WINDINGS_FOR_CORES	Number of windings per meter for cores (conductors)	Integer
.OFFSET_X	X-offset	Linear measure
.OFFSET_Y	Y-offset	Linear measure
.PADDRILL	Pad Drill diameter (PCB)	Linear measure
.PIN_SYMBOL	Symbol for Connector Pin	String
.PINVIEW_SYMBOL	Symbol for pin view	String
.PLCADDR	PLC starting address	String
.PLCPHYSADDR	PLC Physical address	String
.PREFERRED_VIEW	Default View Number	integer
.PREVIEW_SYMBOL	Preview Symbol	String
.REFERENCED_PIN	Referenced Pin	String
.ROUTE_TARGET_NUMBER	Route Target Number	String
.SAPSTATUSTEXT	Status SAP	String
.SCHEMATIC_TYPE	Schematic Type	Integer
.SORT_FORMAT_FILE	sort format file	String
.STRUCTURE_NODE_EXT_NAME	Structure Node ext name	String
.STRUCTURE_NODE_ID	Structure Node ID	String

.STRUCTURE_NODE_VALUE	Structure Node value	String
.STRUCTURE_TEMPLATE_FILE	Structure template file	String
.TOPOLOGY_FOLDER_PATH		String
.TOPOLOGY_SYMBOL	Topology symbol	String
.VALID_COMPOSITE	Valid mating connector	String
.VALID_CONNPART	Valid cavity part	String
.VARIATION_FILE_NAME	Variation File Name	String
.VARIATION_FOLDER_PATH	Variation Folder Path	String
.WIRETYPE_DEFAULT	Default Wire Type	String
.XVL_PROJECT_NAME	XVL ProjectName	String
.XVL_STRUCTURE_CABINET	XVL CabinetName	String
.XVL_STRUCTURE_MAIN	XVL StructureMain	String
.XVL_WIRING_ORDER_NUMBER	XVL WiringOrderNumber	Integer

Attributes beginning with a '.' character are exclusively used to store internal information.

---

## See Also

- [e3Attribute - Overview](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

## Attribute Definition Properties

# Syntax

*2D-Array properties*

## Description

Parameter represents an array of attribute definition properties and their values.

## Possible Values

Property Name	Property Value	Description
"Owner"	"1"	Block connector
"Owner"	"2"	Block device
"Owner"	"3"	Block pin
"Owner"	"4"	Bundle
"Owner"	"5"	Cable
"Owner"	"6"	Cable core
"Owner"	"7"	Cable core end
"Owner"	"8"	Cable end
"Owner"	"9"	Cable type
"Owner"	"10"	Cable type end
"Owner"	"11"	Component
"Owner"	"12"	Component pin
"Owner"	"13"	Connector
"Owner"	"14"	Connector pin
"Owner"	"15"	Core type
"Owner"	"16"	Core type end
"Owner"	"17"	Database symbol

# SetAsMaster - e3Symbol

"Owner"	"18"	Device
"Owner"	"19"	Device pin
"Owner"	"20"	Dimension
"Owner"	"21"	Field symbol
"Owner"	"22"	Functional port
"Owner"	"23"	Functional unit
"Owner"	"24"	Graphic
"Owner"	"25"	Group
"Owner"	"26"	Hose/tube
"Owner"	"27"	Hose/tube end
"Owner"	"28"	Hose/tube inside
"Owner"	"29"	Hose/tube inside end
"Owner"	"30"	Hose/tube inside type
"Owner"	"31"	Hose/tube inside type end
"Owner"	"32"	Hose/tube type
"Owner"	"33"	Hose/tube type end
"Owner"	"34"	Model
"Owner"	"35"	Module
"Owner"	"36"	Net
"Owner"	"37"	Net node
"Owner"	"38"	Net segment
"Owner"	"39"	Project
"Owner"	"40"	Sheet
"Owner"	"41"	Sheet (database)
"Owner"	"42"	Signal
"Owner"	"43"	Signal class
"Owner"	"44"	Signal node
"Owner"	"45"	Symbol
"Owner"	"46"	Text
"Owner"	"47"	Variant/options
"Type"	"1"	Integer
"Type"	"2"	Real
"Type"	"3"	Linear measure
"Type"	"4"	String
"Type"	"5"	Boolean ( yes/no )
"Single instance"	"0"	No

## SetAsMaster - e3Symbol

"Single instance"	"1"	Yes
"Unique value"	"0"	Not unique
"Unique value"	"1"	Object
"Unique value"	"2"	Project
"Unique value"	"3"	Assignment
"Unique value"	"4"	Location
"Unique value"	"5"	Assignment and location
"Must exist"	"0"	No
"Must exist"	"1"	Yes
"Changeable by script only"	"0"	No
"Changeable by script only"	"1"	Yes
"Default value"	"<Value>"	Free text including an empty string
"List of values"	"<Value>"	Attribute list name or an empty string
"Changeable when owner is locked"	"0"	No
"Changeable when owner is locked"	"1"	Yes
"Allow change of lock behaviour"	"0"	No
"Allow change of lock behaviour"	"1"	Yes
"Format"	"<Value>"	Free text including an empty string
"Colour"	"<Color>"	See <a href="#">Colors</a> for a list of values
"Size"	"<Value>"	Positive real value
"Pos x"	"<Value>"	Real value
"Pos y"	"<Value>"	Real value
"Ratio"	"1"	Normal
"Ratio"	"2"	Narrow
"Ratio"	"3"	Wide
"Direction"	"1"	Left aligned
"Direction"	"2"	Center aligned
"Direction"	"3"	Right aligned
"Level"	"<Value>"	Integer between 1 and 256
"Visibility"	"0"	No
"Visibility"	"1"	Yes

## Remarks

The owner property may appear more than once with different property values. All other properties should appear once.

Property names are case-insensitive.

## Version Information

Introduced in v2017-18.00.

---

## See Also

- [e3AttributeDefinition.Create\(\)](#)
  - [e3AttributeDefinition.Get\(\)](#)
  - [e3AttributeDefinition.GetFromDatabase\(\)](#)
  - [e3AttributeDefinition.Set\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



**v2022-23.00**

## Ballooning

# Syntax

*Integer*ballooning

## Description

Parameter represents a ballooning value as an integer.

## Possible Values

Value	Description
0	No ballooning
1	Circle around text
2	Oval around text
4	Rectangle around text
8	Ellipse around text
16	Line to owner
32	Horizontal line on centre of text box
64	Horizontal line on bottom of text box
128	Horizontal line on top of text box
	Available since v2018-19.00

## Remarks

Ballooning flags should be used independently with the exception of Line to owner (16) which can be used in combination with circle (1), oval (2), rectangle (4) or Ellipse (8). For example a ballooning value of 20 represents a rectangle and Line to owner.

Line to owner is only applicable if the text has an owner.

## Version Information

Introduced in v2010-9.10.

Modified in v2018-19.00.

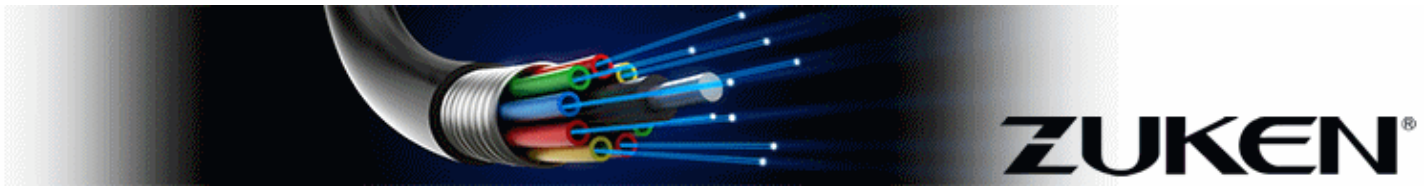
---

## See Also

- [e3DbText.GetBallooning\(\)](#)
  - [e3DbText.SetBallooning\(\)](#)
  - [e3Text.CalculateBoxAt\(\)](#)
  - [e3Text.GetBalllooning\(\)](#)
  - [e3Text.SetBallooning\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).





v2022-23.00

## Colors

# Syntax

*Integer*color

## Description

Parameter represents a color as an integer.

## Possible Predefined Values

Value **Color** Red, Green, Blue Color Space

-1	Automatic
0	0,0,0
1	128, 0, 0
2	0, 128, 0
3	128, 128, 0
4	0, 0, 128
5	128, 0, 128
6	0, 128, 128
7	192, 192, 192
8	192, 220, 192
9	166, 202, 240
10	244, 244, 244
11	160, 160, 160
12	128, 128, 128
13	240, 0, 0
14	0, 240, 0
15	240, 240, 0
16	0, 0, 255

# SetAsMaster - e3Symbol

17	255, 0, 255
18	0, 255, 255
19	255, 255, 255
20	36, 0, 0
21	72, 0, 0
22	109, 0, 0
23	145, 0, 0
24	182, 0, 0
25	218, 0, 0
26	0, 36, 0
27	36, 36, 0
28	72, 36, 0
29	109, 36, 0
30	145, 36, 0
31	182, 36, 0
32	218, 36, 0
33	255, 36, 0
34	0, 72, 0
35	36, 72, 0
36	72, 72, 0
37	109, 72, 0
38	145, 72, 0
39	182, 72, 0
40	218, 72, 0
41	255, 72, 0
42	0, 109, 0
43	36, 109, 0
44	72, 109, 0
45	109, 109, 0
46	145, 109, 0
47	182, 109, 0
48	218, 109, 0
49	255, 109, 0
50	0, 145, 0
51	36, 145, 0
52	72, 145, 0

# SetAsMaster - e3Symbol

53	109, 145, 0
54	145, 145, 0
55	182, 145, 0
56	218, 145, 0
57	255, 145, 0
58	0, 182, 0
59	36, 182, 0
60	72, 182, 0
61	109, 182, 0
62	145, 182, 0
63	182, 182, 0
64	218, 182, 0
65	255, 182, 0
66	0, 218, 0
67	36, 218, 0
68	72, 218, 0
69	109, 218, 0
70	145, 218, 0
71	182, 218, 0
72	218, 218, 0
73	255, 218, 0
74	36, 255, 0
75	72, 255, 0
76	109, 255, 0
77	145, 255, 0
78	182, 255, 0
79	218, 255, 0
80	0, 0, 85
81	36, 0, 85
82	72, 0, 85
83	109, 0, 85
84	145, 0, 85
85	182, 0, 85
86	218, 0, 85
87	255, 0, 85
88	0, 36, 85
89	36, 36, 85

90	72, 36, 85
91	109, 36, 85
92	145, 36, 85
93	182, 36, 85
94	218, 36, 85
95	255, 36, 85
96	0, 72, 85
97	36, 72, 85
98	72, 72, 85
99	109, 72, 85
100	145, 72, 85
101	182, 72, 85
102	218, 72, 85
103	255, 72, 85
104	0, 109, 85
105	36, 109, 85
106	72, 109, 85
107	109, 109, 85
108	145, 109, 85
109	182, 109, 85
110	218, 109, 85
111	255, 109, 85
112	0, 145, 85
113	36, 145, 85
114	72, 145, 85
115	109, 145, 85
116	145, 145, 85
117	182, 145, 85
118	218, 145, 85
119	255, 145, 85
120	0, 182, 85
121	36, 182, 85
122	72, 182, 85
123	109, 182, 85
124	145, 182, 85
125	182, 182, 85

126	218, 182, 85
127	255, 182, 85
128	0, 218, 85
129	36, 218, 85
130	72, 218, 85
131	109, 218, 85
132	145, 218, 85
133	182, 218, 85
134	218, 218, 85
135	255, 218, 85
136	0, 255, 85
137	36, 255, 85
138	72, 255, 85
139	109, 255, 85
140	145, 255, 85
141	182, 255, 85
142	218, 255, 85
143	255, 255, 85
144	0, 0, 170
145	36, 0, 170
146	72, 0, 170
147	109, 0, 170
148	145, 0, 170
149	182, 0, 170
150	218, 0, 170
151	255, 0, 170
152	0, 36, 170
153	36, 36, 170
154	72, 36, 170
155	109, 36, 170
156	145, 36, 170
157	182, 36, 170
158	218, 36, 170
159	255, 36, 170
160	0, 72, 170
161	36, 72, 170
162	72, 72, 170

163	109, 72, 170
164	145, 72, 170
165	182, 72, 170
166	218, 72, 170
167	255, 72, 170
168	0, 109, 170
169	36, 109, 170
170	72, 109, 170
171	109, 109, 170
172	145, 109, 170
173	182, 109, 170
174	218, 109, 170
175	255, 109, 170
176	0, 145, 170
177	36, 145, 170
178	72, 145, 170
179	109, 145, 170
180	145, 145, 170
181	182, 145, 170
182	218, 145, 170
183	255, 145, 170
184	0, 182, 170
185	36, 182, 170
186	72, 182, 170
187	109, 182, 170
188	145, 182, 170
189	182, 182, 170
190	218, 182, 170
191	255, 182, 170
192	0, 218, 170
193	36, 218, 170
194	72, 218, 170
195	109, 218, 170
196	145, 218, 170
197	182, 218, 170
198	218, 218, 170

199	255, 218, 170
200	0, 255, 170
201	36, 255, 170
202	72, 255, 170
203	109, 255, 170
204	145, 255, 170
205	182, 255, 170
206	218, 255, 170
207	255, 255, 170
208	36, 0, 255
209	72, 0, 255
210	109, 0, 255
211	145, 0, 255
212	182, 0, 255
213	218, 0, 255
214	0, 36, 255
215	36, 36, 255
216	72, 36, 255
217	109, 36, 255
218	145, 36, 255
219	182, 36, 255
220	218, 36, 255
221	255, 36, 255
222	0, 72, 255
223	36, 72, 255
224	72, 72, 255
225	109, 72, 255
226	145, 72, 255
227	182, 72, 255
228	218, 72, 255
229	255, 72, 255
230	0, 109, 255
231	36, 109, 255
232	72, 109, 255
233	109, 109, 255
234	145, 109, 255
235	182, 109, 255

236	218, 109, 255
237	255, 109, 255
238	0, 145, 255
239	36, 145, 255
240	72, 145, 255
241	109, 145, 255
242	145, 145, 255
243	182, 145, 255
244	218, 145, 255
245	255, 145, 255
246	0, 182, 255
247	36, 182, 255
248	72, 182, 255
249	109, 182, 255
250	145, 182, 255
251	182, 182, 255
252	218, 182, 255
253	255, 182, 255
254	0, 218, 255
255	36, 218, 255

## Remarks

Color values can be changed using the **E<sup>3</sup>** Database Editor.

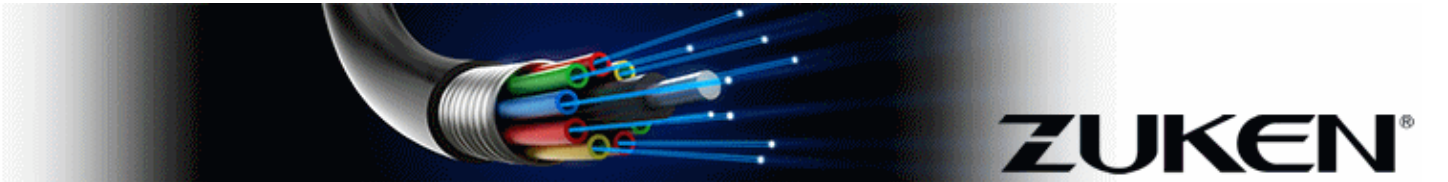
---

## See Also

- [e3Graph.GetColour\(\)](#)
- [e3Graph.GetHatchColour\(\)](#)
- [e3.Graph.GetLineColour\(\)](#)
- [e3.Graph.SetColour\(\)](#)
- [e3.Graph.SetHatchColour\(\)](#)
- [e3.Graph.SetLineColour\(\)](#)
- [e3Text.GetColour\(\)](#)
- [e3Text.SetColour\(\)](#)



More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

Line Styles

Syntax

Integerlinestyles

Description

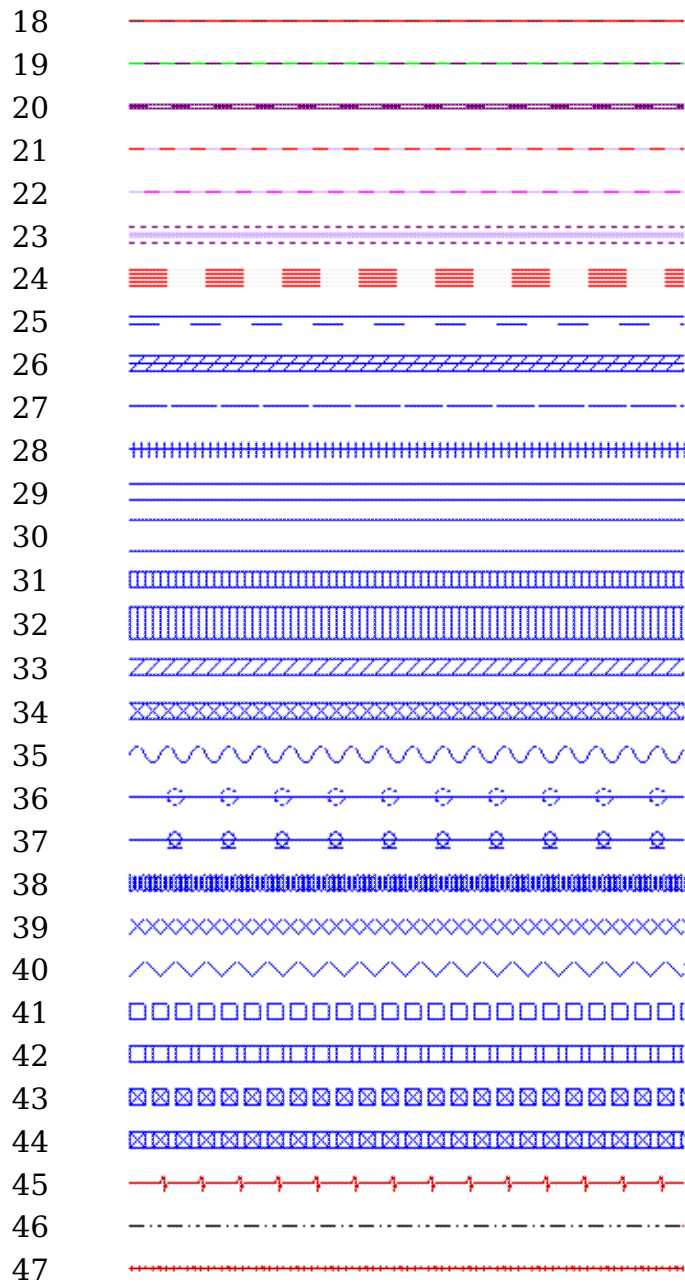
Parameter represents a line style value as an integer.

Possible Fixed Values

Value	Line Style
1	_____
2	-----
3	- - - - -
4	— — — — —

Possible Predefined Values

Value	Line Style
5	_____
6	=====
7	~~~~~
8	=====
9	=====
10	=====
11	=====
12	=====
13	=====
14	=====
15	=====
16	=====
17	=====



## Remarks

Line styles are used by graphic items for lines and hatching.

## Version Information

Introduced in v2010-9.10.

## See Also

- [e3Graph.GetHatchLineStyle\(\)](#)
  - [e3Graph.GetLineStyle\(\)](#)
  - [e3Graph.SetHatchLineStyle\(\)](#)
  - [e3.Graph.SetLineStyle\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

## Menu Item Text

# Syntax

*String text*

## Description

Parameter represents a menu item's display text and menu position.

## Possible Values

Value	Description
"<Menu Item Name>"	Menu item is placed in the menu bar  Example: "User Tool" If a separator menu item is created, it is placed at the end of the defined menu
"<Menu Name>\"	If a menu item is created, it is placed at the end of the defined menu with no name  Example: "Add-ons\ Menu item is placed at the end of the defined menu
"<Menu Name>\<Menu Item Name>"	If the defined menu does not exist, it is created  Example: "Add-ons\User Tool" Menu item is placed under the defined position in the menu
"<Menu Name>\<Menu Item Name>@<Position>"	If <Position> is 0, the menu item is placed at the first position in the menu  Example: "Add-ons\User Tool@3"

	Menu item is placed under the defined existing menu item
"<Menu Item Name>@:<Menu Item Identifier>"	If the defined menu item does not exist, the user tool is created however no menu item is created
	Example: "User Tool@:1"
	Menu item is placed under the defined existing menu item
"<Menu Item Identifier>"	The menu item's name is defined by the currently defined command value
	Example: "1"

## Remarks

The menu names defined in the menu item text should be expressed in the current **E<sup>3</sup>**.series installation language.

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3UserMenuItem.Create\(\)](#)
  - [e3UserMenuItem.CreateContextSeparator\(\)](#)
  - [e3UserMenuItem.CreateContextUserTool\(\)](#)
  - [e3UserMenuItem.CreateSeparator\(\)](#)
  - [e3UserMenuItem.CreateUserTool\(\)](#)
  - [e3UserMenuItem.GetText\(\)](#)
  - [e3UserMenuItem.SetText\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

## Electric Settings

# Syntax

*String name*

## Description

Parameter represents a unique name identifying the **E<sup>3</sup>** setting value.

## Possible Values

If the location of the setting within **E<sup>3</sup>.series** is known and the setting name is required, please reference the following table. If the setting value name is known and the location of the setting within **E<sup>3</sup>.series** is required, please refer to Setting Value Names. For location of values in Fluid Settings see Fluid Project Settings.

Location in  
**E<sup>3</sup>.series**  
Electrical  
Settings

### Setting Value Name

### API Function

#### General □

Working Grid □  
Grid size

SCHEMAGRIDSIZE

e3Job.GetS

• e3Job.Se

e3Job.GetC

e3Job.SetC

e3Job.GetS

#### General □

Working Grid □  
Snap size

GRAPHIC\_SNAPSIZE

SCHEMATRAPSIZE

• e3Job.Se

e3Job.GetT

e3Job.SetT

e3Job.GetS

#### General □

Alternative Grid □

SCHEMAALTGRIDSIZE

Possible Values

607

Grid size

• [e3Job.Set](#)

[e3Job.GetA](#)

[e3Job.SetA](#)

[e3Job.GetS](#)

• [e3Job.Set](#)

[e3Job.GetM](#)

[e3Job.SetM](#)

[e3Job.GetS](#)

• [e3Job.Set](#)

[e3Job.GetM](#)

[e3Job.SetM](#)

[e3Job.GetS](#)

• [e3Job.Set](#)

**General** ☐

Measurement

MEA\_EXTERN\_SCHEMA

Units ☐

Millimeters

**General** ☐

Measurement

MEA\_EXTERN\_SCHEMA

Units ☐ Inches

**General** ☐ Save ☐

Automatically

generate backup AUTOSAVE\_ENABLED

file after (

CheckBox )

**General** ☐ Save ☐

Automatically

generate backup SAVLIMIT

file after (

SpinControl )

**General** ☐ Save ☐

Write messages

and result to file

**General** ☐

Template ☐ File

name

**General** ☐ Layout ☐

Layout measure

**General** ☐

Compatibility

Mode ☐ CR-5000

**General** ☐ Align

Distances ☐

ALIGN\_HORIZONTAL\_DIST

Horizontal

**General** ☐ Align

Distances ☐

ALIGN\_VERTICAL\_DIST

Vertical

[e3Job.GetS](#)

• [e3Job.Set](#)

[e3Job.GetS](#)

• [e3Job.Set](#)



**General** ☐ Snap  
Size ☐ Snap size

**General** ☐  
**Display** ☐ Grid  
view ☐ Points (   
CheckBox )

MODE\_GRID\_OVERLAY

**General** ☐  
**Display** ☐ Grid  
view ☐ Points (   
SpinControl )

SCHEMAOVERSIZE

**General** ☐  
**Display** ☐ Grid  
view ☐ Rulers (   
CheckBox )

PANEL\_MODE\_GRID\_AXIS

SCHEMA\_MODE\_GRID\_AXIS

**General** ☐  
**Display** ☐ Grid  
view ☐ Rulers (   
SpinControl )

SCHEMAAXISGRID

**General** ☐  
**Display** ☐ Sheet  
Reference ☐ Show  
sheet layout

PANEL\_MODE\_GRID\_SHEETLAYOUT

SCHEMA\_MODE\_GRID\_SHEETLAYOUT

**General** ☐  
**Display** ☐ Sheet  
Reference ☐  
Format

SHEETREF\_FORMAT

**General** ☐  
**Display** ☐ Symbol  
Options ☐  
Alternative text  
as component  
code ( CheckBox  
)

ALT\_COMPCODE\_ON

**General** ☐  
**Display** ☐ Symbol

ALT\_COMPCODE

[e3Job.GetS](#)• [e3Job.Se](#)[e3Job.Enal](#)[e3Job.Disa](#)[e3Job.GetS](#)• [e3Job.Se](#)[e3Job.GetF](#)[e3Job.SetF](#)[e3Job.GetS](#)• [e3Job.Se](#)[e3Job.Enal](#)[e3Job.Disa](#)[e3Job.GetS](#)• [e3Job.Se](#)[e3Job.GetF](#)[e3Job.SetF](#)[e3Job.GetS](#)• [e3Job.Se](#)[e3Job.GetS](#)• [e3Job.Se](#)[e3Job.GetF](#)[e3Job.SetF](#)[e3Job.GetS](#)

Options ☐  
 Alternative text  
 as component  
 code ( ComboBox  
 )

**General** ☐

**Display** ☐ Symbol  
 Options ☐ Number  
 of view as an DUPLICATE\_DISPLAY\_OFF  
 extension to the  
 device's device  
 designation

**General** ☐

**Display** ☐ Symbol  
 Options ☐ Mark DISPLAY\_OPEN\_PINS  
 unconnected  
 nodes

**General** ☐

**Display** ☐ Symbol  
 Options ☐ Internal  
 / external / DISPLAY\_IE\_REPRESENTATION  
 jumper / seal  
 representation  
 for nodes

**General** ☐

**Display** ☐ Symbol  
 Options ☐ Add  
 internal device DEVICE\_DESIGNATION\_OF\_CONNECTION\_TARGET  
 designation to  
 connection target

**General** ☐

**Display** ☐ Symbol  
 Options ☐ Mark  
 connected PASS\_WIRE\_MARK  
 wires/conductors  
 at pass wire node  
 ( CheckBox )

**General** ☐

**Display** ☐ Symbol  
 Options ☐ Mark  
 connected PASS\_WIRE\_MARK\_SYMBOL  
 wires/conductors  
 at pass wire node  
 ( ComboBox )

**General** ☐

**Display** ☐ Mark  
 DISPLAY\_OPEN\_NODES

• [e3Job.Se](#)

[e3Job.GetL](#)

[e3Job.SetL](#)

[e3Job.GetS](#)

• [e3Job.Se](#)

[e3Job.GetL](#)

[e3Job.SetL](#)

[e3Job.GetS](#)

• [e3Job.Se](#)

[e3Job.GetL](#)

[e3Job.SetL](#)

[e3Job.GetS](#)

• [e3Job.Se](#)

[e3Job.GetL](#)

[e3Job.SetL](#)

[e3Job.GetS](#)

• [e3Job.Se](#)

[e3Job.GetS](#)

• [e3Job.Se](#)

[e3Job.GetS](#)

• [e3Job.Se](#)

[e3Job.GetS](#)

connect point in  
connection nets ☐  
T-Connections

**General** ☐

**Display** ☐ Mark  
connect point in ☐ DISPLAY\_OPEN\_NODES  
connection nets ☐  
Forced wiring

**General** ☐

**Display** ☐ Mark  
connect point in ☐ DISPLAY\_OPEN\_NODES  
connection nets ☐  
Open line end

**General** ☐

**Display** ☐  
Miscellaneous  
Options ☐ Show  
Tooltips

**General** ☐

**Display** ☐  
Miscellaneous  
Options ☐ Show  
Copilot

**General** ☐

**Display** ☐  
Miscellaneous  
Options ☐ Allow ☐ KEEP\_LINECOLOR  
window  
background color  
as display color

**General** ☐

**Display** ☐  
Miscellaneous  
Options ☐ Display ☐ DISPLAY\_STANDARD\_ROTATED\_TEXTS  
rotated texts acc.  
to standard

**General** ☐

**Display** ☐ FIT\_TEXT  
Miscellaneous  
Options ☐ Resize  
text to fit text box

• [e3Job.Set](#)

[e3Job.GetL](#)

[e3Job.SetL](#)

[e3Job.GetS](#)

• [e3Job.Set](#)

[e3Job.GetL](#)

[e3Job.SetL](#)

[e3Job.GetS](#)

• [e3Job.Set](#)

[e3Job.GetL](#)

[e3Job.SetL](#)

[e3Job.GetS](#)

• [e3Job.Set](#)

[e3Job.GetS](#)

• [e3Job.Set](#)

[e3Job.GetL](#)

[e3Job.SetL](#)

[e3Job.GetS](#)

• [e3Job.Set](#)

**General** □

**Display** □

Miscellaneous  
Options □ Enlarge ENLARGE\_GRID\_POINTS  
grid points when  
zooming

[e3Job.GetS](#)

• [e3Job.Se](#)

**General** □

**Display** □

Miscellaneous  
Options □ Display DISPLAY\_PREVIEW\_SYMBOL\_FOR\_SELECTED\_COMPONENT  
preview symbol  
for selected  
component

[e3Job.GetS](#)

• [e3Job.Se](#)

**General** □

**Display** □

Miscellaneous DISPLAY\_MINIMISE\_DETAILS  
Options □  
Minimize details

[e3Job.GetS](#)

• [e3Job.Se](#)

**General** □

**Display** □

Miscellaneous  
Options □ Invert  
display color

[e3Job.GetI](#)

[e3Job.SetI](#)

**General** □

**Display** □ Suffix

Modification □ SUFFIX\_MODIFICATION\_IS\_ACTIVE  
Suffix  
modification is  
active

[e3Job.GetS](#)

[e3Job.SetS](#)

[e3Job.Disa](#)

[e3Job.Enal](#)

**General** □ **Display** □ Suffix Modification □ Higher level  
assignmentSEPARATOR\_SUFFIX\_MODIFICATION\_ASSIGNMENT

[e3Job.GetSettingValue\(\)](#)

• [e3Job.SetSettingValue\(\)](#)

[e3Job.GetAssignmentSuffixSeparator\(\)](#)

[e3Job.SetAssignmentSuffixSeparator\(\)](#)

**General** □ **Display** □ Suffix Modification □  
LocationSEPARATOR\_SUFFIX\_MODIFICATION\_LOCATION

[e3Job.GetSettingValue\(\)](#)

• [e3Job.SetSettingValue\(\)](#)

e3Job.GetLocationSuffixSeparator()

e3Job.SetLocationSuffixSeparator()

**General** □ **Display** □ Suffix Modification □ Device designation  
SEPARATOR\_SUFFIX\_MODIFICATION\_DEVDES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetDeviceNameSuffixSeparator()

e3Job.SetDeviceNameSuffixSeparator()

**General** □ **Highlight** □ Search □ Highlight found objects when searching  
DO\_HIGHLIGHT

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**General** □ **Highlight** □ Jump □ Keep existing highlights when jumping  
KEEP\_HIGHLIGHT

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetHighlightKeep()

e3Job.SetHighlightKeep()

**General** □ **Highlight** □ Jump □ Zoom faction for 'Jump'  
(%)HIGHLIGHT\_JUMP\_ZOOM\_RATIO

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**General** □ **Highlight** □ Highlight □ Color  
HIGHLIGHT\_COLOR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetHighlightColour()

e3Job.SetHighlightColour()

**General** □ **Highlight** □ Highlight □ Width  
ACTUAL\_HIGHLIGHT\_WIDTH

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetHighlightLineWidth()

e3Job.SetHighlightLineWidth()

**General** □ **Highlight** □ Text Hyperlink □ Use following property ENABLE\_HYPERLINK\_DISPLAY

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**General** □ **Highlight** □ Text Hyperlink □ Underline hyperlinks □ when hovering HYPERLINK\_UNDERLINE\_MODE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**General** □ **Highlight** □ Text Hyperlink □ Underline hyperlinks □ always HYPERLINK\_UNDERLINE\_MODE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**General** □ **Highlight** □ Text Hyperlink □ Underline hyperlinks □ never HYPERLINK\_UNDERLINE\_MODE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**General** □ **Highlight** □ Text Hyperlink □ Color **General** □ **Verify** □ Level □ Release **General** □ **Verify** □ Level □ Development **General** □ **Verify** □ Level □ Draft **General** □ **Verify** □ Use Verification XML file VERIFY\_USE\_XML\_FILE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**General** □ **Language** □ 1st Language LANGUAGES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetLanguages()

e3Job.SetLanguages()

**General** □ **Language** □ 2nd Language LANGUAGES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetLanguages()

e3Job.SetLanguages()

**General** □ **Language** □ 3rd LanguageLANGUAGES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetLanguages()

e3Job.SetLanguages()

**General** □ **Language** □ 4th LanguageLANGUAGES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetLanguages()

e3Job.SetLanguages()

**General** □ **Language** □ 5th LanguageLANGUAGES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetLanguages()

e3Job.SetLanguages()

**General** □ **Language** □ PictogramsPICTOGRAM\_LANGUAGE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**General** □ **Language** □ Language Database **General** □ **Language** Table  
Schema **General** □ **Update in Project** □ Assignment □ Prefer matching symbols,  
conductors and pinsUIP\_ASSIGN\_GATE\_MODE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**General** □ **Update in Project** □ Assignment □ Prefer order of symbol, conductor and  
pinUIP\_ASSIGN\_GATE\_MODE

e3Job.GetSettingValue()

- [e3Job.SetSettingValue\(\)](#)

**General □ Update in Project** □ Attributes □ Overwrite attribute values for devices and symbolsRELOAD\_ATTRIBUTES

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

[e3Job.GetReloadAttributesOnUpdate\(\)](#)

[e3Job.SetReloadAttributesOnUpdate\(\)](#)

**General □ Update in Project** □ Attributes □ Delete unused attributes for devices and symbolsDELETE\_UNUSED\_ATTRIBUTES\_DURING\_UPDATE

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**General □ Update in Project** □ Signals □ Overwrite signals of block connectorsRELOAD\_SIGNALS

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**General □ Update in Project** □ Text Parameters □ Keep text visibility for □ Symbols

KEEP\_TEXT\_VISIBILITY

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

[e3Job.GetKeepSymbolTextVisibilityOnUpdate\(\)](#)

[e3Job.SetKeepSymbolTextVisibilityOnUpdate\(\)](#)

**General □ Update in Project** □ Text Parameters □ Keep text visibility for □ ModelsKEEP\_MODELTEXT\_VISIBILITY

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

[e3Job.GetKeepModelTextVisibilityOnUpdate\(\)](#)

[e3Job.SetKeepModelTextVisibilityOnUpdate\(\)](#)

**General □ Update in Project** □ Text Parameters □ Keep other text parameters for □ SymbolsKEEP\_TEXT\_PARAMETER

[e3Job.GetSettingValue\(\)](#)



- e3Job.SetSettingValue()

e3Job.GetKeepSymbolTextParametersOnUpdate()

e3Job.SetKeepSymbolTextParametersOnUpdate()

**General □ Update in Project □ Text Parameters □ Keep other text parameters for □**  
ModelsKEEP\_MODELTEXT\_PARAMETER

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetKeepModelTextParametersOnUpdate()

e3Job.SetKeepModelTextParametersOnUpdate()

**General □ Update in Project □ Pins □ Restore changed pin namesRESTORE\_PINNAMES**

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetRestoreChangedPinNamesOnUpdate()

e3Job.SetRestoreChangedPinNamesOnUpdate()

**General □ Update in Project □ Pins □ Restore changed physical pin**  
dataUPDATE\_RESTORE\_PHYSICAL\_PIN\_DATA

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**General □ Update in Project □ Pins □ Restore changed logical pin**  
dataUPDATE\_RESTORE\_LOGICAL\_PIN\_DATA

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**General □ Update in Project □ Pins □ Keep preview symbols of**  
devicesKEEP\_PINVIEW\_SYMBOLS\_OF\_DEVICE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**General □ Update in Project □ Connectors □ Keep Connector**  
symbolsKEEP\_CONNECTOR\_SYMBOLS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetKeepConnectorSymbolsOnUpdate()

e3Job.SetKeepConnectorSymbolsOnUpdate()

**General □ Update in Project □ Connectors □** Keep active mating connectors and cavity parts  
KEEP\_COUNTERPARTS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetKeepActiveCounterPartOnUpdate()

e3Job.SetKeepActiveCounterPartOnUpdate()

e3Job.GetKeepActiveConnectorPinTerminalOnUpdate()

e3Job.SetKeepActiveConnectorPinTerminalOnUpdate()

e3Job.GetKeepActiveFittingOnUpdate()

e3Job.SetKeepActiveFittingOnUpdate()

**General □ Update in Project □ Connectors □** Change already plugged mating connectors to the new active mating connector  
CHANGE\_COMPOSITES\_COMPONENTS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**General □ Update in Project □ Connectors □** Place all pins as single pins  
UPDATE\_PLACE\_SINGLE\_PINS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**General □ Update in Project □ Table Symbol for Terminal Plan □** Update table symbol even if the symbol used differs from the one defined in the database  
UPDATE\_TERMPLAN\_TABLE\_SYMBOL

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**General □ Update in Project □ Bundle Symbols □** Update bundle symbol even if the symbol used differs from the one defined in the database  
UPDATE\_BUNDLE\_SYMBOLS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**General □ Update in Project □ Sub-circuit □ Attributes □** Overwrite attribute values  
SUBCIRCUIT\_RELOAD\_ATTRIBUTES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**General □ Update in Project □ Sub-circuit □** Text Parameters □ Keep text visibility for □ SymbolsSUBCIRCUIT\_KEEP\_TEXT\_VISIBILITY

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**General □ Update in Project □ Sub-circuit □** Text Parameters □ Keep text visibility for □ ModelsSUBCIRCUIT\_KEEP\_MODELTEXT\_VISIBILITY

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**General □ Update in Project □ Sub-circuit □** Text Parameters □ Keep other text parameters for □ SymbolsSUBCIRCUIT\_KEEP\_TEXT\_PARAMETER

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**General □ Update in Project □ Sub-circuit □** Text Parameters □ Keep other text parameters for □ ModelsSUBCIRCUIT\_KEEP\_MODELTEXT\_PARAMETER

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**General □ Default Directories □ ( Table ) General □ Calculation □** Electrical calculation □ Activate CalculationGENERAL\_CALCULATION\_ACTIVE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**General □ Calculation □** Electrical calculation □ Alternative attribute for wire lengthCALC\_WIRE\_LENGTH

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**General □ Purge □** Objects □ Unused devicesUNUSED\_DEVICES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**General □ Purge □** Objects □ Unused devices from assembliesUNUSED\_DEVICES\_FROM\_ASSEMBLIES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**General** □ **Purge** □ Objects □ Unused terminals from terminal strips  
UNUSED\_TERMINALS\_FROM\_TERMINAL\_STRIPS

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**General** □ **Purge** □ Objects □ Unused plugged devices  
PURGE\_UNUSED\_PLUGGED\_DEVICES

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**General** □ **Purge** □ Objects □ Unused connected devices  
PURGE\_UNUSED\_CONNECTED\_DEVICES

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**General** □ **Purge** □ Objects □ Unused wires  
UNUSED\_WIRES

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**General** □ **Purge** □ Objects □ Unused cables  
UNUSED\_CABLES

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**General** □ **Purge** □ Objects □ Unused block devices  
UNUSED\_BLOCK\_DEVICES

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**General** □ **Purge** □ Objects □ Unused components  
UNUSED\_COMPONENTS

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**General** □ **Purge** □ Objects □ Unused cable types  
UNUSED\_CABLE\_TYPES

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**General** □ **Purge** □ Objects □ Unused symbol types  
UNUSED\_SYMBOL\_TYPES

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**General** □ **Purge** □ Objects □ Unused signals  
UNUSED\_SIGNALS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**General** □ **Purge** □ Objects □ Unused attribute namesUNUSED\_ATTRIBUTE\_NAMES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**General** □ **Purge** □ Objects □ Unused pin viewsUNPLACED\_PIN\_VIEWS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**General** □ **Purge** □ Objects □ All STEP modelsPURGE\_ALL\_STEP\_MODELS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**General** □ **Purge** □ Objects □ Unused GroupsUNUSED\_GROUPS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**General** □ **Purge** □ Objects □ No longer attribute values **General** □ **Purge** □ Project □ Purge unused objects from project before savingCLEANUP\_WHEN\_SAVING

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetPurgeUnusedBeforeSave()

e3Job.SetPurgeUnusedBeforeSave()

**General** □ **Zoom / Pan / Selection** □ Zoom In/Out □ Ratio (%)ZOOM\_FACTOR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**General** □ **Zoom / Pan / Selection** □ Pan □ Enable panning with arrow keysSCRL\_ENABLE\_ARROWKEYS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**General** □ **Zoom / Pan / Selection** □ Pan □ Ratio (%)SCRL\_FACTOR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**General** □ **Zoom / Pan / Selection** □ Area Selection □ Select all elements inside and intersecting or touching the border of the selection rectangleSELECTION\_BORDER

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**General** □ **Zoom** / **Pan** / **Selection** □ Area Selection □ Included Elements in Area Selection □ SymbolsSELECTION\_SYMBOL

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**General** □ **Zoom** / **Pan** / **Selection** □ Area Selection □ Included Elements in Area Selection □ TextsSELECTION\_TEXT

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**General** □ **Zoom** / **Pan** / **Selection** □ Area Selection □ Included Elements in Area Selection □ GraphicsSELECTION\_GRAPHIC

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**General** □ **Zoom** / **Pan** / **Selection** □ Area Selection □ Included Elements in Area Selection □ Connect LinesSELECTION\_NETSEG

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**General** □ **Zoom** / **Pan** / **Selection** □ Area Selection □ Included Elements in Area Selection □ Net NodesSELECTION\_NETNODE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**General** □ **Zoom** / **Pan** / **Selection** □ Area Selection □ Included Elements in Area Selection □ Attribute Text TemplatesSELECTION\_ATTRIBUTE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**General** □ **Zoom** / **Pan** / **Selection** □ Area Selection □ Included Elements in Area Selection □ DimensionsSELECTION\_DIMENSION

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**General** □ **Locking** □ Password for unlocking objects □ Old password **General** □ **Locking** □ Password for unlocking objects □ New password **General** □ **Locking** □ Password for unlocking objects □ Confirm password **General** □ **Locking** □ Display □ Mark locked objectsMARK\_LOCKED\_OBJECTS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**General** □ **Component Type Attributes** □ Selected  
AttributesCOMPONENT\_TYPE\_ATTRIBUTES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ Autoconnect □ Allow inserting symbol in  
connectionALLOW\_INSERT\_SYMBOL\_IN\_CONNECTION

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ Autoconnect □ Keep signal for all  
connectionsRETAIN\_SIGNAL\_CONNECT\_CELL

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ Autoconnect □ Reconnect after deleting symbolsAUTOCON\_LINES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ Autoconnect □ Preferred Direction □ Vertical connections (top to  
bottom)AUTOCON\_DIR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ Autoconnect □ Preferred Direction □ Horizontal connections (left to  
right)AUTOCON\_DIR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ Net □ Allow net loopsNETLOOPS\_ALLOWED

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ Net □ Specific functionality of inheriting net  
numberINHERIT\_NET\_NUMBER

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ Conductors / Wires □ Unconnect, if conductor is not routed in a connect  
lineUNCONNECT\_CORES

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Connection** □ Conductors / Wires □ Unconnect, if conductor is not routed in a connect line □ Check view and original connections UNCONNECT\_CORES

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Connection** □ Conductors / Wires □ Unconnect, if conductor is not routed in a connect line □ Check only original connections UNCONNECT\_CORES

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Connection** □ Conductors / Wires □ Connect conductors/wires at one end to source cross-reference if cross-reference is unconnected KEEP\_CORE\_AFTER\_UNCONNECT\_REFERENCE

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Connection** □ Conductors / Wires □ Lock new wire pathways LOCK\_WIRES

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Connection** □ Conductors / Wires □ Clear signal after unconnect conductor/ wire at unconnected Pin CLEAR\_SIGNAL\_AT\_PIN\_AFTER\_UNCONNECT\_CORE

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Connection** □ Conductors / Wires □ Assign all conductors within a wire route to a new cable when inserting an inline connector ASSIGN\_ALL\_CORES\_TO\_NEW\_CABLE

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Connection** □ Template Symbol □ Horizontal WCOUNT\_SYMBOL\_HOR

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Connection** □ Template Symbol □ Vertical WCOUNT\_SYMBOL\_VER

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Connection** □ Alternative Template Symbol □ Horizontal WCOUNT\_SYMBOL\_HOR\_ALT

e3Job.GetSettingValue()



- e3Job.SetSettingValue()

**Connection** □ Alternative Template Symbol □ VerticalWCOUNT\_SYMBOL\_VER\_ALT

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ Connect and Wire □ connect only  
graphicallyCONNECT\_AND\_USE\_DEFAULT\_WIRE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetConnectionMode()

e3Job.SetConnectionMode()

**Connection** □ Connect and Wire □ connect and use default  
wireCONNECT\_AND\_USE\_DEFAULT\_WIRE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetConnectionMode()

e3Job.SetConnectionMode()

**Connection** □ Pins □ Keep plug after unplugging  
pinsKEEP\_PLUG\_AFTER\_UNPLUGGING\_PINS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ Pins □ Deny plugging pins of same  
deviceDENY\_PLUG\_PINS\_OF\_SAME\_DEVICE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ Pins □ Allow only valid mating connection to  
plugALLOW\_ONLY\_VALID\_MATING\_CONNECTORS\_TO\_PLUG

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ Pins □ Allow only compatible pin genders to  
plugALLOW\_ONLY\_COMPATIBLE\_PIN\_GENDERS\_TO\_PLUG

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ Pins □ Deny plugging pin with different pin names  
DENY\_PLUG\_PINS\_WITH\_DIFF\_PINNAMES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Connect Lines** □ Lines □ Width  
LINDIA

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetLineWidth()

e3Job.GetLineWidthEx()

e3Job.SetLineWidth()

e3Job.GetBusLineWidth()

e3Job.GetBusLineWidthEx()

e3Job.SetBusLineWidth()

**Connection** □ **Connect Lines** □ Lines □ Style  
LINMOD

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetLineStyle()

e3Job.SetLineStyle()

e3Job.GetBusLineStyle()

e3Job.SetBusLineStyle()

**Connection** □ **Connect Lines** □ Lines □ Color  
LINCOD

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetLineColour()

e3Job.SetLineColour()

e3Job.GetBusLineColour()

e3Job.SetBusLineColour()

**Connection** □ **Connect Lines** □ Lines □ LevelLINLEV

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetLineLevel()

e3Job.SetLineLevel()

e3Job.GetBusLineLevel()

e3Job.SetBusLineLevel()

**Connection** □ **Connect Lines** □ Lines □ Use properties of starting line  
USE\_LINE\_PROPERTIES\_OF\_START\_LINE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Connect Lines** □ Busbars □ WidthBUSBARLINDIA

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Connect Lines** □ Busbars □ StyleBUSBARLINMOD

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Connect Lines** □ Busbars □ ColorBUSBARLINCOD

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Connect Lines** □ Busbars □ LevelBUSBARLINLEV

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Connect Lines** □ Busbars □ Use properties of starting busbar  
USE\_BUSBAR\_PROPERTIES\_OF\_START\_BUSBAR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Connect Lines** □ Ignore for Cabling Table □ Display ignored connect line  
in alternative color ( CheckBox )DISPLAY\_IGNORED\_CONNECT\_LINE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Connect Lines** □ Ignore for Cabling Table □ Display ignored connect line in alternative color ( ComboBox ) COLOUR\_IGNORED\_CONNECT\_LINE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Connect Lines** □ Ignore for Cabling Table □ Ignored new connections in Cabling Table IGNORE\_NEW\_CONNECT\_LINE\_IN\_CABLING\_TABLE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Connect Lines** □ Angled Connections □ Connection angle LIN\_FOLD\_ANGLE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetConnectionInclinationAngle()

e3Job.SetConnectionInclinationAngle()

**Connection** □ **Connect Lines** □ Angled Connections □ Offset distance LIN\_FOLD\_DISTANCE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetConnectionInclinationDistance()

e3Job.SetConnectionInclinationDistance()

**Connection** □ **Connect Lines** □ Delete with graphical representation □ Signal DELETE\_SIGNAL\_ON\_DEL\_CLINE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetDeleteSignalOnDelCline()

e3Job.SetDeleteSignalOnDelCline()

**Connection** □ **Connect Lines** □ Delete with graphical representation □ Conductors / Wires UNCONNECT\_CORES\_ON\_DEL\_CLINE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetUnconnectCoresOnDelCline()

e3Job.SetUnconnectCoresOnDelCline()

**Connection** □ **Connect Lines** □ Delete with graphical representation □ Conductors / Wires from projectDELETE\_CORES\_ON\_DEL\_CLINE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetDeleteCoresOnDelCline()

e3Job.SetDeleteCoresOnDelCline()

**Connection** □ **Connect Lines** □ Template Symbol □ HorizontalCONNECT\_LINE\_SYMBOL\_HOR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Connect Lines** □ Template Symbol □ VerticalCONNECT\_LINE\_SYMBOL\_VER

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **References Style** □ FontREFFONTPTR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **References Style** □ Font StyleREFDIA

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **References Style** □ SizeREFSIZ

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **References Style** □ ColorREFCOD

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **References Style** □ Display Control □ X-OffsetREFOFX

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **References Style** □ Display Control □ Y-OffsetREFOFY

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **References Style** □ Display Control □ GapREFGAP

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **References Style** □ Ratio □ NormalREFMOD

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **References Style** □ Ratio □ NarrowREFMOD

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **References Style** □ Ratio □ WideREFMOD

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **References Style** □ Direction □ SymmetricalREFDIR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **References Style** □ Direction □ UpREFDIR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **References Style** □ Direction □ DownREFDIR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **References Style** □ Function Control □ Allow change reference typeREFCHANGETYPE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **References Format** □ PrefixSHEETREFSETTING

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **References Format** □ SuffixSHEETREFSETTING

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **References Format** □ Sheet textSHEETREFSETTING

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **References Format** □ Reference textSHEETREFSETTING

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **References Format** □ Display references with logically plugged devicesREFERENCE\_BETWEEN\_PLUGGED\_PINS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Signal Logic Lines** □ Display open signal connections ( CheckBox )

AIRLINE\_DISPLAY\_OPEN\_SIGNAL\_CONNECTIONS

AIRLINES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Signal Logic Lines** □ Display open signal connections ( ComboBox )AIRLINES\_LINMOD\_SIG

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Signal Logic Lines** □ Display signal flagsDISPLAY\_SIGNAL\_FLAGS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Signal Logic Lines** □ Mark connect nodes defined as not connected (\*\*NC\*\*)NOCONN\_DISPLAY\_OFF

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Connection** □ **Signal Logic Lines** □ Display negative logic signals with barred line  
lineDISPLAY\_BARRING\_LINES

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Connection** □ **Cable/Conductor Logic Lines** □ Display □ Display unconnected conductors

AIRLINE\_DISPLAY\_UNCONNECTED\_CORES

AIRLINES

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Connection** □ **Cable/Conductor Logic Lines** □ Display □ Display laid conductors also

AIRLINE\_DISPLAY\_LAID\_CORES\_ALSO

AIRLINES

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Connection** □ **Cable/Conductor Logic Lines** □ Lines □ Line style  
styleAIRLINES\_LINMOD\_CAB

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Connection** □ **Cable/Conductor Logic Lines** □ Lines □ Display conductor logic lines as arcs  
arcsAIRLINES

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Connection** □ **Cable/Conductor Logic Lines** □ Lines □ Mark direction

AIRLINE\_MARK\_DIRECTION

AIRLINES

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Connection** □ **Cable/Conductor Logic Lines** □ Lines □ Only for Views □ Used views only

AIRLINE\_USED\_VIEWS\_ONLY



AIRLINES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Cable/Conductor Logic Lines** □ Lines □ Only for Views □ (Table)

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Cable/Conductor Logic Lines** □ Lines □ Show name of conductor

AIRLINE\_SHOW\_NAME\_OF\_CORE

AIRLINES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Cable/Conductor Logic Lines** □ Lines □ Show name of conductor □  
Position □ to endpoints

AIRLINE\_SHOW\_NAME\_OF\_CORE\_POSITION\_CENTERED

AIRLINES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Cable/Conductor Logic Lines** □ Lines □ Show name of conductor □  
Position □ centered

AIRLINE\_SHOW\_NAME\_OF\_CORE\_POSITION\_CENTERED

AIRLINES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Cable/Conductor Logic Lines** □ Lines □ Show name of conductor □  
X-OffsetAIRLINES\_OFFFX\_CABCAB

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Cable/Conductor Logic Lines** □ Lines □ Show name of conductor □  
Y-OffsetAIRLINES\_OFFFY\_CABCAB

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Cable/Conductor Logic Lines** □ Lines □ Show name of conductor □ **Font** □ FontAIRLINES\_TXTFNT\_CABCAB

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Cable/Conductor Logic Lines** □ Lines □ Show name of conductor □ **Font** □ Font Style

AIRLINE\_SHOW\_NAME\_OF\_CORE\_FONT\_BOLD

AIRLINE\_SHOW\_NAME\_OF\_CORE\_FONT\_ITALIC

AIRLINES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Cable/Conductor Logic Lines** □ Lines □ Show name of conductor □ **Font** □ SizeAIRLINE\_TXTSIZ\_CABCAB

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Cable/Conductor Logic Lines** □ Lines □ Show name of conductor □ **Font** □ Ratio □ NormalAIRLINES\_TXTMOD\_CABCAB

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Cable/Conductor Logic Lines** □ Lines □ Show name of conductor □ **Font** □ Ratio □ NarrowAIRLINES\_TXTMOD\_CABCAB

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Cable/Conductor Logic Lines** □ Lines □ Show name of conductor □ **Font** □ Ratio □ WideAIRLINES\_TXTMOD\_CABCAB

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Cable/Conductor Logic Lines** □ Lines □ Show signal name

AIRLINE\_SHOW\_SIGNAL\_NAME

AIRLINES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Cable/Conductor Logic Lines** □ Lines □ Show signal name □ Position □ to endpoints

AIRLINE\_SHOW\_SIGNAL\_NAME\_POSITION\_CENTERED

AIRLINES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Cable/Conductor Logic Lines** □ Lines □ Show signal name □ Position □ centered

AIRLINE\_SHOW\_SIGNAL\_NAME\_POSITION\_CENTERED

AIRLINES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Cable/Conductor Logic Lines** □ Lines □ Show signal name □ X-OffsetAIRLINES\_OFFX\_CABSIG

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Cable/Conductor Logic Lines** □ Lines □ Show signal name □ Y-OffsetAIRLINES\_OFFY\_CABSIG

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Cable/Conductor Logic Lines** □ Lines □ Show signal name □ **Font** □ Font

AIRLINES\_TXTFNT\_CABSIG

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Cable/Conductor Logic Lines** □ Lines □ Show signal name □ **Font** □ Font Style

AIRLINE\_SHOW\_SIGNAL\_NAME\_FONT\_BOLD

AIRLINE\_SHOW\_SIGNAL\_NAME\_FONT\_ITALIC

AIRLINES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Cable/Conductor Logic Lines** □ Lines □ Show signal name □ **Font** □ Size **Connection** □ **Cable/Conductor Logic Lines** □ Lines □ Show signal name □ **Font** □ Ratio □ Normal **Connection** □ **Cable/Conductor Logic Lines** □ Lines □ Show signal name □ **Font** □ Ratio □ Narrow **Connection** □ **Cable/Conductor Logic Lines** □ Lines □ Show signal name □ **Font** □ Ratio □ Wide **Connection** □ **Cable/Conductor Logic Lines** □ Conductor Logic Lines of Signal □ Keep existing conductor logic lines of signalKEEP\_EXISTING\_CORE\_LOGIC\_LINES\_OF\_SIGNAL

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Conductors / Wires** □ Conductor Assignment Procedure □ Use signal equivalence (to select connect pin)PANEL\_CONNECT\_IGNORE\_EQUIVALENT\_PINS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Conductors / Wires** □ Conductor Assignment Procedure □ Use name equivalence additionallyROUTING\_NAME\_EQUIVALENCE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Conductors / Wires** □ Conductor Assignment Procedure □ Ignore minimum cross-section check for multiconductor pinsUSE\_MINIMAL\_CROSECTION\_AS\_GLOBAL\_MINIMUM

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Conductors / Wires** □ Conductor Assignment Procedure □ Show information for option combinationsSHOW\_CORE\_INFO\_OPTION\_COMBINATIONS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Conductors / Wires** □ Conductor Assignment Procedure □ Use physical data of cavity part models instead of device modelFITTING\_DATA\_DOMINATE\_MODELPIN

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetConnectorPinTerminalParameterOverwriteModelPin()

e3Job.SetConnectorPinTerminalParameterOverwriteModelPin()

**Connection** □ **Conductors / Wires** □ Conductor Assignment Procedure □ Use physical data of conductor to select connector pin terminal and wire  
sealUSE\_CORE\_TO\_SELECT\_FITTING

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Connection** □ **Conductors / Wires** □ Conductor Assignment Procedure □ Additional filter attribute for connector pin terminalsCONNECTOR\_PIN\_TERMINAL\_FILTER\_ATTRIBUTE

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Connection** □ **Conductors / Wires** □ Conductor Assignment Procedure □ Additional filter attribute for connector pin terminals □ Default value for filter attributeCONNECTOR\_PIN\_TERMINAL\_FILTER\_ATTRIBUTE\_VALUE

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Connection** □ **Conductors / Wires** □ Conductor Assignment Procedure □ Additional filter attribute for wire sealsWIRE\_SEAL\_ATTRIBUTE

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Connection** □ **Conductors / Wires** □ Conductor Assignment Procedure □ Additional filter attribute for wire seals □ Default value for filter attributeWIRE\_SEAL\_ATTRIBUTE\_VALUE

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Connection** □ **Conductors / Wires** □ Used Wire Type □ Wire group

e3Job.GetDefaultWire()

e3Job.SetDefaultWire()

**Connection** □ **Conductors / Wires** □ Used Wire Type □ Wire

e3Job.GetDefaultWire()

e3Job.SetDefaultWire()

**Connection** □ **Conductors / Wires** □ Used Wire Type □ Color **Connection** □ **Conductors / Wires** □ Used Wire Type □ Cross-section **Connection** □ **Conductors / Wires** □ Used Jumper Type □ Jumper group

e3Job.GetDefaultJumper()

e3Job.SetDefaultJumper()

**Connection** □ **Conductors / Wires** □ Used Jumper Type □ Jumper

e3Job.GetDefaultJumper()

e3Job.SetDefaultJumper()

**Connection** □ **Conductors / Wires** □ Attribute for Schematics Routing □ Use attribute for checking ( CheckBox ) WIRE\_AUTOROUTE\_WITH\_ATTRIBUTES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Conductors / Wires** □ Attribute for Schematics Routing □ Use attribute for checking ( ComboBox ) **Connection** □ **Conductors / Wires** □ Attribute for Schematics Routing □ Use attribute for checking □ Reroute according to new setting WIRE\_AUTOROUTE\_ATTRIBUTES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Conductors / Wires** □ Move Conductor Ends MOVE\_CORE\_ENDS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Conductors / Wires** □ Harness Name HARNESS\_NAME\_ATTRIBUTE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Conductors / Wires** □ **Naming** □ Conductor / Wire Names □ Create unique names CONNECT\_UNIQUE\_CORE\_NUMBER

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Conductors / Wires** □ **Naming** □ Conductor / Wire Names □ Wire Numbers □ Define range WIRE\_NAMING\_RANGE\_ON

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Conductors / Wires** □ **Naming** □ Conductor / Wire Names □ Wire Numbers □ Define range □ from

e3Job.GetWireRange()

e3Job.SetWireRange()

**Connection** □ **Conductors / Wires** □ **Naming** □ Conductor / Wire Names □ Wire Numbers  
□ Define range □ to

[e3Job.GetWireRange\(\)](#)

[e3Job.SetWireRange\(\)](#)

**Connection** □ **Conductors / Wires** □ **Naming** □ Conductor / Wire Names □ Wire Numbers  
□ Format (A<nnn>B)WIRE\_NUMBER\_FORMAT

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Connection** □ **Conductors / Wires** □ **Naming** □ Conductor / Wire Names □ Wire Names □  
Assign wire names automaticallyAUTOMATIC\_GENERATED\_WIRE\_NAMES

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Connection** □ **Conductors / Wires** □ **Naming** □ Conductor / Wire Names □ Wire Names □  
(Table)

[e3Job.GetGeneratedWireNameFormatEx\(\)](#)

[e3Job.SetGeneratedWireNameFormatEx\(\)](#)

**Connection** □ **Conductors / Wires** □ **Calculation** □ Conductor / Wire Calculation □  
Twisted pair □ Warning messages for missing  
attributesTWISTED\_PAIR\_LENGTH\_CALCULATION

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Connection** □ **Conductors / Wires** □ **Calculation** □ Conductor / Wire Calculation □  
Segment diameter □ Ignore unplaced conductors / wires of dynamic  
cablesIGNORE\_UNPLACED\_CORES\_OF\_DYN\_CABLE\_FOR\_NETSEG\_DIAMETER

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Connection** □ **Conductors / Wires** □ **Calculation** □ Conductor / Wire Calculation □  
Segment diameter □ ignore bundle of dynamic  
cablesIGNORE\_BUNDLE\_OF\_DYN\_CABLE\_FOR\_NETSEG\_DIAMETER

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Connection** □ **Conductors / Wires** □ **Calculation** □ Conductor / Wire Calculation □  
Segment diameter □ Use conductors/wires from dynamic cable for cable duct  
fillUSE\_CORES\_WIRES\_OF\_DYN\_CABLE\_FOR\_CABLEDUCTFILL

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Signals** □ Signals on Connections □ Create connections which transfer signalsCONNECT\_SIGNAL

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Signals** □ Signals on Connections □ Transfer signals on connections between viewsCONNECT\_SIGNAL\_VIEWS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Signals** □ Signals on Symbols □ Clear signals when deleting symbolsCLEAR\_SIG\_ON\_DEL\_SYMBOL

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Signals** □ Signals on Symbols □ Even if not connected view symbols are placedCLEAR\_SIG\_ON\_PLACED\_VIEW\_SYMBOL

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Signals** □ Signal Flow on Connectors □ Interrupt signal flow on block connectorsCREATE\_BLCON\_WITH\_FLOW

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Signals** □ Signal Flow on Connectors □ Interrupt signal flow on normal connectorsCREATE\_CONN\_WITH\_FLOW

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Signals** □ Signal Settings for Copy and Import □ Keep user-defined signalsMERGE\_COPY\_AREA\_KEEP\_USER

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Signals** □ Signal Settings for Copy and Import □ Keep system-generated signalsMERGE\_COPY\_AREA\_KEEP\_SYSTEM

e3Job.GetSettingValue()

- e3Job.SetSettingValue()



**Connection** □ **Signals** □ Signal Classes □ Allow signal changes to signals not belonging to same class `ALLOW_SIGNAL_CHANGES_OF_DIFFERENT_CLASSES`

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Connection** □ **Signals** □ Signal Format □ Recalculate signal names according to format specification `RECALC_FORMATTED_SIGNALS`

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Connection** □ **Connectors** □ Pin Names □ Inherit pin names when connecting `INHERIT_PINNAMES`

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Connection** □ **Connectors** □ Mating Connectors □ Use higher level assignment and location of placed devices `USE_ASSIGNMENT_OF_CONN`

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Connection** □ **Connectors** □ Mating Connectors □ Try to assign pins via names first `ASSIGN_PINS_VIA_NAMES`

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Connection** □ **Connectors** □ Mating Connectors □ Ignore pin attribute 'Internal Device Designation' when assigning pins `IGNORE_DOT_CONN`

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Connection** □ **Connectors** □ Mating Connectors □ Generate device designation of mating connector from device designation and pin attribute 'Internal Device Designation' `USE_DOT_CONN_NAME`

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Connection** □ **Connectors** □ Mating Connectors □ Generate device designation of mating connector from device designation of block and device `USE_BLOCK_NAME_FOR_DEVDDES`

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

e3Job.GetUseBlockDesignation()

e3Job.SetUseBlockDesignation()

**Connection** □ **Connectors** □ Mating Connectors □ Separator to use  
USE\_DOT\_CONN\_NAME\_SEPERATOR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Connectors** □ Mating Connectors □ Use same numeric part for connector and mating connector  
USE\_SAME\_NUMERIC\_PART\_FOR\_MATING\_CONNECTOR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Connectors** □ Mating Connectors □ Use automatic connector naming  
USE\_ANSI\_STANDARD\_FOR\_MATING\_CONNECTOR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Connectors** □ Mating Connectors □ Default designation for plugs  
DEFAULT\_DESIGNATION\_FOR\_PLUGS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Connectors** □ Mating Connectors □ Default designation for jacks  
DEFAULT\_DESIGNATION\_FOR\_JACKS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Connection** □ **Connection Target Format** □ Text Type

e3.Job.GetConnectionTargetFormat()

e3.Job.SetConnectionTargetFormat()

**Connection** □ **Connection Target Format** □ Prefix

e3.Job.GetConnectionTargetFormat()

e3.Job.SetConnectionTargetFormat()

**Connection** □ **Connection Target Format** □ Suffix

e3.Job.GetConnectionTargetFormat()

e3.Job.SetConnectionTargetFormat()

**Connection** □ **Connection Target Format** □ Use plugged device as target

[e3.Job.GetConnectionTargetFormat\(\)](#)

[e3.Job.SetConnectionTargetFormat\(\)](#)

**Connection** □ **Connection Target Format** □ Multiline

[e3.Job.GetConnectionTargetFormat\(\)](#)

[e3.Job.SetConnectionTargetFormat\(\)](#)

**Connection** □ **Connection Target Format** □ Number of view as an extension to the device's device designation

[e3.Job.GetConnectionTargetFormat\(\)](#)

[e3.Job.SetConnectionTargetFormat\(\)](#)

**Connection** □ **Connection Target Format** □ (Table)

[e3.Job.GetConnectionTargetFormat\(\)](#)

[e3.Job.SetConnectionTargetFormat\(\)](#)

**Placement** □ Default Designations □ Higher level assignment ( Left EditBox )SEPARATOR\_ASSIGNMENT

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

[e3Job.GetAssignmentSeparator\(\)](#)

[e3Job.SetAssignmentSeparator\(\)](#)

**Placement** □ Default Designations □ Higher level assignment ( Right EditBox )HLA\_DEFAULT

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Placement** □ Default Designations □ Higher level assignment □ Used for unique designationUSE\_HLA\_FOR\_BMK

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Placement** □ Default Designations □ Location ( Left EditBox )SEPARATOR\_LOCATION

[e3Job.GetSettingValue\(\)](#)

- e3Job.SetSettingValue()

e3Job.GetLocationSeparator()

e3Job.SetLocationSeparator()

**Placement** □ Default Designations □ Location ( Right EditBox )LOC\_DEFAULT

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ Default Designations □ Location □ Used for unique designationUSE\_LOC\_FOR\_BMK

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ Default Designations □ Device designation ( Left EditBox )SEPARATOR\_DEVDES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetDeviceNameSeparator()

e3Job.SetDeviceNameSeparator()

**Placement** □ Default Designations □ Device designation ( Right EditBox ) **Placement** □ Default Designations □ Device designation □ Used for unique designationUSE\_DEVDES\_FOR\_BMK

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ Default Designations □ BlocksBLOCK\_DES\_DEFAULT

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ Default Designations □ CablesCABLE\_DES\_DEFAULT

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ Default Designations □ Text order according to standardKEEP\_BMK\_DIN\_ORDER

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ Rules □ Shorten higher level assignment and location as against sheet/fieldSHORT\_HLA\_AND\_LOC

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ Rules □ Use higher level assignment and location of sheet/fieldUSE\_ASSIGNMENT\_OF\_SHEET

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ Rules □ Rename devices when changing the designations for sheet/fieldCHANGE\_DEVICES\_ON\_SHEET

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ Rules □ Use symbol pin attributes when assigning to devicePLACE\_USE\_PIN\_ATTR

e3Job.GetSettingValue()

e3Job.SetSettingValue()

e3Job.GetUsePinAttributesOnAssign()

e3Job.SetUsePinAttributesOnAssign()

**Placement** □ Rules □ Delete pin attributes when symbol is unplacedPLACE\_DEL\_PIN\_ATTR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetDeletePinAttributesOnUnplace()

e3Job.SetDeletePinAttributesOnUnplace()

**Placement** □ Rules □ Delete symbol attributes when symbol is unplacedPLACE\_DEL\_SYMBOL\_ATTR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ Block Devices Options □ Use name of block for devices asBLOCKNAME\_TO\_CONNECTOR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ Block Devices Options □ Use name of block for devices as □ Higher level assignmentBLOCKNAME\_TO\_CONNECTOR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ Block Devices Options □ Use name of block for devices as □ LocationBLOCKNAME\_TO\_CONNECTOR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ Block Devices Options □ Allow same device designation on different blocksALLOW\_SAME\_CONDES\_ON\_BLOCKS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ Block Devices Options □ Use name of block for devices as **Placement** □ IEC 81346 standard □ IEC 81346 is activeIEC\_81346\_IS\_ACTIVE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetIEC61346Setting()

e3Job.SetIEC61346Setting()

**Placement** □ IEC 81346 standard □ Separators □ AssignmentSEPARATOR\_IEC\_81346\_ASSIGNMENT

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ IEC 81346 standard □ Separators □ LocationSEPARATOR\_IEC\_81346\_LOCATION

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ IEC 81346 standard □ Separators □ Device designationSEPARATOR\_IEC\_81346\_DEVDES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ IEC 81346 standard □ Separators □ AttributesSEPARATOR\_IEC\_81346\_ATTRIBUTES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ IEC 81346 standard □ Separators □ Top-level  
prefixUNCUT\_PREFIX\_IEC\_81346

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Change Component** □ Assignment □ Prefer name of  
pinsASSIGN\_GATE\_MODE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Change Component** □ Assignment □ Prefer internal device  
designationASSIGN\_GATE\_MODE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Change Component** □ Assignment □ Prefer order of symbol, conductor and  
pinASSIGN\_GATE\_MODE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Change Component** □ Assignment □ Prefer symbol name, pin name and  
signalASSIGN\_GATE\_MODE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Change Component** □ Attributes □ Overwrite attribute values for devices  
and symbolsCC\_RELOAD\_ATTRIBUTES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Change Component** □ Attributes □ Delete unused attributes for devices  
and symbolsDELETE\_UNUSED\_ATTRIBUTES\_DURING\_CHANGE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Change Component** □ Pins □ Restore changing pin  
namesCC\_RESTORE\_PINNAMES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** ☐ **Change Component** ☐ Connectors ☐ Keep active mating connectors and cavity parts  
CC\_KEEP\_COUNTERPARTS

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Placement** ☐ **Change Component** ☐ Connectors ☐ Keep attribute for pin and block pin  
symbolKEEP\_ATTRIBUTE\_PIN\_BLOCKPIN\_SYMBOL

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Placement** ☐ **Change Component** ☐ Connectors ☐ Place all pins as single pins  
CC\_PLACE\_SINGLE\_PINS

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Placement** ☐ **Change Component** ☐ Table Symbol for Terminal Plan ☐ Update table symbol even if the symbol used differs from the one defined in the database  
CC\_TERMPLAN\_TABLE\_SYMBOL

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Placement** ☐ **Change Component** ☐ Bundle Symbols ☐ Update bundle symbol even if the symbol used differs from the one defined in the database  
CC\_BUNDLE\_SYMBOLS

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Placement** ☐ **Change Component** ☐ Assembly devices ☐ Prefer component code  
ASSIGN\_ASSEMBLY\_DEVICE\_MODE

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Placement** ☐ **Change Component** ☐ Assembly devices ☐ Prefer order of devices in assembly  
ASSIGN\_ASSEMBLY\_DEVICE\_MODE

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Placement** ☐ **Terminal** ☐ Way of Numbering  
☐ ElementWAY\_OF\_TERMINAL\_PINNUMBERING

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Placement** ☐ **Terminal** ☐ Way of Numbering  
InternalWAY\_OF\_TERMINAL\_PINNUMBERING



e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Terminal** □ Terminal strip pin numbering □ Unique pin names  
UNIQUE\_TERMINAL\_STRIP\_PINNAMES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Terminal** □ Terminal plan □ Online update  
TERMPLAN\_ONLINE\_UPDATE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Terminal** □ Terminal port □ Separator for pin and port name  
SEPARATE\_PIN\_PORT

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetPortNameSeparator()

e3Job.SetPortNameSeparator()

**Placement** □ **Terminal** □ Terminal strip sort □ Sort file name.  
SORT\_FORMAT\_FILE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Terminal Plan** □ Plan □ Sheet format  
TERMPLAN\_BORDER

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Terminal Plan** □ Plan □ Table symbol  
TERMPLAN\_ROW\_SYMBOL

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Terminal Plan** □ Jumper □ Jumpers by Connections **Placement** □ **Terminal Plan** □ Jumper □ in line **Placement** □ **Terminal Plan** □ Jumper □ Jumpers by Attributes **Placement** □ **Terminal Plan** □ Jumper □ No Jumpers **Placement** □ **Terminal Plan** □ Options □ Autocompress  
TERMPLAN\_AUTOCOMPRESS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Terminal Plan** □ Options □ Combine same pin names  
TERMPLAN\_COMBINE\_SAME\_PINNAMES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Terminal Plan** □ Options □ Wire in planTERMPLAN\_OUTPUT\_WIRES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Terminal Plan** □ Options □ Only user-defined signalsTERMPLAN\_CONSIDER\_SIGNAL\_EQUIVALENCE\_ONLY\_WITHIN\_A\_SYMBOL

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Terminal Plan** □ Options □ Pin view connectionsTERMPLAN\_PINVIEW\_CONNECTIONS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Terminal Plan** □ Options □ Consider signal equivalence only within a symbol **Placement** □ **Terminal Plan** □ Options □ Show all equivalent pinsTERMPLAN\_SHOW\_ALL\_EQUIVALENT\_PINS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Terminal Plan** □ Internal / External Definition □ Higher level assignment/LocationTERMPLAN\_INTERNAL\_EXTERNAL\_DEFINITION

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Terminal Plan** □ Internal / External Definition □ Higher level assignmentTERMPLAN\_INTERNAL\_EXTERNAL\_DEFINITION

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Terminal Plan** □ Internal / External Definition □ LocationTERMPLAN\_INTERNAL\_EXTERNAL\_DEFINITION

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Terminal Table** □ Jumper □ Jumpers by ConnectionsTERMPLAN\_JUMPER\_ORDER

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Terminal Table** □ Jumper □ in lineTERMPLAN\_JUMPER\_INLINE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Terminal Table** □ Jumper □ Jumpers by AttributesTERMPLAN\_JUMPER\_ORDER

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Terminal Table** □ Jumper □ No JumpersTERMPLAN\_JUMPER\_ORDER

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Terminal Table** □ Options □ AutocompressTERMTABLE\_AUTOCOMPRESS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Terminal Table** □ Options □ Unique connectionsTERMPLAN\_UNIQUE\_CONNECTIONS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Terminal Table** □ Options □ Wires in planTERMTABLE\_OUTPUT\_WIRES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Terminal Table** □ Options □ Only user-defined signalsTERMTABLE\_IGNORE\_SYSTEM\_SIGNALS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Terminal Table** □ Options □ Pin view connectionsTERMTABLE\_PINVIEW\_CONNECTIONS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Terminal Table** □ Options □ Consider signal equivalence only within a symbolTERMTABLE\_CONSIDER\_SIGNAL\_EQUIVALENCE\_ONLY\_WITHIN\_A\_SYMBOL

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Terminal Table** □ Options □ Show all equivalent pins  
TERMTABLE\_SHOW\_ALL\_EQUIVALENT\_PINS

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Placement** □ **Terminal Table** □ Internal / External Definition □ Higher level assignment/Location  
TERMTABLE\_INTERNAL\_EXTERNAL\_DEFINITION

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Placement** □ **Terminal Table** □ Internal / External Definition □ Higher level assignment  
TERMTABLE\_INTERNAL\_EXTERNAL\_DEFINITION

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Placement** □ **Terminal Table** □ Internal / External Definition □ Location  
TERMTABLE\_INTERNAL\_EXTERNAL\_DEFINITION

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Placement** □ **Symbols** □ Placement Parameters □ Level  
CELLEV

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Placement** □ **Symbols** □ Placement Parameters □ Scaling factor  
CELL\_SCAFACTOR

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Placement** □ **Symbols** □ Placement Parameters □ Maintain text size when scaling  
MAINTAIN\_TEXTSIZE

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Placement** □ **Symbols** □ Placement Parameters □ Load symbol graphic from database only if required  
LOAD\_SYMBOL\_GRAPHIC\_ONLY\_IF\_REQUIRED

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Placement** □ **Symbols** □ Text Parameters □ Font  
STXTFONTPTR

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Placement** □ **Symbols** □ Text Parameters □ Change already placed symbols **Placement** □ **Symbols** □ Connector Symbols □ For connectors without componentDEFAULT\_WIRESYM

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetSymbolForConnectorsWithoutCompcode()

e3Job.SetSymbolForConnectorsWithoutCompcode()

**Placement** □ **Symbols** □ Connector Symbols □ For block connectors without componentDEFAULT\_BLCONN

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetSymbolForBlockConnectorsWithoutCompcode()

e3Job.SetSymbolForBlockConnectorsWithoutCompcode()

**Placement** □ **Symbols** □ Connector Symbols □ Gap when placing several single pinsCONN\_CELL\_GAP

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetGapToPlaceSinglePins()

e3Job.SetGapToPlaceSinglePins()

**Placement** □ **Symbols** □ Connector Symbols □ Determine symbol of connectors without component using placed connector symbolsTRY\_AUTO\_GET\_CONNECTOR\_SYMBOL

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetDetermineConnectorSymbol()

e3Job.SetDetermineConnectorSymbol()

**Placement** □ **Symbols** □ Place as graphic □ Create originCREATE\_ORIGIN

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Symbols** □ Symbol text □ Change complete device when changing symbol text for higher level assignment, location and device designation  
CHANGE\_COMPLETE\_DEVICE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Symbols** □ Attribute text □ Use only selected symbols when creating texts  
USE\_SELECTED\_SYMBOLS\_FOR\_ATTRIBUTE\_TEXTS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Symbols** □ **Symbol Views** □ Pin View Symbols □ Determine symbol for pin views using placed connector symbols  
TRY\_AUTO\_GET\_PIN\_VIEW\_SYMBOL

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetDetermineConnectorViewSymbol()

e3Job.SetDetermineConnectorViewSymbol()

**Placement** □ **Symbols** □ **Symbol Views** □ Pin View Symbols □ For device pins  
DEV\_PINVIEW\_SYM\_NAME

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetPinViewSymbolForDevicePins()

e3Job.SetPinViewSymbolForDevicePins()

**Placement** □ **Symbols** □ **Symbol Views** □ Pin View Symbols □ For connector pins  
CONN\_PINVIEW\_SYM\_NAME

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetPinViewSymbolForConnectorPins()

e3Job.SetPinViewSymbolForConnectorPins()

**Placement** □ **Symbols** □ **Symbol Views** □ Pin View Symbols □ For block connector pins  
BLCON\_PINVIEW\_SYM\_NAME

e3Job.GetSettingValue()

- [e3Job.SetSettingValue\(\)](#)

[e3Job.GetPinViewSymbolForBlockConnectorPins\(\)](#)

[e3Job.SetPinViewSymbolForBlockConnectorPins\(\)](#)

**Placement** □ **Symbols** □ **Symbol Views** □ Terminal Strip View Symbol □ For terminal strip views  
TERMSTRIP\_VIEW\_SYM\_NAME

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

[e3Job.GetViewSymbolForTerminalStrips\(\)](#)

[e3Job.SetViewSymbolForTerminalStrips\(\)](#)

**Placement** □ **Symbols** □ **Bundle Symbols** □ Default symbols for Drag&Drop actions □  
Shield symbol  
DEFAULT\_SHIELDSYM

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Placement** □ **Symbols** □ **Bundle Symbols** □ Default symbols for Drag&Drop actions □  
Twisted pair symbol  
DEFAULT\_TWISTED\_PAIRSYM

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Placement** □ **Symbols** □ **Bundle Symbols** □ Default symbols for Drag&Drop actions □  
Grouped symbol  
DEFAULT\_BUNDLESYM

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Placement** □ **Symbols** □ **Bundle Symbols** □ Parameters for automatic placement □  
Minimum distance to pin  
BUNDLE\_DISTANCE\_PIN

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Placement** □ **Symbols** □ **Bundle Symbols** □ Parameters for automatic placement □ Show  
bundle symbol only at connection  
end  
SHOW\_BUNDLE\_SYMBOL\_ONLY\_AT\_CONNECTION\_END

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Placement** □ **Symbols** □ **Bundle Symbols** □ Parameters for automatic placement □  
Minimum distance between bundle symbols  
BUNDLE\_DISTANCE\_MIDDLE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Symbols** □ **Bundle Symbols** □ Parameters for automatic placement □ Line overlapping □ Shield / BottomSHIELD\_BOTTOM\_OVERLAPPING

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Symbols** □ **Bundle Symbols** □ Parameters for automatic placement □ Line overlapping □ Shield / TopSHIELD\_TOP\_OVERLAPPING

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Symbols** □ **Bundle Symbols** □ Parameters for automatic placement □ Line overlapping □ Twisted pair / BottomTWISTED\_PAIR\_BOTTOM\_OVERLAPPING

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Symbols** □ **Bundle Symbols** □ Parameters for automatic placement □ Line overlapping □ Twisted pair / TopTWISTED\_PAIR\_TOP\_OVERLAPPING

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Symbols** □ **Bundle Symbols** □ Parameters for automatic placement □ Line overlapping □ Bundle / Bottom

BUNDLE\_BOTTOM\_OVERLAPPING

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Symbols** □ **Bundle Symbols** □ Parameters for automatic placement □ Line overlapping □ Bundle / TopBUNDLE\_TOP\_OVERLAPPING

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Symbols** □ **Bundle Symbols** □ Twisted Pair □ ArrowTWISTED\_PAIR\_ARROW\_TYP

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Symbols** □ **Bundle Symbols** □ Twisted Pair □ Arrow widthTWISTED\_PAIR\_ARROW\_WIDTH

e3Job.GetSettingValue()



- e3Job.SetSettingValue()

**Placement** □ **Symbols** □ **Bundle Symbols** □ Adjust bundle symbol's size after modifying connection line's

positionADJUST\_BUNDLE\_SYMBOL\_SIZE\_AFTER\_MODIFYING\_CONNECTION\_LINE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Field** □ Text template □ SymbolFIELD\_TEXT\_TEMPLATE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetFieldTextTemplate()

e3Job.SetFieldTextTemplate()

**Placement** □ **Field** □ Outline □ WidthFIELD\_DIA

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetFieldOutlineWidth()

e3Job.SetFieldOutlineWidth()

**Placement** □ **Field** □ Outline □ Line StyleFIELD\_MODE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetFieldOutlineStyle()

e3Job.SetFieldOutlineStyle()

**Placement** □ **Field** □ Outline □ ColorFIELD\_CODE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetFieldOutlineColour()

e3Job.SetFieldOutlineColour()

**Placement** □ **Field** □ Outline □ Interrupt field border when connect line intersects borderCUT\_FIELD\_BORDER

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Field** □ Outline □ Interrupt field border when connect line intersects border  
□ WidthCUT\_FIELD\_BORDER\_GAP

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Field** □ Hatch □ PatternFIELD\_HATCH\_FLAGS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetFieldHatchPattern()

e3Job.SetFieldHatchPattern()

**Placement** □ **Field** □ Hatch □ WidthFIELD\_HATCH\_DIA

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetFieldHatchLineWidth()

e3Job.SetFieldHatchLineWidth()

**Placement** □ **Field** □ Hatch □ DistanceFIELD\_HATCH\_LDIST

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetFieldHatchLineDistance()

e3Job.SetFieldHatchLineDistance()

**Placement** □ **Field** □ Hatch □ ColorFIELD\_HATCH\_CODE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetFieldHatchColour()

e3Job.SetFieldHatchColour()

**Placement** □ **Field** □ Origin □ Place origin in upper left instead of lower  
leftFIELD\_CELL\_POS

Possible Values

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.SetFieldOriginInUpperLeft()

**Placement** □ **Field** □ Usage □ Rename fields when changing the designations in project tree  
RENAME\_FIELDS\_IN\_TREE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Dynamic Symbol** □ Text template □ Symbol  
DYN\_SYM\_TEXT\_TEMPLATE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Dynamic Symbol** □ Outline □ Width  
DYN\_SYM\_DIA

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Dynamic Symbol** □ Outline □ Line Style  
DYN\_SYM\_MODE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Dynamic Symbol** □ Outline □ Color  
DYN\_SYM\_CODE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Dynamic Symbol** □ Hatch □ Pattern  
DYN\_SYM\_HATCH\_FLAGS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Dynamic Symbol** □ Hatch □ Width  
DYN\_SYM\_HATCH\_DIA

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Dynamic Symbol** □ Hatch □ Distance  
DYN\_SYM\_HATCH\_LDIST

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Dynamic Symbol** □ Hatch □ Color  
DYN\_SYM\_HATCH\_CODE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Dynamic Symbol** □ Origin □ Place origin in upper left instead of lower left  
DYN\_SYM\_CELL\_POS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.SetDynamicSymbolOriginInUpperLeft()

**Placement** □ **Block** □ FontBLOCKREFFONTPTR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetBlockTextFont()

e3Job.SetBlockTextFont()

**Placement** □ **Block** □ Font StyleBLOCKREFDIA

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetBlockTextStyle()

e3Job.SetBlockTextStyle()

**Placement** □ **Block** □ SizeBLOCKREFSIZ

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetBlockTextSize()

e3Job.SetBlockTextSize()

**Placement** □ **Block** □ ColorBLOCKREFCOD

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetBlockTextColour()

e3Job.SetBlockTextColour()

**Placement** □ **Block** □ Display Control □ GapBLOCKREFGAP

Possible Values

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetBlockReferenceTextGap()

e3Job.SetBlockReferenceTextGap()

**Placement** □ **Block** □ Display Control □ LevelBLOCKREFLAY

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetBlockReferenceTextLevel()

e3Job.SetBlockReferenceTextLevel()

**Placement** □ **Block** □ Display Control □ RotateBLOCKREFJUST

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetBlockTextAlignment()

e3Job.SetBlockTextAlignment()

e3Job.GetBlockReferenceTextRotate()

e3Job.SetBlockReferenceTextRotate()

**Placement** □ **Block** □ Display Control □ Direction □ UpBLOCKREFGAP

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetBlockReferenceTextDirection()

e3Job.SetBlockReferenceTextDirection()

**Placement** □ **Block** □ Display Control □ Direction □ DownBLOCKREFGAP

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetBlockReferenceTextDirection()

e3Job.SetBlockReferenceTextDirection()

**Placement** □ **Block** □ Ratio □ NormalBLOCKREFMOD

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetBlockTextRatio()

e3Job.SetBlockTextRatio()

**Placement** □ **Block** □ Ratio □ NarrowBLOCKREFMOD

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetBlockTextRatio()

e3Job.SetBlockTextRatio()

**Placement** □ **Block** □ Ratio □ WideBLOCKREFMOD

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetBlockTextRatio()

e3Job.SetBlockTextRatio()

**Placement** □ **Block** □ Direction □ LeftBLOCKREFJUST

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetBlockReferenceTextRotate()

e3Job.SetBlockReferenceTextRotate()

**Placement** □ **Block** □ Direction □ CenterBLOCKREFJUST

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetBlockReferenceTextRotate()

e3Job.SetBlockReferenceTextRotate()

**Placement** □ **Block** □ Direction □ RightBLOCKREFJUST

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetBlockReferenceTextRotate()

e3Job.SetBlockReferenceTextRotate()

**Placement** □ **Block** □ Type □ AllBLOCKREFSORT

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetBlockReferenceType()

e3Job.SetBlockReferenceType()

**Placement** □ **Block** □ Type □ OriginBLOCKREFSORT

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetBlockReferenceType()

e3Job.SetBlockReferenceType()

**Placement** □ **Block** □ Type □ PositionBLOCKREFSORT

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetBlockReferenceType()

e3Job.SetBlockReferenceType()

**Placement** □ **Block** □ Split Block □ Mark split blocks □ NoneBLOCKEDGE\_TYPE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Block** □ Split Block □ Mark split blocks □ LeftBLOCKEDGE\_TYPE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Block** □ Split Block □ Mark split blocks □ RightBLOCKEDGE\_TYPE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Block** □ Split Block □ Mark split blocks □ TopBLOCKEDGE\_TYPE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Block** □ Split Block □ Mark split blocks □ BottomBLOCKEDGE\_TYPE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Block** □ Split Block □ Line □ WidthBLOCKEDGE\_DIA

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Block** □ Split Block □ Line □ Line StyleBLOCKEDGE\_MOD

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Block** □ Split Block □ Line □ ColorBLOCKEDGE\_COD

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Block** □ Split Block □ Split the block and copy the graphic contentsSPLITBLOCKOPTION

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetBlockCopyGraphicInSplit()

e3Job.SetBlockCopyGraphicInSplit()

**Placement** □ **Block** □ Block style □ Ignore default fill colorIGNORE\_FILL\_COLOUR\_BLOCK

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **References Format** □ PrefixBLOCKREFSETTING

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **References Format** □ SuffixBLOCKREFSETTING

e3Job.GetSettingValue()



- [e3Job.SetSettingValue\(\)](#)

**Placement** □ **References Format** □ Sheet textBLOCKREFSETTING

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Placement** □ **References Format** □ Reference textBLOCKREFSETTING

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Placement** □ **Import** □ Merge Sheet Reference Options □ Merge sheet referencesDF\_MERGE\_SHEET\_REFERENCES

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

[e3Job.GetMergeSheetReferences\(\)](#)

[e3Job.SetMergeSheetReferences\(\)](#)

**Placement** □ **Import** □ Merge Sheet Reference Options □ Merge only if reference names contain letters or special charactersDF\_MERGE\_ALPHANUMERIC\_REFERENCES

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

[e3Job.GetMergeAlphanumericReferences\(\)](#)

[e3Job.SetMergeAlphanumericReferences\(\)](#)

**Placement** □ **Import** □ Merge Connect Line Options □ Merge connect lines for import and modificationMERGE\_CONNECTION\_LINES

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

[e3Job.GetImportMergeConnectLines\(\)](#)

[e3Job.SetImportMergeConnectLines\(\)](#)

**Placement** □ **Import** □ Sheets □ Create unique sheet namesUNIQUE\_SHEET\_NAMES

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

[e3Job.GetCreateUniqueSheetNames\(\)](#)

e3Job.SetCreateUniqueSheetNames()

**Placement** ☐ **Import** ☐ Sheets ☐ Ignore sheet borderIGNORE\_SHEET\_BORDER

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** ☐ **Import** ☐ Variants/Options ☐ Rename already existing variants/options from part fileIMPORT\_RENAME\_VARIANTS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** ☐ **Import** ☐ Variants/Options ☐ Use existing variants/options from projectIMPORT\_RENAME\_VARIANTS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** ☐ **Import** ☐ Variants/Options ☐ Ask for each existing variant/optionIMPORT\_RENAME\_VARIANTS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** ☐ **Import** ☐ Variants/Options ☐ Merge inclusive/exclusive definitionsMERGE\_INCLUSIVE\_EXCLUSIVE\_DEFINITIONS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** ☐ **Import** ☐ **Device** ☐ Generate Item Designation ☐ Suffix ( CheckBox )DFI\_USE\_DEFDES\_COPY\_POSTFIX

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetImportUseItemDesignationSuffix()

e3Job.SetImportUseItemDesignationSuffix()

**Placement** ☐ **Import** ☐ **Device** ☐ Generate Item Designation ☐ Suffix ( EditBox )DFI\_DEFDES\_COPY\_POSTFIX

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetImportItemDesignationSuffix()

e3Job.SetImportItemDesignationSuffix()

**Placement** ☐ **Import** ☐ **Device** ☐ Unplaced Devices ☐ Ignore unplaced devices  
IGNORE\_UNPLACED\_DEVICES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** ☐ **Import** ☐ **Device** ☐ Unplaced Devices ☐ Ignore unplaced terminals  
IGNORE\_UNPLACED\_TERMINALS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** ☐ **Import** ☐ **Device** ☐ Unplaced Devices ☐ Ignore unplaced devices of assemblies  
IGNORE\_UNPLACED\_DEVICES\_OF\_ASSEMBLIES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** ☐ **Import** ☐ **Device** ☐ Merge Device Options ☐ Use existing devices  
DFI\_USE\_EXISTING\_DEVICES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetImportMergeExistingDevices()

e3Job.SetImportMergeExistingDevices()

**Placement** ☐ **Import** ☐ **Device** ☐ Merge Device Options ☐ Use existing assemblies  
DFI\_USE\_EXISTING\_ASSEMBLIES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetImportMergeExistingAssemblies()

e3Job.SetImportMergeExistingAssemblies()

**Placement** ☐ **Import** ☐ **Device** ☐ Merge Device Options ☐ Use existing terminal strips  
DFI\_MERGE\_TERMINALS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetImportMergeExistingTerminalStrips()

e3Job.SetImportMergeExistingTerminalStrips()

**Placement** □ **Import** □ **Device** □ Merge Device Options □ Use existing hierarchical blocks  
USE\_EXISTING\_HIERARCHICAL\_BLOCKS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Import** □ **Device** □ Merge Device Options □ Additional Merge Options □ Merge attributes

Available before v2022-23.00

DFI\_MERGE\_ATTRIBUTES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetImportMergeAttributes()

e3Job.SetImportMergeAttributes()

**Placement** □ **Import** □ **Device** □ Merge Device Options □ Additional Merge Options □ Merge using exact conductor connection  
MERGE\_USING\_EXACT\_CORE\_CONNECTION

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetMergeUsingExactCoreConnectionOnImport()

e3Job.SetMergeUsingExactCoreConnectionOnImport()

**Placement** □ **Import** □ **Device** □ Merge Device Options □ Additional Merge Options □ Merge ignoring conductor direction  
MERGE\_IGNOREING\_CORE\_DIRECTION

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Import** □ **Device** □ Merge Device Options □ Additional Merge Options □ Merge options  
MERGE\_OPTIONS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetImportMergeOptions()

e3Job.SetImportMergeOptions()

**Placement** □ **Import** □ **Device** □ Merge Device Options □ Additional Merge Options □ Use pin attributes from subcircuit

Available before v2022-23.00

IMPORT\_USE\_PIN\_ATTR

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

[e3Job.GetUsePinAttributesOnImport\(\)](#)

[e3Job.SetUsePinAttributesOnImport\(\)](#)

**Placement** □ **Import** □ **Device** □ Merge Device Options □ Additional Merge Options □ Ignore component nameIGNORE\_COMPONENT\_CODE\_ON\_IMPORT

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Placement** □ **Import** □ **Device** □ Merge Device Options □ Additional Merge Options □ Add to existing assemblies / create new assembliesKEEP\_ASSEMBLY\_BELONGING

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Placement** □ **Import** □ **Device** □ Use Default View Number □ For original symbolsUSE\_DEFAULT\_VIEW\_OF\_SHEET\_FOR\_SYMBOLS\_WITHOUT\_VIEW

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Placement** □ **Import** □ **Device** □ Use Default View Number □ For view symbolsUSE\_DEFAULT\_VIEW\_OF\_SHEET\_FOR\_SYMBOLS\_WITH\_VIEW

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Placement** □ **Import** □ **Device** □ Merge Attribute Options □ Project preferred

Available from v2022-23.00

DFI\_MERGE\_ATTRIBUTES

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

[e3Job.GetImportMergeAttributes\(\)](#)

### e3Job.SetImportMergeAttributes()

**Placement** □ **Import** □ **Device** □ Merge Attribute Options □ Subcircuit preferred □ Devices □ Merge

Available from v2022-23.00

OPTION\_IMPORT\_DEVICE\_MERGE\_ATTRIBUTES

### e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Import** □ **Device** □ Merge Attribute Options □ Subcircuit preferred □ Devices □ Only

Available from v2022-23.00

OPTION\_IMPORT\_DEVICE\_ONLY\_ATTRIBUTES

### e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Import** □ **Device** □ Merge Attribute Options □ Subcircuit preferred □ Pins □ Merge

Available from v2022-23.00

IMPORT\_USE\_PIN\_ATTR

OPTION\_IMPORT\_PIN\_MERGE\_ATTRIBUTES

### e3Job.GetSettingValue()

- e3Job.SetSettingValue()

### e3Job.GetUsePinAttributesOnImport()

### e3Job.SetUsePinAttributesOnImport()

**Placement** □ **Import** □ **Device** □ Merge Attribute Options □ Subcircuit preferred □ Pins □ Only

Available from v2022-23.00

OPTION\_IMPORT\_PIN\_ONLY\_ATTRIBUTES

### e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Import** □ **Device** □ Merge Attribute Options □ Subcircuit preferred □ Wires □ Merge

Possible Values

Available from v2022-23.00

OPTION\_IMPORT\_WIRE\_MERGE\_ATTRIBUTES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Import** □ **Device** □ Merge Attribute Options □ Subcircuit preferred □ Wires □ Only

Available from v2022-23.00

OPTION\_IMPORT\_WIRE\_ONLY\_ATTRIBUTES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Export/Copy** □ General □ Export structure  
nodesEXPORT\_STRUCTURE\_NODES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Export/Copy** □ Devices □ Unplaced devices (only valid when exporting  
'all')EXPORT\_UNPLACED\_DEVICES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Export/Copy** □ Devices □ Unplaced terminals (only valid when exporting  
'all')EXPORT\_UNPLACED\_TERMINALS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Export/Copy** □ Devices □ All devices of a selected  
assemblyEXPORT\_ALL\_UNPLACED\_DEVICES\_OF\_ASSEMBLIES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Export/Copy** □ Devices □ All terminals of a selected terminal  
stripEXPORT\_ALL\_UNPLACED\_TERMINALS\_OF\_STRIP

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Export/Copy** □ Devices □ Hierarchical block with  
substructuresHIERARCHYBLOCK\_WITH\_STRUCTURE

e3Job.GetSettingValue()

Possible Values

- [e3Job.SetSettingValue\(\)](#)

**Placement** □ **Export/Copy** □ Cables / wires □ Cables / wiresCOPY\_EXPORT\_CABLE

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

[e3Job.GetExportWithCablesAndWires\(\)](#)

[e3Job.SetExportWithCablesAndWires\(\)](#)

**Placement** □ **Export/Copy** □ Cables / wires □ At least one end  
selectedCOPY\_EXPORT\_CABLE\_OPTION

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

[e3Job.GetExportWithCablesAndWiresOption\(\)](#)

[e3Job.SetExportWithCablesAndWiresOption\(\)](#)

**Placement** □ **Export/Copy** □ Cables / wires □ Both ends  
selectedCOPY\_EXPORT\_CABLE\_OPTION

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

[e3Job.GetExportWithCablesAndWiresOption\(\)](#)

[e3Job.SetExportWithCablesAndWiresOption\(\)](#)

**Placement** □ **Export/Copy** □ Cables / wires □ Both ends and the path are  
selectedCOPY\_EXPORT\_CABLE\_OPTION

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

[e3Job.GetExportWithCablesAndWiresOption\(\)](#)

[e3Job.SetExportWithCablesAndWiresOption\(\)](#)

**Placement** □ **Export/Copy** □ Cables / wires □ Full path is  
selectedCOPY\_EXPORT\_CABLE\_OPTION

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)



e3Job.GetExportWithCablesAndWiresOption()

e3Job.SetExportWithCablesAndWiresOption()

**Placement** □ **Export/Copy** □ Busbars □ BusbarsEXPORT\_BUSBAR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Export/Copy** □ Busbars □ All pins in path  
selectedEXPORT\_BUSBAR\_ALL\_PINS\_IN\_PATH\_SELECTED

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Export/Copy** □ Plugging □ Export plugging information for pluggings  
without graphical  
representationEXPORT\_PLUGS\_WITHOUT\_GRAPHICAL\_REPRESENTATIONS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Placement** □ **Connector Symbols** □ Define Text  
VisibilityCONNECTOR\_SYMBOL\_FORMAT

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetConnectorSymbolFormat()

e3Job.SetConnectorSymbolFormat()

**Graphic** □ WidthGRADIA

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetGraphWidth()

e3Job.SetGraphWidth()

**Graphic** □ ArrowsGRAFLG

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetGraphArrows()

e3Job.SetGraphArrows()

**Graphic** □ ColorGRACOD

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetGraphColour()

e3Job.SetGraphColour()

**Graphic** □ LevelGRALEV

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetGraphLevel()

e3Job.SetGraphLevel()

**Graphic** □ Line StyleGRAMOD

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetGraphStyle()

e3Job.SetGraphStyle()

**Graphic** □ Line Style □ DefaultLINESTYLE\_JIS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Graphic** □ Line Style □ Japanese Industrial StandardsLINESTYLE\_JIS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Graphic** □ Redlining Information □ Display non-redliner information in different colorUSE\_NON\_REDLINER\_COLOR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Graphic** □ Redlining Information □ ColorNON\_REDLINER\_COLOR

e3Job.GetSettingValue()

Possible Values

- e3Job.SetSettingValue()

**Graphic** □ Read-Only Level □ Use read-only levelUSE\_READ\_ONLY\_GRAPHIC\_LEVEL

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Graphic** □ Read-Only Level □ ColorREAD\_ONLY\_GRAPHIC\_COLOR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Graphic** □ Read-Only Level □ LevelREAD\_ONLY\_GRAPHIC\_LEVEL

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Graphic** □ **Hatch** □ Pattern

GRAHATCH\_DEGREE1

GRAHATCH\_DEGREE2

GRAHATCH\_FLAGS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetGraphHatchPattern()

e3Job.SetGraphHatchPattern()

**Graphic** □ **Hatch** □ Line StyleGRAHATCH\_GRAMOD

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetGraphHatchStyle()

e3Job.SetGraphHatchStyle()

**Graphic** □ **Hatch** □ WidthGRAHATCH\_GRACOD

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetGraphHatchWidth()

e3Job.SetGraphHatchWidth()

**Graphic ▯ Hatch ▯ Distance**GRAHATCH\_LINDIST

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetGraphHatchDistance()

e3Job.SetGraphHatchDistance()

**Graphic ▯ Hatch ▯ Color**GRAHATCH\_GRACOD

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Graphic ▯ Text ▯ Font ▯ Name**TEXTFONTPTR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetGraphTextFontName()

e3Job.SetGraphTextFontName()

**Graphic ▯ Text ▯ Font ▯ Style**TXTDIA

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetGraphTextStyle()

e3Job.SetGraphTextStyle()

**Graphic ▯ Text ▯ Font ▯ Size**TXTSIZ

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetGraphTextSize()

e3Job.SetGraphTextSize()

e3Job.GetGraphTextHeight()

e3Job.SetGraphTextHeight()

**Graphic ▯ Text ▯ Font ▯ Color**TXTCOD

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetGraphTextColour()

e3Job.SetGraphTextColour()

**Graphic** □ **Text** □ Font □ RatioTXTMOD

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetGraphTextMode()

e3Job.SetGraphTextMode()

**Graphic** □ **Text** □ Font □ AlignmentTXTJUST

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Graphic** □ **Text** □ Font □ LevelTXTLEV

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetGraphTextLevel()

e3Job.SetGraphTextLevel()

**Graphic** □ **Text** □ Effects □ StrikeoutTXTDIA

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetGraphTextStyle()

e3Job.SetGraphTextStyle()

**Graphic** □ **Text** □ Effects □ UnderlineTXTDIA

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetGraphTextStyle()

e3Job.SetGraphTextStyle()

**Graphic** □ **Text** □ Effects □ Subsidiary line to graphicSUBSIDIARY\_LINE\_TO\_GRAPHIC

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Dimensions** □ Lines □ ArrowDIM\_ARROW\_TYP

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Dimensions** □ Lines □ Arrow WidthDIM\_ARROW\_WIDTH

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Dimensions** □ Lines □ Use fixed size to displayDIM\_FIX\_SIZE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Dimensions** □ Lines □ Hide longer part of arrowDIM\_HIDE\_LONGER\_PART

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Dimensions** □ Lines □ ExtensionDIM\_EXTENSION

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Dimensions** □ Lines □ Line width DIM\_WIDTH

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Dimensions** □ Lines □ Extension line offsetDIM\_EXTENSION\_LINE\_OFFSET

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Dimensions** □ Text □ PrecisionDIM\_PRECISION

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Dimensions** □ Text □ OffsetDIM\_TXT\_OFFSET

e3Job.GetSettingValue()

- [e3Job.SetSettingValue\(\)](#)

**Dimensions** □ Text □ PrefixDIM\_PREFIX

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Dimensions** □ Text □ Center textsDIMENSION\_TEXT\_CENTER

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Dimensions** □ Text □ Rotate texts of running dimensionsDIM\_ROTATE\_TEXT\_OF\_RUNNING\_DIM

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Dimensions** □ Text □ Suffix size factor (%)DIM\_SUFFIX\_SIZE\_FACTOR

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Dimensions** □ Text □ SuffixDIM\_SUFFIX

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Dimensions** □ Text □ DisplayDIM\_DISPLAY\_ATTR

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Dimensions** □ General □ LevelDIM\_LEVEL

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Dimensions** □ General □ ColorDIM\_COLOR

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Dimensions** □ **Dimension Text** □ Font □ NameDIM\_TXT\_FONTPTR

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Dimensions** □ **Dimension Text** □ Font □ StyleDIM\_TXT\_DIA

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Dimensions** □ **Dimension Text** □ Font □ SizeDIM\_TXT\_SIZE

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Dimensions** □ **Dimension Text** □ Font □ ColorDIM\_TXT\_COLOR

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Dimensions** □ **Dimension Text** □ Effects □ StrikeoutDIM\_TXT\_DIA

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Dimensions** □ **Dimension Text** □ Effects □ UnderlineDIM\_TXT\_DIA

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Dimensions** □ **Dimension Text** □ Effects □ OpaqueDIM\_TXT\_DIA

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Panel** □ Working Grid □ Grid sizePANELGRIDSIZE

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

[e3Job.GetPanelGridSize\(\)](#)

[e3Job.SetPanelGridSize\(\)](#)

**Panel** □ Working Grid □ Snap sizePANELTRAPSIZE

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

[e3Job.GetPanelTrapSize\(\)](#)

[e3Job.SetPanelTrapSize\(\)](#)

**Panel** □ Alternative Grid □ Grid sizePANELALTGRIDSIZE

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)



e3Job.GetPanelAltGridSize()

e3Job.SetPanelAltGridSize()

**Panel** ☐ Measurement Units ☐ Millimeters MEA\_EXTERN\_PANEL

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ Measurement Units ☐ Inches MEA\_EXTERN\_PANEL

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ Grid View ☐ Points ( CheckBox ) PANEL\_MODE\_GRID\_OVERLAY

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ Grid View ☐ Points ( SpinControl ) PANELOVERSIZE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ Grid View ☐ Rulers ( CheckBox )

PANEL\_MODE\_GRID\_AXIS

SCHEMA\_MODE\_GRID\_AXIS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.EnableRulerGridDisplay()

e3Job.DisableRulerGridDisplay()

**Panel** ☐ Grid View ☐ Rulers ( SpinControl ) PANEL\_AXIS\_GRID

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ Highlight ☐ Color PANEL\_HIGHLIGHT\_COLOUR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ Highlight ☐ Width PANEL\_HIGHLIGHT\_WIDTH

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ Shared sheets □ Display region overviewDISPLAY\_REGION

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ Shared sheets □ Scaling increment of regionPANEL\_REGION\_SCALE\_STEP

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ Placement □ Optimize placement after changesOPTIMIZE\_PLACEMENT\_AFTER\_CHANGES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ 2D □ Display model orientation in 2D2D\_MODEL\_ORIENTATION

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ 2D □ Percentage of visibility to show partially covered modelsPANEL\_2D\_VISIBLE\_PERCENTAGE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ 3D □ Display model orientation in 3D3D\_MODEL\_ORIENTATION

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ 3D □ Display model graphic from 2D in 3D3D\_MODEL\_GRAPHIC

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ 3D □ Display STEP models in 3D3D\_DISPLAY\_STEP\_MODELS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ 3D □ Load STEP models from database only if required3D\_LOAD\_STEP\_MODELS\_IF\_REQUIRED

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ 3D □ Wireframe mode3D\_WIREFRAME

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ 3D □ Orthographic mode3D\_ORTHOGRAPHIC

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ 3D □ Enable lighting3D\_LIGHTING

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ **Renderer Settings** □ Preferred Renderer **Panel** □ **Renderer Settings** □ Anti-Aliasing **Panel** □ **Connection** □ Connection Method □ Signal on pinPANEL\_CONNECT\_METHOD

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ **Connection** □ Connection Method □ Graphical schema connectionsPANEL\_CONNECT\_METHOD

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ **Connection** □ Connection Method □ Use only connections with assigned wires/conductors in schematicPANEL\_CONNECT\_METHOD

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ **Connection** □ Wiring Options □ Allow wire loopsPANEL\_ALLOW\_WIRE\_LOOPS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ **Connection** □ Wiring Options □ Backplane connection distancePANEL\_BACKPLANE\_DISTANCE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ **Connection** □ Wiring Options □ Wire insulation factorPANEL\_WIRE\_INSULATION\_FACTOR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ **Connection** □ Autoconnect Parameter □ Nearest cable ductPANEL\_AUTOCONNECT\_CABLE\_DUCT\_SEARCH\_GAGE\_FLAG

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ **Connection** ☐ Autoconnect Parameter ☐ Bandwidth for searching cable duct ( RadioButton ) **Panel** ☐ **Connection** ☐ Autoconnect Parameter ☐ Bandwidth for searching cable duct ( SpinControl ) PANEL\_AUTOCONNECT\_CABLE\_DUCT\_SEARCH\_GAGE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ **Connection** ☐ Autoconnect Parameter ☐ Use manually defined ports for terminals in schematic PANEL\_AUTOCONNECT\_USE\_SCHEMATIC\_TERMINAL\_PORTS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ **Connection** ☐ Autoconnect Parameter ☐ Delete predefined ports in schematic PANEL\_AUTOCONNECT\_DELETE\_PREFDEF\_PORTS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ **Connection** ☐ Autoconnect Algorithm ☐ Route wire jumpers at the end PANEL\_AUTOCONNECT\_WIRE\_JUMPER\_FINALLY

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ **Connection** ☐ Autoconnect Algorithm ☐ Find suitable connector pin terminal with an additional wiring run DF\_CBHEAD\_PANEL\_AUTOCONNECT\_BIGGER\_CONNECTOR\_PIN\_TERMINAL

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ **Connection** ☐ Autoconnect Algorithm ☐ Current (Chain) PANEL\_AUTOCONNECT\_ALG\_CURRENT

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ **Connection** ☐ Autoconnect Algorithm ☐ Optimized (Pair of Pins) PANEL\_AUTOCONNECT\_ALG\_OPTIMIZE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ **Connection** ☐ Autoconnect Algorithm ☐ Write statistic PANEL\_ROUTING\_STATISTIC

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ **Connection** ☐ Display Options ☐ Use display parameters from wire  
 PANEL\_CONNECTION\_DISPLAY\_WIRE\_PARAMS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ **Connection** ☐ Display Options ☐ Mark sheet comprehensive wires  
 PANEL\_CONNECT\_LEAVING\_WIRE\_MARK

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ **Connection** ☐ Display Options ☐ Mark  
 sizePANEL\_CONNECT\_LEAVING\_WIRE\_MARK\_SIZE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ **Connection** ☐ Display Options ☐ Mark jumper connect  
 pointsPANEL\_CONNECT\_MARK\_JUMPER\_CONNECT\_POINTS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ **Connection** ☐ **Logic Lines** ☐ Display logic linesPANEL\_AIRLINES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ **Connection** ☐ **Logic Lines** ☐ Direct Connection ☐ Display direct  
 connectionsPANEL\_AIRLINES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ **Connection** ☐ **Logic Lines** ☐ Direct Connection ☐ **Advanced** ☐  
 ColorPANEL\_AIRLINES\_DIRECT\_COLOR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ **Connection** ☐ **Logic Lines** ☐ Direct Connection ☐ **Advanced** ☐ Line  
 StylePANEL\_AIRLINES\_DIRECT\_STYLE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel □ Connection □ Logic Lines □ Subnet Connection □ Display subnet connections**PANEL\_AIRLINES

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Panel □ Connection □ Logic Lines □ Subnet Connection □ Advanced □ Color**PANEL\_AIRLINES\_SUBNET\_COLOR

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Panel □ Connection □ Logic Lines □ Subnet Connection □ Advanced □ Line Style**PANEL\_AIRLINES\_SUBNET\_STYLE

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Panel □ Connection □ Logic Lines □ Equivalent Pins □ Display equivalent pins**PANEL\_AIRLINES

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Panel □ Connection □ Logic Lines □ Equivalent Pins □ Advanced □ Color**PANEL\_AIRLINES\_EQUIVALENT\_COLOR

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Panel □ Connection □ Logic Lines □ Equivalent Pins □ Advanced □ Line Style**PANEL\_AIRLINES\_EQUIVALENT\_STYLE

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Panel □ Connection □ Logic Lines □ Signal Carrying Pins □ Display signal carrying pins**PANEL\_AIRLINES

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Panel □ Connection □ Logic Lines □ Signal Carrying Pins □ Advanced □ Color**PANEL\_AIRLINES\_SIGNAL\_COLOR

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Panel □ Connection □ Logic Lines □ Signal Carrying Pins □ Advanced □ Width**PANEL\_AIRLINES\_SIGNAL\_WIDTH

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ **Connection** □ **Logic Lines** □ Sheet Comprehensive Connections □ Display connection logic on pinPANEL\_AIRLINES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ **Restricted Symbol** □ Defined for = All □ LevelPANEL\_RESTRICTED\_ALL\_LEVEL

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ **Restricted Symbol** □ Defined for = All □ Outline □ WidthPANEL\_RESTRICTED\_ALL\_LINE\_WIDTH

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ **Restricted Symbol** □ Defined for = All □ Outline □ ColorPANEL\_RESTRICTED\_ALL\_LINE\_COLOR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ **Restricted Symbol** □ Defined for = All □ Outline □ Line Style

PANEL\_RESTRICTED\_ALL\_LINE\_STYLE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ **Restricted Symbol** □ Defined for = All □ Hatch □ Pattern

PANEL\_RESTRICTED\_ALL\_HATCH\_DEGREE1

PANEL\_RESTRICTED\_ALL\_HATCH\_DEGREE2

PANEL\_RESTRICTED\_ALL\_HATCH\_FLAGS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ **Restricted Symbol** □ Defined for = All □ Hatch □ Line StylePANEL\_RESTRICTED\_ALL\_HATCH\_STYLE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ **Restricted Symbol** ☐ Defined for = All ☐ Hatch ☐  
WidthPANEL\_RESTRICTED\_ALL\_HATCH\_WIDTH

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ **Restricted Symbol** ☐ Defined for = All ☐ Hatch ☐  
DistancePANEL\_RESTRICTED\_ALL\_HATCH\_DIST

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ **Restricted Symbol** ☐ Defined for = All ☐ Hatch ☐  
ColorPANEL\_RESTRICTED\_ALL\_HATCH\_COLOR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ **Restricted Symbol** ☐ Defined for = Components ☐  
LevelPANEL\_RESTRICTED\_DEV\_LEVEL

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ **Restricted Symbol** ☐ Defined for = Components ☐ Outline ☐  
WidthPANEL\_RESTRICTED\_DEV\_LINE\_WIDTH

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ **Restricted Symbol** ☐ Defined for = Components ☐ Outline ☐  
ColorPANEL\_RESTRICTED\_DEV\_LINE\_COLOR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ **Restricted Symbol** ☐ Defined for = Components ☐ Outline ☐ Line  
StylePANEL\_RESTRICTED\_DEV\_LINE\_STYLE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ **Restricted Symbol** ☐ Defined for = Components ☐ Hatch ☐ Pattern

PANEL\_RESTRICTED\_DEV\_HATCH\_DEGREE1

PANEL\_RESTRICTED\_DEV\_HATCH\_DEGREE2

PANEL\_RESTRICTED\_DEV\_HATCH\_FLAGS

e3Job.GetSettingValue()



• e3Job.SetSettingValue()

**Panel** □ **Restricted Symbol** □ Defined for = Components □ Hatch □ Line  
StylePANEL\_RESTRICTED\_DEV\_HATCH\_STYLE

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Panel** □ **Restricted Symbol** □ Defined for = Components □ Hatch □  
WidthPANEL\_RESTRICTED\_DEV\_HATCH\_WIDTH

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Panel** □ **Restricted Symbol** □ Defined for = Components □ Hatch □  
DistancePANEL\_RESTRICTED\_DEV\_HATCH\_DIST

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Panel** □ **Restricted Symbol** □ Defined for = Components □ Hatch □  
ColorPANEL\_RESTRICTED\_DEV\_HATCH\_COLOR

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Panel** □ **Restricted Symbol** □ Defined for = Connections □  
LevelPANEL\_RESTRICTED\_CON\_LEVEL

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Panel** □ **Restricted Symbol** □ Defined for = Connections □ Outline □  
WidthPANEL\_RESTRICTED\_CON\_LINE\_WIDTH

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Panel** □ **Restricted Symbol** □ Defined for = Connections □ Outline □  
ColorPANEL\_RESTRICTED\_CON\_LINE\_COLOR

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Panel** □ **Restricted Symbol** □ Defined for = Connections □ Outline □ Line  
StylePANEL\_RESTRICTED\_CON\_LINE\_STYLE

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Panel** □ **Restricted Symbol** □ Defined for = Connections □ Hatch □ Pattern

PANEL\_RESTRICTED\_CON\_HATCH\_DEGREE1

PANEL\_RESTRICTED\_CON\_HATCH\_DEGREE2

PANEL\_RESTRICTED\_CON\_HATCH\_FLAGS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ **Restricted Symbol** ☐ Defined for = Connections ☐ Hatch ☐ Line  
StylePANEL\_RESTRICTED\_CON\_HATCH\_STYLE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ **Restricted Symbol** ☐ Defined for = Connections ☐ Hatch ☐  
WidthPANEL\_RESTRICTED\_CON\_HATCH\_WIDTH

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ **Restricted Symbol** ☐ Defined for = Connections ☐ Hatch ☐  
DistancePANEL\_RESTRICTED\_CON\_HATCH\_DIST

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ **Restricted Symbol** ☐ Defined for = Connections ☐ Hatch ☐  
ColorPANEL\_RESTRICTED\_CON\_HATCH\_COLOR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ **Restricted Symbol** ☐ Defined for = Cutout area ☐  
LevelPANEL\_RESTRICTED\_CUTOUT\_LEVEL

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ **Restricted Symbol** ☐ Defined for = Cutout area ☐ Outline ☐  
WidthPANEL\_RESTRICTED\_CUTOUT\_LINE\_WIDTH

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ **Restricted Symbol** ☐ Defined for = Cutout area ☐ Outline ☐  
ColorPANEL\_RESTRICTED\_CUTOUT\_LINE\_COLOR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ **Restricted Symbol** ☐ Defined for = Cutout area ☐ Outline ☐ Line  
StylePANEL\_RESTRICTED\_CUTOUT\_LINE\_STYLE

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Panel** ☐ **Restricted Symbol** ☐ Defined for = Cutout area ☐ Hatch ☐ Pattern

PANEL\_RESTRICTED\_CUTOUT\_HATCH\_DEGREE1

PANEL\_RESTRICTED\_CUTOUT\_HATCH\_DEGREE2

PANEL\_RESTRICTED\_CUTOUT\_HATCH\_FLAGS

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Panel** ☐ **Restricted Symbol** ☐ Defined for = Cutout area ☐ Hatch ☐ Line  
StylePANEL\_RESTRICTED\_CUTOUT\_HATCH\_STYLE

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Panel** ☐ **Restricted Symbol** ☐ Defined for = Cutout area ☐ Hatch ☐  
WidthPANEL\_RESTRICTED\_CUTOUT\_HATCH\_WIDTH

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Panel** ☐ **Restricted Symbol** ☐ Defined for = Cutout area ☐ Hatch ☐  
DistancePANEL\_RESTRICTED\_CUTOUT\_HATCH\_DIST

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Panel** ☐ **Restricted Symbol** ☐ Defined for = Cutout area ☐ Hatch ☐  
ColorPANEL\_RESTRICTED\_CUTOUT\_HATCH\_COLOR

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Panel** ☐ **Restricted Symbol** ☐ Defined for = Drill-hole ☐  
LevelPANEL\_RESTRICTED\_HOLE\_LEVEL

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Panel** ☐ **Restricted Symbol** ☐ Defined for = Drill-hole ☐ Outline ☐  
WidthPANEL\_RESTRICTED\_HOLE\_LINE\_WIDTH

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Panel** □ **Restricted Symbol** □ Defined for = Drill-hole □ Outline □ Color  
 PANEL\_RESTRICTED\_HOLE\_LINE\_COLOR

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Panel** □ **Restricted Symbol** □ Defined for = Drill-hole □ Outline □ Line  
 Style  
 PANEL\_RESTRICTED\_HOLE\_LINE\_STYLE

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Panel** □ **Restricted Symbol** □ Defined for = Drill-hole □ Hatch □ Pattern

PANEL\_RESTRICTED\_HOLE\_HATCH\_DEGREE1

PANEL\_RESTRICTED\_HOLE\_HATCH\_DEGREE2

PANEL\_RESTRICTED\_HOLE\_HATCH\_FLAGS

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Panel** □ **Restricted Symbol** □ Defined for = Drill-hole □ Hatch □ Line  
 Style  
 PANEL\_RESTRICTED\_HOLE\_HATCH\_STYLE

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Panel** □ **Restricted Symbol** □ Defined for = Drill-hole □ Hatch □  
 Width  
 PANEL\_RESTRICTED\_HOLE\_HATCH\_WIDTH

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Panel** □ **Restricted Symbol** □ Defined for = Drill-hole □ Hatch □  
 Distance  
 PANEL\_RESTRICTED\_HOLE\_HATCH\_DIST

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Panel** □ **Restricted Symbol** □ Defined for = Drill-hole □ Hatch □  
 Color  
 PANEL\_RESTRICTED\_HOLE\_HATCH\_COLOR

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Panel** □ **Restricted Symbol** □ **Display** □ Objects □ Restricted for  
 all  
 PANEL\_RESTRICTED\_DISPLAY

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ **Restricted Symbol** □ **Display** □ Objects □ Restricted for Components  
PANEL\_RESTRICTED\_DISPLAY

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ **Restricted Symbol** □ **Display** □ Objects □ Restricted for Connections  
PANEL\_RESTRICTED\_DISPLAY

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ **Restricted Symbol** □ **Display** □ Objects □ Cutout area  
PANEL\_RESTRICTED\_DISPLAY

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ **Restricted Symbol** □ **Display** □ Objects □ Drill-hole  
PANEL\_RESTRICTED\_DISPLAY

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ **Mount Symbol** □ Level  
PANEL\_MOUNT\_SYM\_LEVEL

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ **Mount Symbol** □ Outline □ Width  
PANEL\_MOUNT\_SYM\_LINE\_WIDTH

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ **Mount Symbol** □ Outline □ Color  
PANEL\_MOUNT\_SYM\_LINE\_COLOR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ **Mount Symbol** □ Outline □ Line Style  
PANEL\_MOUNT\_SYM\_LINE\_STYLE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ **Mount Symbol** □ Hatch □ Pattern

PANEL\_MOUNT\_SYM\_HATCH\_DEGREE1

PANEL\_MOUNT\_SYM\_HATCH\_DEGREE2

PANEL\_MOUNT\_SYM\_HATCH\_FLAGS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ **Mount Symbol** □ Hatch □ ColorPANEL\_MOUNT\_SYM\_HATCH\_COLOR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ **Cable Duct Symbol** □ LevelPANEL\_CABLE\_DUCT\_SYM\_LEVEL

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ **Cable Duct Symbol** □ Outline □ WidthPANEL\_CABLE\_DUCT\_SYM\_LINE\_WIDTH

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ **Cable Duct Symbol** □ Outline □ ColorPANEL\_CABLE\_DUCT\_SYM\_LINE\_COLOR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ **Cable Duct Symbol** □ Outline □ Line  
StylePANEL\_CABLE\_DUCT\_SYM\_LINE\_STYLE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ **Cable Duct Symbol** □ Hatch □ Pattern

PANEL\_CABLE\_DUCT\_SYM\_HATCH\_DEGREE1

PANEL\_CABLE\_DUCT\_SYM\_HATCH\_DEGREE2

PANEL\_CABLE\_DUCT\_SYM\_HATCH\_FLAGS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ **Cable Duct Symbol** □ Hatch □  
ColorPANEL\_CABLE\_DUCT\_SYM\_HATCH\_COLOR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ **Cable Duct Symbol** □ **Display** □ Fill Size □ Display fill sizePANEL\_CABLE\_DUCT

e3Job.GetSettingValue()

- e3Job.SetSettingValue()
- \_

e3Job.GetDisplayDuctFillSize()

e3Job.SetDisplayDuctFillSize()

**Panel** □ **Cable Duct Symbol** □ **Display** □ Fill Size □ **Advanced** □  
ColorPANEL\_CABLE\_DUCT\_FILL\_COLOR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetDisplayDuctFillSize()

e3Job.SetDisplayDuctFillSize()

**Panel** □ **Cable Duct Symbol** □ **Display** □ Fill Size □ **Advanced** □ Line  
StylePANEL\_CABLE\_DUCT\_FILL\_STYLE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetDisplayDuctFillSize()

e3Job.SetDisplayDuctFillSize()

**Panel** □ **Cable Duct Symbol** □ **Display** □ Fill Size □ Cable duct fill limit  
(%)PANEL\_CABLE\_DUCT\_FILL\_LIMIT

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Device.GetCableDuctFillLimit()

e3Device.SetCableDuctFillLimit()

**Panel** □ **Cable Duct Symbol** □ **Display** □ Fill Size □ Cable duct warning limit  
(%)PANEL\_CABLE\_DUCT\_CRITICAL\_FILL\_LIMIT

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Device.GetCableDuctWarningLimit()

e3Device.SetCableDuctWarningLimit()

**Panel** □ **Cable Duct Symbol** □ **Display** □ Fill Size □ Connection factor for space requirementsCORRECTION\_FACTOR\_FOR\_SPACE\_REQUIREMENTS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ **Cable Duct Symbol** □ **Display** □ Docking Point □ Display docking pointPANEL\_CABLE\_DUCT

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetDisplayDuctDockingPoints()

e3Job.SetDisplayDuctDockingPoints()

**Panel** □ **Cable Duct Symbol** □ **Display** □ Docking Point □ **Advanced** □ ColorPANEL\_CABLE\_DUCT\_DOCK\_COLOR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetDisplayDuctDockingPoints()

e3Job.SetDisplayDuctDockingPoints()

**Panel** □ **Cable Duct Symbol** □ **Display** □ Docking Point □ **Advanced** □ SizePANEL\_CABLE\_DUCT\_DOCK\_WIDTH

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetDisplayDuctDockingPoints()

e3Job.SetDisplayDuctDockingPoints()

**Panel** □ **Cable Duct Symbol** □ **Display** □ Lateral Punching Width □ Display lateral punching widthPANEL\_CABLE\_DUCT

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ **Cable Duct Symbol** □ **Display** □ Lateral Punching Width □ **Advanced** □ Color **Panel** □ **Cable Duct Symbol** □ **Display** □ Lateral Punching Width □ **Advanced** □ Size **Panel** □ **Cable Duct Symbol** □ **Display** □ Break Line □ Display break linePANEL\_CABLE\_DUCT



e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ **Cable Duct Symbol** □ **Display** □ Break Line □ **Advanced** □ Color **Panel** □ **Cable Duct Symbol** □ **Display** □ Break Line □ **Advanced** □ Size **Panel** □ **Checks** □ Placement □ Mounting description <-> Slot description □ OnPANEL\_CHECKS\_OUTLINE\_TO\_SLOT

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ **Checks** □ Placement □ Mounting description <-> Slot description □ WarningPANEL\_CHECKS\_OUTLINE\_TO\_SLOT

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ **Checks** □ Placement □ Mounting description <-> Slot description □ OffPANEL\_CHECKS\_OUTLINE\_TO\_SLOT

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ **Checks** □ Placement □ Complete component -> Slot Area/Line □ OnPANEL\_CHECKS\_FIT\_TO\_TARGET

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ **Checks** □ Placement □ Complete component -> Slot Area/Line □ WarningPANEL\_CHECKS\_FIT\_TO\_TARGET

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ **Checks** □ Placement □ Complete component -> Slot Area/Line □ OffPANEL\_CHECKS\_FIT\_TO\_TARGET

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ **Checks** □ Placement □ Component <-> Component □ OnPANEL\_CHECKS\_OUTLINE\_TO\_OUTLINE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ **Checks** □ Placement □ Component <-> Component □ WarningPANEL\_CHECKS\_OUTLINE\_TO\_OUTLINE

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Panel** ☐ **Checks** ☐ Placement ☐ Component <-> Component ☐  
OffPANEL\_CHECKS\_OUTLINE\_TO\_OUTLINE

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Panel** ☐ **Checks** ☐ Placement ☐ Component <-> Restricted ☐  
OnPANEL\_CHECKS\_OUTLINE\_TO\_RESTRICTED

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Panel** ☐ **Checks** ☐ Placement ☐ Component <-> Restricted ☐  
WarningPANEL\_CHECKS\_OUTLINE\_TO\_RESTRICTED

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Panel** ☐ **Checks** ☐ Placement ☐ Component <-> Restricted ☐  
OffPANEL\_CHECKS\_OUTLINE\_TO\_RESTRICTED

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Panel** ☐ **Checks** ☐ Placement ☐ Cutout <-> Component ☐  
OnPANEL\_CHECKS\_OUTLINE\_TO\_CUTOUT

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Panel** ☐ **Checks** ☐ Placement ☐ Cutout <-> Component ☐  
WarningPANEL\_CHECKS\_OUTLINE\_TO\_CUTOUT

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Panel** ☐ **Checks** ☐ Placement ☐ Cutout <-> Component ☐  
OffPANEL\_CHECKS\_OUTLINE\_TO\_CUTOUT

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Panel** ☐ **Checks** ☐ Placement ☐ Cutout <-> Restricted ☐  
OnPANEL\_CHECKS\_RESTRICTED\_TO\_CUTOUT

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Panel** ☐ **Checks** ☐ Placement ☐ Cutout <-> Restricted ☐  
WarningPANEL\_CHECKS\_RESTRICTED\_TO\_CUTOUT

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ **Checks** ☐ Placement ☐ Cutout <-> Restricted ☐  
OffPANEL\_CHECKS\_RESTRICTED\_TO\_CUTOUT

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ **Checks** ☐ Placement ☐ Variants/Options <-> Variants/Options ☐  
OnPANEL\_CHECKS\_COMPONENT\_OPTIONS\_VARIANTS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ **Checks** ☐ Placement ☐ Variants/Options <-> Variants/Options ☐  
WarningPANEL\_CHECKS\_COMPONENT\_OPTIONS\_VARIANTS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ **Checks** ☐ Placement ☐ Variants/Options <-> Variants/Options ☐  
OffPANEL\_CHECKS\_COMPONENT\_OPTIONS\_VARIANTS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ **Checks** ☐ Autoconnect ☐ Wire/Conductor from Pin <-> Component/Restricted ☐  
OnPANEL\_CHECKS\_WIRE\_TO\_OUTLINE\_1

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ **Checks** ☐ Autoconnect ☐ Wire/Conductor from Pin <-> Component/Restricted ☐  
WarningPANEL\_CHECKS\_WIRE\_TO\_OUTLINE\_1

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ **Checks** ☐ Autoconnect ☐ Wire/Conductor from Pin <-> Component/Restricted ☐  
OffPANEL\_CHECKS\_WIRE\_TO\_OUTLINE\_1

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** ☐ **Checks** ☐ Autoconnect ☐ Wire/Conductor to cable duct<-> Component/Restricted ☐  
☐ OnPANEL\_CHECKS\_WIRE\_TO\_OUTLINE\_2

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Panel** □ **Checks** □ Autoconnect □ Wire/Conductor to

cable duct<-> Component/Restricted □ Warning

PANEL\_CHECKS\_WIRE\_TO\_OUTLINE\_2

[e3Job.GetSettingValue\(\)](#)

[e3Job.SetSettingValue\(\)](#)

e

**Panel** □ **Checks** □ Autoconnect □ Wire/Conductor to cable duct<-> Component/Restricted  
□ OffPANEL\_CHECKS\_WIRE\_TO\_OUTLINE\_2

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Panel** □ **Checks** □ All attributesPANEL\_AUTOROUTE\_WITH\_EXPLICIT\_ATTRIBUTES

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Panel** □ **Checks** □ Only attribute ( RadioButton  
)PANEL\_AUTOROUTE\_WITH\_EXPLICIT\_ATTRIBUTES

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Panel** □ **Checks** □ Only attribute ( ComboBox )PANEL\_AUTOROUTE\_ATTRIBUTES

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Panel** □ **Checks** □ Allow crossing cable ducts with different connection  
classesPANEL\_CHECKS\_ALLOW\_CROSSING\_CABLEDUCTS

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Variants/Options** □ Project Setting □ Separator for variant  
textsVARIANT\_TEXT\_SEPARATOR

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Variants/Options** □ Project Setting □ Unique Names for  
Variants/OptionsOPTION\_VARIANT\_UNIQUE\_NAMES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Variants/Options** □ Create New Alias □ ( Table ) **Variants/Options** □ **Display** □  
Activation of Variants / Options □ Variants □ defaultVARIANT\_ACTIVE\_VARIANT

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetActiveVariantId()

e3Job.SetActiveVariantId()

**Variants/Options** □ **Display** □ Activation of Variants / Options □ Variants □  
allVARIANT\_ACTIVE\_VARIANT

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetActiveVariantId()

e3Job.SetActiveVariantId()

**Variants/Options** □ **Display** □ Activation of Variants / Options □ Options □  
defaultVARIANT\_ACTIVE\_VARIANT

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetActiveVariantId()

e3Job.SetActiveVariantId()

**Variants/Options** □ **Display** □ Activation of Variants / Options □ Options □  
allVARIANT\_ACTIVE\_VARIANT

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetActiveVariantId()

e3Job.SetActiveVariantId()

e3Job.GetDisplayOptionsAll()

e3Job.SetDisplayOptionsAll()

**Variants/Options** □ **Display** □ Activation of Variants / Options □ Options □  
noneVARIANT\_ACTIVE\_VARIANT

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetActiveVariantId()

e3Job.SetActiveVariantId()

e3Job.GetDisplayOptionsNone()

e3JobSetDisplayOptionsNone()

**Variants/Options** □ **Display** □ Activation of Variants / Options □ Display elements without  
variants / optionsDISPLAY\_OBJECTS\_WITHOUT\_VARIANTS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Variants/Options** □ **Display** □ Display Settings □ Display info in  
tooltipsSHOW\_VARIANT\_TOOLTIP

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Variants/Options** □ **Display** □ Display Settings □ Display type in  
expressionsDISPLAY\_TYPE\_EXPRESSION

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Variants/Options** □ **Display** □ Display Settings □ Display all values in  
textsVAR\_SHOW\_ALL\_VALUES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Variants/Options** □ **Display** □ Display Settings □ Mark availability of different active  
attribute values ( CheckBox )MARK\_DIFFERENT\_ACTIVE\_VARIANT\_VALUES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Variants/Options** □ **Display** □ Display Settings □ Mark availability of different active  
attribute values ( EditBox )ALTERNATIVE\_VARIANT\_STRING

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Variants/Options** □ **Display** □ Display Settings □ Highlight color for inactive variants/options ( CheckBox )DISPLAY\_OTHER\_VARIANTS\_GRAYED

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Variants/Options** □ **Display** □ Display Settings □ Highlight color for inactive variants/options ( ComboBox )VARIANT\_INACTIVE\_COLOR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Variants/Options** □ **Display** □ Display Settings □ Display elements with variants/options in another color ( CheckBox )

DRAW\_VAR\_OTHER\_COLOR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetDisplayOptionsColoured()

e3Job.SetDisplayOptionsColoured()

**Variants/Options** □ **Display** □ Display Settings □ Display elements with variants/options in another color ( ComboBox )VARIANT\_COLOR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Variants/Options** □ **Display** □ Display Settings □ Display elements without variants/options in another color ( CheckBox )DRAW\_OBJECTS\_WITHOUT\_VARIANTS\_OPTIONS\_IN\_OTHER\_COLOR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Variants/Options** □ **Display** □ Display Settings □ Display elements without variants/options in another color ( ComboBox )NO\_VARIANT\_COLOR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Variants/Options** □ **Display** □ Display Settings □ Highlight variants/options in the following colorVARIANT\_HIGHLIGHT\_COLOR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Variants/Options** □ **Variant Text** □ Font □ NameVARIANT\_TEXT\_FONT

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Variants/Options** □ **Variant Text** □ Font □ Style VARIANT\_TEXT\_DIA

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Variants/Options** □ **Variant Text** □ Font □ Size VARIANT\_TEXT\_SIZE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Variants/Options** □ **Variant Text** □ Font □ Color VARIANT\_TEXT\_COLOR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Variants/Options** □ **Variant Text** □ Font □ Ratio VARIANT\_TEXT\_MODE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Variants/Options** □ **Variant Text** □ Font □ Alignment VARIANT\_TEXT\_JUST

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Variants/Options** □ **Variant Text** □ Effects □ Strikeout VARIANT\_TEXT\_DIA

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Variants/Options** □ **Variant Text** □ Effects □ Underline VARIANT\_TEXT\_DIA

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Variants/Options** □ **Option Inheritance** □ Inherit Options of Sheet/Field/Hierarchical block □ No inheritance for symbols/devices from sheet from hierarchical block INHERIT\_OPTIONS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Variants/Options** □ **Option Inheritance** □ Inherit Options of Sheet/Field/Hierarchical block □ Add sheet options for symbols/devices / hierarchical block options for sheets INHERIT\_OPTIONS

e3Job.GetSettingValue()



• e3Job.SetSettingValue()

**Variants/Options** □ **Option Inheritance** □ Inherit Options of Sheet/Field/Hierarchical block □ Only sheet options for symbols/devices hierarchical block options for sheets INHERIT\_OPTIONS

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Variants/Options** □ **Option Inheritance** □ Inherit Options of Sheet/Field/Hierarchical block □ No inheritance from field INHERIT\_OPTIONS\_FIELD

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Variants/Options** □ **Option Inheritance** □ Inherit Options of Sheet/Field/Hierarchical block □ Add field options INHERIT\_OPTIONS\_FIELD

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Variants/Options** □ **Option Inheritance** □ Inherit Options of Sheet/Field/Hierarchical block □ Only field options INHERIT\_OPTIONS\_FIELD

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Variants/Options** □ **Option Inheritance** □ Option Inherit when Placing/Moving □ Symbols INHERIT\_OPTIONS\_OBJECTS

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Variants/Options** □ **Option Inheritance** □ Option Inherit when Placing/Moving □ Devices INHERIT\_OPTIONS\_OBJECTS

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Variants/Options** □ **Option Inheritance** □ Visibility of Symbols in Tree □ Depending on visibility of sheet INHERIT\_OPTIONS\_VISIBILITY

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Variants/Options** □ **Locking** □ Password for unlocking the variant / option structure □ Old password **Variants/Options** □ **Locking** □ Password for unlocking the variant / option structure □ New password **Variants/Options** □ **Locking** □ Password for unlocking the variant / option structure □ Confirm password **Formboard** □ General □ Display unconnected conductors

FORMBOARD\_SHOW\_UNCONNECTED\_CORES

## AIRLINES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetDisplayFormboardUnconnectedCores()

e3Job.SetDisplayFormboardUnconnectedCores()

**Formboard** □ General □ Formboard name as an extension to device designation  
FORMBOARD\_SHOW\_NAME\_OF\_FORMBOARD\_AS\_EXTENSION

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetDisplayAppendFormboardNameToDeviceName()

e3Job.SetDisplayAppendFormboardNameToDeviceName()

**Formboard** □ Table □ Table SymbolTABLESYMBOLPTR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetTableSymbol()

e3Job.SetTableSymbol()

**Formboard** □ Table □ AutoplaceFORMBOARD\_AUTOPLACE\_TABLE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetFormboardAutoplaceTableSymbol()

e3Job.SetFormboardAutoplaceTableSymbol()

**Formboard** □ Table □ Display subsidiary linesFORMBOARD\_SHOW\_TABLE\_AIRLINES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetDisplayFormboardTableSubsidiaryLines()

e3Job.SetDisplayFormboardTableSubsidiaryLines()

**Formboard** □ Table □ Display one row for each  
conductorFORMBOARD\_DISPLAY\_ONE\_TABLE\_ROW\_FOR\_EACH\_CORE

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Formboard** □ Table □ Display pins without  
conductorsFORMBOARD\_TABLE\_DISPLAY\_PINS\_WITHOUT\_CORES

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Formboard** □ Table □ Break table after ( CheckBox  
)FORMBOARD\_WRAP\_TABLE\_ROWS

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Formboard** □ Table □ Break table after ( SpinControl  
)FORMBOARD\_WRAP\_TABLE\_ROWS\_COUNT

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

**Formboard** □ Connections / Segments □ Mark segment with different manufacturing  
and displayed lengthsFORMBOARD\_SHOW\_SEGMENT\_LENGTHS\_DIFF

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

e3Job.GetDisplayFormboardMarkDifferenLength()

e3Job.SetDisplayFormboardMarkDifferenLength()

**Formboard** □ Connections / Segments □ Display effective direction on selected  
segmentsFORMBOARD\_SHOW\_EFFECTIVE\_DIRECTION

e3Job.GetSettingValue()

• e3Job.SetSettingValue()

e3Job.GetDisplayFormboardEffectiveDirection()

e3Job.SetDisplayFormboardEffectiveDirection()

**Formboard** □ Connections / Segments □ Display  
nodesFORMBOARD\_SHOW\_NET\_NODES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetDisplayFormboardNodes()

e3Job.SetDisplayFormboardNodes()

**Formboard** □ Connector □

AutorotateFORMBOARD\_AUTOROTATE\_FORMBOARD\_SYMBOLS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetFormboardAutorotateConnectorSymbols()

e3Job.SetFormboardAutorotateConnectorSymbols()

**Formboard** □ Rotate Branch □ Angle to rotateFORMBOARD\_ROTATE\_BRANCH\_ANGLE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetFormboardBranchAngleStep()

e3Job.SetFormboardBranchAngleStep()

**Formboard** □ Shared sheets □ Display region overviewDISPLAY\_FORMBOARD\_REGION

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Formboard** □ Shared sheets □ Scaling increment of regionFORMBOARD\_REGION\_SCALE\_STEP

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**MIL-Standard** □ Break-up Edge □ BottomDISPLAY\_MIL\_STANDARD

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetDisplayMILStandard()

e3Job.SetDisplayMILStandard()

**MIL-Standard** □ Break-up Edge □ Top and bottomDISPLAY\_MIL\_STANDARD

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetDisplayMILStandard()

e3Job.SetDisplayMILStandard()

**MIL-Standard** □ Display Options □ Display complete connector when final pin is contained in the groupDISPLAY\_MIL\_STANDARD

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetDisplayMILStandard()

e3Job.SetDisplayMILStandard()

**MIL-Standard** □ Display Options □ Display complete connector when all placed pins are contained in one groupDISPLAY\_MIL\_STANDARD

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetDisplayMILStandard()

e3Job.SetDisplayMILStandard()

**MIL-Standard** □ Display Options □ Always display complete groupDISPLAY\_MIL\_STANDARD

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetDisplayMILStandard()

e3Job.SetDisplayMILStandard()

**MIL-Standard** □ Display Options □ Display pin type (m/f)DISPLAY\_MIL\_STANDARD

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetDisplayMILStandard()

e3Job.SetDisplayMILStandard()

**MIL-Standard** □ Display Options □ Hide mating connector pin  
namesDISPLAY\_MIL\_STANDARD

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetDisplayMILStandard()

e3Job.SetDisplayMILStandard()

**MIL-Standard** □ Place Block Connectors □ on the block's  
outlineDISPLAY\_MIL\_STANDARD

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetDisplayMILStandard()

e3Job.SetDisplayMILStandard()

**MIL-Standard** □ Place Block Connectors □ outside blockDISPLAY\_MIL\_STANDARD

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetDisplayMILStandard()

e3Job.SetDisplayMILStandard()

**MIL-Standard** □ Place Block Connectors □ inside blockDISPLAY\_MIL\_STANDARD

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetDisplayMILStandard()

e3Job.SetDisplayMILStandard()

**MIL-Standard** □ Assembly □ Treat connectors of assembly as  
unitDISPLAY\_MIL\_STANDARD

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetDisplayMILStandard()

e3Job.SetDisplayMILStandard()

**MIL-Standard** □ Assembly □ Ignore view  
assignmentMIL\_DISPLAY\_CONNECTOR\_IGNORE\_VIEWS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetDisplayMILStandard()

e3Job.SetDisplayMILStandard()

**MIL-Standard** □ Backshell Pins □ Treat backshell pins as unitDISPLAY\_MIL\_STANDARD

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetDisplayMILStandard()

e3Job.SetDisplayMILStandard()

**MIL-Standard** □ Graphic □ Use line style from MIL symbol

DISPLAY\_MIL\_STANDARD

USE\_MIL\_LINE\_STYLE\_FROM\_SYMBOL

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetDisplayMILStandard()

e3Job.SetDisplayMILStandard()

**MIL-Standard** □ Graphic □ WidthMIL\_GRADIA

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

e3Job.GetMILGraphicLineWidth()

e3Job.SetMILGraphicLineWidth()

**MIL-Standard** □ Graphic □ ColorMIL\_GRACOD

e3Job.GetSettingValue()

- [e3Job.SetSettingValue\(\)](#)

[e3Job.GetMILGraphicLineColour\(\)](#)

[e3Job.SetMILGraphicLineColour\(\)](#)

**MIL-Standard** □ Graphic □ Line StyleMIL\_GRAMOD

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

[e3Job.GetMILGraphicLineStyle\(\)](#)

[e3Job.SetMILGraphicLineStyle\(\)](#)

**Functional Design** □ Options □ Show graphical representation of functional unitsFU\_SHOW\_ATTRIBUTE\_TEMPLATE

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Functional Design** □ Options □ Display logic lines on block connectionsFD\_SIGNAL\_AIRLINES

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Functional Design** □ Display □ Fit color for assigned installation spacesASSIGNED\_FUBLOCK\_FILL\_COLOR

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Functional Design** □ Display □ Fit color for assigned disconnecting pointsASSIGNED\_FUDPOINT\_FILL\_COLOR

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Functional Design** □ Block Connection Symbols □ BidirectionalFD\_BLNODE\_BI

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Functional Design** □ Block Connection Symbols □ InFD\_BLNODE\_IN

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Functional Design** □ Block Connection Symbols □ OutFD\_BLNODE\_OUT



e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Functional Design** □ Block Connection Symbols □ GNDFD\_BLNODE\_GND

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Functional Design** □ Block Connection Symbols □ PlusFD\_BLNODE\_PLUS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Functional Design** □ Block Connection Symbols □ MinusFD\_BLNODE\_MINUS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Functional Design** □ Block Connection Symbols □ UndeterminedFD\_BLNODE\_UNDETERMINED

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Functional Design** □ Symbols □ Topology symbolTOPOLOGY\_SYMBOL

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Functional Design** □ Symbols □ Installation space symbolDEFAULT\_INSTALLATION\_SPACE\_SYMBOL

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Functional Design** □ Shared sheets □ Display region overviewDISPLAY\_TOPOLOGY\_REGION

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Functional Design** □ Shared sheets □ Scaling increment of regionTOPOLOGY\_REGION\_SCALE\_STEP

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Topology** □ Options □ Show graphical representation of functional unitsFU\_SHOW\_ATTRIBUTE\_TEMPLATE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Topology** □ Options □ Display logic lines on block connectionsFD\_SIGNAL\_AIRLINES

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Topology** □ Display □ Fit color for assigned installation spacesASSIGNED\_FUBLOCK\_FILL\_COLOR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Topology** □ Display □ Fit color for assigned disconnecting pointsASSIGNED\_FUDPOINT\_FILL\_COLOR

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Topology** □ Block Connection Symbols □ BidirectionalFD\_BLNODE\_BI

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Topology** □ Block Connection Symbols □ InFD\_BLNODE\_IN

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Topology** □ Block Connection Symbols □ OutFD\_BLNODE\_OUT

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Topology** □ Block Connection Symbols □ GNDFD\_BLNODE\_GND

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Topology** □ Block Connection Symbols □ PlusFD\_BLNODE\_PLUS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Topology** □ Block Connection Symbols □ MinusFD\_BLNODE\_MINUS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Topology** □ Block Connection Symbols □ UndeterminedFD\_BLNODE\_UNDETERMINED

e3Job.GetSettingValue()

- [e3Job.SetSettingValue\(\)](#)

**Topology** □ Symbols □ Topology symbol `TOPOLOGY_SYMBOL`

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Topology** □ Symbols □ Installation space  
symbol `DEFAULT_INSTALLATION_SPACE_SYMBOL`

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Topology** □ Shared sheets □ Display region overview `DISPLAY_TOPOLOGY_REGION`

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Topology** □ Shared sheets □ Scaling increment of  
region `TOPOLOGY_REGION_SCALE_STEP`

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Electrical Checks** □ General □ Activate electrical checks `ECHECK_CHECKS`

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Electrical Checks** □ General □ Activate signal on error `ECHECK_SIGNAL_ON_ERROR`

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Electrical Checks** □ General □ Display current flow `ECHECK_DISP_CUR_FLOW`

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Electrical Checks** □ General □ Add attributes to new net segments and update physical  
length of used conductor/wire `ECHECK_ATTR_TO_NEW_CONN`

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Electrical Checks** □ General □ Check fuse melting time and wire ignition  
time `ECHECK_FUSE_MELTING_WIRE_IGNITION` [e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Electrical Checks** □ Default Colors □ Load active color `ECHECK_LOAD_COLOUR`

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Electrical Checks** □ Default Colors □ Coil active colorECHECK\_COIL\_COLOUR

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Electrical Checks** □ Default Colors □ LED active colorECHECK\_LED\_COLOUR

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Electrical Checks** □ Default Colors □ Fuse blown colorECHECK\_FUSE\_COLOUR

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Electrical Checks** □ Default Colors □ All other error colorECHECK\_ALL\_COLOUR

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Electrical Checks** □ Default Values □ Wire lengthECHECK\_WIRE\_LENGTH

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Electrical Checks** □ Default Values □ Wire cross-sectionECHECK\_WIRE\_CROSSECTION

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Electrical Checks** □ Default Values □ Wire colorECHECK\_WIRE\_COLOUR

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Electrical Checks** □ Default Values □ Wire weight [kg/km]ECHECK\_WIRE\_WEIGHT

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Electrical Checks** □ Default Values □ Wire specific resistance [Ohm x mm<sup>2</sup>/m]ECHECK\_WIRE\_SPEC\_RESISTANCE

[e3Job.GetSettingValue\(\)](#)

- [e3Job.SetSettingValue\(\)](#)

**Electrical Checks** □ Default Values □ Ambient temperature [°C]ECHECK\_AMB\_TEMP

[e3Job.GetSettingValue\(\)](#)

- e3Job.SetSettingValue()

**Electrical Checks** □ Default Values □ Fuse derating factor  
[%]ECHECK\_FUSE\_DER\_FACT

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Electrical Checks** □ Default Values □ Diode voltage drop  
[V]ECHECK\_DIODE\_VOLT\_DROP

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Electrical Checks** □ Default Values □ internal resistance of connect elements  
[Ohm]ECHECK\_INTERNAL\_RESISTANCE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Auto Routing** □ Router Mode □ Horizontally after  
placingAUTOROUTE\_AFTER\_PLACE\_HORIZONTAL

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Auto Routing** □ Router Mode □ Vertically after  
placingAUTOROUTE\_AFTER\_PLACE\_VERTICAL

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Auto Routing** □ Router Mode □ After movingAUTOROUTE\_AFTER\_MOVE

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Auto Routing** □ Router Settings □ Crossing □ Allow crossing  
connectionsAUTOROUTE\_ALLOW\_CROSS\_CONNECTIONS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Auto Routing** □ Router Settings □ Crossing □ Allow crossing connections □  
CostAUTOROUTE\_COST\_CROSS\_CONNECTIONS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Auto Routing** □ Router Settings □ Crossing □ Allow crossing  
symbolsAUTOROUTE\_ALLOW\_CROSS\_SYMBOLS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Auto Routing** □ Router Settings □ Crossing □ Allow crossing symbols □  
CostAUTOROUTE\_COST\_CROSS\_SYMBOLS

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Auto Routing** □ Router Settings □ Minimum Distance to First Turn □ Top [Grid size x  
value]AUTOROUTE\_BEND\_DISTANCE\_FROM\_TOP

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Auto Routing** □ Router Settings □ Minimum Distance to First Turn □ Bottom [Grid size x  
value]AUTOROUTE\_BEND\_DISTANCE\_FROM\_BOTTOM

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Auto Routing** □ Router Settings □ Minimum Distance to First Turn □ Left [Grid size x  
value]AUTOROUTE\_BEND\_DISTANCE\_FROM\_LEFT

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Auto Routing** □ Router Settings □ Minimum Distance to First Turn □ Right [Grid size x  
value]AUTOROUTE\_BEND\_DISTANCE\_FROM\_RIGHT

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Auto Routing** □ Router Settings □ Minimum Distance to First Turn □ Direction change  
costAUTOROUTE\_BEND\_COST

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

**Auto Routing** □ Router Settings □ Sort □ Sort  
connectionsAUTOROUTE\_CONNECTION\_LENGTH

e3Job.GetSettingValue()

- e3Job.SetSettingValue()

If no setting functions are shown for a **E<sup>3</sup>.series** setting location, no functions are  
available and the setting value can only be retrieved or assigned using **E<sup>3</sup>.series**  
interactively.

## See Also

- Fluid Project Settings
  - Setting Value Names
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

## Fluid Settings

# Syntax

*String name*

## Description

Parameter represents a unique name identifying the **E<sup>3</sup>** setting value.

## Possible Values

If the location of the setting within **E<sup>3</sup>.series** is known and the setting name is required, please reference the following table. If the setting value name is known and the location of the setting within **E<sup>3</sup>.series** is required, please refer to Setting Value Names. For location of values in Electrical Settings see Electric Project Settings.

Location in <b>E<sup>3</sup>.series</b>	Setting Value Name
Electrical Settings	

<b>General</b> □ Working Grid □	FLUIDGRIDSIZE
Grid size	

<b>General</b> □ Working Grid □	FLUIDTRAPSIZE
Snap size	

<b>General</b> □ Alternative	FLUIDALTGRIDSIZE
Grid □ Grid size	



**General** ☐ Measurement Units ☐ Millimeters MEA\_EXTERN\_SCHEMA

**General** ☐ Measurement Units ☐ Inches MEA\_EXTERN\_SCHEMA

**General** ☐ Save ☐ Automatically generate backup file after ( ☐ CheckBox ) AUTOSAVE\_ENABLED

**General** ☐ Save ☐ Automatically generate backup file after ( ☐ SpinControl ) SAVLIMIT

**General** ☐ Save ☐ Write messages and result to file

**General** ☐ Template ☐ File name

**General** ☐ Align Distances ☐ Horizontal ALIGN\_HORIZONTAL\_DIST

**General** ☐ Align Distances ☐ Vertical ALIGN\_VERTICAL\_DIST

**General** ☐ Snap Size ☐ Snap size

**General** ☐ **Display** ☐ Grid view ☐ Points ( ☐ CheckBox ) MODE\_GRID\_OVERLAY

SCHEMAOVERSIZE

**General** ▯ **Display** ▯ Grid

view ▯ Points ( SpinControl )

**General** ▯ **Display** ▯ GridPANEL\_MODE\_GRID\_AXIS

view ▯ Rulers ( CheckBox ) SCHEMA\_MODE\_GRID\_AXIS

**General** ▯ **Display** ▯ Grid

view ▯ Rulers ( SpinControl ) SCHEMAAXISGRID

**General** ▯ **Display** ▯ SheetPANEL\_MODE\_GRID\_SHEETLAYOUT

Reference ▯ Show sheet layout SCHEMA\_MODE\_GRID\_SHEETLAYOUT

**General** ▯ **Display** ▯ Sheet SHEETREF\_FORMAT

Reference ▯ Format

**General** ▯ **Display** ▯

Symbol Options ▯ Alternative text as component code ( CheckBox ) ALT\_COMPCODE\_ON

**General** ▯ **Display** ▯

Symbol Options ▯ Alternative text as component code ( ComboBox ) ALT\_COMPCODE

**General** ▯ **Display** ▯

Symbol Options ▯ Number of view as an extension to the device's device designation DUPLICATE\_DISPLAY\_OFF

**General** ▯ **Display** ▯ DISPLAY\_OPEN\_PINS

Symbol Options ☐ Mark  
unconnected nodes

**General ☐ Display ☐**

Symbol Options ☐ Internal  
/ external / jumper / seal FLUID\_DISPLAY\_IE\_REPRESENTATION  
representation for nodes

**General ☐ Display ☐**

Symbol Options ☐ Add  
internal device DEVICE\_DESIGNATION\_OF\_CONNECTION\_TARGET  
designation to  
connection target

**General ☐ Display ☐ Mark**

connect point in  
connection nets ☐ DISPLAY\_OPEN\_NODES  
T-Connections

**General ☐ Display ☐ Mark**

connect point in  
connection nets ☐ Forced DISPLAY\_OPEN\_NODES  
wiring

**General ☐ Display ☐ Mark**

connect point in  
connection nets ☐ Open DISPLAY\_OPEN\_NODES  
line end

**General ☐ Display ☐**

Miscellaneous Options ☐  
Show Tooltips

**General ☐ Display ☐**

Miscellaneous Options ☐  
Show Copilot

**General ☐ Display ☐**

Miscellaneous Options ☐  
Allow window KEEP\_LINECOLOR  
background color as  
display color

DISPLAY\_STANDARD\_ROTATED\_TEXTS

**General** □ **Display** □  
 Miscellaneous Options □  
 Display rotated texts  
 acc. to standard

**General** □ **Display** □  
 Miscellaneous Options □ FIT\_TEXT  
 Resize text to fit text box

**General** □ **Display** □  
 Miscellaneous Options □ ENLARGE\_GRID\_POINTS  
 Enlarge grid points when  
 zooming

**General** □ **Display** □  
 Miscellaneous Options □ DISPLAY\_PREVIEW\_SYMBOL\_FOR\_SELECTED\_COMPONENT  
 Display preview symbol  
 for selected component

**General** □ **Display** □  
 Miscellaneous Options □ DISPLAY\_MINIMISE\_DETAILS  
 Minimize details

**General** □ **Display** □  
 Miscellaneous Options □  
 Invert display color

**General** □ **Display** □ Suffix  
 Modification □ Suffix SUFFIX\_MODIFICATION\_IS\_ACTIVE  
 modification is active

**General** □ **Display** □ Suffix  
 Modification □ Higher SEPARATOR\_SUFFIX\_MODIFICATION\_ASSIGNMENT  
 level assignment

**General** □ **Display** □ Suffix  
 Modification □ Location SEPARATOR\_SUFFIX\_MODIFICATION\_LOCATION

**General** □ **Display** □ Suffix  
 Modification □ Device SEPARATOR\_SUFFIX\_MODIFICATION\_DEVDES  
 designation

**General □ Highlight □**

Search □ Highlight found DO\_HIGHLIGHT  
objects when searching

**General □ Highlight □**

Jump □ Keep existing KEEP\_HIGHLIGHT  
highlights when jumping

**General □ Highlight □**

Jump □ Zoom faction for HIGHLIGHT\_JUMP\_ZOOM\_RATIO  
'Jump' (%)

**General □ Highlight □**

Highlight □ Color HIGHLIGHT\_COLOR

**General □ Highlight □**

Highlight □ Width ACTUAL\_HIGHLIGHT\_WIDTH

**General □ Highlight □**

Text Hyperlink □ Use ENABLE\_HYPERLINK\_DISPLAY  
following property

**General □ Highlight □**

Text Hyperlink □ HYPERLINK\_UNDERLINE\_MODE  
Underline hyperlinks □  
when hovering

**General □ Highlight □**

Text Hyperlink □ HYPERLINK\_UNDERLINE\_MODE  
Underline hyperlinks □  
always

**General □ Highlight □**

Text Hyperlink □ HYPERLINK\_UNDERLINE\_MODE  
Underline hyperlinks □  
never

**General □ Highlight □**

Text Hyperlink □ Color HYPERLINK\_COLOUR

**General** □ **Verify** □ Level □  
Release

**General** □ **Verify** □ Level □  
Development

**General** □ **Verify** □ Level □  
Draft

**General** □ **Verify** □ Use  
Verification XML file      VERIFY\_USE\_XML\_FILE

**General** □ **Language** □ 1<sup>st</sup>  
Language      LANGUAGES

**General** □ **Language** □  
2nd Language      LANGUAGES

**General** □ **Language** □  
3rd Language      LANGUAGES

**General** □ **Language** □ 4<sup>th</sup>  
Language      LANGUAGES

**General** □ **Language** □ 5<sup>th</sup>  
Language      LANGUAGES

**General** □ **Language** □      PICTOGRAM\_LANGUAGE  
Pictograms

**General □ Language □**

Language Database

**General □ Language**

Table Schema

**General □ Update in****Project □ Assignment □**Prefer matching  
symbols, conductors and  
pins

UIP\_ASSIGN\_GATE\_MODE

**General □ Update in****Project □ Assignment □**Prefer order of symbol,  
conductor and pin

UIP\_ASSIGN\_GATE\_MODE

**General □ Update in****Project □ Attributes □**Overwrite attribute  
values for devices and  
symbols

RELOAD\_ATTRIBUTES

**General □ Update in****Project □ Attributes □**Delete unused attributes  
for devices and symbols

DELETE\_UNUSED\_ATTRIBUTES\_DURING\_UPDATE

**General □ Update in****Project □ Signals □**Overwrite signals of  
block connectors

RELOAD\_SIGNALS

**General □ Update in****Project □ Text**Parameters □ Keep text  
visibility for □ Symbols

KEEP\_TEXT\_VISIBILITY

**General □ Update in****Project □ Text**Parameters □ Keep text  
visibility for □ Models

KEEP\_MODELTEXT\_VISIBILITY

**General □ Update in****Project □ Text**Parameters □ Keep other  
text parameters for □

KEEP\_TEXT\_PARAMETER

## Symbols

**General** □ **Update in****Project** □ TextParameters □ Keep other KEEP\_MODELTEXT\_PARAMETER  
text parameters for □  
Models**General** □ **Update in****Project** □ Pins □ Restore RESTORE\_PINNAMES  
changed pin names**General** □ **Update in****Project** □ Pins □ Restore UPDATE\_RESTORE\_PHYSICAL\_PIN\_DATA  
changed physical pin  
data**General** □ **Update in****Project** □ Pins □ Restore UPDATE\_RESTORE\_LOGICAL\_PIN\_DATA  
changed logical pin data**General** □ **Update in****Project** □ Pins □ Keep KEEP\_PINVIEW\_SYMBOLS\_OF\_DEVICES  
preview symbols of  
devices**General** □ **Update in****Project** □ Fitting □ Keep KEEP\_CONNECTOR\_SYMBOLS  
fitting symbol**General** □ **Update in****Project** □ Fitting □ Keep KEEP\_COUNTERPARTS  
active mating and fitting  
parts



**General □ Update in****Project □ Fitting □ Change**

already used mating part CHANGE\_COMPOSITES\_COMPONENTS  
to the new active mating  
part

**General □ Update in****Project □ Fitting □ Place** UPDATE\_PLACE\_SINGLE\_PINS

all pins as single pins

**General □ Update in****Project □ Subcircuit □**

Attributes □ Overwrite SUBCIRCUIT\_RELOAD\_ATTRIBUTES  
attribute values

**General □ Update in****Project □ Subcircuit □**

Text Parameters □ Keep SUBCIRCUIT\_KEEP\_TEXT\_VISIBILITY  
text visibility for □  
Symbols

**General □ Update in****Project □ Subcircuit □**

Text Parameters □ Keep SUBCIRCUIT\_KEEP\_MODELTEXT\_VISIBILITY  
text visibility for □ Models

**General □ Update in****Project □ Subcircuit □**

Text Parameters □ Keep SUBCIRCUIT\_KEEP\_TEXT\_PARAMETER  
other text parameters  
for □ Symbols

**General □ Update in****Project □ Subcircuit □**

Text Parameters □ Keep SUBCIRCUIT\_KEEP\_MODELTEXT\_PARAMETER  
other text parameters  
for □ Models

**General □ Default****Directories □ ( Table )****General □ Purge □ Objects**

□ Unused devices UNUSED\_DEVICES

**General □ Purge □ Objects**

□ Unused devices from UNUSED\_DEVICES\_FROM\_ASSEMBLIES  
assemblies

**General □ Purge □ Objects**

□ Unused plugged devices PURGE\_UNUSED\_PLUGGED\_DEVICES

**General □ Purge □ Objects**

□ Unused connected PURGE\_UNUSED\_CONNECTED\_DEVICES  
devices

**General** ☐ **Purge** ☐ Objects  
☐ Unused hoses / tubes **UNUSED\_CABLES**

**General** ☐ **Purge** ☐ Objects  
☐ Unused block devices **UNUSED\_BLOCK\_DEVICES**

**General** ☐ **Purge** ☐ Objects  
☐ Unused components **UNUSED\_COMPONENTS**

**General** ☐ **Purge** ☐ Objects  
☐ Unused hose / tube types **UNUSED\_CABLE\_TYPES**

**General** ☐ **Purge** ☐ Objects  
☐ Unused symbol types **UNUSED\_SYMBOL\_TYPES**

**General** ☐ **Purge** ☐ Objects  
☐ Unused signals **UNUSED\_SIGNALS**

**General** ☐ **Purge** ☐ Objects  
☐ Unused attribute names **UNUSED\_ATTRIBUTE\_NAMES**

**General** ☐ **Purge** ☐ Objects  
☐ Unused pin views **UNPLACED\_PIN\_VIEWS**

**General** ☐ **Purge** ☐ Objects  
☐ All STEP models **PURGE\_ALL\_STEP\_MODELS**

**General** ☐ **Purge** ☐ Objects  
☐ Unused Groups **UNUSED\_GROUPS**

**General** ☐ **Purge** ☐ Objects  
☐ No longer attribute values

**General** ☐ **Purge** ☐ Project  
☐ Purge unused objects from project before saving **CLEANUP\_WHEN\_SAVING**

**General** ☐ **Zoom / Pan / Selection** ☐ Zoom In/Out ☒ **Zoom Factor**  
Ratio (%) **ZOOM\_FACTOR**

**General** ☐ **Zoom / Pan / Selection** ☐ Pan ☐ Enable **SCRL\_ENABLE\_ARROWKEYS**

panning with arrow keys

**General □ Zoom / Pan /**

**Selection** □ Pan □ Ratio SCRL\_FACTOR  
(%)

**General □ Zoom / Pan /**

**Selection** □ Area

Selection □ Select all  
elements inside and SELECTION\_BORDER  
intersecting or touching  
the border of the  
selection rectangle

**General □ Zoom / Pan /**

**Selection** □ Area

Selection □ Included SELECTION\_SYMBOL  
Elements in Area  
Selection □ Symbols

**General □ Zoom / Pan /**

**Selection** □ Area

Selection □ Included SELECTION\_TEXT  
Elements in Area  
Selection □ Texts

**General □ Zoom / Pan /**

**Selection** □ Area

Selection □ Included SELECTION\_GRAPHIC  
Elements in Area  
Selection □ Graphics

**General □ Zoom / Pan /**

**Selection** □ Area

Selection □ Included SELECTION\_NETSEG  
Elements in Area  
Selection □ Connect Lines

**General □ Zoom / Pan /**

**Selection** □ Area

Selection □ Included SELECTION\_NETNODE  
Elements in Area  
Selection □ Net Nodes

**General □ Zoom / Pan /**

**Selection** □ Area

Selection □ Included SELECTION\_ATTRIBUTE  
Elements in Area  
Selection □ Attribute Text  
Templates

**General □ Zoom / Pan / SELECTION\_DIMENSION**

**Selection** □ Area

Selection □ Included  
Elements in Area  
Selection □ Dimensions

**General** □ **Locking** □

Password for unlocking  
objects □ Old password

**General** □ **Locking** □

Password for unlocking  
objects □ New password

**General** □ **Locking** □

Password for unlocking  
objects □ Confirm  
password

**General** □ **Locking** □

Display □ Mark locked objects MARK\_LOCKED\_OBJECTS

**General** □ **Component****Type Attributes** □

Selected Attributes COMPONENT\_TYPE\_ATTRIBUTES

**Connection** □

Autoconnect □ Allow  
inserting symbol in  
connection ALLOW\_INSERT\_SYMBOL\_IN\_CONNECTION

**Connection** □

Autoconnect □ Keep signal for all connections RETAIN\_SIGNAL\_CONNECT\_CELL

**Connection** □

Autoconnect □ Reconnect after deleting symbols AUTOCON\_LINES

**Connection** □

Autoconnect □ Preferred  
Direction □ Vertical connections (top to bottom) AUTOCON\_DIR

**Connection** □

Autoconnect □ Preferred  
Direction □ Horizontal connections (left to right) AUTOCON\_DIR

**Connection** □ Net □ Allow  
net loops

NETLOOPS\_ALLOWED

**Connection** □ Net □

Specific functionality of  
inheriting net number INHERIT\_NET\_NUMBER

**Connection** □

Hoses/Tubes □ UNCONNECT\_CORES  
Unconnect, if conductor  
is not routed in a

connect line

**Connection** □

Hoses/Tubes □

Unconnect, if conductor is not routed in a connect line □ Check view and original connections UNCONNECT\_CORES

**Connection** □

Hoses/Tubes □

Unconnect, if conductor is not routed in a connect line □ Check only original connections UNCONNECT\_CORES

**Connection** □

Hoses/Tubes □ Connect

hoses/tubes at one end

to source

cross-reference if

cross-reference is

unconnected

KEEP\_CORE\_AFTER\_UNCONNECT\_REFERENCE

**Connection** □

Hoses/Tubes □ Lock new LOCK\_WIRES

wire pathways

**Connection** □

Hoses/Tubes □ Check

pneumatic/hydraulic pins

CONNECT\_WIRE\_FLUID\_CHECK

**Connection** □

Hoses/Tubes □ Clear

signal after unconnect

hose/tube at

unconnected Pin

CLEAR\_SIGNAL\_AT\_PIN\_AFTER\_UNCONNECT\_CORE

**Connection** □ Template

Symbol □ Horizontal

WCOUNT\_SYMBOL\_HOR

**Connection** □ Template

Symbol □ Vertical

WCOUNT\_SYMBOL\_VER

**Connection** □ Alternative

Template Symbol □

Horizontal

WCOUNT\_SYMBOL\_HOR\_ALT

**Connection** □ Alternative

Template Symbol □

Vertical

WCOUNT\_SYMBOL\_VER\_ALT

**Connection** □ Connect

and use Hose/Tube □

connect only graphically

CONNECT\_AND\_USE\_DEFAULT\_HOSE

**Connection** □ Connect  
and use Hose/Tube □  
connect and use default  
wire CONNECT\_AND\_USE\_DEFAULT\_HOSE

**Connection** □ Pins □ Keep  
plug after unplugging KEEP\_PLUG\_AFTER\_UNPLUGGING\_PINS  
pins

**Connection** □ Pins □ Deny  
plugging pins of same DENY\_PLUG\_PINS\_OF\_SAME\_DEVICE  
device

**Connection** □ Pins □ Allow  
only valid mating ALLOW\_ONLY\_VALID\_MATING\_CONNECTORS\_TO\_PLUG  
connection to plug

**Connection** □ Pins □ Allow  
only compatible pin ALLOW\_ONLY\_COMPATIBLE\_PIN\_GENDERS\_TO\_PLUG  
genders to plug

**Connection** □ Pins □ Deny  
plugging pin with DENY\_PLUG\_PINS\_WITH\_DIFF\_PINNAMES  
different pin names

**Connection** □ **Connect**  
**Lines** □ Lines □ Width LINDIA

**Connection** □ **Connect** LINMOD  
**Lines** □ Lines □ Style

**Connection** □ **Connect**  
**Lines** □ Lines □ Color      LINCOD

**Connection** □ **Connect**  
**Lines** □ Lines □ Level      LINLEV

**Connection** □ **Connect**  
**Lines** □ Lines □ Use  
properties of starting      USE\_LINE\_PROPERTIES\_OF\_START\_LINE  
line

**Connection** □ **Connect**  
**Lines** □ Angled  
Connections □ Connection      LIN\_FOLD\_ANGLE  
angle

**Connection** □ **Connect**  
**Lines** □ Angled  
Connections □ Offset      LIN\_FOLD\_DISTANCE  
distance

**Connection** □ **Connect**      DELETE\_SIGNAL\_ON\_DEL\_CLINE  
**Lines** □ Delete with  
graphical representation  
□ Signal

**Connection** ☐ **Connect**

**Lines** ☐ Delete with graphical representation UNCONNECT\_CORES\_ON\_DEL\_CLINE  
☐ Hose/Tube

**Connection** ☐ **Connect**

**Lines** ☐ Delete with graphical representation DELETE\_CORES\_ON\_DEL\_CLINE  
☐ Hose/Tube from project

**Connection** ☐ **Connect**

**Lines** ☐ Template Symbol ☐ CONNECT\_LINE\_SYMBOL\_HOR  
 Horizontal

**Connection** ☐ **Connect**

**Lines** ☐ Template Symbol ☐ CONNECT\_LINE\_SYMBOL\_VER  
 Vertical

**Connection** ☐

**References Style** ☐ Font REFFONTPTR

**Connection** ☐

**References Style** ☐ Font REF DIA  
 Style

**Connection** ☐

**References Style** ☐ Size REFSIZ

**Connection** ☐

**References Style** ☐ Color REFCOD

**Connection** ☐

**References Style** ☐ REFOFX  
 Display Control ☐ X-Offset

**Connection** ☐

**References Style** ☐ REFOFY  
 Display Control ☐ Y-Offset

**Connection** ☐

**References Style** ☐ REFGAP  
 Display Control ☐ Gap

**Connection** ☐

**References Style** ☐ Ratio REFMOD  
☐ Normal

REFMOD



**Connection** ☐**References Style** ☐ Ratio☐ Narrow**Connection** ☐**References Style** ☐ RatioREFMOD☐ Wide**Connection** ☐**References Style** ☐ REFDIRDirection ☐ Symmetrical**Connection** ☐**References Style** ☐ REFDIRDirection ☐ Up**Connection** ☐**References Style** ☐ REFDIRDirection ☐ Down**Connection** ☐**References Style** ☐ REFCHANGETYPEFunction Control ☐ Allow  
change reference type**Connection** ☐**References Format** ☐ SHEETREFSETTING

Prefix

**Connection** ☐**References Format** ☐ SHEETREFSETTING

Suffix

**Connection** ☐**References Format** ☐ SHEETREFSETTING

Sheet text

**Connection** ☐**References Format** ☐ SHEETREFSETTING

Reference text

**Connection** ☐**References Format** ☐ REFERENCE\_BETWEEN\_PLUGGED\_PINSDisplay references with  
logically plugged devices**Connection** ☐ **Signal****Logic Lines** ☐ Display  
open signal connections  
( CheckBox ) AIRLINE\_DISPLAY\_OPEN\_SIGNAL\_CONNECTIONS**Connection** ☐ **Signal****Logic Lines** ☐ Display  
open signal connections  
( ComboBox ) AIRLINES\_LINMOD\_SIG**Connection** ☐ **Signal****Logic Lines** ☐ Display  
signal flags DISPLAY\_SIGNAL\_FLAGS

**Connection** ☐ **Signal****Logic Lines** ☐ Mark

connect nodes defined as  
not connected (\*\*NC\*\*) NOCONN\_DISPLAY\_OFF

**Connection** ☐**Hose/Tube Logic Lines**

☐ Display ☐ Display

unconnected hoses/tubes AIRLINES

**Connection** ☐**Hose/Tube Logic Lines**

☐ Display ☐ Display laid

hoses/tubes also AIRLINES

**Connection** ☐**Hose/Tube Logic Lines**

☐ Lines ☐ Line style

AIRLINES\_LINMOD\_CAB  
AIRLINES

**Connection** ☐**Hose/Tube Logic Lines**

☐ Lines ☐ Display hose/tube

logic lines as arcs

**Connection** ☐**Hose/Tube Logic Lines**

☐ Lines ☐ Mark direction AIRLINES

**Connection** ☐**Hose/Tube Logic Lines**

☐ Lines ☐ Only for Views ☐

Used views only

AIRLINE\_USED\_VIEWS\_ONLY

AIRLINES

**Connection** ☐**Hose/Tube Logic Lines**

☐ Lines ☐ Only for Views ☐

(Table)

**Connection** ☐**Hose/Tube Logic Lines**

☐ Lines ☐ Show name of

hose/tube

AIRLINE\_SHOW\_NAME\_OF\_CORE

AIRLINES

**Connection** ☐**Hose/Tube Logic Lines**

☐ Lines ☐ Show name of

hose/tube ☐ Position ☐ to

endpoints

AIRLINE\_SHOW\_NAME\_OF\_CORE\_POSITION\_CENTERED

AIRLINES

**Connection** ☐**Hose/Tube Logic Lines**

☐ Lines ☐ Show name of

hose/tube ☐ Position ☐

centered

AIRLINE\_SHOW\_NAME\_OF\_CORE\_POSITION\_CENTERED

AIRLINES

AIRLINES\_OFFX\_CABCAB

**Connection** □**Hose/Tube Logic Lines**

□ Lines □ Show name of  
hose/tube □ X-Offset

**Connection** □**Hose/Tube Logic Lines**

□ Lines □ Show name of AIRLINES\_OFFY\_CABCAB  
hose/tube □ Y-Offset

**Connection** □**Hose/Tube Logic Lines**

□ Lines □ Show name of AIRLINES\_TXTFNT\_CABCAB  
hose/tube □ **Font** □ Font

**Connection** □**Hose/Tube Logic Lines**

□ Lines □ Show name of AIRLINE\_SHOW\_NAME\_OF\_CORE\_FONT\_BOLD  
hose/tube □ **Font** □ Font  
Style AIRLINES

**Connection** □**Hose/Tube Logic Lines**

□ Lines □ Show name of AIRLINE\_TXTSIZ\_CABCAB  
hose/tube □ **Font** □ Size

**Connection** □**Hose/Tube Logic Lines**

□ Lines □ Show name of AIRLINES\_TXTMOD\_CABCAB  
hose/tube □ **Font** □ Ratio  
□ Normal

**Connection** □**Hose/Tube Logic Lines**

□ Lines □ Show name of AIRLINES\_TXTMOD\_CABCAB  
hose/tube □ **Font** □ Ratio □  
Narrow

**Connection** □**Hose/Tube Logic Lines**

□ Lines □ Show name of AIRLINES\_TXTMOD\_CABCAB  
conductor □ **Font** □ Ratio □  
Wide

**Connection** □**Hose/Tube Logic Lines**

□ Lines □ Show signal AIRLINE\_SHOW\_SIGNAL\_NAME  
name AIRLINES

**Connection** □**Hose/Tube Logic Lines**

AIRLINE\_SHOW\_SIGNAL\_NAME\_POSITION\_CENTERED  
□ Lines □ Show signal  
name □ Position □ to AIRLINES  
endpoints

**Connection** □**Hose/Tube Logic Lines**

AIRLINE\_SHOW\_SIGNAL\_NAME\_POSITION\_CENTERED

☐ Lines ☐ Show signal AIRLINES  
name ☐ Position ☐ centered

**Connection ☐**

**Hose/Tube Logic Lines** AIRLINES\_OFFFX\_CABSIG  
☐ Lines ☐ Show signal  
name ☐ X-Offset

**Connection ☐**

**Hose/Tube Logic Lines** AIRLINES\_OFFFY\_CABSIG  
☐ Lines ☐ Show signal  
name ☐ Y-Offset

**Connection ☐**

**Hose/Tube Logic Lines** AIRLINES\_TXTFNT\_CABSIG  
☐ Lines ☐ Show signal  
name ☐ **Font** ☐ Font

**Connection ☐** AIRLINE\_SHOW\_SIGNAL\_NAME\_FONT\_BOLD  
**Hose/Tube Logic Lines** AIRLINE\_SHOW\_SIGNAL\_NAME\_FONT\_ITALIC  
☐ Lines ☐ Show signal  
name ☐ **Font** ☐ Font Style AIRLINES

**Connection ☐**

**Hose/Tube Logic Lines**  
☐ Lines ☐ Show signal  
name ☐ **Font** ☐ Size

**Connection ☐**

**Hose/Tube Logic Lines**  
☐ Lines ☐ Show signal  
name ☐ **Font** ☐ Ratio ☐  
Normal

**Connection ☐**

**Hose/Tube Logic Lines**  
☐ Lines ☐ Show signal  
name ☐ **Font** ☐ Ratio ☐  
Narrow

**Connection ☐**

**Hose/Tube Logic Lines**  
☐ Lines ☐ Show signal  
name ☐ **Font** ☐ Ratio ☐ Wide

**Connection ☐**

**Hose/Tube Logic Lines**  
☐ Hose/Tube logic lines of signal ☐ Keep existing hose/tube logic lines of signal KEEP\_EXISTING\_CORE\_LOGIC\_LINES\_OF\_SIGNAL

**Connection ☐** PANEL\_DEFAULT\_FLUID  
**Hoses/Tubes** ☐ Used  
Hose/Tube Type ☐  
Hose/Tube

**Connection □ Signals □**

Signals on Connections □  
 Create connections  
 which transfer signals

CONNECT\_SIGNAL

**Connection □ Signals □**

Signals on Connections □  
 Transfer signals on  
 connections between  
 views

CONNECT\_SIGNAL\_VIEWS

**Connection □ Signals □**

Signals on Symbols □  
 Clear signals when  
 deleting symbols

CLEAR\_SIG\_ON\_DEL\_SYMBOL

**Connection □ Signals □**

Signals on Symbols □  
 Even if not connected  
 view symbols are placed

CLEAR\_SIG\_ON\_PLACED\_VIEW\_SYMBOL

**Connection □ Signals □**

Signal Flow on Fittings □  
 Interrupt signal flow on  
 block fittings

CREATE\_BLCON\_WITH\_FLOW

**Connection □ Signals □**

Signal Flow on Fittings □  
 Interrupt signal flow on  
 fittings

CREATE\_CONN\_WITH\_FLOW

**Connection □ Signals □**

Signal Settings for Copy  
 and Import □ Keep  
 user-defined signals

MERGE\_COPY\_AREA\_KEEP\_USER

**Connection □ Signals □**

Signal Settings for Copy  
 and Import □ Keep  
 system-generated  
 signals

MERGE\_COPY\_AREA\_KEEP\_SYSTEM

**Connection □ Signals □**

Signal Classes □ Allow  
 signal changes to signals  
 not belonging to same  
 class

ALLOW\_SIGNAL\_CHANGES\_OF\_DIFFERENT\_CLASSES

**Connection □ Signals □**

Signal Format □  
 Recalculate signal  
 names according to  
 format specification

RECALC\_FORMATTED\_SIGNALS

**Connection** ☐ **Fitting** ☐

Pin Names ☐ Inherit pin names when connecting INHERIT\_PINNAMES

**Connection** ☐ **Fitting** ☐

Mating parts ☐ Use higher level assignment and location of placed devices USE\_ASSIGNMENT\_OF\_CONN

**Connection** ☐ **Fitting** ☐

Mating parts ☐ Try to assign pins via names first ASSIGN\_PINS\_VIA\_NAMES

**Connection** ☐ **Fitting** ☐

Mating parts ☐ Ignore pin attribute 'Internal Device Designation' when assigning pins IGNORE\_DOT\_CONN

**Connection** ☐ **Fitting** ☐

Mating parts ☐ Generate device designation of mating parts from device designation and pin attribute 'Internal Device Designation' USE\_DOT\_CONN\_NAME

**Connection** ☐ **Fitting** ☐

Mating parts ☐ Generate device designation of mating parts from device designation of block and device USE\_BLOCK\_NAME\_FOR\_DEVDES

**Connection** ☐ **Fitting** ☐

Mating parts ☐ Separator to use USE\_DOT\_CONN\_NAME\_SEPERATOR

**Connection** ☐ **Fitting** ☐

Mating parts ☐ Use same numeric part for fitting and mating part USE\_SAME\_NUMERIC\_PART\_FOR\_MATING\_CONNECTOR

**Connection** ☐ **Fitting** ☐

Mating parts ☐ Use automatic fitting naming USE\_ANSI\_STANDARD\_FOR\_MATING\_CONNECTOR

**Connection** ☐ **Fitting** ☐

Mating parts ☐ Default designation for fittings DEFAULT\_DESIGNATION\_FOR JACKS

**Connection** ☐ **Fitting** ☐

Mating parts ☐ Default DEFAULT\_DESIGNATION\_FOR\_PLUGS

designation for mating  
parts

**Connection** □

**Connection Target**

**Format** □ Text Type

**Connection** □

**Connection Target**

**Format** □ Prefix

**Connection** □

**Connection Target**

**Format** □ Suffix

**Connection** □

**Connection Target**

**Format** □ Use plugged  
device as target

**Connection** □

**Connection Target**

**Format** □ Multiline

**Connection** □

**Connection Target**

**Format** □ Number of view  
as an extension to the  
device's device  
designation

**Connection** □

**Connection Target**

**Format** □ (Table)

**Placement** □ Default  
Designations □ Higher  
level assignment ( Left  
EditBox )

SEPARATOR\_ASSIGNMENT

**Placement** □ Default  
Designations □ Higher  
level assignment ( Right  
EditBox )

HLA\_DEFAULT

**Placement** □ Default  
Designations □ Higher  
level assignment □ Used  
for unique designation

USE\_HLA\_FOR\_BMK

**Placement** □ Default  
Designations □ Location ( Left  
EditBox )

SEPARATOR\_LOCATION

**Placement** □ Default  
 Designations □ Location ( LOC\_DEFAULT  
 Right EditBox )

**Placement** □ Default  
 Designations □ Location □ USE\_LOC\_FOR\_BMK  
 Used for unique  
 designation

**Placement** □ Default  
 Designations □ Device  
 designation ( Left  
 EditBox ) SEPARATOR\_DEVDES

**Placement** □ Default  
 Designations □ Device  
 designation ( Right  
 EditBox )

**Placement** □ Default  
 Designations □ Device  
 designation □ Used for  
 unique designation USE\_DEVDES\_FOR\_BMK

**Placement** □ Default  
 Designations □ Blocks BLOCK\_DES\_DEFAULT

**Placement** □ Default  
 Designations □ Hose/Tube CABLE\_DES\_DEFAULT

**Placement** □ Default  
 Designations □ Text order KEEP\_BMK\_DIN\_ORDER  
 according to standard

**Placement** □ Rules □  
 Shorten higher level  
 assignment and location  
 as against sheet/field SHORT\_HLA\_AND\_LOC

**Placement** □ Rules □ Use  
 higher level assignment  
 and location of  
 sheet/field USE\_ASSIGNMENT\_OF\_SHEET

**Placement** □ Rules □  
 Rename devices when  
 changing the  
 designations for CHANGE\_DEVICES\_ON\_SHEET



sheet/field

**Placement** □ Rules □ Use  
 symbol pin attributes PLACE\_USE\_PIN\_ATTR  
 when assigning to device

**Placement** □ Rules □  
 Delete pin attributes PLACE\_DEL\_PIN\_ATTR  
 when symbol is unplaced

**Placement** □ Rules □  
 Delete symbol attributes PLACE\_DEL\_SYMBOL\_ATTR  
 when symbol is unplaced

**Placement** □ Block  
 Devices Options □ Use BLOCKNAME\_TO\_CONNECTOR  
 name of block for  
 devices as

**Placement** □ Block  
 Devices Options □ Use BLOCKNAME\_TO\_CONNECTOR  
 name of block for  
 devices as □ Higher level  
 assignment

**Placement** □ Block  
 Devices Options □ Use BLOCKNAME\_TO\_CONNECTOR  
 name of block for  
 devices as □ Location

**Placement** □ Block  
 Devices Options □ Allow ALLOW\_SAME\_CONDES\_ON\_BLOCKS  
 same device designation  
 on different blocks

**Placement** □ Block  
 Devices Options □ Use  
 name of block for  
 devices as

**Placement** □ IEC 81346  
 standard □ IEC 81346 is IEC\_81346\_IS\_ACTIVE  
 active

SEPARATOR\_IEC\_81346\_ASSIGNMENT

**Placement** □ IEC 81346  
standard □ Separators □  
Assignment

**Placement** □ IEC 81346  
standard □ Separators □ SEPARATOR\_IEC\_81346\_LOCATION  
Location

**Placement** □ IEC 81346  
standard □ Separators □ SEPARATOR\_IEC\_81346\_DEVDES  
Device designation

**Placement** □ IEC 81346  
standard □ Separators □ SEPARATOR\_IEC\_81346\_ATTRIBUTES  
Attributes

**Placement** □ IEC 81346  
standard □ Separators □ UNCUT\_PREFIX\_IEC\_81346  
Top-level prefix

**Placement** □ **Change**  
**Component** □ ASSIGN\_GATE\_MODE  
Assignment □ Prefer name  
of pins

**Placement** □ **Change**  
**Component** □  
Assignment □ Prefer ASSIGN\_GATE\_MODE  
internal device  
designation

**Placement** □ **Change**  
**Component** □  
Assignment □ Prefer order ASSIGN\_GATE\_MODE  
of symbol, hose/tube and  
pin

**Placement** □ **Change**  
**Component** □  
Assignment □ Prefer ASSIGN\_GATE\_MODE  
symbol name, pin name  
and signal

**Placement** □ **Change**  
**Component** □ Attributes □  
Overwrite attribute CC\_RELOAD\_ATTRIBUTES  
values for devices and  
symbols

**Placement** □ **Change**  
**Component** □ Attributes □ DELETE\_UNUSED\_ATTRIBUTES\_DURING\_CHANGE  
Delete unused attributes  
for devices and symbols

**Placement** □ **Change** CC\_RESTORE\_PINNAMES  
**Component** □ Pins □  
Restore changing pin

names

**Placement □ Change**

**Component** □ Fitting □  
Keep active mating and  
fitting parts

CC\_KEEP\_COUNTERPARTS

**Placement □ Change**

**Component** □ Fitting □  
Keep attribute for pin  
and block pin symbol

KEEP\_ATTRIBUTE\_PIN\_BLOCKPIN\_SYMBOL

**Placement □ Change**

**Component** □ Fitting □  
Place all pins as single  
pins

CC\_PLACE\_SINGLE\_PINS

**Placement □ Change**

**Component** □ Assembly  
devices □ Prefer  
component code

ASSIGN\_ASSEMBLY\_DEVICE\_MODE

**Placement □ Change**

**Component** □ Assembly  
devices □ Prefer order of  
devices in assembly

ASSIGN\_ASSEMBLY\_DEVICE\_MODE

**Placement □ Symbols**

□ Placement Parameters □  
Level

CELLEV

**Placement □ Symbols**

□ Placement Parameters □  
Scaling factor

CELL\_SCAFACTOR

**Placement □ Symbols**

□ Placement Parameters □  
Maintain text size when  
scaling

MAINTAIN\_TEXTSIZE

**Placement □ Symbols**

□ Placement Parameters □  
Load symbol graphic  
from database only if  
required

LOAD\_SYMBOL\_GRAPHIC\_ONLY\_IF\_REQUIRED

**Placement □ Symbols □**

Text Parameters □ Font

STXTFONTPTR

**Placement □ Symbols □**

Text Parameters □  
Change already placed  
symbols

**Placement □ Symbols □**

Place as graphic □ Create  
origin

CREATE\_ORIGIN

**Placement □ Symbols □**

Symbol text □ Change  
complete device when  
changing symbol text for CHANGE\_COMPLETE\_DEVICE  
higher level assignment,  
location and device  
designation

**Placement □ Symbols □**

Attribute text □ Use only USE\_SELECTED\_SYMBOLS\_FOR\_ATTRIBUTE\_TEXTS  
selected symbols when  
creating texts

**Placement □ Symbols □**

**Symbol Views □ Pin View**

Symbols □ Determine TRY\_AUTO\_GET\_PIN\_VIEW\_SYMBOL  
symbol for pin views  
using placed fitting  
symbols

**Placement □ Symbols □**

**Symbol Views □ Pin View** DEV\_PINVIEW\_SYM\_NAME

Symbols □ For device pins

**Placement □ Symbols □**

**Symbol Views □ Pin View** CONN\_PINVIEW\_SYM\_NAME

Symbols □ For fitting pins

**Placement □ Symbols □**

**Symbol Views □ Pin View** BLCON\_PINVIEW\_SYM\_NAME

Symbols □ For block  
fitting pins

**Placement □ Field □ Text**

template □ Symbol FIELD\_TEXT\_TEMPLATE

**Placement □ Field □** FIELD\_DIA

Outline □ Width

**Placement** □ **Field** □  
Outline □ Line Style FIELD\_MODE

**Placement** □ **Field** □  
Outline □ Color FIELD\_CODE

**Placement** □ **Field** □  
Outline □ Interrupt field  
border when connect  
line intersects border CUT\_FIELD\_BORDER

**Placement** □ **Field** □  
Outline □ Interrupt field  
border when connect  
line intersects border □  
Width CUT\_FIELD\_BORDER\_GAP

**Placement** □ **Field** □  
Hatch □ Pattern FIELD\_HATCH\_FLAGS

**Placement** □ **Field** □  
Hatch □ Width FIELD\_HATCH\_DIA

**Placement** □ **Field** □  
Hatch □ Distance FIELD\_HATCH\_LDIST

**Placement** □ **Field** □  
Hatch □ Color FIELD\_HATCH\_CODE

**Placement** □ **Field** □  
Origin □ Place origin in  
upper left instead of  
lower left FIELD\_CELL\_POS

**Placement** □ **Field** □  
Usage □ Rename fields  
when changing the  
designations in project  
tree RENAME\_FIELDS\_IN\_TREE

**Placement** □ **Dynamic**  
**Symbol** □ Text template □ DYN\_SYM\_TEXT\_TEMPLATE  
Symbol

**Placement** □ **Dynamic**  
**Symbol** □ Outline □ Width DYN\_SYM\_DIA

**Placement** □ **Dynamic**  
**Symbol** □ Outline □ Line DYN\_SYM\_MODE  
Style

**Placement** □ **Dynamic**  
**Symbol** □ Outline □ Color DYN\_SYM\_CODE

**Placement** □ **Dynamic**  
**Symbol** □ Hatch □ Pattern DYN\_SYM\_HATCH\_FLAGS

**Placement** □ **Dynamic**  
**Symbol** □ Hatch □ Width DYN\_SYM\_HATCH\_DIA

**Placement** □ **Dynamic**  
**Symbol** □ Hatch □ Distance DYN\_SYM\_HATCH\_LDIST

**Placement** □ **Dynamic**  
**Symbol** □ Hatch □ Color DYN\_SYM\_HATCH\_CODE

**Placement** □ **Dynamic**  
**Symbol** □ Origin □ Place  
origin in upper left  
instead of lower left DYN\_SYM\_CELL\_POS

**Placement** □ **Block** □ Font **BLOCKREFFONTPTR**

**Placement** □ **Block** □ Font  
Style **BLOCKREFDIA**

**Placement** □ **Block** □ Size **BLOCKREFSIZ**

**Placement** □ **Block** □  
Color **BLOCKREFCOD**

**Placement** □ **Block** □  
Display Control □ Gap **BLOCKREFGAP**

**Placement** □ **Block** □  
Display Control □ Level **BLOCKREFLAY**

**Placement** □ **Block** □ **BLOCKREFJUST**  
Display Control □ Rotate

**Placement** □ **Block** □  
Display Control □ BLOCKREFGAP  
Direction □ Up

**Placement** □ **Block** □  
Display Control □ BLOCKREFGAP  
Direction □ Down

**Placement** □ **Block** □  
Ratio □ Normal BLOCKREFMOD

**Placement** □ **Block** □  
Ratio □ Narrow BLOCKREFMOD

**Placement** □ **Block** □  
Ratio □ Wide BLOCKREFMOD

**Placement** □ **Block** □  
Direction □ Left BLOCKREFJUST

BLOCKREFJUST



**Placement** □ **Block** □  
 Direction □ Center

**Placement** □ **Block** □ BLOCKREFJUST  
 Direction □ Right

**Placement** □ **Block** □ Type BLOCKREFSORT  
 □ All

**Placement** □ **Block** □ Type BLOCKREFSORT  
 □ Origin

**Placement** □ **Block** □ Type BLOCKREFSORT  
 □ Position

**Placement** □ **Block** □ Split  
 Block □ Mark split blocks □ BLOCKEDGE\_TYPE  
 None

**Placement** □ **Block** □ Split  
 Block □ Mark split blocks □ BLOCKEDGE\_TYPE  
 Left

**Placement** □ **Block** □ Split  
 Block □ Mark split blocks □ BLOCKEDGE\_TYPE  
 Right

**Placement** □ **Block** □ Split  
 Block □ Mark split blocks □ BLOCKEDGE\_TYPE  
 Top

**Placement** □ **Block** □ Split BLOCKEDGE\_TYPE  
 Block □ Mark split blocks □

Bottom

**Placement** □ **Block** □ Split  
Block □ Line □ Width BLOCKEDGE\_DIA

**Placement** □ **Block** □ Split  
Block □ Line □ Line Style BLOCKEDGE\_MOD

**Placement** □ **Block** □ Split  
Block □ Line □ Color BLOCKEDGE\_COD

**Placement** □ **Block** □ Split  
Block □ Split the block  
and copy the graphic  
contents SPLITBLOCKOPTION

**Placement** □ **Block** □  
Block style □ Ignore  
default fill color IGNORE\_FILL\_COLOUR\_BLOCK

**Placement** □ **References**  
**Format** □ Prefix BLOCKREFSETTING

**Placement** □ **References**  
**Format** □ Suffix BLOCKREFSETTING

**Placement** □ **References**  
**Format** □ Sheet text BLOCKREFSETTING

**Placement** □ **References**  
**Format** □ Reference text BLOCKREFSETTING

**Placement** □ **Import** □  
Merge Sheet Reference  
Options □ Merge sheet  
references DF\_MERGE\_SHEET\_REFERENCES

**Placement** □ **Import** □ DF\_MERGE\_ALPHANUMERIC\_REFERENCES  
Merge Sheet Reference  
Options □ Merge only if  
reference names contain  
letters or special  
characters

**Placement** ☐ **Import** ☐

Merge Connect Line

Options ☐ Merge connect MERGE\_CONNECTION\_LINES  
lines for import and  
modification**Placement** ☐ **Import** ☐Sheets ☐ Create unique UNIQUE\_SHEET\_NAMES  
sheet names**Placement** ☐ **Import** ☐Sheets ☐ Ignore sheet IGNORE\_SHEET\_BORDER  
border**Placement** ☐ **Import** ☐Variants/Options ☐Rename already existing IMPORT\_RENAME\_VARIANTS  
variants/options from  
part file**Placement** ☐ **Import** ☐Variants/Options ☐ Use IMPORT\_RENAME\_VARIANTS  
existing variants/options  
from project**Placement** ☐ **Import** ☐Variants/Options ☐ Ask for IMPORT\_RENAME\_VARIANTS  
each existing  
variant/option**Placement** ☐ **Import** ☐Variants/Options ☐ Merge MERGE\_INCLUSIVE\_EXCLUSIVE\_DEFINITIONS  
inclusive/exclusive  
definitions**Placement** ☐ **Import** ☐**Device** ☐ Generate Item DFI\_USE\_DEFDES\_COPY\_POSTFIX  
Designation ☐ Suffix (  
CheckBox )**Placement** ☐ **Import** ☐**Device** ☐ Generate Item DFI\_DEFDES\_COPY\_POSTFIX  
Designation ☐ Suffix (  
EditBox )

**Placement** ☐ **Import** ☐**Device** ☐ UnplacedDevices ☐ Ignore unplaced devices  
IGNORE\_UNPLACED\_DEVICES**Placement** ☐ **Import** ☐**Device** ☐ UnplacedDevices ☐ Ignore unplaced devices of assemblies  
IGNORE\_UNPLACED\_DEVICES\_OF\_ASSEMBLIES**Placement** ☐ **Import** ☐**Device** ☐ Merge DeviceOptions ☐ Use existing devices

DFI\_USE\_EXISTING\_DEVICES

**Placement** ☐ **Import** ☐**Device** ☐ Merge DeviceOptions ☐ Use existing assemblies

DFI\_USE\_EXISTING\_ASSEMBLIES

**Placement** ☐ **Import** ☐**Device** ☐ Merge DeviceOptions ☐ Use existing hierarchical blocks

USE\_EXISTING\_HIERARCHICAL\_BLOCKS

**Placement** ☐ **Import** ☐**Device** ☐ Merge DeviceOptions ☐ Additional Merge Options ☐ Merge attributes

DFI\_MERGE\_ATTRIBUTES

Available before  
v2022-23.00**Placement** ☐ **Import** ☐**Device** ☐ Merge DeviceOptions ☐ Additional Merge Options ☐ Merge using exact hose/tube connection

MERGE\_USING\_EXACT\_CORE\_CONNECTION

**Placement** ☐ **Import** ☐**Device** ☐ Merge DeviceOptions ☐ Additional Merge Options ☐ Merge ignoring hose/tube

MERGE\_IGNOREING\_CORE\_DIRECTION

direction

**Placement** ☐ **Import** ☐  
**Device** ☐ Merge Device  
Options ☐ Additional MERGE\_OPTIONS  
Merge Options ☐ Merge  
options

**Placement** ☐ **Import** ☐  
**Device** ☐ Merge Device  
Options ☐ Additional  
Merge Options ☐ Use pin  
attributes from IMPORT\_USE\_PIN\_ATTR  
subcircuit

Available before  
v2022-23.00

**Placement** ☐ **Import** ☐  
**Device** ☐ Merge Device  
Options ☐ Additional IGNORE\_COMPONENT\_CODE\_ON\_IMPORT  
Merge Options ☐ Ignore  
component name

**Placement** ☐ **Import** ☐  
**Device** ☐ Merge Device  
Options ☐ Additional  
Merge Options ☐ Add to  
existing assemblies /  
create new assemblies

**Placement** ☐ **Import** ☐  
**Device** ☐ Use Default  
View Number ☐ For  
original symbols

**Placement** ☐ **Import** ☐  
**Device** ☐ Use Default  
View Number ☐ For view  
symbols

**Placement** ☐ **Import** ☐  
**Device** ☐ Merge Attribute  
Options ☐ Project  
preferred DFI\_MERGE\_ATTRIBUTES

Available from  
v2022-23.00

**Placement** ☐ **Import** ☐ OPTION\_IMPORT\_DEVICE\_MERGE\_ATTRIBUTES  
**Device** ☐ Merge Attribute  
Options ☐ Subcircuit  
preferred ☐ Devices ☐

## Merge

Available from  
v2022-23.00

**Placement** □ **Import** □

**Device** □ Merge Attribute

Options □ Subcircuit

preferred □ Devices □ Only `OPTION_IMPORT_DEVICE_ONLY_ATTRIBUTES`

Available from  
v2022-23.00

**Placement** □ **Import** □

**Device** □ Merge Attribute

Options □ Subcircuit `IMPORT_USE_PIN_ATTR`

preferred □ Pins □ Merge `OPTION_IMPORT_PIN_MERGE_ATTRIBUTES`

Available from  
v2022-23.00

**Placement** □ **Import** □

**Device** □ Merge Attribute

Options □ Subcircuit

preferred □ Pins □ Only `OPTION_IMPORT_PIN_ONLY_ATTRIBUTES`

Available from  
v2022-23.00

**Placement** □ **Import** □

**Device** □ Merge Attribute

Options □ Subcircuit

preferred □ Wires □ Merge `OPTION_IMPORT_WIRE_MERGE_ATTRIBUTES`

Available from  
v2022-23.00

**Placement** □ **Import** □

**Device** □ Merge Attribute

Options □ Subcircuit

preferred □ Wires □ Only `OPTION_IMPORT_WIRE_ONLY_ATTRIBUTES`

Available from  
v2022-23.00

**Placement** □

**Export/Copy** □ General □ `EXPORT_STRUCTURE_NODES`

Export structure nodes

**Placement** □

**Export/Copy** □ Devices □

Unplaced devices (only `EXPORT_UNPLACED_DEVICES`

valid when exporting

'all')

**Placement** □

**Export/Copy** □ Devices □ EXPORT\_ALL\_UNPLACED\_DEVICES\_OF\_ASSEMBLIES  
All devices of a selected assembly

**Placement** □

**Export/Copy** □ Devices □ HIERARCHYBLOCK\_WITH\_STRUCTURE  
Hierarchical block with substructures

**Placement** □

**Export/Copy** □ Hoses/Tubes □ COPY\_EXPORT\_CABLE  
Hoses/Tubes

**Placement** □

**Export/Copy** □ Hoses/Tubes □ At least one end selected COPY\_EXPORT\_CABLE\_OPTION

**Placement** □

**Export/Copy** □ Hoses/Tubes □ Both ends selected COPY\_EXPORT\_CABLE\_OPTION

**Placement** □

**Export/Copy** □ Hoses/Tubes □ Both ends and the path are selected COPY\_EXPORT\_CABLE\_OPTION

**Placement** □

**Export/Copy** □ Hoses/Tubes □ Full path is selected COPY\_EXPORT\_CABLE\_OPTION

**Placement** □

**Export/Copy** □ Plugging □ EXPORT\_PLUGS\_WITHOUT\_GRAPHICAL\_REPRESENTATIONS  
Export plugging information for pluggings without

graphical representation

**Graphic** □ Width GRADIA

**Graphic** □ Arrows GRAFLG

**Graphic** □ Color GRACOD

**Graphic** □ Level GRALEV

**Graphic** □ Line Style GRAMOD

**Graphic** □ Line Style □  
Default LINESTYLE\_JIS

**Graphic** □ Line Style □  
Japanese Industrial Standards LINESTYLE\_JIS

**Graphic** □ Redlining  
Information □ Display  
non-redliner information  
in different color USE\_NON\_REDLINER\_COLOR

NON\_REDLINER\_COLOR



**Graphic** □ Redlining  
Information □ Color

**Graphic** □ Read-Only  
Level □ Use read-only  
level USE\_READ\_ONLY\_GRAPHIC\_LEVEL

**Graphic** □ Read-Only  
Level □ Color READ\_ONLY\_GRAPHIC\_COLOR

**Graphic** □ Read-Only  
Level □ Level READ\_ONLY\_GRAPHIC\_LEVEL

GRAHATCH\_DEGREE1  
**Graphic** □ **Hatch** □  
Pattern GRAHATCH\_DEGREE2  
GRAHATCH\_FLAGS

**Graphic** □ **Hatch** □ Line  
Style GRAHATCH\_GRAMOD

**Graphic** □ **Hatch** □ Width GRAHATCH\_GRADIA

**Graphic** □ **Hatch** □  
Distance GRAHATCH\_LINDIST

**Graphic** □ **Hatch** □ Color GRAHATCH\_GRACOD

**Graphic** □ **Text** □ Font □  
Name TXTFONTPTR

**Graphic** □ **Text** □ Font □  
Style                                   TXTDIA

**Graphic** □ **Text** □ Font □  
Size                                   TXTSIZ

**Graphic** □ **Text** □ Font □  
Color                                TXTCOD

**Graphic** □ **Text** □ Font □  
Ratio                                TXTMOD

**Graphic** □ **Text** □ Font □  
Alignment                           TXTJUST

**Graphic** □ **Text** □ Font □  
Level                                TXTLEV

**Graphic** □ **Text** □ Effects □  
Strikeout                           TXTDIA

**Graphic** □ **Text** □ Effects □  
Underline                      TXTDIA

**Graphic** □ **Text** □ Effects □  
Subsidiary line to                      SUBSIDIARY\_LINE\_TO\_GRAPHIC  
graphic

**Dimensions** □ Lines □  
Arrow                      DIM\_ARROW\_TYP

**Dimensions** □ Lines □  
Arrow Width                      DIM\_ARROW\_WIDTH

**Dimensions** □ Lines □ Use  
fixed size to display                      DIM\_FIX\_SIZE

**Dimensions** □ Lines □  
Hide longer part of                      DIM\_HIDE\_LONGER\_PART  
arrow

**Dimensions** □ Lines □  
Extension                      DIM\_EXTENSION

**Dimensions** □ Lines □ Line  
width                      DIM\_WIDTH

**Dimensions** □ Lines □  
Extension line offset                      DIM\_EXTENSION\_LINE\_OFFSET

**Dimensions** □ Text □  
Precision                      DIM\_PRECISION

**Dimensions** □ Text □  
Offset                      DIM\_TXT\_OFFSET

**Dimensions** □ Text □  
Prefix                      DIM\_PREFIX

**Dimensions** □ Text □  
Center texts                      DIMENSION\_TEXT\_CENTER

DIM\_ROTATE\_TEXT\_OF\_RUNNING\_DIM

**Dimensions** □ Text □  
 Rotate texts of running  
 dimensions

**Dimensions** □ Text □ DIM\_SUFFIX\_SIZE\_FACTOR  
 Suffix size factor (%)

**Dimensions** □ Text □ DIM\_SUFFIX  
 Suffix

**Dimensions** □ Text □ DIM\_DISPLAY\_ATTR  
 Display

**Dimensions** □ General □ DIM\_LEVEL  
 Level

**Dimensions** □ General □ DIM\_COLOR  
 Color

**Dimensions** □  
**Dimension Text** □ Font □ DIM\_TXT\_FONTPTR  
 Name

**Dimensions** □  
**Dimension Text** □ Font □ DIM\_TXT\_DIA  
 Style

**Dimensions** □  
**Dimension Text** □ Font □ DIM\_TXT\_SIZE  
 Size

**Dimensions** □  
**Dimension Text** □ Font □ DIM\_TXT\_COLOR  
 Color

**Dimensions** □  
**Dimension Text** □ DIM\_TXT\_DIA  
 Effects □ Strikeout

**Dimensions** □  
**Dimension Text** □ DIM\_TXT\_DIA  
 Effects □ Underline

**Dimensions** □  
**Dimension Text** □ DIM\_TXT\_DIA  
 Effects □ Opaque

**Panel** □ Working Grid □ PANELGRIDSIZE  
 Grid size

**Panel** ☐ Working Grid ☐ PANELTRAPSIZE  
Snap size

**Panel** ☐ Alternative Grid ☐ PANELALTGRIDSIZE  
Grid size

**Panel** ☐ Measurement  
Units ☐ Millimeters MEA\_EXTERN\_PANEL

**Panel** ☐ Measurement  
Units ☐ Inches MEA\_EXTERN\_PANEL

**Panel** ☐ Grid View ☐ Points  
( CheckBox ) ANEL\_MODE\_GRID\_OVERLAY

**Panel** ☐ Grid View ☐ Points  
( SpinControl ) PANELOVERSIZE

**Panel** ☐ Grid View ☐ Rulers  
( CheckBox ) PANEL\_MODE\_GRID\_AXIS  
SCHEMA\_MODE\_GRID\_AXIS

**Panel** ☐ Grid View ☐ Rulers  
( SpinControl ) PANELAXISGRID

**Panel** ☐ Highlight ☐ Color PANEL\_HIGHLIGHT\_COLOUR

**Panel** ☐ Highlight ☐ Width PANEL\_HIGHLIGHT\_WIDTH

**Panel** ☐ Shared sheets ☐  
Display region overview DISPLAY\_REGION

PANEL\_REGION\_SCALE\_STEP

**Panel** □ Shared sheets □  
Scaling increment of  
region

**Panel** □ Placement □  
Optimize placement after changes OPTIMIZE\_PLACEMENT\_AFTER\_CHANGES

**Panel** □ 2D □ Display  
model orientation in 2D 2D\_MODEL\_ORIENTATION

**Panel** □ 2D □ Percentage  
of visibility to show partially covered models PANEL\_2D\_VISIBLE\_PERCENTAGE

**Panel** □ 3D □ Display  
model orientation in 3D 3D\_MODEL\_ORIENTATION

**Panel** □ 3D □ Display  
model graphic from 2D in 3D 3D\_MODEL\_GRAPHIC

**Panel** □ 3D □ Display STEP  
models in 3D 3D\_DISPLAY\_STEP\_MODELS

**Panel** □ 3D □ Load STEP  
models from database only if required 3D\_LOAD\_STEP\_MODELS\_IF\_REQUIRED

**Panel** □ 3D □ Wireframe  
mode 3D\_WIREFRAME

**Panel** □ 3D □ Orthographic  
mode 3D\_ORTHOGRAPHIC

**Panel** □ 3D □ Enable  
lighting 3D\_LIGHTING

**Panel** □ **Renderer**  
**Settings** □ Preferred  
Renderer

**Panel** □ **Renderer**  
**Settings** □ Anti-Aliasing

**Panel** □ **Connection** □  
Connection Method □ PANEL\_CONNECT\_METHOD  
Signal on pin

**Panel** □ **Connection** □ PANEL\_CONNECT\_METHOD  
Connection Method □  
Graphical schema  
connections

**Panel ▯ Connection ▯**

Connection Method ▯ Use  
only connections with  
assigned  
wires/conductors in  
schematic

PANEL\_CONNECT\_METHOD

**Panel ▯ Connection ▯**

Wiring Options ▯ Allow  
wire loops

PANEL\_ALLOW\_WIRE\_LOOPS

**Panel ▯ Connection ▯**

Wiring Options ▯  
Backplane connection  
distance

PANEL\_BACKPLANE\_DISTANCE

**Panel ▯ Connection ▯**

Wiring Options ▯ Wire  
insulation factor

PANEL\_WIRE\_INSULATION\_FACTOR

**Panel ▯ Connection ▯**

Autoconnect Parameter ▯  
Nearest cable duct

PANEL\_AUTOCONNECT\_CABLE\_DUCT\_SEARCH\_GAGE\_FLAG

**Panel ▯ Connection ▯**

Autoconnect Parameter ▯  
Bandwidth for searching  
cable duct ( RadioButton  
)

PANEL\_AUTOCONNECT\_CABLE\_DUCT\_SEARCH\_GAGE\_FLAG

**Panel ▯ Connection ▯**

Autoconnect Parameter ▯  
Bandwidth for searching  
cable duct ( SpinControl  
)

PANEL\_AUTOCONNECT\_CABLE\_DUCT\_SEARCH\_GAGE

**Panel ▯ Connection ▯**

Autoconnect Parameter ▯  
Use manually defined  
ports for terminals in  
schematic

PANEL\_AUTOCONNECT\_USE\_SCHEMATIC\_TERMINAL\_PORTS

**Panel ▯ Connection ▯**

Autoconnect Parameter ▯  
Delete predefined ports  
in schematic

PANEL\_AUTOCONNECT\_DELETE\_PREFDEF\_PORTS

**Panel ▯ Connection ▯**

Autoconnect Algorithm ▯  
Route wire jumpers at  
the end

PANEL\_AUTOCONNECT\_WIRE\_JUMPER\_FINALLY

**Panel ▯ Connection ▯**

Autoconnect Algorithm ▯  
Find suitable fitting with  
an additional wiring run

DF\_CBHEAD\_PANEL\_AUTOCONNECT\_BIGGER\_CONNECTOR\_PIN

**Panel □ Connection □**

Autoconnect Algorithm □ PANEL\_AUTOCONNECT\_ALG\_CURRENT  
Current (Chain)

**Panel □ Connection □**

Autoconnect Algorithm □ PANEL\_AUTOCONNECT\_ALG\_OPTIMIZE  
Optimized (Pair of Pins)

**Panel □ Connection □**

Autoconnect Algorithm □ PANEL\_ROUTING\_STATISTIC  
Write statistic

**Panel □ Connection □**

Display Options □ Use PANEL\_CONNECTION\_DISPLAY\_WIRE\_PARAMS  
display parameters from wire

**Panel □ Connection □**

Display Options □ Mark PANEL\_CONNECT\_LEAVING\_WIRE\_MARK  
sheet comprehensive wires

**Panel □ Connection □**

Display Options □ Mark PANEL\_CONNECT\_LEAVING\_WIRE\_MARK\_SIZE  
size

**Panel □ Connection □**

Display Options □ Mark PANEL\_CONNECT\_MARK\_JUMPER\_CONNECT\_POINTS  
jumper connect points

**Panel □ Connection □**

**Logic Lines** □ Display PANEL\_AIRLINES  
logic lines

**Panel □ Connection □**

**Logic Lines** □ Direct PANEL\_AIRLINES  
Connection □ Display direct connections

**Panel □ Connection □**

**Logic Lines** □ Direct PANEL\_AIRLINES\_DIRECT\_COLOR  
Connection □ **Advanced** □ Color

**Panel □ Connection □**

**Logic Lines** □ Direct PANEL\_AIRLINES\_DIRECT\_STYLE  
Connection □ **Advanced** □ Line Style

**Panel □ Connection □**

**Logic Lines** □ Subnet PANEL\_AIRLINES  
Connection □ Display subnet connections

**Panel □ Connection □**

PANEL\_AIRLINES\_SUBNET\_COLOR  
**Logic Lines** □ Subnet  
Connection □ **Advanced** □ Color



**Panel** □ **Connection** □  
**Logic Lines** □ Subnet  
 Connection □ **Advanced** □ PANEL\_AIRLINES\_SUBNET\_STYLE  
 Line Style

**Panel** □ **Connection** □  
**Logic Lines** □ Equivalent  
 Pins □ Display equivalent pins PANEL\_AIRLINES

**Panel** □ **Connection** □  
**Logic Lines** □ Equivalent PANEL\_AIRLINES\_EQUIVALENT\_COLOR  
 Pins □ **Advanced** □ Color

**Panel** □ **Connection** □  
**Logic Lines** □ Equivalent  
 Pins □ **Advanced** □ Line PANEL\_AIRLINES\_EQUIVALENT\_STYLE  
 Style

**Panel** □ **Connection** □  
**Logic Lines** □ Signal  
 Carrying Pins □ Display signal carrying pins PANEL\_AIRLINES

**Panel** □ **Connection** □  
**Logic Lines** □ Signal  
 Carrying Pins □ PANEL\_AIRLINES\_SIGNAL\_COLOR  
**Advanced** □ Color

**Panel** □ **Connection** □  
**Logic Lines** □ Signal  
 Carrying Pins □ PANEL\_AIRLINES\_SIGNAL\_WIDTH  
**Advanced** □ Size

**Panel** □ **Connection** □  
**Logic Lines** □ Sheet  
 Comprehensive PANEL\_AIRLINES  
 Connections □ Display connection logic on pin

**Panel** □ **Restricted**  
**Symbol** □ Defined for = PANEL\_RESTRICTED\_ALL\_LEVEL  
 All □ Level

**Panel** □ **Restricted**  
**Symbol** □ Defined for = PANEL\_RESTRICTED\_ALL\_LINE\_WIDTH  
 All □ Outline □ Width

**Panel** □ **Restricted**  
**Symbol** □ Defined for = PANEL\_RESTRICTED\_ALL\_LINE\_COLOR  
 All □ Outline □ Color

**Panel** □ **Restricted**  
**Symbol** □ Defined for = PANEL\_RESTRICTED\_ALL\_LINE\_STYLE  
 All □ Outline □ Line Style

**Panel** □ **Restricted** PANEL\_RESTRICTED\_ALL\_HATCH\_DEGREE1  
**Symbol** □ Defined for =

All ▢ Hatch ▢ Pattern	PANEL_RESTRICTED_ALL_HATCH_DEGREE2
	PANEL_RESTRICTED_ALL_HATCH_FLAGS
<b>Panel ▢ Restricted</b> <b>Symbol</b> ▢ Defined for = All ▢ Hatch ▢ Line Style	PANEL_RESTRICTED_ALL_HATCH_STYLE
<b>Panel ▢ Restricted</b> <b>Symbol</b> ▢ Defined for = All ▢ Hatch ▢ Width	PANEL_RESTRICTED_ALL_HATCH_WIDTH
<b>Panel ▢ Restricted</b> <b>Symbol</b> ▢ Defined for = All ▢ Hatch ▢ Distance	PANEL_RESTRICTED_ALL_HATCH_DIST
<b>Panel ▢ Restricted</b> <b>Symbol</b> ▢ Defined for = All ▢ Hatch ▢ Color	PANEL_RESTRICTED_ALL_HATCH_COLOR
<b>Panel ▢ Restricted</b> <b>Symbol</b> ▢ Defined for = Components ▢ Level	PANEL_RESTRICTED_DEV_LEVEL
<b>Panel ▢ Restricted</b> <b>Symbol</b> ▢ Defined for = Components ▢ Outline ▢ Width	PANEL_RESTRICTED_DEV_LINE_WIDTH
<b>Panel ▢ Restricted</b> <b>Symbol</b> ▢ Defined for = Components ▢ Outline ▢ Color	PANEL_RESTRICTED_DEV_LINE_COLOR
<b>Panel ▢ Restricted</b> <b>Symbol</b> ▢ Defined for = Components ▢ Outline ▢ Line Style	PANEL_RESTRICTED_DEV_LINE_STYLE
<b>Panel ▢ Restricted</b> <b>Symbol</b> ▢ Defined for = Components ▢ Hatch ▢ Pattern	PANEL_RESTRICTED_DEV_HATCH_DEGREE1
	PANEL_RESTRICTED_DEV_HATCH_DEGREE2
	PANEL_RESTRICTED_DEV_HATCH_FLAGS
<b>Panel ▢ Restricted</b> <b>Symbol</b> ▢ Defined for = Components ▢ Hatch ▢ Line Style	PANEL_RESTRICTED_DEV_HATCH_STYLE
<b>Panel ▢ Restricted</b> <b>Symbol</b> ▢ Defined for = Components ▢ Hatch ▢ Width	PANEL_RESTRICTED_DEV_HATCH_WIDTH
<b>Panel ▢ Restricted</b> <b>Symbol</b> ▢ Defined for = Components ▢ Hatch ▢	PANEL_RESTRICTED_DEV_HATCH_DIST

Distance

**Panel** ☐ **Restricted**

**Symbol** ☐ Defined for = PANEL\_RESTRICTED\_DEV\_HATCH\_COLOR  
Components ☐ Hatch ☐ Color

**Panel** ☐ **Restricted**

**Symbol** ☐ Defined for = PANEL\_RESTRICTED\_CON\_LEVEL  
Connections ☐ Level

**Panel** ☐ **Restricted**

**Symbol** ☐ Defined for = PANEL\_RESTRICTED\_CON\_LINE\_WIDTH  
Connections ☐ Outline ☐ Width

**Panel** ☐ **Restricted**

**Symbol** ☐ Defined for = PANEL\_RESTRICTED\_CON\_LINE\_COLOR  
Connections ☐ Outline ☐ Color

**Panel** ☐ **Restricted**

**Symbol** ☐ Defined for = PANEL\_RESTRICTED\_CON\_LINE\_STYLE  
Connections ☐ Outline ☐ Line Style

**Panel** ☐ **Restricted**

**Symbol** ☐ Defined for = PANEL\_RESTRICTED\_CON\_HATCH\_DEGREE1  
Connections ☐ Hatch ☐ Pattern  
PANEL\_RESTRICTED\_CON\_HATCH\_DEGREE2  
PANEL\_RESTRICTED\_CON\_HATCH\_FLAGS

**Panel** ☐ **Restricted**

**Symbol** ☐ Defined for = PANEL\_RESTRICTED\_CON\_HATCH\_STYLE  
Connections ☐ Hatch ☐ Line Style

**Panel** ☐ **Restricted**

**Symbol** ☐ Defined for = PANEL\_RESTRICTED\_CON\_HATCH\_WIDTH  
Connections ☐ Hatch ☐ Width

**Panel** ☐ **Restricted**

**Symbol** ☐ Defined for = PANEL\_RESTRICTED\_CON\_HATCH\_DIST  
Connections ☐ Hatch ☐ Distance

**Panel** ☐ **Restricted**

**Symbol** ☐ Defined for = PANEL\_RESTRICTED\_CON\_HATCH\_COLOR  
Connections ☐ Hatch ☐ Color

**Panel** ☐ **Restricted**

**Symbol** ☐ Defined for = PANEL\_RESTRICTED\_CUTOUT\_LEVEL  
Cutout area ☐ Level

**Panel** ☐ **Restricted**

**Symbol** ☐ Defined for = PANEL\_RESTRICTED\_CUTOUT\_LINE\_WIDTH

Cutout area □ Outline □  
Width

**Panel □ Restricted**

**Symbol** □ Defined for = PANEL\_RESTRICTED\_CUTOUT\_LINE\_COLOR  
Cutout area □ Outline □  
Color

**Panel □ Restricted**

**Symbol** □ Defined for = PANEL\_RESTRICTED\_CUTOUT\_LINE\_STYLE  
Cutout area □ Outline □  
Line Style

**Panel □ Restricted**

**Symbol** □ Defined for = PANEL\_RESTRICTED\_CUTOUT\_HATCH\_DEGREE1  
Cutout area □ Hatch □  
Pattern  
PANEL\_RESTRICTED\_CUTOUT\_HATCH\_DEGREE2  
PANEL\_RESTRICTED\_CUTOUT\_HATCH\_FLAGS

**Panel □ Restricted**

**Symbol** □ Defined for = PANEL\_RESTRICTED\_CUTOUT\_HATCH\_STYLE  
Cutout area □ Hatch □ Line  
Style

**Panel □ Restricted**

**Symbol** □ Defined for = PANEL\_RESTRICTED\_CUTOUT\_HATCH\_WIDTH  
Cutout area □ Hatch □  
Width

**Panel □ Restricted**

**Symbol** □ Defined for = PANEL\_RESTRICTED\_CUTOUT\_HATCH\_DIST  
Cutout area □ Hatch □  
Distance

**Panel □ Restricted**

**Symbol** □ Defined for = PANEL\_RESTRICTED\_CUTOUT\_HATCH\_COLOR  
Cutout area □ Hatch □  
Color

**Panel □ Restricted**

**Symbol** □ Defined for = PANEL\_RESTRICTED\_HOLE\_LEVEL  
Drill-hole □ Level

**Panel □ Restricted**

**Symbol** □ Defined for = PANEL\_RESTRICTED\_HOLE\_LINE\_WIDTH  
Drill-hole □ Outline □ Width

**Panel □ Restricted**

**Symbol** □ Defined for = PANEL\_RESTRICTED\_HOLE\_LINE\_COLOR  
Drill-hole □ Outline □ Color

**Panel □ Restricted**

**Symbol** □ Defined for = PANEL\_RESTRICTED\_HOLE\_LINE\_STYLE  
Drill-hole □ Outline □ Line  
Style

**Panel □ Restricted**

**Symbol** □ Defined for = PANEL\_RESTRICTED\_HOLE\_HATCH\_DEGREE1

Drill-hole □ Hatch □ Pattern PANEL\_RESTRICTED\_HOLE\_HATCH\_DEGREE2

PANEL\_RESTRICTED\_HOLE\_HATCH\_FLAGS

**Panel □ Restricted**

**Symbol** □ Defined for = PANEL\_RESTRICTED\_HOLE\_HATCH\_STYLE  
Drill-hole □ Hatch □ Line  
Style

**Panel □ Restricted**

**Symbol** □ Defined for = PANEL\_RESTRICTED\_HOLE\_HATCH\_WIDTH  
Drill-hole □ Hatch □ Width

**Panel □ Restricted**

**Symbol** □ Defined for = PANEL\_RESTRICTED\_HOLE\_HATCH\_DIST  
Drill-hole □ Hatch □  
Distance

**Panel □ Restricted**

**Symbol** □ Defined for = PANEL\_RESTRICTED\_HOLE\_HATCH\_COLOR  
Drill-hole □ Hatch □ Color

**Panel □ Restricted**

**Symbol** □ **Display** □ PANEL\_RESTRICTED\_DISPLAY  
Objects □ Restricted for  
all

**Panel □ Restricted**

**Symbol** □ **Display** □ PANEL\_RESTRICTED\_DISPLAY  
Objects □ Restricted for  
Components

**Panel □ Restricted**

**Symbol** □ **Display** □ PANEL\_RESTRICTED\_DISPLAY  
Objects □ Restricted for  
Connections

**Panel □ Restricted**

**Symbol** □ **Display** □ PANEL\_RESTRICTED\_DISPLAY  
Objects □ Cutout area

**Panel □ Restricted**

**Symbol** □ **Display** □ PANEL\_RESTRICTED\_DISPLAY  
Objects □ Drill-hole

**Panel □ Mount Symbol** □ PANEL\_MOUNT\_SYM\_LEVEL  
Level

**Panel □ Mount Symbol** □ PANEL\_MOUNT\_SYM\_LINE\_WIDTH  
Outline □ Width

**Panel □ Mount Symbol** □ PANEL\_MOUNT\_SYM\_LINE\_COLOR  
Outline □ Color

**Panel □ Mount Symbol** □ PANEL\_MOUNT\_SYM\_LINE\_STYLE  
Outline □ Line Style

PANEL\_MOUNT\_SYM\_HATCH\_DEGREE1

**Panel** □ **Mount Symbol** □ PANEL\_MOUNT\_SYM\_HATCH\_DEGREE2  
Hatch □ Pattern

PANEL\_MOUNT\_SYM\_HATCH\_FLAGS

**Panel** □ **Mount Symbol** □ PANEL\_MOUNT\_SYM\_HATCH\_COLOR  
Hatch □ Color

**Panel** □ **Cable Duct**  
**Symbol** □ Level PANEL\_CABLE\_DUCT\_SYM\_LEVEL

**Panel** □ **Cable Duct**  
**Symbol** □ Outline □ Width PANEL\_CABLE\_DUCT\_SYM\_LINE\_WIDTH

**Panel** □ **Cable Duct**  
**Symbol** □ Outline □ Color PANEL\_CABLE\_DUCT\_SYM\_LINE\_COLOR

**Panel** □ **Cable Duct**  
**Symbol** □ Outline □ Line Style PANEL\_CABLE\_DUCT\_SYM\_LINE\_STYLE

PANEL\_CABLE\_DUCT\_SYM\_HATCH\_DEGREE1

**Panel** □ **Cable Duct**  
**Symbol** □ Hatch □ Pattern PANEL\_CABLE\_DUCT\_SYM\_HATCH\_DEGREE2

PANEL\_CABLE\_DUCT\_SYM\_HATCH\_FLAGS

**Panel** □ **Cable Duct**  
**Symbol** □ Hatch □ Color PANEL\_CABLE\_DUCT\_SYM\_HATCH\_COLOR

**Panel** □ **Cable Duct**  
**Symbol** □ **Display** □ Fill Size □ Display fill size PANEL\_CABLE\_DUCT

**Panel** □ **Cable Duct**  
**Symbol** □ **Display** □ Fill Size □ **Advanced** □ Color PANEL\_CABLE\_DUCT\_FILL\_COLOR

**Panel** □ **Cable Duct**  
**Symbol** □ **Display** □ Fill Size □ **Advanced** □ Line PANEL\_CABLE\_DUCT\_FILL\_STYLE

## Style

**Panel □ Cable Duct**

**Symbol □ Display □ Fill** PANEL\_CABLE\_DUCT\_FILL\_LIMIT  
 Size □ Cable duct fill limit (%)

**Panel □ Cable Duct**

**Symbol □ Display □ Fill** PANEL\_CABLE\_DUCT\_CRITICAL\_FILL\_LIMIT  
 Size □ Cable duct warning limit (%)

**Panel □ Cable Duct**

**Symbol □ Display □ Fill** CORRECTION\_FACTOR\_FOR\_SPACE\_REQUIREMENTS  
 Size □ Connection factor for space requirements

**Panel □ Cable Duct**

**Symbol □ Display □** PANEL\_CABLE\_DUCT  
 Docking Point □ Display docking point

**Panel □ Cable Duct**

**Symbol □ Display □** PANEL\_CABLE\_DUCT\_DOCK\_COLOR  
 Docking Point □  
**Advanced □ Color**

**Panel □ Cable Duct**

**Symbol □ Display □** PANEL\_CABLE\_DUCT\_DOCK\_WIDTH  
 Docking Point □  
**Advanced □ Size**

**Panel □ Cable Duct**

**Symbol □ Display □**  
 Lateral Punching Width □ PANEL\_CABLE\_DUCT  
 Display lateral punching width

**Panel □ Cable Duct****Symbol □ Display □**

Lateral Punching Width □

**Advanced □ Color****Panel □ Cable Duct****Symbol □ Display □**

Lateral Punching Width □

**Advanced □ Size****Panel □ Cable Duct****Symbol □ Display □ Break** PANEL\_CABLE\_DUCT

Line □ Display break line

**Panel □ Cable Duct****Symbol □ Display □ Break**Line □ **Advanced □ Color****Panel □ Cable Duct****Symbol □ Display □ Break**Line □ **Advanced □ Size****Panel □ Checks □**

Placement □ Mounting

description &lt;-&gt; Slot

PANEL\_CHECKS\_OUTLINE\_TO\_SLOT

description □ On

**Panel □ Checks □**

Placement □ Mounting

description &lt;-&gt; Slot

PANEL\_CHECKS\_OUTLINE\_TO\_SLOT

description □ Warning

**Panel □ Checks □**

Placement □ Mounting

description &lt;-&gt; Slot

PANEL\_CHECKS\_OUTLINE\_TO\_SLOT

description □ Off

**Panel □ Checks □**

Placement □ Complete

component -&gt; Slot

PANEL\_CHECKS\_FIT\_TO\_TARGET

Area/Line □ On

**Panel □ Checks □**

Placement □ Complete

component -&gt; Slot

PANEL\_CHECKS\_FIT\_TO\_TARGET

Area/Line □ Warning

**Panel □ Checks □**

Placement □ Complete

component -&gt; Slot

PANEL\_CHECKS\_FIT\_TO\_TARGET

Area/Line □ Off

**Panel □ Checks □**

Placement □ Component

&lt;-&gt; Component □ On

PANEL\_CHECKS\_OUTLINE\_TO\_OUTLINE

**Panel □ Checks □**

Placement □ Component

PANEL\_CHECKS\_OUTLINE\_TO\_OUTLINE



<-> Component ☐  
Warning

**Panel ☐ Checks ☐**  
Placement ☐ Component PANEL\_CHECKS\_OUTLINE\_TO\_OUTLINE  
<-> Component ☐ Off

**Panel ☐ Checks ☐**  
Placement ☐ Component PANEL\_CHECKS\_OUTLINE\_TO\_RESTRICTED  
<-> Restricted ☐ On

**Panel ☐ Checks ☐**  
Placement ☐ Component PANEL\_CHECKS\_OUTLINE\_TO\_RESTRICTED  
<-> Restricted ☐ Warning

**Panel ☐ Checks ☐**  
Placement ☐ Component PANEL\_CHECKS\_OUTLINE\_TO\_RESTRICTED  
<-> Restricted ☐ Off

**Panel ☐ Checks ☐**  
Placement ☐ Cutout <-> PANEL\_CHECKS\_OUTLINE\_TO\_CUTOUT  
Component ☐ On

**Panel ☐ Checks ☐**  
Placement ☐ Cutout <-> PANEL\_CHECKS\_OUTLINE\_TO\_CUTOUT  
Component ☐ Warning

**Panel ☐ Checks ☐**  
Placement ☐ Cutout <-> PANEL\_CHECKS\_OUTLINE\_TO\_CUTOUT  
Component ☐ Off

**Panel ☐ Checks ☐**  
Placement ☐ Cutout <-> PANEL\_CHECKS\_RESTRICTED\_TO\_CUTOUT  
Restricted ☐ On

**Panel ☐ Checks ☐**  
Placement ☐ Cutout <-> PANEL\_CHECKS\_RESTRICTED\_TO\_CUTOUT  
Restricted ☐ Warning

**Panel ☐ Checks ☐**  
Placement ☐ Cutout <-> PANEL\_CHECKS\_RESTRICTED\_TO\_CUTOUT  
Restricted ☐ Off

**Panel ☐ Checks ☐**  
Placement ☐  
Variants/Options <-> PANEL\_CHECKS\_COMPONENT\_OPTIONS\_VARIANTS  
Variants/Options ☐ On

**Panel ☐ Checks ☐**  
Placement ☐  
Variants/Options <-> PANEL\_CHECKS\_COMPONENT\_OPTIONS\_VARIANTS  
Variants/Options ☐  
Warning

**Panel ☐ Checks ☐**  
Placement ☐  
Variants/Options <-> PANEL\_CHECKS\_COMPONENT\_OPTIONS\_VARIANTS  
Variants/Options ☐ Off

**Panel □ Checks □**

Autoconnect □

Wire/Conductor from Pin &lt;-&gt; PANEL\_CHECKS\_WIRE\_TO\_OUTLINE\_1

Component/Restricted □  
On**Panel □ Checks □**

Autoconnect □

Wire/Conductor from Pin &lt;-&gt; PANEL\_CHECKS\_WIRE\_TO\_OUTLINE\_1

Component/Restricted □  
Warning**Panel □ Checks □**

Autoconnect □

Wire/Conductor from Pin &lt;-&gt; PANEL\_CHECKS\_WIRE\_TO\_OUTLINE\_1

Component/Restricted □  
Off**Panel □ Checks □**

Autoconnect □

Wire/Conductor to cable duct &lt;-&gt; PANEL\_CHECKS\_WIRE\_TO\_OUTLINE\_2

Component/Restricted □  
On**Panel □ Checks □**

Autoconnect □

Wire/Conductor to cable duct &lt;-&gt; PANEL\_CHECKS\_WIRE\_TO\_OUTLINE\_2

Component/Restricted □  
Warning**Panel □ Checks □**

Autoconnect □

Wire/Conductor to cable duct &lt;-&gt; PANEL\_CHECKS\_WIRE\_TO\_OUTLINE\_2

Component/Restricted □  
Off**Panel □ Checks □ All  
attributes****Panel □ Checks □ Only  
attribute ( RadioButton )** PANEL\_AUTOROUTE\_WITH\_EXPLICIT\_ATTRIBUTES**Panel □ Checks □ Only  
attribute ( ComboBox )** PANEL\_AUTOROUTE\_ATTRIBUTES**Panel □ Checks □ Allow  
crossing cable ducts  
with different connection** PANEL\_CHECKS\_ALLOW\_CROSSING\_CABLEDUCTS

classes

**Variants/Options** □

Project Setting □  
Separator for variant  
texts

VARIANT\_TEXT\_SEPARATOR

**Variants/Options** □

Project Setting □ Unique  
Names for  
Variants/Options

OPTION\_VARIANT\_UNIQUE\_NAMES

**Variants/Options** □

Create New Alias □ (   
Table )

**Variants/Options** □

**Display** □ Activation of  
Variants / Options □  
Variants □ default

VARIANT\_ACTIVE\_VARIANT

**Variants/Options** □

**Display** □ Activation of  
Variants / Options □  
Variants □ all

VARIANT\_ACTIVE\_VARIANT

**Variants/Options** □

**Display** □ Activation of  
Variants / Options □  
Options □ default

VARIANT\_ACTIVE\_VARIANT

**Variants/Options** □

**Display** □ Activation of  
Variants / Options □  
Options □ all

VARIANT\_ACTIVE\_VARIANT

**Variants/Options** □

**Display** □ Activation of  
Variants / Options □

VARIANT\_ACTIVE\_VARIANT

Options ☐ none

**Variants/Options** ☐

**Display** ☐ Activation of

Variants / Options ☐ DISPLAY\_OBJECTS\_WITHOUT\_VARIANTS

Display elements without  
variants / options

**Variants/Options** ☐

**Display** ☐ Display

Settings ☐ Display info in  
tooltips SHOW\_VARIANT\_TOOLTIP

**Variants/Options** ☐

**Display** ☐ Display

Settings ☐ Display type in  
expressions DISPLAY\_TYPE\_EXPRESSION

**Variants/Options** ☐

**Display** ☐ Display

Settings ☐ Display all  
values in texts VAR\_SHOW\_ALL\_VALUES

**Variants/Options** ☐

**Display** ☐ Display

Settings ☐ Mark  
availability of different  
active attribute values (  
CheckBox ) MARK\_DIFFERENT\_ACTIVE\_VARIANT\_VALUES

**Variants/Options** ☐

**Display** ☐ Display

Settings ☐ Mark  
availability of different  
active attribute values (  
EditBox ) ALTERNATIVE\_VARIANT\_STRING

**Variants/Options** ☐

**Display** ☐ Display

Settings ☐ Highlight color  
for inactive  
variants/options (  
CheckBox ) DISPLAY\_OTHER\_VARIANTS\_GRAYED

**Variants/Options** ☐

**Display** ☐ Display

Settings ☐ Highlight color  
for inactive  
variants/options (  
Variant\_Inactive\_Color ) VARIANT\_INACTIVE\_COLOR

ComboBox )

**Variants/Options** □

**Display** □ Display

Settings □ Display

elements with

variants/options in

another color (

CheckBox )

DRAW\_VAR\_OTHER\_COLOR

**Variants/Options** □

**Display** □ Display

Settings □ Display

elements with

variants/options in

another color (

ComboBox )

VARIANT\_COLOR

**Variants/Options** □

**Display** □ Display

Settings □ Display

elements without

variants/options in

another color (

CheckBox )

DRAW\_OBJECTS\_WITHOUT\_VARIANTS\_OPTIONS\_IN\_OTHER\_CO

**Variants/Options** □

**Display** □ Display

Settings □ Display

elements without

variants/options in

another color (

ComboBox )

NO\_VARIANT\_COLOR

**Variants/Options** □

**Display** □ Display

Settings □ Highlight

variants/options in the

following color

VARIANT\_HIGHLIGHT\_COLOR

**Variants/Options** □

**Variant Text** □ Font □

Name

VARIANT\_TEXT\_FONT

**Variants/Options** □

**Variant Text** □ Font □

Style

VARIANT\_TEXT\_DIA

**Variants/Options** □

**Variant Text** □ Font □ Size

VARIANT\_TEXT\_SIZE

**Variants/Options** □

**Variant Text** □ Font □

Color

VARIANT\_TEXT\_COLOR

VARIANT\_TEXT\_MODE

**Variants/Options** □**Variant Text** □ Font □  
Ratio**Variants/Options** □**Variant Text** □ Font □ VARIANT\_TEXT\_JUST  
Alignment**Variants/Options** □**Variant Text** □ Effects □ VARIANT\_TEXT\_DIA  
Strikeout**Variants/Options** □**Variant Text** □ Effects □ VARIANT\_TEXT\_DIA  
Underline**Variants/Options** □**Option Inheritance** □Inherit Options of  
Sheet/Field/Hierarchical  
block □ No inheritance for INHERIT\_OPTIONS  
symbols/devices from  
sheet / for sheets from  
hierarchical block**Variants/Options** □**Option Inheritance** □Inherit Options of  
Sheet/Field/Hierarchical  
block □ Add sheet options INHERIT\_OPTIONS  
for symbols/devices /  
hierarchical block  
options for sheets**Variants/Options** □**Option Inheritance** □Inherit Options of  
Sheet/Field/Hierarchical  
block □ Only sheet options INHERIT\_OPTIONS  
for symbols/devices  
hierarchical block  
options for sheets**Variants/Options** □**Option Inheritance** □Inherit Options of  
Sheet/Field/Hierarchical INHERIT\_OPTIONS\_FIELD  
block □ No inheritance  
from field**Variants/Options** □

INHERIT\_OPTIONS\_FIELD

**Option Inheritance** □Inherit Options of  
Sheet/Field/Hierarchical  
block □ Add field options

**Variants/Options** □

**Option Inheritance** □

Inherit Options of INHERIT\_OPTIONS\_FIELD  
Sheet/Field/Hierarchical  
block □ Only field options

**Variants/Options** □

**Option Inheritance** □

Option Inherit when INHERIT\_OPTIONS\_OBJECTS  
Placing/Moving □ Symbols

**Variants/Options** □

**Option Inheritance** □

Option Inherit when INHERIT\_OPTIONS\_OBJECTS  
Placing/Moving □ Devices

**Variants/Options** □

**Option Inheritance** □

Visibility of Symbols in INHERIT\_OPTIONS\_VISIBILITY  
Tree □ Depending on  
visibility of sheet

**Variants/Options** □

**Locking** □ Password for  
unlocking the variant /  
option structure □ Old  
password

**Variants/Options** □

**Locking** □ Password for  
unlocking the variant /  
option structure □ New  
password

**Variants/Options** □

**Locking** □ Password for  
unlocking the variant /  
option structure □  
Confirm password

**Auto Routing** □ Router

Mode □ Horizontally after AUTOROUTE\_AFTER\_PLACE\_HORIZONTAL  
placing

**Auto Routing** □ Router

Mode □ Vertically after AUTOROUTE\_AFTER\_PLACE\_VERTICAL  
placing

**Auto Routing** □ Router

Mode □ After moving AUTOROUTE\_AFTER\_MOVE

**Auto Routing** □ Router

Settings □ Crossing □ Allow AUTOROUTE\_ALLOW\_CROSS\_CONNECTIONS  
crossing connections

AUTOROUTE\_COST\_CROSS\_CONNECTIONS

**Auto Routing** □ Router  
 Settings □ Crossing □ Allow  
 crossing connections □  
 Cost

**Auto Routing** □ Router  
 Settings □ Crossing □ Allow **AUTOROUTE\_ALLOW\_CROSS\_SYMBOLS**  
 crossing symbols

**Auto Routing** □ Router  
 Settings □ Crossing □ Allow **AUTOROUTE\_COST\_CROSS\_SYMBOLS**  
 crossing symbols □ Cost

**Auto Routing** □ Router  
 Settings □ Minimum **AUTOROUTE\_BEND\_DISTANCE\_FROM\_TOP**  
 Distance to First Turn □  
 Top [Grid size x value]

**Auto Routing** □ Router  
 Settings □ Minimum **AUTOROUTE\_BEND\_DISTANCE\_FROM\_BOTTOM**  
 Distance to First Turn □  
 Bottom [Grid size x value]

**Auto Routing** □ Router  
 Settings □ Minimum **AUTOROUTE\_BEND\_DISTANCE\_FROM\_LEFT**  
 Distance to First Turn □  
 Left [Grid size x value]

**Auto Routing** □ Router  
 Settings □ Minimum **AUTOROUTE\_BEND\_DISTANCE\_FROM\_RIGHT**  
 Distance to First Turn □  
 Right [Grid size x value]

**Auto Routing** □ Router  
 Settings □ Minimum **AUTOROUTE\_BEND\_COST**  
 Distance to First Turn □  
 Direction change cost

**Auto Routing** □ Router  
 Settings □ Sort □ Sort **AUTOROUTE\_CONNECTION\_LENGTH**  
 connections

If no setting functions are shown for a **E<sup>3</sup>.series** setting location, no functions are available and the setting value can only be retrieved or assigned using **E<sup>3</sup>.series** interactively.

---

## See Also

- [Electric Project Settings](#)
  - [Setting Value Names](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).





v2022-23.00

## Rotation String

# Syntax

*String* *rot*

## Description

Parameter represents a rotation value as a *String*.

## Possible Values

Value	Description
	Numerical values
"-n.n" to "n.n"	For example - "90", "-20.0", "380.5"
"x"	Mirror on the x-axis
	Value is case insensitive
"y"	Mirror on the y-axis
	Value is case insensitive
"/1"	Mirror on the y-axis
"/2"	Rotate 90 degrees
"/3"	Rotate 90 degrees and mirror on the x-axis
"/4"	Rotate 180 degrees
"/5"	Mirror on the x-axis
"/6"	Rotate 270 degrees
"/7"	Rotate 270 degrees and mirror on the x-axis

## Remarks

Positive values will cause anticlockwise rotation. Negative values will cause clockwise rotation.

Numerical values can be combined with "x" and "y". The mirroring and rotating is done in left to right order in the rotation [String](#). For example:

"x20y80" will mirror on the x-axis, rotate 20 degrees, mirror on the y-axis and rotate 80 degrees respectively.

The rotation parameter must not contain any invalid characters or formatting including white space characters.

## Version Information

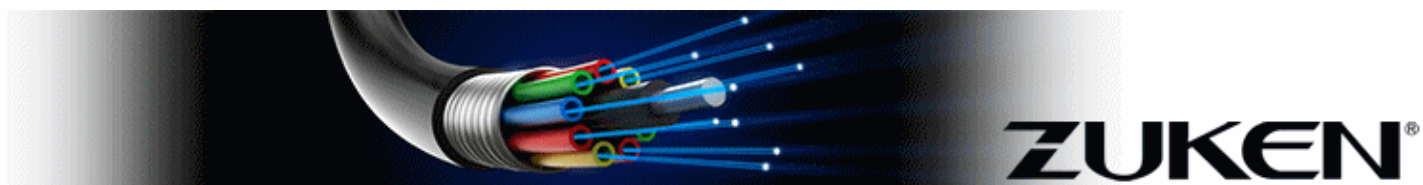
Introduced in v2018-19.41 and v2019-20.11.

---

## See Also

- [e3DbeModel.PlaceSymbol\(\)](#)
  - [e3DbeSymbol.PlaceSymbol\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



**v2022-23.00**

## Schematic

# Syntax

*IntegerSchematic*

## Description

Parameter represents a schematic value as an integer.

## Possible Values

Value	Description
0	Electric
1	Hydraulic
2	Pneumatic
3	Process, measurement and control
4	Tubes + instruments
5	Single Line Diagram
6	Panel Symbol
>= 100	User-defined types

## Remarks

The schematic types are defined in the database whether further types may be user-defined.

## Version Information

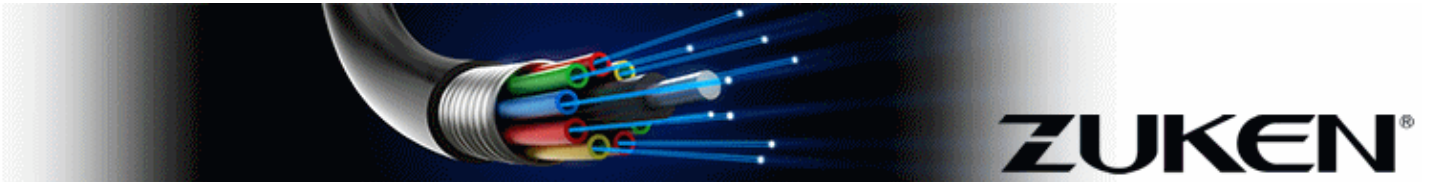
Introduced in v2009-8.50.

---

## See Also

- [e3Device.GetSchematicTypes\(\)](#)
  - [e3Sheet.GetSchematicTypes\(\)](#)
  - [e3Sheet.SetSchematicTypes\(\)](#)
  - [e3Symbol.GetSchematicTypes\(\)](#)
  - [e3Tree.GetVisibleInfoTypesEx\(\)](#)
  - [e3Tree.SetVisibleInfoTypesEx\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

Setting Value Names

Syntax

String *name*

Description

Parameter represents a unique name identifying the **E<sup>3</sup>** setting value.

Possible Values

If the setting name is known and the location of the setting within **E<sup>3</sup>.series** is required, please reference the following table. If the location of the setting within **E<sup>3</sup>.series** is known and the setting value name is required, please refer to Electric Project Settings or Fluid Project Settings.

Name	Location in
2D_MODEL_ORIENTATION	<b>Settings</b> □ Display mode orientation
3D_DISPLAY_STEP_MODELS	<b>Settings</b> □ Display STEP 3D
3D_LIGHTING	<b>Settings</b> □ Enable light
3D_LOAD_STEP_MODELS_IF_REQUIRED	<b>Settings</b> □ Load STEP database on

3D\_MODEL\_GRAPHIC

**Settings** ☐ **3D Model Graphic**  
Display model from 2D in 3D

3D\_MODEL\_ORIENTATION

**Settings** ☐ **3D Model Orientation**  
Display model orientation

3D\_ORTHOGRAPHIC

**Settings** ☐ **3D Orthographic**  
Orthographic

3D\_WIREFRAME

**Settings** ☐ **3D Wireframe**  
Wireframe

ACTUAL\_HIGHLIGHT\_WIDTH

**Settings** ☐ **Highlight Width**  
Highlight Width

ADJUST\_BUNDLE\_SYMBOL\_SIZE\_AFTER\_MODIFYING\_CONNECTION\_LINE

**Electric Symbol Placement**  
**Bundle Symbol**  
Adjust bundle size after modifying connection line position

AIRLINE\_DISPLAY\_LAID\_CORES\_ALSO

**Electric Symbol**  
**Connection**  
**Cable/Conductor Lines** ☐ **Display**  
laid conductors

AIRLINE\_DISPLAY\_OPEN\_SIGNAL\_CONNECTIONS

AIRLINE\_DISPLAY\_UNCONNECTED\_CORES

AIRLINE\_MARK\_DIRECTION

AIRLINES\_OFFX\_CABCAB

AIRLINES\_OFFX\_CABSIG

**Fluid Settings**  
**Connection**  
**Hose/Tube**  
□ Display □  
hoses/tubes

**Settings** □  
**Signal Log**  
Display open  
connections

**Electric Settings**  
**Connection**  
**Cable/Conc**  
**Lines** □ Disj  
unconnected

**Fluid Settings**  
**Connection**  
**Hose/Tube**  
□ Display □  
unconnected

**Electric Settings**  
**Connection**  
**Cable/Conc**  
**Lines** □ Line  
direction

**Fluid Settings**  
**Connection**  
**Hose/Tube**  
□ Lines □ M

**Electric Settings**  
**Connection**  
**Cable/Conc**  
**Lines** □ Line  
name of con  
X-Offset

**Fluid Settings**  
**Connection**  
**Hose/Tube**  
□ Lines □ Sh  
hose/tube □

**Electric Settings**  
**Connection**  
**Cable/Conc**  
**Lines** □ Line  
signal name

AIRLINES\_OFFY\_CABCAB

AIRLINES\_OFFY\_CABSIG

AIRLINES

**Fluid Settings**  
**Connection**  
**Hose/Tube**  
☐ Lines ☐ Sh  
name ☐ X-Of  
**Electric Settings**  
**Connection**  
**Cable/Conc**  
**Lines** ☐ Line  
name of con  
Y-Offset

**Fluid Settings**  
**Connection**  
**Hose/Tube**  
☐ Lines ☐ Sh  
hose/tube ☐  
**Electric Settings**  
**Connection**  
**Cable/Conc**  
**Lines** ☐ Line  
signal name

**Fluid Settings**  
**Connection**  
**Hose/Tube**  
☐ Lines ☐ Sh  
name ☐ Y-Of

**Electric Settings**  
**Connection**  
**Logic Lines**  
open signal

**Electric Settings**  
**Connection**  
**Cable/Conc**  
**Lines** ☐ Disj  
unconnected

**Electric Settings**  
**Connection**  
**Cable/Conc**  
**Lines** ☐ Disj  
conductors

**Electric Settings**  
**Connection**  
**Cable/Conc**  
**Lines** ☐ Line



unconnected  
logic lines a

**Electric Se**  
**Connection**  
**Cable/Conc**  
**Lines** □ Line  
direction

**Electric Se**  
**Connection**  
**Cable/Conc**  
**Lines** □ Line  
Views □ Use

**Electric Se**  
**Connection**  
**Cable/Conc**  
**Lines** □ Line  
name of con

**Electric Se**  
**Connection**  
**Cable/Conc**  
**Lines** □ Line  
name of con  
Position

**Electric Se**  
**Connection**  
**Cable/Conc**  
**Lines** □ Line  
name of con  
□ Font Style

**Electric Se**  
**Connection**  
**Cable/Conc**  
**Lines** □ Line  
signal name

**Electric Se**  
**Connection**  
**Cable/Conc**  
**Lines** □ Line  
signal name

**Electric Se**  
**Connection**  
**Cable/Conc**

**Lines** ☐ Line  
signal name  
Font Style

**Fluid Setting**  
**Connection**  
**Hose/Tube**  
☐ Display ☐   
unconnected

**Fluid Setting**  
**Connection**  
**Hose/Tube**  
☐ Display ☐   
also

**Fluid Setting**  
**Connection**  
**Hose/Tube**  
☐ Lines ☐ Di  
logic lines a

**Fluid Setting**  
**Connection**  
**Hose/Tube**  
☐ Lines ☐ M

**Fluid Setting**  
**Connection**  
**Hose/Tube**  
☐ Lines ☐ Or  
Used views

**Fluid Setting**  
**Connection**  
**Hose/Tube**  
☐ Lines ☐ Sh  
hose/tube

**Fluid Setting**  
**Connection**  
**Hose/Tube**  
☐ Lines ☐ Sh  
hose/tube ☐

**Fluid Setting**  
**Connection**  
**Hose/Tube**  
☐ Lines ☐ Sh  
hose/tube ☐

AIRLINE\_SHOW\_NAME\_OF\_CORE

AIRLINE\_SHOW\_NAME\_OF\_CORE\_FONT\_BOLD

AIRLINE\_SHOW\_NAME\_OF\_CORE\_FONT\_ITALIC

Possible Values

Font Style

**Fluid Settings**  
**Connection**  
**Hose/Tube**  
☐ Lines ☐ Show  
name

**Fluid Settings**  
**Connection**  
**Hose/Tube**  
☐ Lines ☐ Show  
name ☐ Position

**Fluid Settings**  
**Connection**  
**Hose/Tube**  
☐ Lines ☐ Show  
name ☐ Font Style

**Electric Settings**  
**Formboard**  
Display unconnected  
conductors

**Electric Settings**  
**Connection**  
**Cable/Conductor**  
**Lines** ☐ Line  
name of conductor

**Fluid Settings**  
**Connection**  
**Tube Logic**  
Lines ☐ Show  
hose/tube

**Electric Settings**  
**Connection**  
**Cable/Conductor**  
**Lines** ☐ Line  
name of conductor  
☐ Font Style

**Fluid Settings**  
**Connection**  
**Hose/Tube**  
☐ Lines ☐ Show  
hose/tube ☐  
Style

**Electric Settings**

AIRLINE\_SHOW\_NAME\_OF\_CORE\_POSITION\_CENTERED

**Connection  
Cable/Conc  
Lines** □ Line  
name of con  
□ Font Style

**Fluid Setti  
Connection  
Hose/Tube**  
□ Lines □ Sh  
hose/tube □  
Style

**Electric Se  
Connection  
Cable/Conc  
Lines** □ Line  
name of con  
Position

**Electric Se  
Connection  
Cable/Conc  
Lines** □ Line  
of conductor  
endpoints

**Electric Se  
Connection  
Cable/Conc  
Lines** □ Line  
name of con  
centered

**Fluid Setti  
Connection  
Hose/Tube**  
□ Lines □ Sh  
hose/tube □

**Fluid Setti  
Connection  
Hose/Tube**  
□ Lines □ Sh  
hose/tube □

**Fluid Setti  
Connection  
Hose/Tube**  
□ Lines □ Sh  
hose/tube □

AIRLINE\_SHOW\_SIGNAL\_NAME

AIRLINE\_SHOW\_SIGNAL\_NAME\_FONT\_BOLD

AIRLINE\_SHOW\_SIGNAL\_NAME\_FONT\_ITALIC

AIRLINE\_SHOW\_SIGNAL\_NAME\_POSITION\_CENTERED

**Electric Se**  
**Connection**  
**Cable/Conc**  
**Lines** □ Line  
signal name

**Fluid Setti**  
**Connection**  
**Hose/Tube**  
□ Lines □ Sh  
name

**Electric Se**  
**Connection**  
**Cable/Conc**  
**Lines** □ Line  
name of con  
□ Font Style

**Fluid Setti**  
**Connection**  
**Hose/Tube**  
□ Lines □ Sh  
hose/tube □  
Style

**Electric Se**  
**Connection**  
**Cable/Conc**  
**Lines** □ Line  
name of con  
□ Font Style

**Fluid Setti**  
**Connection**  
**Hose/Tube**  
□ Lines □ Sh  
hose/tube □  
Style

**Electric Se**  
**Connection**  
**Cable/Conc**  
**Lines** □ Line  
signal name

**Electric Se**  
**Connection**  
**Cable/Conc**  
**Lines** □ Line  
signal name  
endpoints

AIRLINES\_LINMOD\_CAB

AIRLINES\_LINMOD\_SIG

AIRLINES\_TXTFNT\_CABCAB

AIRLINES\_TXTFNT\_CABSIG

Possible Values

**Electric Se**  
**Connection**  
**Cable/Conc**  
**Lines** □ Line  
signal name

**Fluid Setti**  
**Connection**  
**Hose/Tube**  
□ Lines □ Sh  
name □ Posi

**Fluid Setti**  
**Connection**  
**Hose/Tube**  
□ Lines □ Sh  
name □ To e

**Fluid Setti**  
**Connection**  
**Hose/Tube**  
□ Lines □ Sh  
name □ cent

**Electric Se**  
**Connection**  
**Cable/Conc**  
**Lines** □ Line

**Fluid Setti**  
**Connection**  
**Hose/Tube**  
□ Line □ Lin

**Settings** □ (   
**Signal Log**  
Display ope  
connections

**Electric Se**  
**Connection**  
**Cable/Conc**  
**Lines** □ Line  
name of con  
□ Font

**Fluid Setti**  
**Connection**  
**Hose/Tube**  
□ Lines □ Sh  
hose/tube □

AIRLINES\_TXTMOD\_CABCAB

AIRLINE\_TXTSIZ\_CABCAB

AIRLINE\_USED\_VIEWS\_ONLY

ALIGN\_HORIZONTAL\_DIST

Possible Values

**Electric Se**  
**Connection**  
**Cable/Conc**  
**Lines** □ Line  
name of con  
□ Font

**Fluid Setti**  
**Connection**  
**Hose/Tube**  
□ Lines □ Sh  
hose/tube □

**Electric Se**  
**Connection**  
**Cable/Conc**  
**Lines** □ Line  
name of con  
□ Ratio

**Fluid Setti**  
**Connection**  
**Hose/Tube**  
□ Lines □ Sh  
hose/tube □

**Electric Se**  
**Connection**  
**Cable/Conc**  
**Lines** □ Line  
name of con  
□ Size

**Fluid Setti**  
**Connection**  
**Hose/Tube**  
□ Lines □ Sh  
hose/tube □

**Electric Se**  
**Connection**  
**Cable/Conc**  
**Lines** □ Line  
Views □ Use

**Fluid Setti**  
**Connection**  
**Hose/Tube**  
□ Lines □ Or  
Used views

**Settings** □  
Align Distan

ALIGN\_VERTICAL\_DIST

**Settings** ☐ **Align Distance**

ALLOW\_INSERT\_SYMBOL\_IN\_CONNECTION

**Settings** ☐ **Autoconnect**  
Autoconnecting symbols  
inserting symbols in  
connection

ALLOW\_ONLY\_COMPATIBLE\_PIN\_GENDERS\_TO\_PLUG

**Settings** ☐ **Pins** ☐ **Allow**  
Pins ☐ Allow  
compatible pins  
to plug

ALLOW\_ONLY\_VALID\_MATING\_CONNECTORS\_TO\_PLUG

**Settings** ☐ **Pins** ☐ **Allow**  
Pins ☐ Allow  
mating connectors  
plug

ALLOW\_SAME\_CONDES\_ON\_BLOCKS

**Settings** ☐ **Block Device**  
Block Device  
Allow same  
designation  
blocks

ALLOW\_SIGNAL\_CHANGES\_OF\_DIFFERENT\_CLASSES

**Settings** ☐ **Signals** ☐ **S**  
**Signals** ☐ S  
Allow signal  
signals not in  
same class

ALT\_COMPCODE

**Settings** ☐ **Display** ☐ **S**  
**Display** ☐ S  
☐ Alternative  
component  
ComboBox )



ALT\_COMPCODE\_ON

**Settings** □  
**Display** □ S  
 □ Alternativ  
 component  
 CheckBox )

ALTERNATIVE\_VARIANT\_STRING

**Variants/O**  
**Display** □ D  
 Settings □ M  
 availability c  
 active attrib  
 EditBox )

ASSIGN\_ALL\_CORES\_TO\_NEW\_CABLE

**Electric Se**  
**Connection**  
 / Wires □ As  
 conductors  
 route to a n  
 when insert  
 connector

ASSIGN\_ASSEMBLY\_DEVICE\_MODE

**Settings** □ .  
**Change Co**  
 Assembly de  
 component

ASSIGNED\_FUBLOCK\_FILL\_COLOR

**Settings** □ .  
**Change Co**  
 Assembly de  
 order of dev  
 assembly

ASSIGNED\_FUDPOINT\_FILL\_COLOR

**Electric Se**  
**Topology** □  
 color for ass  
 Installation

**Electric Se**  
**Functional**  
 Display □ Fi  
 assigned fun  
 blocks

**Electric Se**  
**Topology** □  
 color for ass  
 disconnectin

ASSIGN\_GATE\_MODE

ASSIGN\_PINS\_VIA\_NAMES

**Electric Se**  
**Functional**  
Display □ Fi  
assigned dis  
points

**Settings □**  
**Change Co**  
Assignment  
of pins

**Settings □**  
**Change Co**  
Assignment  
internal dev  
designation

**Electric Se**  
**Placement**  
**Component**  
Assignment  
of symbol, c  
pin

**Fluid Setti**  
**Placement**  
**Component**  
Assignment  
of symbol, h  
pin

**Settings □**  
**Change Co**  
Assignment  
symbol nam  
and signal

**Electric Se**  
**Connection**  
**Connectors**  
Connectors  
assign pins  
first

**Fluid Setti**  
**Connection**  
Mating part  
assign pins  
first

AUTOCON\_DIR

**Settings** ☐ **Autoconnect**  
direction ☐ V  
connections  
bottom)

AUTOCON\_LINES

**Settings** ☐ **Autoconnect**  
direction ☐ H  
connections  
right)

AUTOMATIC\_GENERATED\_WIRE\_NAMES

**Settings** ☐ **Autoconnect**  
after deleting

AUTOROUTE\_AFTER\_MOVE

**Electric Se**  
**Connection**  
**Conductor**  
**Naming** ☐ V  
Assign wire  
automatical

AUTOROUTE\_AFTER\_PLACE\_HORIZONTAL

**Settings** ☐ **Router Mo**  
☐ Router Mo  
moving

AUTOROUTE\_AFTER\_PLACE\_VERTICAL

**Settings** ☐ **Router Mo**  
☐ Router Mo  
Horizontally  
placing

AUTOROUTE\_ALLOW\_CROSS\_CONNECTIONS

**Settings** ☐ **Router Mo**  
☐ Router Mo  
after placing

AUTOROUTE\_ALLOW\_CROSS\_SYMBOLS

**Settings** ☐ **Router Se**  
☐ Router Se  
Crossing ☐ A  
connections

AUTOROUTE\_BEND\_COST

**Settings** ☐ .  
☐ Router Se  
Minimum D  
First Turn ☐  
change cost

AUTOROUTE\_BEND\_DISTANCE\_FROM\_BOTTOM

**Settings** ☐ .  
☐ Router Se  
Minimum D  
First Turn ☐  
size x value,

AUTOROUTE\_BEND\_DISTANCE\_FROM\_LEFT

**Settings** ☐ .  
☐ Router Se  
Minimum D  
First Turn ☐  
size x value,

AUTOROUTE\_BEND\_DISTANCE\_FROM\_RIGHT

**Settings** ☐ .  
☐ Router Se  
Minimum D  
First Turn ☐  
size x value,

AUTOROUTE\_BEND\_DISTANCE\_FROM\_TOP

**Settings** ☐ .  
☐ Router Se  
Minimum D  
First Turn ☐  
size x value,

AUTOROUTE\_CONNECTION\_LENGTH

**Settings** ☐ .  
☐ Router Se  
Sort connec

AUTOROUTE\_COST\_CROSS\_CONNECTIONS

**Settings** ☐ .  
☐ Router Se  
Crossing ☐ A  
connections

AUTOROUTE\_COST\_CROSS\_SYMBOLS

**Settings** ☐ .  
☐ Router Se  
Crossing ☐ A  
symbols ☐ C

AUTOSAVE\_ENABLED

**Settings** ☐ .  
☐ Automatic  
backup file  
CheckBox )

BLCON\_PINVIEW\_SYM\_NAME

**Electric Se  
Placement**

BLOCK\_DES\_DEFAULT

BLOCKEDGE\_COD

BLOCKEDGE\_DIA

BLOCKEDGE\_MOD

BLOCKEDGE\_TYPE

**Symbol Vie**  
Symbols □ F  
connector p

**Fluid Setti**  
**Placement**  
**Symbol Vie**  
Symbols □ F  
fitting pins

**Settings □ .**  
Default Des  
Blocks

**Settings □ .**  
**Block □ Spl**  
Color

**Settings □ .**  
**Block □ Spl**  
Width

**Settings □ .**  
**Block □ Spl**  
Line Style

**Settings □ .**  
**Block □ Spl**  
split blocks

**Settings □ .**  
**Block □ Spl**  
split blocks

**Settings □ .**  
**Block □ Spl**  
split blocks

**Settings □ .**  
**Block □ Spl**  
split blocks

**Settings □ .**  
**Block □ Spl**  
split blocks

BLOCKNAME\_TO\_CONNECTOR

**Settings** ☐   
Block Device   
Use name of   
devices as

**Settings** ☐   
Block Device   
Use name of   
devices as ☐   
assignment

**Settings** ☐   
Block Device   
Use name of   
devices as ☐

BLOCKREFCOD

**Settings** ☐   
**Block** ☐ Col

BLOCKREFDIA

**Settings** ☐   
**Block** ☐ For

BLOCKREFFONTPTR

**Settings** ☐   
**Block** ☐ For

BLOCKREFGAP

**Settings** ☐ .  
**Block** ☐ Dis  
Gap

**Settings** ☐ .  
**Block** ☐ Dis  
Direction ☐

**Settings** ☐ .  
**Block** ☐ Dis  
Direction ☐

BLOCKREFJUST


**Settings** ☐ .  
**Block** ☐ Dis  
Rotate

**Settings** ☐ .  
**Block** ☐ Dir

**Settings** ☐ .  
**Block** ☐ Dir

**Settings** ☐ .  
**Block** ☐ Dir

BLOCKREFLAY

**Settings**  **Block**  Dis  
Level

BLOCKREFMOD

**Settings**  **Block**  Rat

**Settings**  **Block**  Rat

**Settings**  **Block**  Rat

BLOCKREFSETTING

**Settings**  **References**  
Prefix

**Settings**  **References**  
Suffix

**Settings**  **References**  
Sheet text

**Settings**  **References**  
Reference to

BLOCKREFSIZ

**Settings**  **Block**  Siz



BLOCKREFSORT

**Settings** □  
**Block** □ Type

**Settings** □  
**Block** □ Type

**Settings** □  
**Block** □ Type

BUNDLE\_BOTTOM\_OVERLAPPING

**Electric Se**  
**Placement**  
**Bundle Sym**  
Parameters  
automatic p  
Line overlap  
/ Bottom

BUNDLE\_DISTANCE\_MIDDLE

**Electric Se**  
**Placement**  
**Bundle Sym**  
Parameters  
automatic p  
Minimum di  
between bu

BUNDLE\_DISTANCE\_PIN

**Electric Se**  
**Placement**  
**Bundle Sym**  
Parameters  
automatic p  
Minimum di

BUNDLE\_TOP\_OVERLAPPING

**Electric Se**  
**Placement**  
**Bundle Sym**  
Parameters  
automatic p  
Line overlap  
/ Top

BUSBARLINCOD

BUSBARLINDIA

BUSBARLINLEV

BUSBARLINMOD

CABLE\_DES\_DEFAULT

CALC\_WIRE\_LENGTH

CC\_BUNDLE\_SYMBOLS

CC\_KEEP\_COUNTERPARTS

**Electric Se**  
**Connection**  
**Lines** □ Bus

**Electric Se**  
**Connection**  
**Lines** □ Bus

**Electric Se**  
**Connection**  
**Lines** □ Bus

**Electric Se**  
**Connection**  
**Lines** □ Bus

**Electric Se**  
**Placement**  
Designation

**Fluid Setti**  
**Placement**  
Designation

**Electric Se**  
**General** □ C  
Electrical ca  
Alternative  
wire length

**Electric Se**  
**Placement**  
**Component**  
Symbols □ U  
symbol even  
symbol used  
the one defi  
database

**Electric Se**  
**Placement**  
**Component**  
□ Keep activ  
connectors  
parts

**Fluid Setti**  
**Placement**  
**Component**  
Keep active  
fitting parts

CC\_PLACE\_SINGLE\_PINS

CC\_RELOAD\_ATTRIBUTES

CC\_RESTORE\_PINNAMES

CC\_TERMPLAN\_TABLE\_SYMBOL

CELLEV

CELL\_SCAFACTOR

CHANGE\_COMPLETE\_DEVICE

CHANGE\_COMPOSITES\_COMPONENTS

**Electric Se**  
**Placement**  
**Component**  
□ Place all p  
pins

**Fluid Setti**  
**Placement**  
**Component**  
Place all pin  
pins

**Settings** □  
**Change Co**  
Attributes □  
attribute va  
devices and

**Settings** □  
**Change Co**  
Pins □ Resto  
pin names

**Electric Se**  
**Placement**  
**Component**  
Symbol for "  
Plan □ Upda  
symbol even  
symbol is ch  
differing fro  
symbol in th

**Settings** □  
**Symbols** □  
Parameters

**Settings** □  
**Symbols** □  
Parameters  
factor

**Settings** □  
**Symbols** □  
Change com  
when chang  
text for high  
assignment,  
device desig

**Electric Se**  
**General** □ **U**  
**Project** □ **C**

Change already existing connecting connector to new active connector

**Fluid Settings**  
**General** ☐ **Units**  
**Project** ☐ For already used parts to the new part

**Settings** ☐ **Rules** ☐ Rename when changing designation sheet/field

**Settings** ☐ **Purge** ☐ Remove unused objects from project before

**Electric Settings**  
**Connections** ☐ / Wires ☐ Clear after unconnected conductor/wire disconnected

**Fluid Settings**  
**Connections** ☐ Hoses/Tubes ☐ Clear signal after hose/tube at disconnected

**Settings** ☐ **Signals** ☐ Symbols ☐ Clear when deleting

**Settings** ☐ **Signals** ☐ Symbols ☐ For connected view

CHANGE\_DEVICES\_ON\_SHEET

CLEANUP\_WHEN\_SAVING

CLEAR\_SIGNAL\_AT\_PIN\_AFTER\_UNCONNECT\_CORE

CLEAR\_SIG\_ON\_DEL\_SYMBOL

CLEAR\_SIG\_ON\_PLACED\_VIEW\_SYMBOL

COLOUR\_IGNORED\_CONNECT\_LINE

COMPONENT\_TYPE\_ATTRIBUTES

CONN\_CELL\_GAP

CONNECT\_AND\_USE\_DEFAULT\_HOSE

CONNECT\_AND\_USE\_DEFAULT\_WIRE

are placed

**Electric Se  
Connection  
Lines** □ Ign  
Cabling Tab  
ignored con  
alternative c  
ComboBox )

**Settings** □ (   
**Component  
Attributes**  
Attributes

**Electric Se  
Placement**  
Connector S  
when placin  
single pins

**Fluid Setti  
Connection**  
and use Hos  
connect only

**Fluid Setti  
Connection**  
and use Hos  
connect and  
hose/tube

**Electric Se  
Connection**  
and Wire □ c  
graphically

**Electric Se  
Connection**  
and Wire □ c  
use default

CONNECT\_LINE\_SYMBOL\_HOR

CONNECT\_LINE\_SYMBOL\_VER

CONNECTOR\_PIN\_TERMINAL\_FILTER\_ATTRIBUTE

CONNECTOR\_PIN\_TERMINAL\_FILTER\_ATTRIBUTE\_VALUE

CONNECTOR\_SYMBOL\_FORMAT

**Settings** □  
**Connect Li**  
Template Sy  
Horizontal

**Settings** □  
**Connect Li**  
Template Sy  
Vertical

**Electric Se**  
**Connection**  
**Conductor**  
Conductor A  
Procedure □  
filter attribu  
connector p

**Electric Se**  
**Connection**  
**Conductor**  
Conductor A  
Procedure □  
filter attribu  
connector p  
Default valu  
attribute

**Electric Se**  
**Placement**  
**Symbols** □  
Visibility

CONNECT\_SIGNAL

CONNECT\_SIGNAL\_VIEWS

CONNECT\_UNIQUE\_CORE\_NUMBER

CONNECT\_WIRE\_FLUID\_CHECK

CONN\_PINVIEW\_SYM\_NAME

COPY\_EXPORT\_CABLE

**Settings** ☐ **Signals** ☐ S  
Connections  
connections  
transfer sign

**Settings** ☐ **Signals** ☐ S  
Connections  
signals on c  
between vie

**Electric Se**  
**Connection**  
**Conductors**  
**Naming** ☐ C  
Wire Names  
unique nam

**Fluid Setti**  
**Connection**  
Hoses/Tube  
pneumatic/h

**Electric Se**  
**Placement**  
**Symbol Vie**  
Symbols ☐ F  
pins

**Fluid Setti**  
**Placement**  
**Symbol Vie**  
Symbols ☐ F

**Electric Se**  
**Placement**  
**Export/Cop**  
wires ☐ Cab

COPY\_EXPORT\_CABLE\_OPTION

**Fluid Settings  
Placement  
Export/Copy**  
Hoses/Tubes  
Hoses/Tubes  
**Electric Settings  
Placement  
Export/Copy**  
wires ☐ At least one  
selected

**Electric Settings  
Placement  
Export/Copy**  
wires ☐ Both  
selected

**Electric Settings  
Placement  
Export/Copy**  
wires ☐ Both  
path are selected

**Electric Settings  
Placement  
Export/Copy**  
wires ☐ Full  
selected

**Fluid Settings  
Placement  
Export/Copy**  
Hoses/Tubes  
one end selected

**Fluid Settings  
Placement  
Export/Copy**  
Hoses/Tubes  
selected

**Fluid Settings  
Placement  
Export/Copy**  
Hoses/Tubes  
and the path  
selected

**Fluid Settings  
Placement**



CORRECTION\_FACTOR\_FOR\_SPACE\_REQUIREMENTS

CREATE\_BLCON\_WITH\_FLOW

CREATE\_CONN\_WITH\_FLOW

CREATE\_ORIGIN

CUT\_FIELD\_BORDER

CUT\_FIELD\_BORDER\_GAP

Possible Values

**Export/Cop**  
Hoses/Tube  
selected

**Settings** ☐ **Duct Symbol**  
Fill Size ☐ C  
factor for sp  
requiremen

**Electric Se**  
**Connection**  
Signal Flow  
Connectors  
signal flow c  
connectors

**Fluid Setti**  
**Connection**  
Signal Flow  
Interrupt sig  
block fitting

**Electric Se**  
**Connection**  
Signal Flow  
Connectors  
signal flow c  
connectors

**Fluid Setti**  
**Connection**  
Signal Flow  
Interrupt sig  
fittings

**Settings** ☐ **Symbols** ☐  
graphic ☐ C

**Settings** ☐ **Field** ☐ Out.  
field border  
connect line  
border

**Settings** ☐ **Field** ☐ Out.  
field border  
connect line

DEFAULT\_BLCONN

**Electric Se  
Placement**  
Connector S  
block conne  
component

DEFAULT\_BUNDLESYM

**Electric Se  
Placement  
Bundle Sym**  
Default sym  
DragDrop a  
Grouped sym

DEFAULT\_DESIGNATION\_FOR\_JACKS

**Electric Se  
Connection  
Connectors**  
designation  
Default desi  
jacks

DEFAULT\_DESIGNATION\_FOR\_PLUGS

**Fluid Setti  
Connection**  
Device desi  
Generation  
designation  
**Electric Se  
Connection  
Connectors**  
designation  
Default desi  
plugs

DEFAULT\_INSTALLATION\_SPACE\_SYMBOL

**Fluid Setti  
Connection**  
Device desi  
Generation  
designation  
parts  
**Electric Se  
Topology □**  
Installation

DEFAULT\_SHIELDSYM

DEFAULT\_TWISTED\_PAIRSYM

DEFAULT\_WIRESYM

DELETE\_CORES\_ON\_DEL\_CLINE

DELETE\_SIGNAL\_ON\_DEL\_CLINE

**Electric Se**  
**Functional**  
Symbols □ I  
space symbol

**Electric Se**  
**Placement**  
**Bundle Sym**  
Default sym  
DragDrop a  
symbol

**Electric Se**  
**Placement**  
**Bundle Sym**  
Default sym  
DragDrop a  
Twisted pair

**Electric Se**  
**Placement**  
Connector S  
connector w  
component

**Electric Se**  
**Connection**  
**Lines** □ Del  
graphical re  
□ Conductor  
project

**Fluid Setti**  
**Connection**  
**Lines** □ Del  
graphical re  
□ Hose/Tube

**Settings** □  
**Connect Li**  
with graphi  
representat

DELETE\_UNUSED\_ATTRIBUTES\_DURING\_CHANGE

**Settings** □  
**Change Co**  
Attributes □  
unused attri  
devices and

DELETE\_UNUSED\_ATTRIBUTES\_DURING\_UPDATE

**Settings** □  
**Update in**  
Attributes □  
unused attri  
devices and

DENY\_PLUG\_PINS\_OF\_SAME\_DEVICE

**Settings** □  
Pins □ Deny  
of same dev

DENY\_PLUG\_PINS\_WITH\_DIFF\_PINNAMES

**Settings** □  
Pins □ Deny  
with differ

DEVICE\_DESIGNATION\_OF\_CONNECTION\_TARGET

**Settings** □  
**Display** □ S  
□ Add intern  
designation  
connection t

DEV\_PINVIEW\_SYM\_NAME

**Settings** □  
**Symbols** □  
**Views** □ Pin  
□ For device

DF\_CBHEAD\_PANEL\_AUTOCONNECT\_BIGGER\_CONNECTOR\_PIN\_TERMINAL

**Electric Se**  
**Panel** □ Co  
Autoconnec  
Find suitabl  
pin terminal  
additional w

**Fluid Setti**  
**Connection**  
Autoconnec  
Find suitabl  
an additiona

DFI\_DEFDES\_COPY\_POSTFIX

**Settings** ☐ **Import** ☐ **D**  
Generate It  
Designation  
EditBox )

Before v202

DFI\_MERGE\_ATTRIBUTES

**Settings** ☐ **Import** ☐ **D**  
Additional M  
Options ☐ M  
attributes

From v2022

DFI\_MERGE\_TERMINALS

**Settings** ☐ **Import** ☐ **D**  
Attribute Op  
Project pref

**Electric Se**  
**Placement**  
**Device** ☐ M  
Options ☐ U  
terminal str

DFI\_USE\_DEFDES\_COPY\_POSTFIX

**Settings** ☐ **Import** ☐ **D**  
Generate It  
Designation  
Checkbox )

DFI\_USE\_EXISTING\_ASSEMBLIES

**Settings** ☐ **Import** ☐ **D**  
Device Opti  
existing ass

DFI\_USE\_EXISTING\_DEVICES

**Settings** ☐ **Import** ☐ **D**  
Device Opti  
existing dev

DF\_MERGE\_ALPHANUMERIC\_REFERENCES

**Settings** ☐ **Import** ☐ M  
Reference C  
Merge only  
names conta  
special char

DF\_MERGE\_SHEET\_REFERENCES

**Settings** ☐ **Import** ☐ M  
Reference C  
Merge shee

DIM\_ARROW\_TYP

**Settings** ☐ **Import** ☐ **Lines** ☐ **Arro**

DIM\_ARROW\_WIDTH

**Settings** ☐ **Import** ☐ **Lines** ☐ **Arro**



DIM\_COLOR

**Settings**    
General 

DIM\_DISPLAY\_ATTR

**Settings**    
Text 

DIMENSION\_TEXT\_CENTER

**Settings**    
Text 




DIM\_EXTENSION

**Settings**    
Lines 

DIM\_EXTENSION\_LINE\_OFFSET

**Settings**    
Lines  Extension  
offset

DIM\_FIX\_SIZE

**Settings**    
Lines  Use  
display




DIM\_HIDE\_LONGER\_PART

**Settings**    
Lines  Hide  
of arrow


DIM\_LEVEL

**Settings**    
General 

DIM\_PRECISION

**Settings**    
Text 

DIM\_PREFIX

**Settings**    
Text 

DIM\_ROTATE\_TEXT\_OF\_RUNNING\_DIM

DIM\_SUFFIX

**Settings** [ ]  
Text [ ] Rotat  
running dim

DIM\_SUFFIX\_SIZE\_FACTOR

**Settings** [ ]  
Text [ ] Suffi  
(%)

DIM\_TXT\_COLOR

**Settings** [ ]  
**Dimension**  
Color

DIM\_TXT\_DIA

**Settings** [ ]  
**Dimension**  
Style

DIM\_TXT\_FONTPTR

**Settings** [ ]  
**Dimension**  
Effects [ ] Str

DIM\_TXT\_OFFSET

**Settings** [ ]  
**Dimension**  
Effects [ ] Un

DIM\_TXT\_SIZE

**Settings** [ ]  
**Dimension**  
Effects [ ] Op

DIM\_WIDTH

**Settings** [ ]  
**Dimension**  
Name

**Settings** [ ]  
Text [ ] Offse

**Settings** [ ]  
**Dimension**  
Size

**Settings** [ ]  
Lines [ ] Line



DISPLAY\_BARRING\_LINES

**Electric Se**  
**Connection**  
**Logic Lines**  
negative log  
with barred

DISPLAY\_CONNECTION\_TYPES

**Menu** ☐ **Vie**  
Connection

DISPLAY\_FORMBOARD\_REGION

**Electric Se**  
**Formboard**  
sheets ☐ Dis  
overview

DISPLAY\_IE\_REPRESENTATION

**Electric Se**  
**General** ☐ **I**  
Symbol Opt  
/ external / j  
representat

DISPLAY\_IGNORED\_CONNECT\_LINE

**Electric Se**  
**Connection**  
**Lines** ☐ Ign  
Cabling Tab  
ignored con  
alternative c  
CheckBox )

DISPLAY\_MIL\_STANDARD

**Electric Se**  
**MIL-Stand**  
Break-up Ec

**Electric Se**  
**MIL-Stand**  
Break-up Ec  
bottom

**Electric Se**  
**MIL-Stand**  
Options □ D  
complete co  
when final p  
contained in

**Electric Se**  
**MIL-Stand**  
Options □ D  
complete co  
when all pla  
contained in

**Electric Se**  
**MIL-Stand**  
Options □ A  
complete gr

**Electric Se**  
**MIL-Stand**  
Options □ D  
(m/f)

**Electric Se**  
**MIL-Stand**  
Options □ H  
connector p

**Electric Se**  
**MIL-Stand**  
Block Conne  
block's outli

**Electric Se**  
**MIL-Stand**  
Block Conne  
outside bloc

**Electric Se**  
**MIL-Stand**  
Block Conne  
block

**Electric Se**  
**MIL-Stand**  
Assembly □  
connectors  
as unit

DISPLAY\_MINIMISE\_DETAILS

DISPLAY\_OBJECTS\_WITHOUT\_VARIANTS

DISPLAY\_OPEN\_NODES

**Electric Se**  
**MIL-Stand**  
Backshell P  
backshell pi

**Electric Se**  
**MIL-Stand**  
Use line sty  
symbol

**Settings** ☐  
**Display** ☐ M  
Options ☐ M  
details

**Settings** ☐  
**Variants/O**  
**Display** ☐ A  
Variants / O  
Display elen  
variants / op

**Electric Se**  
**General** ☐ I  
connect poin  
connection  
T-Connectio

**Electric Se**  
**General** ☐ I  
connect poin  
connection  
wiring

**Electric Se**  
**General** ☐ I  
connect poin  
connection  
line ends

**Fluid Setti**  
**General** ☐ I  
connect poin  
connection  
T-Connectio

**Fluid Setti**  
**General** ☐ I  
connect poin  
connection  
wiring

DISPLAY\_OPEN\_PINS

**Fluid Settings**  
**General** ☐ **Display** ☐ **Mark**  
connect points  
connection lines  
line ends

**Settings** ☐ **Display** ☐ **Mark**  
☐ Mark unc  
nodes

DISPLAY\_OTHER\_VARIANTS\_GRAYED

**Settings** ☐ **Variants/Options**  
**Display** ☐ **Display** ☐ **Mark**  
Settings ☐ H  
for inactive  
variants/opti  
CheckBox )

DISPLAY\_PREVIEW\_SYMBOL\_FOR\_SELECTED\_COMPONENT

**Settings** ☐ **Display** ☐ **Mark**  
Options ☐ D  
symbol for s  
component

DISPLAY\_REGION

**Settings** ☐ **Display** ☐ **Mark**  
sheets ☐ Dis  
overview

DISPLAY\_SIGNAL\_FLAGS

**Settings** ☐ **Variants/Options**  
**Signal Log**  
Display sign

DISPLAY\_STANDARD\_ROTATED\_TEXTS

**Settings** ☐ **Display** ☐ **Mark**  
Options ☐ D  
texts acc. to

DISPLAY\_TOPOLOGY\_REGION

DISPLAY\_TYPE\_EXPRESSION

DO\_HIGHLIGHT

DRAW\_OBJECTS\_WITHOUT\_VARIANTS\_OPTIONS\_IN\_OTHER\_COLOR

DRAW\_VAR\_OTHER\_COLOR

DUPLICATE\_DISPLAY\_OFF

**Electric Se**  
**Topology** ☐  
Sheets ☐ Dis  
overview

**Electric Se**  
**Functional**  
Shared Shee  
region over

**Settings** ☐  
**Variants/O**  
**Display** ☐ D  
Settings ☐ D  
expressions



**Settings** ☐  
**Highlight** ☐  
Highlight fo  
when search

**Settings** ☐  
**Variants/O**  
**Display** ☐ D  
Settings ☐ D  
elements wi  
variants / op  
another col  
CheckBox )



**Settings** ☐  
**Variants/O**  
**Display** ☐ D  
Settings ☐ D  
elements wi  
options in a  
CheckBox )

**Settings** ☐  
**Display** ☐ S  
☐ Number o  
extension to  
device desig



DYNSYM\_CELL\_POS

**Settings**  **Dynamic S**  
Origin  Pla  
upper left in  
lower left


DYNSYM\_CODE

**Settings**  **Dynamic S**  
Outline  Co

DYNSYM\_DIA

**Settings**  **Dynamic S**  
Outline  W

DYNSYM\_HATCH\_CODE

**Settings**  **Dynamic S**  
Hatch  Col


DYNSYM\_HATCH\_DIA

**Settings**  **Dynamic S**  
Hatch  Wid

DYNSYM\_HATCH\_FLAGS

**Settings**  **Dynamic S**  
Hatch  Pat

DYNSYM\_HATCH\_LDIST

**Settings**  **Dynamic S**  
Hatch  Dis

DYNSYM\_MODE

DYNSYM\_TEXT\_TEMPLATE

ECHECK\_ALL\_COLOUR

ECHECK\_AMB\_TEMP

ECHECK\_ATTR\_TO\_NEW\_CONN

ECHECK\_CHECKS

ECHECK\_COIL\_COLOUR

ECHECK\_DIODE\_VOLT\_DROP

ECHECK\_DISP\_CUR\_FLOW

ECHECK\_FUSE\_COLOUR

ECHECK\_FUSE\_DER\_FACT

Possible Values

**Settings** □ **Dynamic S**  
Outline □ Li

**Settings** □ **Dynamic S**  
template □ S

**Electric Se**  
**Electrical C**  
Default Col  
error color

**Electric Se**  
**Electrical C**  
Default Valu  
temperature

**Electric Se**  
**Electrical C**  
**General** □ A  
to new net s  
update phys  
used conduc

**Electric Se**  
**Electrical C**  
**General** □ A  
electrical ch

**Electric Se**  
**Electrical C**  
Default Col  
active color

**Electric Se**  
**Electrical C**  
Default Valu  
voltage drop

**Electric Se**  
**Electrical C**  
**General** □ D  
flow

**Electric Se**  
**Electrical C**  
Default Col  
blown color

ECHECK\_FUSE\_MELTING\_WIRE\_IGNITION

ECHECK\_INTERNAL\_RESISTANCE

ECHECK\_LED\_COLOUR

ECHECK\_LOAD\_COLOUR

ECHECK\_SIGNAL\_ON\_ERROR

ECHECK\_WIRE\_COLOUR

ECHECK\_WIRE\_CROSSECTION

ECHECK\_WIRE\_LENGTH

ECHECK\_WIRE\_SPEC\_RESISTANCE

Possible Values

**Electric Se**  
**Electrical C**  
Default Valu  
derating fac

**Electric Se**  
**Electrical C**  
General ☐ C  
melting time  
ignition time

**Electric Se**  
**Electrical C**  
Default Valu  
resistance o  
elements [O

**Electric Se**  
**Electrical C**  
Default Col  
LED active c

**Electric Se**  
**Electrical C**  
Default Col  
active color

**Electric Se**  
**Electrical C**  
General ☐ A  
on error

**Electric Se**  
**Electrical C**  
Default Valu  
color

**Electric Se**  
**Electrical C**  
Default Valu  
cross-section

**Electric Se**  
**Electrical C**  
Default Valu  
length



ECHECK\_WIRE\_WEIGHT

ENABLE\_HYPERLINK\_DISPLAY

ENLARGE\_GRID\_POINTS

EXPORT\_ALL\_UNPLACED\_DEVICES\_OF\_ASSEMBLIES

EXPORT\_ALL\_UNPLACED\_TERMINALS\_OF\_STRIP

EXPORT\_BUSBAR

EXPORT\_BUSBAR\_ALL\_PINS\_IN\_PATH\_SELECTED

EXPORT\_STRUCTURE\_NODES

EXPORT\_UNPLACED\_DEVICES

**Electric Se**  
**Electrical C**  
Default Valu  
specific resi  
x mm<sup>2</sup>/m]

**Electric Se**  
**Electrical C**  
Default Valu  
weight [kg/l

**Settings** ☐  
**Highlight** ☐  
Hyperlink ☐  
property

**Settings** ☐  
**Display** ☐ M  
Options ☐ E  
points when

**Settings** ☐  
**Export/Cop**  
All devices o  
assembly

**Electric Se**  
**Placement**  
**Export/Cop**  
All terminal  
selected ter

**Electric Se**  
**Placement**  
**Export/Cop**  
Busbars

**Electric Se**  
**Placement**  
**Export/Cop**  
All pins in p

**Settings** ☐  
**Export/Cop**  
Export struc

**Settings** ☐  
**Export/Cop**  
Unplaced de

EXPORT\_UNPLACED\_TERMINALS

FD\_BLNODE\_BI

FD\_BLNODE\_GND

FD\_BLNODE\_IN

FD\_BLNODE\_MINUS

FD\_BLNODE\_OUT

valid when c  
'all')

**Electric Se  
Placement  
Export/Cop**  
Unplaced te  
valid when c  
'all')

**Electric Se  
Topology** □  
Connection  
Bidirectiona

**Electric Se  
Functional**  
Block Conne  
Symbols □ E

**Electric Se  
Topology** □  
Connection  
GND

**Electric Se  
Functional**  
Block Conne  
Symbols □ C

**Electric Se  
Topology** □  
Connection

**Electric Se  
Functional**  
Block Conne  
Symbols □ I

**Electric Se  
Topology** □  
Connection  
Minus

**Electric Se  
Functional**  
Block Conne  
Symbols □ M

**Electric Se  
Topology** □  
Connection

FD\_BLNODE\_PLUS

FD\_BLNODE\_UNDETERMINED

FD\_SIGNAL\_AIRLINES

FIELD\_CELL\_POS

FIELD\_CODE

**Electric Se**  
**Functional**  
Block Conne  
Symbols □ C  
**Electric Se**  
**Topology** □  
Connection  
Plus

**Electric Se**  
**Functional**  
Block Conne  
Symbols □ P  
**Electric Se**  
**Topology** □  
Connection  
Undetermin

**Electric Se**  
**Functional**  
Block Conne  
Symbols □ U  
**Electric Se**  
**Topology** □  
Display logi  
block conne

**Electric Se**  
**Functional**  
Options □ D  
lines on blo  
connections

**Settings** □  
**Field** □ Orig  
origin in up  
instead of lo

**Settings** □  
**Field** □ Out

FIELD\_DIA

**Settings** □  
**Field** □ Out

FIELD\_HATCH\_CODE

**Settings** □  
**Field** □ Hat

FIELD\_HATCH\_DIA

**Settings** □  
**Field** □ Hat

FIELD\_HATCH\_FLAGS

**Settings** □  
**Field** □ Hat

FIELD\_HATCH\_LDIST

**Settings** □ **Field** □ Hatch

FIELD\_MODE

**Settings** □ **Field** □ Outline Style

FIELD\_TEXT\_TEMPLATE

**Settings** □ **Field** □ Text Symbol

FIT\_TEXT

**Settings** □ **Display** □ Model Options □ Right text box

FITTING\_DATA\_DOMINATE\_MODELPIN

**Electric Se**  
**Connection**  
**Conductors**  
Conductor A  
Procedure □  
data of cavi  
models inste  
model

FLUIDALTGRIDSIZE

**Fluid Settings**  
**General** ▢ **Alt**  
Grid ▢ Grid

FLUID\_DISPLAY\_IE\_REPRESENTATION

**Fluid Settings**  
**General** ▢ **IE**  
Symbol Options  
/ External representation  
for nodes

FLUIDGRIDSIZE

**Fluid Settings**  
**General** ▢ **V**  
Grid size

FLUIDTRAPSIZE

**Fluid Settings**  
**General** ▢ **V**  
Snap size

FORMBOARD\_AUTOPLACE\_TABLE

**Electric Settings**  
**Formboard**  
Autoplace

FORMBOARD\_AUTOROTATE\_FORMBOARD\_SYMBOLS

**Electric Se**  
**Formboard**  
Autorotate

FORMBOARD\_DISPLAY\_ONE\_TABLE\_ROW\_FOR\_EACH\_CORE

**Electric Se**  
**Formboard**  
Display one  
conductor

FORMBOARD\_REGION\_SCALE\_STEP

**Electric Se**  
**Formboard**  
sheets □ Sca  
increment o

FORMBOARD\_ROTATE\_BRANCH\_ANGLE

**Electric Se**  
**Formboard**  
Branch □ An

FORMBOARD\_SHOW\_EFFECTIVE\_DIRECTION

**Electric Se**  
**Formboard**  
Connections  
Display effe  
direction on  
segments

FORMBOARD\_SHOW\_NAME\_OF\_FORMBOARD\_AS\_EXTENSION

**Electric Se**  
**Formboard**

Formboard  
extension to  
designation

FORMBOARD\_SHOW\_NET\_NODES

**Electric Se**  
**Formboard**  
Connections  
Display nod

FORMBOARD\_SHOW\_SEGMENT\_LENGTHS\_DIFF

**Electric Se**  
**Formboard**  
Connections  
Mark segme  
different ma  
and displaye

FORMBOARD\_SHOW\_TABLE\_AIRLINES

**Electric Se**  
**Formboard**  
Display sub

FORMBOARD\_SHOW\_UNCONNECTED\_CORES

**Electric Se**  
**Formboard**  
Display unco  
conductors

FORMBOARD\_TABLE\_DISPLAY\_PINS\_WITHOUT\_CORES

**Electric Se**  
**Formboard**



	Display pins conductors
FORMBOARD_WRAP_TABLE_ROWS	<b>Electric Se</b> <b>Formboard</b> Break table CheckBox )
FORMBOARD_WRAP_TABLE_ROWS_COUNT	<b>Electric Se</b> <b>Formboard</b> Break table SpinControl
FU_SHOW_ATTRIBUTE_TEMPLATE	<b>Electric Se</b> <b>Topology</b> <input type="checkbox"/> Show graph representat. functional u
GENERAL_CALCULATION_ACTIVE	<b>Electric Se</b> <b>Functional</b> Options <input type="checkbox"/> S representat. functional u
GRACOD	<b>Electric Se</b> <b>General</b> <input type="checkbox"/> C Electrical ca Activate Cal
GRADIA	<b>Settings</b> <input type="checkbox"/> Color
	<b>Settings</b> <input type="checkbox"/> Width

GRAFLG

**Settings** ☐ **Arrows**

GRAHATCH\_DEGREE1

**Settings** ☐  
**Hatch** ☐ **Pat**

GRAHATCH\_DEGREE2

**Settings** ☐  
**Hatch** ☐ **Pat**

GRAHATCH\_FLAGS

**Settings**   
**Hatch**  Pat

GRAHATCH\_GRACOD

**Settings**   
**Hatch**  Co

GRAHATCH\_GRACOD

**Settings**   
**Hatch**  Wi

GRAHATCH\_GRAMOD

**Settings**   
**Hatch**  Lin

GRAHATCH\_LINDIST

**Settings**   
**Hatch**  Dis

GRALEV

**Settings**  **Level**

GRAMOD

**Settings**  **Style**


GRAPHIC\_SNAPSIZE

**Electric Se**  
**General** 

HARNESS\_NAME\_ATTRIBUTE

**Electric Se**  
**Connection**  
**Conductor**  
Harness nam

HIERARCHYBLOCK\_WITH\_STRUCTURE

**Settings**  **Export/Cop**  
Hierarchica  
substructur

HIGHLIGHT\_COLOR

**Settings**  **Highlight**  **Color**

HIGHLIGHT\_JUMP\_ZOOM\_RATIO

**Settings** ☐  
**Highlight** ☐  
factor for 'Jump'

HLA\_DEFAULT

**Settings** ☐  
Default Description  
Higher level  
(Right Edit)

HYPERLINK\_COLOUR

**Settings** ☐  
**Highlight** ☐  
Hyperlink ☐

HYPERLINK\_UNDERLINE\_MODE

**Settings** ☐  
**Highlight** ☐  
Hyperlink ☐  
hyperlinks ☐  
hovering

**Settings** ☐  
**Highlight** ☐  
Hyperlink ☐  
hyperlinks ☐

**Settings** ☐  
**Highlight** ☐  
Hyperlink ☐  
hyperlinks ☐

IEC\_81346\_IS\_ACTIVE

**Settings** ☐  
IEC 81346 standard  
81346 is active

IGNORE\_BUNDLE\_OF\_DYN\_CABLE\_FOR\_NETSEG\_DIAMETER

**Electric Symbols**  
**Connection**  
**Conductors**  
**Calculation**  
/ Wire Calculation

IGNORE\_COMPONENT\_CODE\_ON\_IMPORT

IGNORE\_DOT\_CONN

IGNORE\_FILL\_COLOUR\_BLOCK

IGNORE\_NEW\_CONNECT\_LINE\_IN\_CABLING\_TABLE

IGNORE\_SHEET\_BORDER

IGNORE\_UNPLACED\_CORES\_OF\_DYN\_CABLE\_FOR\_NETSEG\_DIAMETER

IGNORE\_UNPLACED\_DEVICES

Possible Values

Segment dia  
Ignore bund  
dynamic cab

**Settings** □  
**Import** □ **D**  
Additional M  
Options □ Ig  
component

**Electric Se**  
**Connection**  
**Connectors**  
Connectors  
attribute 'In  
Designation  
assigning pi

**Fluid Setti**  
**Connection**  
Mating part  
attribute 'In  
Designation  
assigning pi

**Settings** □  
**Block** □ Blo  
Ignore defa

**Electric Se**  
**Connection**  
**Lines** □ Ign  
Cabling Tab  
new connec  
Cabling Tab

**Settings** □  
**Import** □ SH  
sheet borde

**Electric Se**  
**Connection**  
**Conductor**  
**Calculation**  
/ Wire Calcu  
Segment dia  
Ignore unpla  
conductors  
dynamic cab

IGNORE\_UNPLACED\_DEVICES\_OF\_ASSEMBLIES

IGNORE\_UNPLACED\_TERMINALS

IMPORT\_RENAME\_VARIANTS

IMPORT\_USE\_PIN\_ATTR

INHERIT\_NET\_NUMBER

Possible Values

**Settings** □  
**Import** □ **D**  
Unplaced D  
Ignore unpla

**Settings** □  
**Import** □ **D**  
Unplaced D  
Ignore unpla  
of assemblie

**Electric Se**  
**Placement**  
**Device** □ Un  
Devices □ Ig  
terminals

**Settings** □  
**Import** □  
Variants/Op  
Rename alre  
variants/opt  
part file

**Settings** □  
**Import** □  
Variants/Op  
existing vari  
from projec

**Settings** □  
**Import** □  
Variants/Op  
each existin  
variant/opti  
Before v202

**Settings** □  
**Import** □ **D**  
Additional M  
Options □ U  
attributes fr  
subcircuit

From v2022

**Settings** □  
**Import** □ **D**  
Attribute Op  
Subcircuit p  
Pins □ Merg

INHERIT\_OPTIONS

INHERIT\_OPTIONS\_FIELD

**Settings** ☐ (Net ☐ Specif  
functionality  
inheriting n  
**Settings** ☐  
**Variants/O**  
**Option Inh**  
Inherit Opti  
Sheet/Field/  
block ☐ No i  
symbols/dev  
sheet / for s  
hierarchical  
**Settings** ☐  
**Variants/O**  
**Option Inh**  
Inherit Opti  
Sheet/Field/  
block ☐ Add  
for symbols/  
hierarchical  
options for s  
**Settings** ☐  
**Variants/O**  
**Option Inh**  
Inherit Opti  
Sheet/Field/  
block ☐ Only  
for symbols/  
hierarchical  
options for s  
**Settings** ☐  
**Variants/O**  
**Option Inh**  
Inherit Opti  
Sheet/Field/  
block ☐ No i  
from field  
**Settings** ☐  
**Variants/O**  
**Option Inh**  
Inherit Opti  
Sheet/Field/  
block ☐ Add  
**Settings** ☐  
**Variants/O**



INHERIT\_OPTIONS\_OBJECTS

INHERIT\_OPTIONS\_VISIBILITY

INHERIT\_PINNAMES

KEEP\_ASSEMBLY\_BELONGING

KEEP\_ATTRIBUTE\_PIN\_BLOCKPIN\_SYMBOL

**Option Inheritance**  
Inherit Options from Parent  
Sheet/Field/Block Only

**Settings**  
**Variants/Options**  
**Option Inheritance**  
Option Inheritance  
Placing/Moving

**Settings**  
**Variants/Options**  
**Option Inheritance**  
Option Inheritance  
Placing/Moving

**Settings**  
**Variants/Options**  
**Option Inheritance**  
Visibility of  
Tree ☐ Depend  
visibility of s

**Electric Settings**  
**Connections**  
**Connectors**  
Inherit pin names  
connecting

**Fluid Settings**  
**Connections**  
Pin Names [ ]  
names when

**Settings** ☐ **Import** ☐ **Default**  
Additional M  
Options ☐ As  
assemblies /  
assemblies

**Electric Settings**  
**Placement**  
**Components**  
☐ Keep attri  
and block p

**Fluid Settings**  
**Placement**  
**Components**  
Keep attribu  
and block p

KEEP\_BMK\_DIN\_ORDER

KEEP\_CONNECTOR\_SYMBOLS

KEEP\_CORE\_AFTER\_UNCONNECT\_REFERENCE

KEEP\_COUNTERPARTS

**Settings** ☐ **U**  
Default Des  
Text order a  
standard

**Electric Se**  
**General** ☐ **U**  
**Project** ☐ C  
Keep connec

**Fluid Setti**  
**General** ☐ **U**  
**Project** ☐ F  
fitting symb

**Electric Se**  
**Connection**  
/ Wires ☐ Co  
conductors/  
end to sourc  
cross-refere  
cross-refere  
unconnecte

**Fluid Setti**  
**Connection**  
Hose/Tubes  
hoses/tubes  
to source  
cross-refere  
cross-refere  
unconnecte

**Electric Se**  
**General** ☐ **U**  
**Project** ☐ C  
Keep active  
connectors  
parts

**Fluid Setti**  
**General** ☐ **U**  
**Project** ☐ F  
active matir  
parts

KEEP\_EXISTING\_CORE\_LOGIC\_LINES\_OF\_SIGNAL

**Electric Settings**  
**Connection**  
**Cable/Conductor**  
**Lines** □ Core  
Lines of Signal □  
existing conductor  
lines of signal

KEEP\_HIGHLIGHT

**Fluid Settings**  
**Connection**  
**Hose/Tube**  
□ Hose/tube  
of Signal □  
hose/tube loc  
signal

KEEP\_LINECOLOR

**Settings** □  
**Highlight** □  
existing highlight  
jumping

KEEP\_MODELTEXT\_PARAMETER

**Settings** □  
**Display** □ M  
Options □ A  
background  
display color

KEEP\_MODELTEXT\_VISIBILITY

**Settings** □  
**Update in** M  
Parameters  
text parameter  
Models

KEEP\_PINVIEW\_SYMBOLS\_OF\_DEVICE

**Settings** [ ]  
**Update in** [ ]  
[ ] Keep prev  
of devices

KEEP\_PLUG\_AFTER\_UNPLUGGING\_PINS

**Settings** [ ]  
Pins [ ] Keep  
unplugging

KEEP\_TEXT\_PARAMETER

**Settings** [ ]  
**Update in** [ ]  
Parameters  
text paramet  
Symbols

KEEP\_TEXT\_VISIBILITY














**Settings** [ ]  
**Update in** [ ]  
Parameters  
visibility for

LANGUAGES

**Settings** [ ]  
**Language** [ ]  
Language


**Settings** [ ]  
**Language** [ ]  
Language

**Settings** [ ]  
**Language** [ ]  
Language

	<div>Settings  <b>Language</b>  Language</div>
	<div>Settings  <b>Language</b>  Language</div>
LINCOD	<div>Settings  <b>Connect Li</b>  Color</div>
LINDIA	<div>Settings  <b>Connect Li</b>  Width</div>
LINESTYLE_JIS	<div>Settings  <b>Style</b>  Defa</div>
LIN_FOLD_ANGLE	<div>Settings  <b>Style</b>  Japa Industrial S</div> <div>Settings  <b>Connect Li</b></div>

Connections  
angle

LIN\_FOLD\_DISTANCE

**Settings**   
**Connect Li**  
Connections  
distance

LINLEV

**Settings**   
**Connect Li**  
Level

LINMOD

**Settings**   
**Connect Li**  
Style

LOAD\_SYMBOL\_GRAPHIC\_ONLY\_IF\_REQUIRED

**Settings**   
**Symbols**   
Parameters

LOC\_DEFAULT

LOCK\_WIRES

MAINTAIN\_TEXTSIZE

MARK\_DIFFERENT\_ACTIVE\_VARIANT\_VALUES

MARK\_LOCKED\_OBJECTS

MEA\_EXTERN\_PANEL

MEA\_EXTERN\_SCHEMA

symbol graph  
database on

**Settings** ☐ **Default Des**  
Default Des  
Location (Ri

**Electric Se**  
**Connection**  
/ Wires ☐ Lo  
pathways

**Fluid Setti**  
**Connection**  
Hoses/Tube  
wire pathwa

**Settings** ☐ **Symbols** ☐  
Parameters  
text size wh

**Settings** ☐ **Variants/O**  
**Display** ☐ D  
Settings ☐ M  
availability  
active attrib  
CheckBox )

**Settings** ☐ **Locking** ☐ I  
locked obje

**Settings** ☐ **Measureme**  
Measureme  
Millimeters

**Settings** ☐ **Measureme**  
Measureme  
Inches

**Settings** ☐ **Measureme**  
Measureme  
Millimeters

**Settings** ☐ **Measureme**  
Measureme  
Inches

MERGE\_CONNECTION\_LINES

**Settings** ☐ **Import** ☐ Merge Options  
Line Option  
connect line  
and modify

MERGE\_COPY\_AREA\_KEEP\_SYSTEM

**Settings** ☐ **Signals** ☐ Merge Options  
for Copy and  
Keep system  
signals

MERGE\_COPY\_AREA\_KEEP\_USER

**Settings** ☐ **Signals** ☐ Merge Options  
for Copy and  
Keep user-defined  
signals

MERGE\_IGNOREING\_CORE\_DIRECTION

**Electric Settings**  
**Placement**  
**Device** ☐ Additional Merge Options  
Merge Options  
ignoring core  
direction

MERGE\_INCLUSIVE\_EXCLUSIVE\_DEFINITIONS

**Fluid Settings**  
**Placement**  
**Device** ☐ Additional Merge Options  
Merge Options  
ignoring host  
direction

MERGE\_OPTIONS

**Settings** ☐ **Import** ☐ Merge Options  
Variants/Options  
inclusive/exclusive  
definitions

**Settings** ☐ **Import** ☐ Merge Options  
Additional Merge  
Options ☐ Merge Options



MERGE\_USING\_EXACT\_CORE\_CONNECTION

**Electric Se**  
**Placement**  
**Device** □ Ac  
Merge Optic  
using exact  
connection

**Fluid Setti**  
**Placement**  
**Device** □ M  
exact hose/t  
connection

MIL\_DISPLAY\_CONNECTOR\_IGNORE\_VIEWS

**Electric Se**  
**MIL-Stand**  
Assembly □  
assignment

MIL\_GRACOD

**Electric Se**  
**MIL-Stand**  
Color

MIL\_GRADIA

**Electric Se**  
**MIL-Stand**  
Width

MIL\_GRAMOD

**Electric Se**  
**MIL-Stand**  
Line Style

MODE\_GRID\_OVERLAY

**Settings** ☐ **Display** ☐ G  
Points ( Che

MOVE\_CORE\_ENDS

**Electric Se**  
**Connection**  
**Conductors**  
Move condu

NETLOOPS\_ALLOWED

**Settings** ☐ **Net** ☐ Allow

NOCONN\_DISPLAY\_OFF

**Settings** ☐ **Signal Log**  
Mark conne  
defined as n  
(\*\*NC\*\*)

NON\_REDLINER\_COLOR

**Settings** ☐ **Redlining In**  
Color

NO\_VARIANT\_COLOR

## OPTION IMPORT DEVICE MERGE ATTRIBUTES

## OPTION IMPORT DEVICE ONLY ATTRIBUTES

## OPTION IMPORT PIN MERGE ATTRIBUTES

### OPTION IMPORT PIN ONLY ATTRIBUTES

## OPTION IMPORT WIRE MERGE ATTRIBUTES

## OPTION IMPORT WIRE ONLY ATTRIBUTES

## OPTION VARIANT UNIQUE NAMES

### Possible Values

**Settings** ☐  
**Variants/Options** ☐  
**Display** ☐ Display  
 Settings ☐ Display  
 elements with  
 variants/options  
 another color  
 (ComboBox)

Settings ☐   
 Import ☐ Devices ☐   
 Attribute Op   
 Subcircuit p   
 Devices ☐ M

Settings ☐   
 Import ☐ D   
 Attribute Op   
 Subcircuit p   
 Devices ☐ O

**Settings** ☐ **Import** ☐ **D**  
Attribute Op  
Subcircuit p  
Pins ☐ Merg

**Settings** ☐ **Import** ☐ **Display** ☐

Attribute Op ☐ Subcircuit p ☐ Pins ☐ Only ☐

Settings ☐   
 Import ☐ Design ☐   
 Attribute Op ☐   
 Subcircuit p ☐   
 Wires ☐ Men

**Settings** ☐ **Import** ☐ **Debug** ☐

Attribute Op ☐ Subcircuit p ☐ Wires ☐ Only ☐

## Settings

OPTIMIZE\_PLACEMENT\_AFTER\_CHANGES

**Settings** ☐ **Placement** ☐  
Placement a  
placement a

PANEL\_2D\_VISIBLE\_PERCENTAGE

**Settings** ☐ **Percentage** ☐  
Percentage  
to show par  
models

**Settings** ☐ **Connection** ☐  
**Lines** ☐ **Disj** ☐

**Settings** ☐ **Connection** ☐  
**Lines** ☐ **Dir** ☐  
☐ Display di  
connections

**Settings** ☐ **Connection** ☐  
**Lines** ☐ **Sub** ☐  
Connection  
subnet conn

PANEL\_AIRLINES

**Settings** ☐ **Connection** ☐  
**Lines** ☐ **Equ** ☐  
Display equ

**Settings** ☐ **Connection** ☐  
**Lines** ☐ **Sig** ☐  
Pins ☐ Displ  
carrying pin

**Settings** ☐ **Connection** ☐  
**Lines** ☐ **She** ☐  
Comprehens  
Connections  
connection l

PANEL\_AIRLINES\_DIRECT\_COLOR

**Settings** ☐ **Connection** ☐  
**Lines** ☐ **Dir** ☐  
☐ Display di  
connections  
Color

PANEL\_AIRLINES\_DIRECT\_STYLE

PANEL\_AIRLINES\_EQUIVALENT\_COLOR

PANEL\_AIRLINES\_EQUIVALENT\_STYLE

PANEL\_AIRLINES\_SIGNAL\_COLOR

PANEL\_AIRLINES\_SIGNAL\_WIDTH

PANEL\_AIRLINES\_SUBNET\_COLOR

PANEL\_AIRLINES\_SUBNET\_STYLE

PANEL\_ALLOW\_WIRE\_LOOPS

PANELALTGRIDSIZE

Possible Values

**Settings** [ ]  
**Connection**  
**Lines** [ ] Dir  
[ ] Display di  
connections  
Line Style

**Settings** [ ]  
**Connection**  
**Lines** [ ] Equ  
Display equ  
**Advanced** [ ]

**Settings** [ ]  
**Connection**  
**Lines** [ ] Equ  
Display equ  
**Advanced** [ ]

**Settings** [ ]  
**Connection**  
**Lines** [ ] Sig  
Pins [ ] Displ  
carrying pin  
[ ] Color

**Settings** [ ]  
**Connection**  
**Lines** [ ] Sig  
Pins [ ] Displ  
carrying pin  
[ ] Width

**Settings** [ ]  
**Connection**  
**Lines** [ ] Sub  
Connection  
subnet conn  
**Advanced** [ ]

**Settings** [ ]  
**Connection**  
**Lines** [ ] Sub  
Connection  
subnet conn  
**Advanced** [ ]

**Settings** [ ]  
**Connection**  
Options [ ] Al  
loops

	<b>Settings</b>  <b>Connection</b> Alternative size
PANEL_AUTOCONNECT_ALG_CURRENT	<b>Settings</b>  <b>Connection</b> Autoconnect Current (Ch
PANEL_AUTOCONNECT_ALG_OPTIMIZE	<b>Settings</b>  <b>Connection</b> Autoconnect Optimized (
PANEL_AUTOCONNECT_CABLE_DUCT_SEARCH_GAGE	<b>Settings</b>  <b>Connection</b> Autoconnect Bandwidth f cable duct ( )
PANEL_AUTOCONNECT_CABLE_DUCT_SEARCH_GAGE_FLAG	<b>Settings</b>  <b>Connection</b> Autoconnect Nearest cab
PANEL_AUTOCONNECT_DELETE_PREFDEF_PORTS	<b>Settings</b>  <b>Connection</b> Autoconnect Bandwidth f cable duct ( Button )
PANEL_AUTOCONNECT_USE_SCHEMATIC_TERMINAL_PORTS	<b>Settings</b>  <b>Connection</b> Autoconnect Delete pred in schemati
	<b>Settings</b>  <b>Connection</b> Autoconnect Use manual

PANEL\_AUTOCONNECT\_WIRE\_JUMPER\_FINALLY

PANEL\_AUTOROUTE\_ATTRIBUTES

PANEL\_AUTOROUTE\_WITH\_EXPLICIT\_ATTRIBUTES

PANELAXISGRID

PANEL\_BACKPLANE\_DISTANCE

ports for ter  
schematic

**Settings** □  
**Connections**  
Autoconnect  
Route wire j  
end

**Settings** □  
**Checks** □ T  
attribute(s)  
during auto-  
enable/disab  
cable ducts  
attribute ( C

**Settings** □  
**Checks** □ T  
attribute(s)  
during auto-  
enable/disab  
cable ducts  
attributes

**Settings** □  
**Checks** □ T  
attribute(s)  
during auto-  
enable/disab  
cable ducts  
attribute ( F

**Settings** □  
View □ Rule  
SpinControl

**Settings** □  
**Connections**  
Options □ B  
connection c

PANEL\_CABLE\_DUCT

**Settings** [ ... ]  
**Duct Symbol**  
Fill Size [ ... ] D

**Settings** [ ... ]  
**Duct Symbol**  
Docking Poi  
docking poi

**Settings** [ ... ]  
**Duct Symbol**  
Lateral Pun  
Display late  
width

**Settings** [ ... ]  
**Duct Symbol**  
Break Line [ ... ]  
break line

PANEL\_CABLE\_DUCT\_CRITICAL\_FILL\_LIMIT

**Settings** [ ... ]  
**Duct Symbol**  
Fill Size [ ... ] C  
warning lim

PANEL\_CABLE\_DUCT\_DOCK\_COLOR

**Settings** [ ... ]  
**Duct Symbol**  
Docking Poi  
docking poi  
**Advanced** [ ... ]

PANEL\_CABLE\_DUCT\_DOCK\_WIDTH

**Settings** [ ... ]  
**Duct Symbol**  
Docking Poi  
docking poi  
**Advanced** [ ... ]



PANEL\_CABLE\_DUCT\_FILL\_COLOR

**Settings** ☐ **Advanced** ☐  
**Duct Symbol**  
Fill Size ☐ D

PANEL\_CABLE\_DUCT\_FILL\_LIMIT

**Settings** ☐ **Advanced** ☐  
**Duct Symbol**  
Fill Size ☐ C  
limit (%)

PANEL\_CABLE\_DUCT\_FILL\_STYLE

**Settings** ☐ **Advanced** ☐  
**Duct Symbol**  
Fill Size ☐ D

PANEL\_CABLE\_DUCT\_SYM\_HATCH\_COLOR

**Settings** ☐ **Advanced** ☐  
**Duct Symbol**  
Color

PANEL\_CABLE\_DUCT\_SYM\_HATCH\_DEGREE1

**Settings** ☐ **Advanced** ☐  
**Duct Symbol**  
Pattern

PANEL\_CABLE\_DUCT\_SYM\_HATCH\_DEGREE2

**Settings**   
**Duct Symbol**  
Pattern

PANEL\_CABLE\_DUCT\_SYM\_HATCH\_FLAGS

**Settings**   
**Duct Symbol**  
Pattern

PANEL\_CABLE\_DUCT\_SYM\_LEVEL

**Settings**   
**Duct Symbol**

PANEL\_CABLE\_DUCT\_SYM\_LINE\_COLOR

**Settings**   
**Duct Symbol**  
Color



PANEL\_CABLE\_DUCT\_SYM\_LINE\_STYLE

**Settings**   
**Duct Symbol**  
Line Style

PANEL\_CABLE\_DUCT\_SYM\_LINE\_WIDTH

**Settings**   
**Duct Symbol**  
Width

PANEL\_CHECKS\_ALLOW\_CROSSING\_CABLEDUCTS

**Settings**   
**Checks**   
attribute(s)  
during auto-  
enable/disable  
cable ducts  
crossing cable  
with different  
classes

PANEL\_CHECKS\_COMPONENT\_OPTIONS\_VARIANTS

PANEL\_CHECKS\_FIT\_TO\_TARGET

PANEL\_CHECKS\_OUTLINE\_TO\_CUTOUT

PANEL\_CHECKS\_OUTLINE\_TO\_OUTLINE

**Settings** ☐ .  
**Checks** ☐  
Variants/Op  
Variants/Op

**Settings** ☐  
**Checks** ☐  
Variants/Op  
Variants/Op  
Warning

**Settings** ☐ .  
**Checks** ☐  
Variants/Op  
Variants/Op

**Settings** ☐ .  
**Checks** ☐ C  
component  
Area/Line ☐

**Settings** ☐ .  
**Checks** ☐ C  
component  
Area/Line ☐

**Settings** ☐ .  
**Checks** ☐ C  
component  
Area/Line ☐

**Settings** ☐ .  
**Checks** ☐ C  
Component

**Settings** ☐ .  
**Checks** ☐ C  
Component

**Settings** ☐ .  
**Checks** ☐ C  
Component

**Settings** ☐ .  
**Checks** ☐ C  
<--> Compo

**Settings** ☐ .  
**Checks** ☐ C  
<--> Compo  
Warning

PANEL\_CHECKS\_OUTLINE\_TO\_RESTRICTED

PANEL\_CHECKS\_OUTLINE\_TO\_SLOT

PANEL\_CHECKS\_RESTRICTED\_TO\_CUTOUT

PANEL\_CHECKS\_WIRE\_TO\_OUTLINE\_1

**Settings** ☐   
**Checks** ☐ C  
<--> Comp

**Settings** ☐   
**Checks** ☐ C  
Restricted ☐

**Settings** ☐   
**Checks** ☐ C  
Restricted ☐

**Settings** ☐   
**Checks** ☐ C  
Restricted ☐

**Settings** ☐   
**Checks** ☐ M  
description  
description

**Settings** ☐   
**Checks** ☐ M  
description  
description

**Settings** ☐   
**Checks** ☐ M  
description  
description

**Settings** ☐   
**Checks** ☐ C  
Restricted ☐

**Settings** ☐   
**Checks** ☐ C  
Restricted ☐

**Settings** ☐   
**Checks** ☐ C  
Restricted ☐

**Settings** ☐   
**Checks** ☐ A  
Wire/Condu  
<-->  
Component/  
On

**Settings** ☐   
**Checks** ☐ A  
Wire/Condu

PANEL\_CHECKS\_WIRE\_TO\_OUTLINE\_2

PANEL\_CONNECT\_IGNORE\_EQUIVALENT\_PINS

PANEL\_CONNECTION\_DISPLAY\_WIRE\_PARAMS


PANEL\_CONNECT\_LEAVING\_WIRE\_MARK

PANEL\_CONNECT\_LEAVING\_WIRE\_MARK\_SIZE

Possible Values



<->  
Component/  
Warning  
  
**Settings** □  
**Checks** □ A  
Wire/Condu  
<->  
Component/  
Off  
  
**Settings** □  
**Checks** □ A  
Wire/Condu  
duct<->  
Component/  
On  
  
**Settings** □  
**Checks** □ A  
Wire/Condu  
duct<->  
Component/  
Warning  
  
**Settings** □  
**Checks** □ A  
Wire/Condu  
duct<->  
Component/  
Off  
  
**Electric Se**  
**Connection**  
**Conductor**  
Conductor A  
Procedure □  
equivalence  
connect pin  
  
**Settings** □  
**Connection**  
Options □ U  
parameters  
  
**Settings** □  
**Connection**  
Options □ M  
comprehens  
  
**Settings** □  
**Connection**



PANEL\_CONNECT\_MARK\_JUMPER\_CONNECT\_POINTS

**Panel  Con**  
Display Opt  
jumper conn



PANEL\_CONNECT\_METHOD

**Settings  .**  
**Connection**  
Method  Si

**Settings  .**  
**Connection**  
Method  G  
schema con

**Settings  .**  
**Connection**  
Method  U  
connection v  
wires/condu  
schematic

PANEL\_DEFAULT\_FLUID

**Fluid Setti**  
**Connection**  
**Tubes  Us**  
Type  Hose

PANELGRIDSIZE

**Settings  .**  
Working Gr

PANEL\_HIGHLIGHT\_COLOUR

	<b>Settings</b> □ Highlight □
PANEL_HIGHLIGHT_WIDTH	<b>Settings</b> □ Highlight □
PANEL_MODE_GRID_AXIS	<b>Settings</b> □ <b>Display</b> □ G Rulers (Che
PANEL_MODE_GRID_OVERLAY	<b>Settings</b> □ View □ Rule )
PANEL_MODE_GRID_SHEETLAYOUT	<b>Settings</b> □ <b>Display</b> □ S Reference □ layout
PANEL_MOUNT_SYM_HATCH_COLOR	<b>Settings</b> □ <b>Symbol</b> □ H
PANEL_MOUNT_SYM_HATCH_DEGREE1	<b>Settings</b> □ <b>Symbol</b> □ H

PANEL\_MOUNT\_SYM\_HATCH\_DEGREE2

**Settings**  **Symbol**  Hatch

PANEL\_MOUNT\_SYM\_HATCH\_FLAGS

**Settings**  **Symbol**  Hatch

PANEL\_MOUNT\_SYM\_LEVEL

**Settings**  **Symbol**  Level

PANEL\_MOUNT\_SYM\_LINE\_COLOR

**Settings**  **Symbol**  Outline

PANEL\_MOUNT\_SYM\_LINE\_STYLE

**Settings**  **Symbol**  Outline  
Style

PANEL\_MOUNT\_SYM\_LINE\_WIDTH

**Settings**  **Symbol**  Outline

PANELOVERSIZE


**Settings**  **View**  Point  
SpinControl

PANEL\_REGION\_SCALE\_STEP

**Settings**  sheets  Scale  
increment of




PANEL\_RESTRICTED\_ALL\_HATCH\_COLOR

**Settings**  **Restricted**  
Defined for

**Settings**  **Restricted**  
Hatch  Col

PANEL\_RESTRICTED\_ALL\_HATCH\_DEGREE1

**Settings**  **Restricted**  
Defined for


**Settings**  **Restricted**  
Hatch  Pat

PANEL\_RESTRICTED\_ALL\_HATCH\_DEGREE2

**Settings**  **Restricted**  
Defined for

**Settings**  **Restricted**  
Hatch  Pat

PANEL\_RESTRICTED\_ALL\_HATCH\_DIST

**Settings**  **Restricted**  
Defined for


**Settings**  **Restricted**  
Hatch  Dis

PANEL\_RESTRICTED\_ALL\_HATCH\_FLAGS

**Settings**  **Restricted**  
Defined for

**Settings**  **Restricted**  
Hatch  Pat

PANEL\_RESTRICTED\_ALL\_HATCH\_STYLE

**Settings**  **Restricted**  
Defined for

PANEL\_RESTRICTED\_ALL\_HATCH\_WIDTH


PANEL\_RESTRICTED\_ALL\_LEVEL


PANEL\_RESTRICTED\_ALL\_LINE\_COLOR


PANEL\_RESTRICTED\_ALL\_LINE\_STYLE


PANEL\_RESTRICTED\_ALL\_LINE\_WIDTH

PANEL\_RESTRICTED\_CON\_HATCH\_COLOR


**Settings**  **Restricted**  
Hatch  Lin

**Settings**  **Restricted**  
Defined for


**Settings**  **Restricted**  
Hatch  Wid

**Settings**  **Restricted**  
Defined for

**Settings**  **Restricted**  
Level

**Settings**  **Restricted**  
Defined for


**Settings**  **Restricted**  
Outline  Co

**Settings**  **Restricted**  
Defined for

**Settings**  **Restricted**  
Outline  Li


**Settings**  **Restricted**  
Defined for

**Settings**  **Restricted**  
Outline  W

**Settings**  **Restricted**  
Defined for  
Connections


**Settings**  **Restricted**  
Hatch  Col



PANEL\_RESTRICTED\_CON\_HATCH\_DEGREE1

**Settings**  **Restricted**  
Defined for  
Connections


**Settings**  **Restricted**  
Hatch  Pat

PANEL\_RESTRICTED\_CON\_HATCH\_DEGREE2

**Settings**  **Restricted**  
Defined for  
Connections


**Settings**  **Restricted**  
Hatch  Pat

PANEL\_RESTRICTED\_CON\_HATCH\_DIST

**Settings**  **Restricted**  
Defined for  
Connections


**Settings**  **Restricted**  
Hatch  Dis


PANEL\_RESTRICTED\_CON\_HATCH\_FLAGS

**Settings**  **Restricted**  
Defined for  
Connections

**Settings**  **Restricted**  
Hatch  Pat

PANEL\_RESTRICTED\_CON\_HATCH\_STYLE

**Settings**  **Restricted**  
Defined for  
Connections


**Settings**  **Restricted**  
Hatch  Lin



PANEL\_RESTRICTED\_CON\_HATCH\_WIDTH

**Settings**  **Restricted**


	Defined for Connections
	<b>Settings</b> □ <b>Restricted</b> Hatch □ Width
PANEL_RESTRICTED_CON_LEVEL	<b>Settings</b> □ <b>Restricted</b> Defined for Connections
	<b>Settings</b> □ <b>Restricted</b> Level
PANEL_RESTRICTED_CON_LINE_COLOR	<b>Settings</b> □ <b>Restricted</b> Defined for Connections
	<b>Settings</b> □ <b>Restricted</b> Outline □ Color
PANEL_RESTRICTED_CON_LINE_STYLE	<b>Settings</b> □ <b>Restricted</b> Defined for Connections
	<b>Settings</b> □ <b>Restricted</b> Outline □ Line
PANEL_RESTRICTED_CON_LINE_WIDTH	<b>Settings</b> □ <b>Restricted</b> Defined for Connections
	<b>Settings</b> □ <b>Restricted</b> Outline □ Width
PANEL_RESTRICTED_CUTOUT_HATCH_COLOR	<b>Settings</b> □ <b>Restricted</b> Defined for area
	<b>Settings</b> □ <b>Restricted</b> Hatch □ Color

PANEL\_RESTRICTED\_CUTOUT\_HATCH\_DEGREE1

**Settings**  **Restricted**  
Defined for  
area


**Settings**  **Restricted**  
Hatch  Pat

PANEL\_RESTRICTED\_CUTOUT\_HATCH\_DEGREE2

**Settings**  **Restricted**  
Defined for  
area


**Settings**  **Restricted**  
Hatch  Pat

PANEL\_RESTRICTED\_CUTOUT\_HATCH\_DIST

**Settings**  **Restricted**  
Defined for  
area


**Settings**  **Restricted**  
Hatch  Dis

PANEL\_RESTRICTED\_CUTOUT\_HATCH\_FLAGS

**Settings**  **Restricted**  
Defined for  
area

**Settings**  **Restricted**  
Hatch  Pat

















PANEL\_RESTRICTED\_CUTOUT\_HATCH\_STYLE

**Settings**  **Restricted**  
Defined for  
area


**Settings**  **Restricted**  
Hatch  Lin


PANEL\_RESTRICTED\_CUTOUT\_HATCH\_WIDTH

**Settings**  **Restricted**


	Defined for area
	<b>Settings</b>  <b>Restricted</b> Hatch  Width
PANEL_RESTRICTED_CUTOUT_LEVEL	<b>Settings</b>  <b>Restricted</b> Defined for area
	<b>Settings</b>  <b>Restricted</b> Level
PANEL_RESTRICTED_CUTOUT_LINE_COLOR	<b>Settings</b>  <b>Restricted</b> Defined for area
	<b>Settings</b>  <b>Restricted</b> Outline  Color
PANEL_RESTRICTED_CUTOUT_LINE_STYLE	<b>Settings</b>  <b>Restricted</b> Defined for area
	<b>Settings</b>  <b>Restricted</b> Outline  Line
PANEL_RESTRICTED_CUTOUT_LINE_WIDTH	<b>Settings</b>  <b>Restricted</b> Defined for area
	<b>Settings</b>  <b>Restricted</b> Outline  Width
PANEL_RESTRICTED_DEV_HATCH_COLOR	<b>Settings</b>  <b>Restricted</b> Defined for Components
	<b>Settings</b>  <b>Restricted</b> Hatch  Color

PANEL\_RESTRICTED\_DEV\_HATCH\_DEGREE1

**Settings**  **Restricted**  
Defined for  
Components


**Settings**  **Restricted**  
Hatch  Pat

PANEL\_RESTRICTED\_DEV\_HATCH\_DEGREE2

**Settings**  **Restricted**  
Defined for  
Components


**Settings**  **Restricted**  
Hatch  Pat

PANEL\_RESTRICTED\_DEV\_HATCH\_DIST

**Settings**  **Restricted**  
Defined for  
Components


**Settings**  **Restricted**  
Hatch  Dis

PANEL\_RESTRICTED\_DEV\_HATCH\_FLAGS

**Settings**  **Restricted**  
Defined for  
Components

**Settings**  **Restricted**  
Hatch  Pat

PANEL\_RESTRICTED\_DEV\_HATCH\_STYLE

**Settings**  **Restricted**  
Defined for  
Components

**Settings**  **Restricted**  
Hatch  Lin

PANEL\_RESTRICTED\_DEV\_HATCH\_WIDTH

**Settings**  **Restricted**

PANEL\_RESTRICTED\_DEV\_LEVEL

PANEL\_RESTRICTED\_DEV\_LINE\_COLOR

PANEL\_RESTRICTED\_DEV\_LINE\_STYLE

PANEL\_RESTRICTED\_DEV\_LINE\_WIDTH

PANEL\_RESTRICTED\_DISPLAY

Defined for Components

**Settings** ☐ **Restricted**  
Hatch ☐ Width

**Settings** ☐ **Restricted**  
Defined for Components

**Settings** ☐ **Restricted**  
Level

**Settings** ☐ **Restricted**  
Defined for Components

**Settings** ☐ **Restricted**  
Outline ☐ Color

**Settings** ☐ **Restricted**  
Defined for Components

**Settings** ☐ **Restricted**  
Outline ☐ Line

**Settings** ☐ **Restricted**  
Defined for Components

**Settings** ☐ **Restricted**  
Outline ☐ Width

**Settings** ☐ **Restricted**  
**Display** ☐ On  
Restricted for

**Settings** ☐ **Restricted**  
**Display** ☐ On  
Restricted for Components







PANEL\_RESTRICTED\_HOLE\_HATCH\_COLOR



PANEL\_RESTRICTED\_HOLE\_HATCH\_DEGREE1

PANEL\_RESTRICTED\_HOLE\_HATCH\_DEGREE2


PANEL\_RESTRICTED\_HOLE\_HATCH\_DIST


**Settings**  **Restricted**  
**Display**  O  
Restricted f  
Connections

**Settings**  **Restricted**  
**Display**  O  
area


**Settings**  **Restricted**  
**Display**  O  
Drill-hole

**Settings**  **Restricted**  
Defined for

**Settings**  **Restricted**  
Hatch  Col


**Settings**  **Restricted**  
Defined for

**Settings**  **Restricted**  
Hatch  Pat


**Settings**  **Restricted**  
Defined for



**Settings**  **Restricted**  
Hatch  Pat

**Settings**  **Restricted**  
Defined for


**Settings**  **Restricted**  
Hatch  Dis



PANEL\_RESTRICTED\_HOLE\_HATCH\_FLAGS

**Settings**  **Restricted**  
Defined for


**Settings**  **Restricted**  
Hatch  Pat



PANEL\_RESTRICTED\_HOLE\_HATCH\_STYLE

**Settings**  **Restricted**  
Defined for


**Settings**  **Restricted**  
Hatch  Lin

PANEL\_RESTRICTED\_HOLE\_HATCH\_WIDTH

**Settings**  **Restricted**  
Defined for


**Settings**  **Restricted**  
Hatch  Wid



PANEL\_RESTRICTED\_HOLE\_LEVEL

**Settings**  **Restricted**  
Defined for

**Settings**  **Restricted**  
Level

PANEL\_RESTRICTED\_HOLE\_LINE\_COLOR

**Settings**  **Restricted**  
Defined for

**Settings**  **Restricted**  
Outline  Co

PANEL\_RESTRICTED\_HOLE\_LINE\_STYLE

**Settings**  **Restricted**  
Defined for

**Settings**  **Restricted**  
Outline  Li

PANEL\_RESTRICTED\_HOLE\_LINE\_WIDTH

**Settings**  **Restricted**

	Defined for
	<b>Settings</b> <a href="#">[...]</a> <b>Restricted</b> Outline <a href="#">[...]</a> W
PANEL_ROUTING_STATISTIC	<b>Settings</b> <a href="#">[...]</a> <b>Connection</b> Autoconnec Write statist
PANELTRAPSIZE	<b>Settings</b> <a href="#">[...]</a> Working Gr
PANEL_WIRE_INSULATION_FACTOR	<b>Settings</b> <a href="#">[...]</a> <b>Connection</b> Options <a href="#">[...]</a> W factor
PASS_WIRE_MARK	<b>Electric Se</b> <b>General</b> <a href="#">[...]</a> <b>I</b> Symbol Opt. connected wires/condu wire node (
PASS_WIRE_MARK_SYMBOL	<b>Electric Se</b> <b>General</b> <a href="#">[...]</a> <b>I</b> Symbol Opt. connected wires/condu wire node (
PICTOGRAM_LANGUAGE	<b>Settings</b> <a href="#">[...]</a> <b>Language</b> <a href="#">[...]</a>
PLACE_DEL_PIN_ATTR	<b>Settings</b> <a href="#">[...]</a> Rules <a href="#">[...]</a> Dele attributes w is unplaced

PLACE\_DEL\_SYMBOL\_ATTR

**Settings** □  
Rules □ Delete  
attributes w  
is unplaced

PLACE\_USE\_PIN\_ATTR

**Settings** □  
Rules □ Use  
attributes w  
assigning to

PURGE\_ALL\_STEP\_MODELS

**Settings** □  
**Purge** □ Ob  
models

PURGE\_UNUSED\_CONNECTED\_DEVICES

**Settings** □  
**Purge** □ Ob  
connected d

PURGE\_UNUSED\_PLUGGED\_DEVICES

**Settings** □  
**Purge** □ Ob  
plugged dev

READ\_ONLY\_GRAPHIC\_COLOR

**Settings** □  
Read Only L

READ\_ONLY\_GRAPHIC\_LEVEL

**Settings** □  
Read Only L

RECALC\_FORMATTED\_SIGNALS

**Settings** □  
**Signals** □ S  
Recalculate  
names acco  
format spec

REFCHANGETYPE

**Settings** [\[...\]](#)  
**References**  
Functional C  
Allow chang  
type

REFCOD

**Settings** [\[...\]](#)  
**References**

REFDIA

**Settings** [\[...\]](#)  
**References**  
Style

REFDIR

**Settings** [\[...\]](#)  
**References**  
Direction [\[...\]](#)

**Settings** [\[...\]](#)  
**References**  
Direction [\[...\]](#)

**Settings** [\[...\]](#)  
**References**  
Direction [\[...\]](#)

REFERENCE\_BETWEEN\_PLUGGED\_PINS

**Settings** [\[...\]](#)  
**Reference**  
Display refe  
logically plu

REFFONTPTR

**Settings** [\[...\]](#)  
**References**

REFGAP

**Settings** [\[...\]](#)  
**References**  
Display Con

REFMOD

REFOFX

REFOFY

REFSIZ

RELOAD\_ATTRIBUTES

RELOAD\_SIGNALS

RENAME\_FIELDS\_IN\_TREE

**Settings**   
**References**  
 Normal



**Settings**   
**References**  
 Narrow



**Settings**   
**References**  
 Wide



**Settings**   
**References**  
Display Con

**Settings**   
**References**  
Display Con

**Settings**   
**References**

**Settings**   
**Update in**  
Attributes   
attribute va  
devices and

**Settings**   
**Update in**  
Signals   
signals of bl  
connectors

**Settings**   
**Field**   
fields when  
designation

tree

RESTORE\_PINNAMES

**Settings** ☐ **Update in I**  
☐ Restore ch  
names

RETAIN\_SIGNAL\_CONNECT\_CELL

**Settings** ☐ **Autoconnec**  
signal for al

ROUTING\_NAME\_EQUIVALENCE

**Electric Se**  
**Connection**  
**Conductor**  
Conductor A  
Procedure ☐  
equivalence

SAVLIMIT

**Settings** ☐ **Automatic**  
backup file  
SpinControl

SCHEMAALTGRIDSIZE

**Electric Se**  
**General** ☐ A  
Grid ☐ Grid

SCHEMAAXISGRID

**Settings** ☐ **Display** ☐ G  
Rules (Spin

SCHEMAGRIDSIZE

**Electric Se**  
**General** □ V  
Grid size

SCHEMA\_MODE\_GRID\_AXIS

**Settings** □  
**Display** □ G  
Rules ( Che

**Settings** □  
View □ Rule  
)

SCHEMA\_MODE\_GRID\_SHEETLAYOUT

**Settings** □  
**Display** □ S  
Reference □  
layout

SCHEMAOVERSIZE

**Settings** □  
**Display** □ G  
Points ( Spin

SCHEMATRAPSIZE

**Electric Se**  
**General** □ V



SCRL\_ENABLE\_ARROWKEYS

**Settings** ☐ **Zoom / Pan**  
☐ Pan ☐ Enable  
with arrow keys

SCRL\_FACTOR

**Settings** ☐ **Zoom / Pan**  
☐ Pan ☐ Ratio

SELECTION\_ATTRIBUTE

**Settings** ☐ **Zoom / Pan**  
☐ Area Selection  
Elements in  
Selection ☐  
Templates

SELECTION\_BORDER

**Settings** ☐ **Zoom / Pan**  
☐ Area Selection  
all elements  
intersecting  
the border of  
selection rectangle

SELECTION\_DIMENSION

**Settings** ☐ **Zoom / Pan**  
☐ Area Selection  
Elements in  
Selection ☐

SELECTION\_GRAPHIC

**Settings** ☐ **Zoom / Pan**  
☐ Area Selection  
Elements in  
Selection ☐

SELECTION\_NETNODE

**Settings** ☐ **Zoom / Pan**  
☐ Area Selection  
Elements in  
Selection ☐

SELECTION\_NETSEG

SELECTION\_SYMBOL

SELECTION\_TEXT




SEPARATE\_PIN\_PORT




SEPARATOR\_ASSIGNMENT




SEPARATOR\_DEVDES

SEPARATOR\_IEC\_81346\_ASSIGNMENT


SEPARATOR\_IEC\_81346\_ATTRIBUTES


**Settings**  **Zoom / Pan**  
 Area Selection  
Elements in  
Selection 

**Settings**  **Zoom / Pan**  
 Area Selection  
Elements in  
Selection 


**Settings**  **Zoom / Pan**  
 Area Selection  
Elements in  
Selection 

**Electric Symbols**  
**Placement**  
Terminal position  
for pin and port

**Settings**  **Zoom / Pan**  
Default Description  
Higher level  
( Left EditBox )

**Settings**  **Zoom / Pan**  
Default Description  
Device design  
EditBox )

**Settings**  **Zoom / Pan**  
IEC 81346 symbols  
Separators |

**Settings**  **Zoom / Pan**  
IEC 81346 symbols  
Separators |

SEPARATOR\_IEC\_81346\_DEVDES

**Settings** □  
IEC 81346 s  
Separators |  
designation

SEPARATOR\_IEC\_81346\_LOCATION

**Settings** □  
IEC 81346 s  
Separators |

SEPARATOR\_LOCATION

**Settings** □  
Default Des  
Location ( L

SEPARATOR\_SUFFIX\_MODIFICATION\_ASSIGNMENT

**Settings** □  
**Display** □ S  
Modification  
level assign

SEPARATOR\_SUFFIX\_MODIFICATION\_DEVDES

**Settings** □  
**Display** □ S  
Modification  
designation

SEPARATOR\_SUFFIX\_MODIFICATION\_LOCATION

**Settings** □  
**Display** □ S  
Modification


SHEETREF\_FORMAT


**Settings**   
**Display**   
Reference 

**Settings**   
**Reference**  
Prefix


**Settings**   
**Reference**  
Suffix

SHEETREFSETTING


**Settings**   
**Reference**  
Sheet text

**Settings**   
**Reference**  
Reference t

SHIELD\_BOTTOM\_OVERLAPPING

**Electric Se**  
**Placement**  
**Bundle Sy**  
Parameters  
automatic p  
Line overlap  
/ Bottom

SHIELD\_TOP\_OVERLAPPING

**Electric Se**  
**Placement**  
**Bundle Sy**  
Parameters  
automatic p  
Line overlap  
/ Top

SHOW\_BUNDLE\_SYMBOL\_ONLY\_AT\_CONNECTION\_END

**Electric Se**  
**Placement**

SHOW\_CORE\_INFO\_OPTION\_COMBINATIONS

SHOW\_VARIANT\_TOOLTIP

.SORT\_FORMAT\_FILE

SPLITBLOCKOPTION

STXTFONTPTR

SUBCIRCUIT\_KEEP\_MODELTEXT\_PARAMETER

SUBCIRCUIT\_KEEP\_MODELTEXT\_VISIBILITY

SUBCIRCUIT\_KEEP\_TEXT\_PARAMETER

Possible Values

**Bundle Sy**  
Parameters  
automatic p  
Show bundl  
only at conn

**Electric Se**  
**Connection**  
**Conductor**  
Conductor A  
Procedure [   
information  
combination

**Settings [**  
**Variants/O**  
**Display [ D**  
Settings [ D  
tooltips

**Electric Se**  
**Placement**  
Terminal str  
file name

**Settings [**  
**Block [ Spl**  
the block an  
graphic con

**Settings [**  
**Symbols [**  
Parameters

**Settings [**  
**Update in l**  
**Subcircuit**  
text paramet  
Models

**Settings [**  
**Update in l**  
**Subcircuit**  
visibility for

**Settings [**  
**Update in l**

SUBCIRCUIT\_KEEP\_TEXT\_VISIBILITY

SUBCIRCUIT\_RELOAD\_ATTRIBUTES

SUBSIDIARY\_LINE\_TO\_GRAPHIC

SUFFIX\_MODIFICATION\_IS\_ACTIVE

TABLESYMBOLPTR

TERMPPLAN\_AUTOCOMPRESS

TERMPPLAN\_BORDER

TERMPPLAN\_COMBINE\_SAME\_PINNAMES

**Subcircuit**  
text param  
Symbols

**Settings** ☐  
**Update in** ☐  
**Subcircuit**  
visibility for

**Settings** ☐  
**Update in** ☐  
**Subcircuit**  
Overwrite a  
values

**Settings** ☐  
☐ Effects ☐ S  
to graphic

**Settings** ☐  
**Display** ☐ S  
Modification  
modification

**Electric Se**  
**Formboard**  
Table Symb

**Electric Se**  
**Placement**  
**Plan** ☐ Opti  
Autocompre

**Electric Se**  
**Placement**  
**Plan** ☐ Plan  
**Electric Se**

TERMPLAN\_CONSIDER\_SIGNAL\_EQUIVALENCE\_ONLY\_WITHIN\_A\_SYMBOL

TERMPLAN\_IGNORE\_SYSTEM\_SIGNALS

TERMPLAN\_INTERNAL\_EXTERNAL\_DEFINITION

TERMPLAN\_JUMPER\_INLINE

TERMPLAN\_JUMPER\_ORDER

**Placement**  
**Plan** □ Opti  
same pin na

**Electric Se**  
**Placement**  
**Plan** □ Opti  
signal equiv  
within a syn

**Electric Se**  
**Placement**  
**Plan** □ Opti  
user-defined

**Electric Se**  
**Placement**  
**Plan** □ Inter  
Definition □  
assignment/

**Electric Se**  
**Placement**  
**Plan** □ Inter  
Definition □  
assignment

**Electric Se**  
**Placement**  
**Plan** □ Inter  
Definition □

**Electric Se**  
**Placement**  
**Plan** □ Jump

**Electric Se**  
**Placement**  
**Plan** □ Jump  
Connections

**Electric Se**  
**Placement**  
**Plan** □ Jump  
by Attribute

**Electric Se**  
**Placement**  
**Plan** □ Jump  
Jumpers

TERMPLAN\_ONLINE\_UPDATE

**Electric Se**  
**Placement**  
Terminal pla  
update

TERMPLAN\_OUTPUT\_WIRES

**Electric Se**  
**Placement**  
**Plan** □ Opti  
plan

TERMPLAN\_PINVIEW\_CONNECTIONS

**Eleelectric S**  
**Placement**  
**Plan** □ Opti  
connections

TERMPLAN\_ROW\_SYMBOL

**Electric Se**  
**Placement**  
**Plan** □ Plan

TERMPLAN\_SHOW\_ALL\_EQUIVALENT\_PINS

**Eleelectric S**  
**Placement**  
**Plan** □ Opti  
equivalent p

TERMPLAN\_UNIQUE\_CONNECTIONS

**Electric Se**  
**Placement**  
**Plan** □ Opti  
connections

TERMSTRIP\_VIEW\_SYM\_NAME

**Electric Se**  
**Placement**  
**Symbol Vie**  
Strip View S  
terminal str

TERMTABLE\_AUTOCOMPRESS

**Electric Se**  
**Placement**  
**Table** □ Opt  
Autocompre

TERMTABLE\_CONSIDER\_SIGNAL\_EQUIVALENCE\_ONLY\_WITHIN\_A\_SYMBOL

**Electric Se**  
**Placement**  
**Table** □ Opt  
signal equiv



TERMTABLE\_IGNORE\_SYSTEM\_SIGNALS

within a syn

**Electric Se  
Placement  
Table** □ Opt  
user-defined

TERMTABLE\_INTERNAL\_EXTERNAL\_DEFINITION

**Electric Se  
Placement  
Table** □ Inte  
External De  
Higher leve  
assignment/

**Electric Se  
Placement  
Table** □ Inte  
External De  
Higher leve

TERMTABLE\_JUMPER\_INLINE

**Electric Se  
Placement  
Table** □ Inte  
External De  
Location

**Electric Se  
Placement  
Table** □ Jum

TERMTABLE\_JUMPER\_ORDER

**Electric Se  
Placement  
Table** □ Jum  
by Connecti

**Electric Se  
Placement  
Table** □ Jum  
by Attribute

TERMTABLE\_PINVIEW\_CONNECTIONS

**Electric Se  
Placement  
Table** □ Jum  
Jumpers

**Electric Se  
Placement  
Table** □ Opt  
view connec

TERMTABLE\_OUTPUT\_WIRES

**Electric Se**  
**Placement**  
**Table** □ Opt  
plan

TERMTABLE\_SHOW\_ALL\_EQUIVALENT\_PINS

**Electric Se**  
**Placement**  
**Table** □ Opt  
equivalent p

TERMTABLE\_UNIQUE\_CONNECTIONS

**Eelctric Se**  
**Placement**  
**Table** □ Opt  
connections

TOPOLOGY\_DESIGNER\_CROSS\_PROBING

**Settings** □  
**Highlight** □  
Hyperlink □  
Topology De  
probing

TOPOLOGY\_REGION\_SCALE\_STEP

**Electric Se**  
**Topology** □  
Sheets □ Sc  
increment o

TOPOLOGY\_SYMBOL

**Electric Se**  
**Functional**  
Shared Shee  
increment o

TRY\_AUTO\_GET\_CONNECTOR\_SYMBOL

**Electric Se**  
**Topology** □  
Topology sy

**Electric Se**  
**Functional**  
Symbols □ T  
symbol

**Electric Se**  
**Placement**  
Connector S  
Determine s  
connector w  
component  
connector sy

TRY\_AUTO\_GET\_PIN\_VIEW\_SYMBOL

**Electric Set**  
**Placement**  
**Symbol Vie**  
Symbols □ D  
symbol for p  
using placed  
symbols

**Fluid Setti**  
**Placement**  
**Symbol Vie**  
Symbols □ D  
symbol for p  
using placed  
symbols

TWISTED\_PAIR\_ARROW\_TYP

**Electric Se**  
**Placement**  
**Bundle Sym**  
Twisted Pair

TWISTED\_PAIR\_ARROW\_WIDTH

**Electric Se**  
**Placement**  
**Bundle Sym**  
Twisted Pair  
width

TWISTED\_PAIR\_BOTTOM\_OVERLAPPING

**Electric Se**  
**Placement**  
**Bundle Sym**  
Parameters  
automatic p  
Line overlap  
Twisted Pair

TWISTED\_PAIR\_LENGTH\_CALCULATION

**Electric Se**  
**Connection**  
**Conductor**  
**Calculation**  
/ Wire Calcu  
Twisted pair  
messages fo

TWISTED\_PAIR\_TOP\_OVERLAPPING

attributes  
**Electric Se**  
**Placement**  
**Bundle Sym**  
Parameters  
automatic p  
Line overlap  
Twisted Pair

TXTCOD

**Settings**   
 Font  Col

TXTDIA

**Settings**   
 Font  Sty

**Settings**   
 Effects  S

**Settings**   
 Effects  U

TXTFONTPTR

**Settings**   
 Font  Na

TXTJUST

**Settings**   
 Font  Ali

TXTLEV

**Settings**   
 Font  Lev

TXTMOD

**Settings** □ □  
□ Font □ Rat

TXTSIZ

**Settings** □ □  
□ Font □ Siz

UIP\_ASSIGN\_GATE\_MODE

**Settings** □ □  
**Update in** □  
Assignment  
matching sy  
conductors

**Settings** □ □  
**Update in** □  
Assignment  
of symbols,  
and pins

UNCONNECT\_CORES

UNCONNECT\_CORES\_ON\_DEL\_CLINE

UNCUT\_PREFIX\_IEC\_81346

UNIQUE\_SHEET\_NAMES

**Electric Se**  
**Connection**  
/ Wires □ Un  
conductor is  
in a connect

**Electric Se**  
**Connection**  
/ Wires □ Ch  
original con

**Electric Se**  
**Connection**  
/ Wires □ Ch  
original con

**Fluid Setti**  
**Connection**  
Tubes □ Und  
conductor is  
in a connect

**Fluid Setti**  
**Connection**  
Tubes □ Che  
original con

**Fluid Setti**  
**Connection**  
Tubes □ Che  
original con

**Electric Se**  
**Connection**  
**Lines** □ Del  
graphical re  
□ Conductor

**Fluid Setti**  
**Connection**  
**Lines** □ Del  
graphical re  
□ Hose/Tube

**Settings** □  
IEC 81346 s  
Separators |  
prefix

**Settings** □  
**Import** □ S  
unique shee

UNIQUE\_TERMINAL\_STRIP\_PINNAMES

**Electric Se**  
**Placement**  
Terminal strip  
numbering [  
names

UNPLACED\_PIN\_VIEWS

**Settings** [ ]  
**Purge** [ ] Ob  
pin views

UNUSED\_ATTRIBUTE\_NAMES

**Settings** [ ]  
**Purge** [ ] Ob  
attribute na

UNUSED\_BLOCK\_DEVICES

**Settings** [ ]  
**Purge** [ ] Ob  
block device

UNUSED\_CABLES

**Fluid Setti**  
**General** [ ]  
[ ] Unused ho

UNUSED\_CABLE\_TYPES

**Electric Se**  
**General** [ ]  
[ ] Unused ca

UNUSED\_COMPONENTS

**Fluid Setti**  
**General** [ ]  
[ ] Unused ho

UNUSED\_DEVICES

**Settings** [ ]  
**Purge** [ ] Ob  
components

UNUSED\_DEVICES\_FROM\_ASSEMBLIES

**Settings** ☐ **Purge** ☐ Ob  
devices from

UNUSED\_GROUPS

**Settings** ☐ **Purge** ☐ Ob  
Groups

UNUSED\_SIGNALS

**Settings** ☐ **Purge** ☐ Ob  
signals

UNUSED\_SYMBOL\_TYPES

**Settings** ☐ **Purge** ☐ Ob  
symbol type

UNUSED\_TERMINALS\_FROM\_TERMINAL\_STRIP

**Electric Se**  
**General** ☐ **I**  
☐ Unused te  
terminal str

UNUSED\_WIRES

**Electric Se**  
**General** ☐ **I**  
☐ Unused w

UPDATE\_BUNDLE\_SYMBOLS

**Electric Se**  
**General** ☐ **U**  
**Project** ☐ **B**  
☐ Update bu  
even if the s  
differs from  
defined in tl

UPDATE\_PLACE\_SINGLE\_PINS

**Electric Se**  
**General** ☐ **U**  
**Project** ☐ **C**  
Place all pin  
pins

**Fluid Setti**  
**General** ☐ **U**  
**Project** ☐ **F**



UPDATE\_RESTORE\_LOGICAL\_PIN\_DATA

all pins as s

**Settings** ☐ **Update in**  
☐ Restore ch  
pin data

UPDATE\_RESTORE\_PHYSICAL\_PIN\_DATA

**Settings** ☐ **Update in**  
☐ Restore ch  
physical pin

UPDATE\_TERMPLAN\_TABLE\_SYMBOL

**Electric Se**  
**General** ☐ **U**  
**Project** ☐ T  
for Termina  
Update tabl  
even if the s  
differs from  
defined in th

USE\_ANSI\_STANDARD\_FOR\_MATING\_CONNECTOR

**Electric Se**  
**Connection**  
**Connectors**  
designation  
Use automa  
naming

USE\_ASSIGNMENT\_OF\_CONN

**Fluid Setti**  
**Connection**  
Mating part  
automatic c  
naming

USE\_ASSIGNMENT\_OF\_SHEET

**Electric Se**  
**Connection**  
**Connectors**  
Connectors  
level assign  
location of p**Fluid Setti**  
**Connection**  
Mating Part  
level assign  
location of p**Settings** ☐ **U**  
Rules ☐ Use  
assignment  
of sheet/fiel

USE\_BLOCK\_NAME\_FOR\_DEVDES

**Electric Se**  
**Connection**  
**Connectors**  
designation  
Generate de  
designation  
connector fr  
designation  
device

USE\_BUSBAR\_PROPERTIES\_OF\_START\_BUSBAR

**Fluid Setti**  
**Connection**  
Mating part  
device desig  
mating conn  
device desig  
block and d

USE\_CORES\_WIRES\_OF\_DYN\_CABLE\_FOR\_CABLEDUCTFILL

**Electric Se**  
**Connection**  
**Lines** □ Bus  
properties o  
busbar

USE\_CORE\_TO\_SELECT\_FITTING

**Electric Se**  
**Connection**  
**Conductors**  
**Calculation**  
/ Wire Calcul  
Segment dia  
conductors/  
dynamic cal  
duct fill

USE\_DEFAULT\_VIEW\_OF\_SHEET\_FOR\_SYMBOLS\_WITHOUT\_VIEW

**Electric Se**  
**Connection**  
**Conductors**  
Conductor A  
Procedure □  
data of conc  
select connec  
terminal and

USE\_DEFAULT\_VIEW\_OF\_SHEET\_FOR\_SYMBOLS\_WITH\_VIEW

**Settings** □  
**Import** □ **D**  
Default View  
For original

USE\_DEVDES\_FOR\_BMK

**Settings** ☐ **Default Des**  
Default Des  
Device design  
for unique d

**Electric Se**  
**Connection**  
**Connectors**  
designation  
Generate de  
designation  
connector fr  
designation  
attribute 'In  
Designation

USE\_DOT\_CONN\_NAME

**Fluid Setti**  
**Connection**  
Mating part  
device design  
mating part  
designation  
attribute 'In  
Designation

**Electric Se**  
**Connection**  
**Connectors**  
designation  
Separator to

USE\_DOT\_CONN\_NAME\_SEPERATOR

**Fluid Setti**  
**Connection**  
Mating part  
to use

**Settings** ☐ **Import** ☐ **D**  
**Import** ☐ **D**  
Device Opti  
existing hier  
blocks

USE\_EXISTING\_HIERARCHICAL\_BLOCKS

**Settings** ☐ **Default Des**  
Default Des  
Higher leve  
Used for un  
designation

USE\_HLA\_FOR\_BMK

**Settings** ☐ **Connect Li**  
**Connect Li**  
Use propert

USE\_LINE\_PROPERTIES\_OF\_START\_LINE

USE\_LOC\_FOR\_BMK

starting line

**Settings** □  
Default Designation  
Location □ U  
unique design

USE\_MIL\_LINE\_STYLE\_FROM\_SYMBOL

**Electric Se**  
**MIL-Stand**  
Use line styl  
symbol

USE\_MINIMAL\_CROSSECTION\_AS\_GLOBAL\_MINIMUM

**Electric Se**  
**Connection**  
**Conductor**  
Conductor A  
Procedure □  
minimum cr  
check for m

USE\_NON\_REDLINER\_COLOR

**Settings** □  
Redlining In  
Display non  
information  
color

USE\_READ\_ONLY\_GRAPHIC\_LEVEL

**Settings** □  
Read Only L  
read-only le

USE\_SAME\_NUMERIC\_PART\_FOR\_MATING\_CONNECTOR

**Electric Se**  
**Connection**  
**Connectors**  
designation  
Use same n  
for connect  
mating conn

USE\_SELECTED\_SYMBOLS\_FOR\_ATTRIBUTE\_TEXTS

**Fluid Setti**  
**Connection**  
Mating part  
numeric par  
and mating

		<b>Settings</b> <input type="checkbox"/> <b>Symbol</b> <input type="checkbox"/> Use only selected symbols when creating texts
		<b>Settings</b> <input type="checkbox"/> <b>Variants/Options</b> <input type="checkbox"/> <b>Display</b> <input type="checkbox"/> Variants / Options Variants <input type="checkbox"/> display
		<b>Settings</b> <input type="checkbox"/> <b>Variants/Options</b> <input type="checkbox"/> <b>Display</b> <input type="checkbox"/> Variants / Options Variants <input type="checkbox"/> additional
VARIANT_ACTIVE_VARIANT		<b>Settings</b> <input type="checkbox"/> <b>Variants/Options</b> <input type="checkbox"/> <b>Display</b> <input type="checkbox"/> Variants / Options Options <input type="checkbox"/> display
		<b>Settings</b> <input type="checkbox"/> <b>Variants/Options</b> <input type="checkbox"/> <b>Display</b> <input type="checkbox"/> Variants / Options Options <input type="checkbox"/> additional
		<b>Settings</b> <input type="checkbox"/> <b>Variants/Options</b> <input type="checkbox"/> <b>Display</b> <input type="checkbox"/> Variants / Options Options <input type="checkbox"/> none
VARIANT_COLOR		<b>Settings</b> <input type="checkbox"/> <b>Variants/Options</b> <input type="checkbox"/> <b>Display</b> <input type="checkbox"/> elements with variants/options or another color (ComboBox )
VARIANT_HIGHLIGHT_COLOR		<b>Settings</b> <input type="checkbox"/> <b>Variants/Options</b> <input type="checkbox"/> <b>Display</b> <input type="checkbox"/> Display

VARIANT\_INACTIVE\_COLOR

VARIANT\_TEXT\_COLOR

VARIANT\_TEXT\_DIA

VARIANT\_TEXT\_FONT

VARIANT\_TEXT\_JUST

VARIANT\_TEXT\_MODE

Settings □ H  
variants/opt  
following co

**Settings □**  
**Variants/O**  
**Display □ D**  
Settings □ H  
for inactive  
variants/opt  
ComboBox )

**Settings □**  
**Variants/O**  
**Variant Te**  
Color  
**Settings □**  
**Variants/O**  
**Variant Te**  
Style

**Settings □**  
**Variants/O**  
**Variant Te**  
Effects □ St


**Settings □**  
**Variants/O**  
**Variant Te**  
Effects □ Un

**Settings □**  
**Variants/O**  
**Variant Te**  
Name

**Settings □**  
**Variants/O**  
**Variant Te**  
Alignment

**Settings □**  
**Variants/O**  
**Variant Te**  
Ratio




VARIANT\_TEXT\_SEPARATOR

**Settings**   
**Variants/O**  
Project Sett  
Separator fo  
texts



VARIANT\_TEXT\_SIZE

**Settings**   
**Variants/O**  
**Variant Tex**

VAR\_SHOW\_ALL\_VALUES

**Settings**   
**Variants/O**  
**Display**  D  
Settings  D  
values in tex

VERIFY\_USE\_XML\_FILE

**Settings**   
**Verify**  Us  
XML file


WAY\_OF\_TERMINAL\_PINNUMBERING

**Electric Se**  
**Placement**  
Way of Num  
Element



WCOUNT\_SYMBOL\_HOR

**Electric Se**  
**Placement**  
Way of Num  
Internal

WCOUNT\_SYMBOL\_HOR\_ALT

**Settings**   
Template Sy  
Horizontal



WCOUNT\_SYMBOL\_VER

**Settings**   
Alternative '  
Symbol  H

WCOUNT\_SYMBOL\_VER\_ALT

**Settings**   
Template Sy  
Vertical

WIRE\_AUTOROUTE\_ATTRIBUTES

**Settings**   
Alternative '  
Symbol  Ve

WIRE\_AUTOROUTE\_WITH\_ATTRIBUTES

WIRE\_NAMING\_RANGE\_ON

WIRE\_NUMBER\_FORMAT

WIRE\_SEAL\_ATTRIBUTE

WIRE\_SEAL\_ATTRIBUTE\_VALUE

ZOOM\_FACTOR

If no alternative API functions are listed, no alternative functions are available and the setting value can only be retrieved or assigned using [e3Job.GetSettingValue\(\)](#) and [e3Job.SetSettingValue\(\)](#) respectively.

---

**Electric Seals**  
**Connection**  
**Conductor**  
Attribute for  
Routing □ R  
according to

**Electric Seals**  
**Connection**  
**Conductor**  
Attribute for  
Routing □ U  
for checking

**Electric Seals**  
**Connection**  
**Conductor**  
**Naming** □ V  
Define rang

**Electric Seals**  
**Connection**  
**Conductor**  
**Naming** □ V  
Format (A<

**Electric Seals**  
**Connection**  
**Conductor**  
Core Assign  
Procedure □  
filter attribu  
seals

**Electric Seals**  
**Connection**  
**Conductor**  
Core Assign  
Procedure □  
filter attribu  
seals □ Defa  
filter attribu

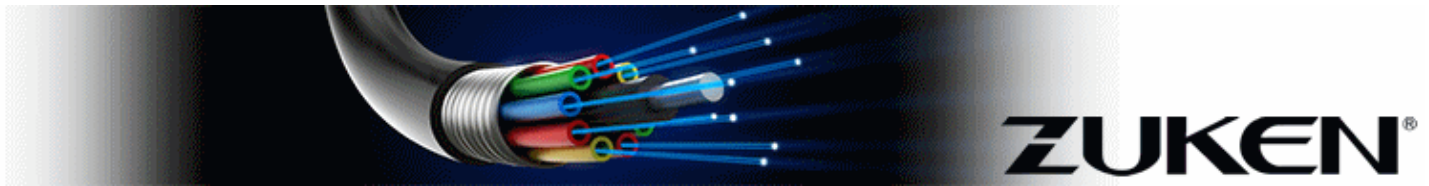
**Settings** □  
**Zoom / Pan**  
□ Zoom In/C



## See Also

- [Electric Project Settings](#)
  - [Fluid Project Settings](#)
  - [Colors](#)
  - [Line Styles](#)
  - [e3Job.GetSettingValue\(\)](#)
  - [e3Job.SetSettingValue\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

## Short Cut

## Syntax

*String shortcut*

## Description

Parameter represents a menu item's shortcut key combination.

## Possible Values

The short cut definition format is "<**Modifier Keys**><**Key**>"

For example:

"W" - for the **w** key

"^G" for **Ctrl-G** key combination

"^!I" for **Ctrl-Alt-I** key combination

"+^Home" for **Shift-Ctrl-Home** key combination

## Remarks

Combinations of the following modifier key values are possible:

Modifier Value	Description
"<Empty>"	No modifier key
"!"	Alt key
"^"	Control (Ctrl) key
"+"	Shift key

The following key values are possible:

Key Value	Description
"<Empty>"	No key
"<Single Alphabetic Letter>"	A to Z keys
"<Single Normal Character>"	Non-alphanumeric keys
"<Single Numeric Character>"	0 to 9 keys
"End"	End key
"ESC"	Escape key
"Home"	Home key
"Ins"	Insert key
"PgDn"	Page Down key
"PgUp"	Page Up key
	Function keys
"F<1..24>"	For example: "F2"
"Down"	Down cursor key
"Left"	Left cursor key
"Right"	Right cursor key
"Up"	Up cursor key
	Numerical keypad number keys
"Numpad<0..9>"	For example: "Numpad4"
"NumpadAdd"	Add numerical keypad key
"NumpadDiv"	Divide numerical keypad key
"NumpadDot"	Decimal point numerical keypad key
"NumpadEnter"	Enter numerical keypad key
"NumpadMult"	Multiply numerical keypad key
"NumpadSub"	Subtract numerical keypad key

## Version Information

Introduced in v2009-8.50.

---

## See Also

- [e3UserMenuItem.Create\(\)](#)
  - [e3UserMenuItem.GetShortCut\(\)](#)
  - [e3UserMenuItem.SetShortCut\(\)](#)
-

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

## Terminal Plan Settings

### Syntax

*String*"<SettingName> = <SettingValue>"

*Dictionary*: [*String*]key/[*String*]value pair

### Description

Parameter represents a setting for a terminal plan.

### Possible Values

Setting Key	Value	Description
"AutoCompress"	"True" or "1"	Optimize the terminal plan per connection 1 will activate
	"False" or "0"	0 will deactivate setting
"CombineSamePinNames"	"True" or "1"	If not defined or not defined in <b>E<sup>3</sup></b> . settings is used Connections to pins with the same name are handled as one
	"False" or "0"	1 will activate 0 will deactivate setting

			<div>If not defined defined in<math>E^3</math>. settings is us</div> <div>Signal equival double-deck t are displayed terminal on th level on whic plugged</div> <div>If not active, the equivalen equipotential displayed in t plan as if they plugged on th level</div>
"ConsiderSignalEquivalenceOnlyWithinOneSymbol"	"True" or "1 " "False" or "0"		<div>This is only in Jumper value "Connection"</div> <div>1 will activat</div> <div>0 will deactiv setting</div>
			<div>If not defined defined in<math>E^3</math>. settings is us</div>
"InLine"	"True" or "1 " "False" or "0"		<div>Available sinc v2018-19.42 a v2019-20.13</div> <div>Graphically c terminal pins crossing equi will be put to jumpers crea sequentially</div> <div>This might re terminal plan a jumper from although the connection do</div>
			<div>This is only in Jumper value "Connection"</div>

		1 will activate
		0 will deactivate setting
		If not defined in <b>E3</b> , settings is used
		Indicates the option for internal or external definition
	"UseAssignmentLocation"	"UseAssignment" displays the assignment and location
"InternalExternalDefinition"	"UseAssignment"	"UseAssignment" displays the assignment
	"UseLocation"	"UseLocation" displays the location
		Available since v2018-19.00
"Jumpers"	"Connections"	Indicates how jumpers on the terminal plan are handled
	"Attributes"	
	"None"	Valid values are "Connections", "Attributes", or "None"
		"Connections" means jumpers on ground connections
		"Attributes" means jumpers on the ".BRIDGE" attribute (jumper for terminal a conductor connection)
		"None" means jumpers are ignored in terminal plan
		Default value is "Connections"
		Available since v2018-19.00

		Text node is u display the system-gener (#) in the term
"OnlyUser-DefinedSignals"	"True" or "1"	1 will activate
	"False" or "0"	0 will deactiv setting
		If not defined defined in <b>E<sup>3</sup>.</b> settings is use Graphically d duplicate con between the s
	"True" or "1"	1 will activate
"PinViewConnections"	"False" or "0"	0 will deactiv setting
		If not defined defined in <b>E<sup>3</sup>.</b> settings is use Sheet format plan
"SheetFormat"	"<SheetFormatName>"	Can be any of available shee in the active s database
		Must be defin terminal plan created Sheet name o terminal plan
"SheetName"	"<SheetName>"	If not defined terminal plan created, <b>E<sup>3</sup>.s</b> generates a s
		Not stored as setting
"ShowAllEquivalentPins"	"True" or "1"	If set to "True"
	"False" or "0"	pins including pins are displ



			<div>If not defined defined in <b>E<sup>3</sup></b>. settings is us</div> <div>Available since v2018-19.50, v2019-20.31, v2020-21.21 a v2021-2201</div> <div>Symbol used connection lin representation terminal plan</div>
"TableSymbol"	"<TableSymbolName>"		<div>Can be any of available tabl the active sym database</div> <div>Can be left un table symbol for the sheet</div> <div>Uses unique point-to-point connections t the terminal p</div>
"UniqueConnections"	"True" or "1" "False" or "0"		<div>1 will activat</div> <div>0 will deactiv setting</div>
			<div>If not defined defined in <b>E<sup>3</sup></b>. settings is us</div> <div>Wires used in will be output terminal plan</div>
"WiresInPlan"	"True" or "1" "False" or "0"		<div>1 will activat</div> <div>0 will deactiv setting</div>
			<div>If not defined defined in <b>E<sup>3</sup></b>. settings is us</div>

## Remarks

Terminal plan settings are usually passed in and out of functions as elements of a [String Array](#) or [Dictionary](#).

## Version Information

Introduced in v2009-8.50.

Modified in v2018-19.00.

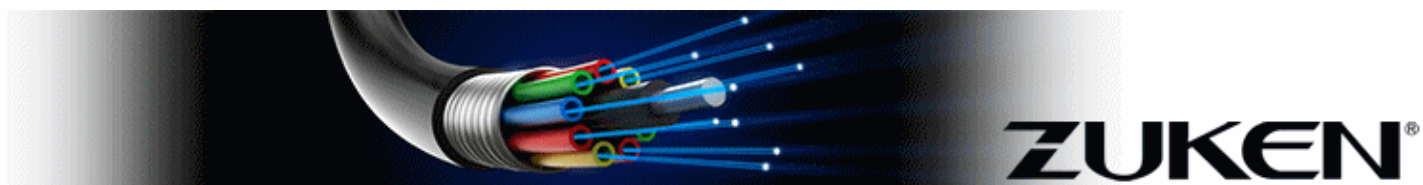
Modified in v2018-19.50, v2019-20.31, v2020-21.21 and v2021-2201.

---

## See Also

- [e3Device.GetTerminalPlanSettings\(\)](#)
  - [e3Device.InsertTerminalPlan\(\)](#)
  - [e3Device.SetTerminalPlanSettings\(\)](#)
  - [e3Job.GetTerminalPlanSettings\(\)](#)
  - [e3Job.SetTerminalPlanSettings\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).



v2022-23.00

## Text Type

## Syntax

*Integer* *text* *type*

## Description

Parameter represents an **E<sup>3</sup>** text type as an integer.

## Possible Predefined Values

Value	Maximum Length	Text Name	Information
0	-	No type	
1	255	Position	Display of jumper position
2	-	Comment	
3	64	Pin number	Pin name
4	12	Symbol designation	Symbol designation or sequential number of symbol
5	64	Component code	Component name from database
6	8	PCB grid designation	
7	200	Signal name	Signal name on pin
9	255	Pin symbol text - fix	
11	255	Text fix	Fixed text on symbol
12	252	Device designation	Device designation of device
13	255	Text variable	
14	252	Higher level assignment	Higher level assignment of device

## SetAsMaster - e3Symbol

15	252	Location	Location of device
16	255	Reference to master	Reference to master symbol
18	255	Reference to slave	Reference to slave symbol
19	64	External pin number	
20	40	Drawn by	Sheet text
21	20	Document number	Sheet text
22	50	Customer	Sheet text
23	20	Drawing number	Sheet text
24	55	Sheet number	Sheet name
25	15	Special note	Sheet text
26	255	Date	Sheet text
27	6	Issue	Sheet text
28	10	Change	Sheet text
29	10	Date of modification	Sheet text
30	6	Modified by	Sheet text
31	255	Number of sheets in project	Sheet text
32	255	Device designation	Device designation of sheet
33	50	Function	Sheet text
34	252	Higher level assignment	Higher level assignment of sheet
35	252	Location	Location of sheet
36	15	Origin	Sheet text
37	15	Replaces	Sheet text
38	15	Replaced by	Sheet text
39	255	Document type	Document or sheet type
40	30	Name 1	Name 1 of the drawing
41	10	Date of modification (1)	Sheet text
42	15	User	Sheet text
43	15	Approved	Sheet text
44	15	Standard	Sheet text
45	255	Project	Project name
46	50	Order number	Sheet text
47	50	Order	Sheet text
48	25	Name 2	Name 2 of the drawing
49	255	Reserve	Sheet text
50	50	Technical data 1	
51	50	Technical data 2	

# SetAsMaster - e3Symbol

52	50	Technical data 3	
53	255	Technical data 4	
70	60	PLC - Physical address	
71	60	PLC - Symbolic address	
72	60	PLC - Comment 1	
73	60	PLC - Comment 2	
74	255	PLC Topology Network	
75	255	PLC Topology	
76	255	PLC Network Type	
77	255	PLC Position Number	
86	255	Attribute for use from active pin terminal	
87	255	Attribute for use from View Symbol	
88	255	Text Function Attribute	
89	255	Article Type	
203	16	Connector pin name	
205	508	Block file	Block file type
206	255	Sheet reference	Sheet cross-reference with multiply displayed (split) blocks
212	252	Block designation	Project designation of a block or module
213	16	Connector pin name / total	
220	25	SAP Document Number	Sheet text
221	5	SAP Document Version	Sheet text
222	5	SAP Document Part	Sheet text
223	5	SAP Document Type	Sheet text
224	50	SAP Document Description	Sheet text
225	5	SAP Document Status	Sheet text
226	30	SAP Document Status text	Sheet text
230	255	rmCISHeaderChangeCount	
231	255	rmCISHeaderDevice	
232	255	rmCISHeaderChangeType	
233	255	rmCISChangeCount	
234	255	rmCISDevice	
235	255	rmCISChangeType	
236	255	rmCISMoreChanges	
250	255	rmCLSHeaderChangeNumber	

251	255	rmCLSHeaderPath
252	255	rmCLSHeaderChangeType
253	255	rmCLSHeaderQuantity
254	255	rmCLSHeaderMaterial
255	255	rmCLSHeaderAdditional
256	255	rmCLSHeaderOldValue
257	255	rmCLSHeaderNewValue
258	255	rmCLSHeaderLocation
259	255	rmCLSHeaderDevice
260	255	rmCLSHeaderPin
261	255	rmCLSHeaderFrom
262	255	rmCLSHeaderTo
263	255	rmCLSHeaderCoreNumber
264	255	rmCLSHeaderSignal
280	255	rmCLSChangeNumber
281	255	rmCLSPath
282	255	rmCLSChangeType
283	255	rmCLSQuantity
284	255	rmCLSMaterial
285	255	rmCLSAdditional
286	255	rmCLSOldValue
287	255	rmCLSNewValue
288	255	rmCLSFromLocation
289	255	rmCLSFromDevice
290	255	rmCLSFromPin
291	255	rmCLSToLocation
292	255	rmCLSToDevice
293	255	rmCLSToPin
294	255	rmCLSCoreNumber
295	255	CLS signal name
303	32	Module port name
312	255	rmCLSFooter
320	255	rmCRPHeaderDocumentType
321	255	rmCRPHeaderAssignment
322	255	rmCRPHeaderLocation
323	255	rmCRPHeaderSheet

## SetAsMaster - e3Symbol

324	255	rmCRPHeaderChangeIndex	
325	255	rmCRPHeaderChangeReason	
326	255	rmCRPHeaderChangeDate	
327	255	rmCRPHeaderChangeUser	
335	255	rmCRPDocumentType	
336	255	rmCRPAssignment	
337	255	rmCRPLocation	
338	255	rmCRPSheet	
339	255	rmCRPChangeIndex	
340	255	rmCRPChangeReason	
341	255	rmCRPChangeDate	
342	255	rmCRPChangeUser	
349	255	rmCRPFooter	
350	64	Pin number	Terminal pin name
351	200	Signal	Signal on terminal
352	255	Reference to schema	Cross-reference to terminal symbol's placement in the circuit diagram
353	255	Item designation (ext.)	Item designation of external device
354	64	Pin name (ext.)	Pin name of external device
355	64	Conductor name (ext.)	Conductor or wire name of external cable wire
356	255	Item designation (int.)	Item designation of internal device
357	64	Pin name (int.)	Pin name of internal device
358	64	Conductor name (int.)	Conductor or wire name of internal cable wire
359	255	Connection class	
370	80	Component code (Terminal plan sheet symbol)	Sheet text
371	64	Cable type (ext.)	Cable type of external cable
372	64	Cable type (int.)	Sheet text
373	252	Item designation of cable (ext.)	Cable type of internal cable
			Sheet text
			Item designation of external cable
			Sheet text

			Item designation of internal cable
374	252	Item designation of cable (int.)	
			Sheet text
400	255	Graph. Terminal plan (Term. Cross reference)	
401	255	Graph. Terminal plan (Term. Pin name)	
402	255	Graph. Terminal plan (Term. Device designation)	
403	255	Graph. Terminal plan (Function for destination)	
404	255	Graph. Terminal plan (Conductor name)	
405	255	Graph. Terminal plan (Attribute 1 for destination)	
406	255	Graph. Terminal plan (Attribute 2 for destination)	
407	255	Graph. Terminal plan (Attribute 3 for destination)	
408	255	Graph. Terminal plan (Attribute 4 for destination)	
409	255	Graph. Terminal plan (Attribute 5 for destination)	
410	255	Wire number	
411	255	BOM Position	Bill of materials position
412	255	Wire number (conductor)	
413	255	Additional wire number	
414	255	Internal Item Designation	
415	255	Symbol pin text extended	
416	255	Pin description	
417	255	Net segment cross-section	
418	255	Net segment circumference	
420	25	Device function	
421	255	Device sub-function	
423	255	Amount of insulation to be cut back/stripped (mm)	
424	255	Insulation	
425	255	Amount of insulation to be cut back/stripped Shield (mm)	
430	3	Nominal width	



## SetAsMaster - e3Symbol

440	255	PLC Function Type	
443	255	Length	
450	255	Bundle Type	
468	255	PID EMSR-Points-Function	
469	255	PID EMSR-Points-Item Designation	
470	255	Item Designation Electric	
471	255	Item Designation Fluid	
472	20	Item Designation Pneumatic	
473	255	Item Designation (Text Instance)	
482	255	Additional Part	
483	255	Increment	
484	255	State	
490	255	Wire identifier	
1000	252	Attribute	
1001	164	Variant	Variant name
1002	255	Connection Target	
1003	255	Conductor in connection	Conductor or wire name in the connection
1004	64	Variant value	
1005	255	Cable type in connection	
1006	255	Free symbol text	
1007	255	Signal on connection	
1008	255	Length of segment	
1009	255	Bus name	
1010	255	Device name	
1011	255	Number of cables in connection	
1012	255	Cable in connection	Device designation of the cable in the connection
1013	255	Number of conductors in connection	
1014	255	Higher level assignment of cable (connection)	
1015	255	Location of cable (connection)	
1016	255	DDS-C text type 39	Sheet text
1017	255	DDS-C text type 34	Sheet text
1018	255	DDS-C text type 35	Sheet text
1020	64	Following sheet in project	Sheet text
1021	128	Project name	Sheet text

# SetAsMaster - e3Symbol

1022	255	Project path name	Sheet text
1023	255	Project file name	Sheet text
1024	255	Wire group	Wire group of the wire in the connection
1025	255	Wire	Wire type of the wire in the connection
1026	255	Cross-section	Cross-section of the conductor or wire in the connection
1027	255	Color	Color of the conductor or wire in the connection
1028	255	Test point	
1029	255	Conductor color (int.)	Conductor or wire color of internal cable or wire
1030	255	Conductor cross-section (int.)	Conductor or wire cross-section of internal cable or wire
1031	255	Conductor length (int.)	Conductor or wire length of internal cable or wire
1032	255	Conductor type (int.)	Conductor or wire type of internal cable or wire
1033	255	Conductor color (ext.)	Conductor or wire color of external cable or wire
1034	255	Conductor cross-section (ext.)	Conductor or wire cross-section of external cable or wire
1035	255	Conductor length (ext.)	Conductor or wire length of external cable or wire
1036	255	Conductor type (ext.)	Conductor or wire type of external cable or wire
1037	255	Sheet reference	
1041	255	Plot date	Sheet text
1042	255	Plot time	Sheet text
1043	255	Plot user name	Sheet text
1044	255	Project store date	Sheet text
1045	255	Shield name (int.)	Shield name of internal cable
1046	255	Shield name (ext.)	Shield name of external cable
1047	252	Device designation (ext.)	Device designation of external device
1048	252	Higher level assignment (ext.)	

## SetAsMaster - e3Symbol

			Higher level assignment of external device
1049	252	Location (ext.)	Location of external device
1050	252	Device designation (int.)	Device designation of internal device
1051	252	Higher level assignment (int.)	Higher level assignment of internal device
1052	252	Location (int.)	Location of internal device
1053	252	Device designation of cable (ext.)	Sheet text
1054	252	Higher level assignment of cable (ext.)	Sheet text
1055	252	Location of cable (ext.)	Sheet text
1056	252	Device designation of cable (int.)	Sheet text
1057	252	Higher level assignment of cable (int.)	Sheet text
1058	252	Location of cable (int.)	Sheet text
1059	80	Component code (Terminal plan table)	
1060	80	Wire group name (int.)	
1061	80	Wire group name (ext.)	
1062	255	Item designation	
1063	255	Conductor / Wire marker position (ext. )	
1064	255	Conductor wire marker position (int.)	
1065	255	Generation time	Sheet text
1066	255	User name of draftsman	Sheet text
1067	255	Date of modification	Sheet text
1068	255	Time of modification	Sheet text
1069	255	User name of modifier	Sheet text
1070	255	Reference to pin number	
1071	255	Terminal plan table item designation	
1072	252	Terminal plan table device designation	
1073	252	Terminal plan table higher level assignment	
1074	252	Terminal plan table location	
1075	255	Connection Target device designation	

## SetAsMaster - e3Symbol

1076	255	Connection Target location	
1077	255	Connection Target higher level assignment	
1078	255	Assembly device name	
1079	252	Assembly device designation	
1080	252	Assembly location	
1081	252	Assembly higher level assignment	
1082	255	Assembly item designation	
1083	64	Assembly component code	
1085	255	Connection Target Format 1	
1086	255	Connection Target Format 2	
1087	255	Valid cavity part	
1088	255	Active connector pin terminal	
1089	255	Sheet Column	Sheet text
1090	255	Sheet Row	Sheet text
1091	255	Sheet Reference Name	
1092	255	Reference from original to view	
1093	255	Reference from view to original	
1094	255	Port name	
1095	255	Port name from panel	
1096	255	Pin and port name	
1097	255	Reference to assembly master	
1098	255	Reference to assembly slave	
1099	255	Reference between device views	
1100	255	Reference between pin views	
1101	64	Port name of terminal pin at internal side	
1102	64	Port name of terminal pin at external side	
1103	64	Port name (ext.)	
1104	64	Pin and port name (ext.)	
1105	64	Port name (int.)	
1106	64	Pin and port name (int.)	
1107	255	Connection Target Format 1 (multiline)	
1108	255	Connection Target Format 2 (multiline)	
1109	255		Sheet text

## SetAsMaster - e3Symbol

		Time of modification (sheet content only)	
1110	255	DDS-C text type 23	Sheet text
1111	255	DDS-C text type 48	Sheet text
1112	255	DDS-C text type 49	Sheet text
1113	255	Functional unit	
1114	255	Functional port	
1115	255	Conductor manufacturing length	
1116	255	Number of windings per meter for conductors	
1117	252	Name of connector insert	
1118	200	Signal class	
1119	255	Signal class on connection	
1120	255	eCheck ambient temperature	
1121	255	eCheck maximum temperature	
1122	255	eCheck actual temperature	
1123	255	eCheck resistance	
1124	255	eCheck minimum current	
1125	255	eCheck nominal current	
1126	255	eCheck maximum current	
1127	255	eCheck actual current	
1128	255	eCheck nominal voltage	
1129	255	eCheck minimum voltage	
1130	255	eCheck maximum voltage	
1131	255	eCheck actual voltage	
1132	255	eCheck internal pin resistance	
1133	255	eCheck voltage drop	
1134	255	eCheck rating	
1135	255	eCheck power	
1136	255	eCheck specific resistance	
1137	255	eCheck weight	
1138	255	eCheck cross-section	
1139	255	eCheck color	
1140	255	eCheck internal resistance	
1141	255	Reference to schematic	
1142	255	Reference to panel	
1150	255	Number of sheets in folder	Sheet text

## SetAsMaster - e3Symbol

1151	64	Previous sheet in folder	Sheet text
1152	64	Following sheet in folder	Sheet text
1153	64	First sheet in folder	Sheet text
1154	64	Last sheet in folder	Sheet text
1155	255	Following sheet exists in folder?	Sheet text
1156	64	Previous sheet in project	Sheet text
1157	64	First sheet in project	Sheet text
1158	64	Last sheet in project	Sheet text
1159	255	Has following sheet in project?	Sheet text
1161	200	Signal of conductors/wires	
1162	255	Variants/options text of assembly	
1163	255	Variants/options text of device	
1164	255	Variants/options text of symbol	
1165	255	Variants/options text of connection	
1166	255	Variants/options text combined	
1167	255	Active Wire seal	
1168	255	Default wire group	
1169	255	Default wire type	
1170	255	Default wire color	
1171	255	Default wire cross-section	
1172	255	Default wire outer diameter	
1173	255	Default wire name	
1174	255	Outer diameter	Outside diameter of wires and cables
1175	255	Model size (X)	Model size on the x-axis
1176	255	Model size (Y)	Model size on the y-axis
1177	255	Model size (Z)	Model size on the z-axis
1178	255	Device letter code	
1179	255	Sheet Region Scaling	Sheet text
1180	255	Sheet Region Reference X	Sheet position of sheet region on the x-axis
1181	255	Sheet Region Reference Y	Sheet position of sheet region on the y-axis
1182	255	Sheet Region Reference Z	Sheet position of sheet region on the z-axis
1183	200	Signal (ext.)	
1184	200	Signal (int.)	
1185	255	Outer segment diameter	

5000	255	Saber Model Name
5001	255	Saber Library
5002	255	Saber Parameter 1
5003	255	Saber Pin
5100	255	E3.ePLM FunctionName
5101	255	E3.ePLM Description
5102	255	E3.ePLM Function
5200	255	E3.WiDGen Option
5201	255	E3.WiDGen Pinfunction

---

## See Also

- [e3Job.GetTextTypes\(\)](#)
  - [e3Text.GetType\(\)](#)
  - [e3Text.GetTypeId\(\)](#)
- 

More questions? Please contact your local support office or Zuken Global Support (ZGS) if support calls can be logged by the user (<https://support.zuken.com/global/>).