

COMOS

Platform
Class documentation POptions_dll

Programming Manual

Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

DANGER

indicates that death or severe personal injury **will** result if proper precautions are not taken.

WARNING

indicates that death or severe personal injury **may** result if proper precautions are not taken.

CAUTION

indicates that minor personal injury can result if proper precautions are not taken.

NOTICE

indicates that property damage can result if proper precautions are not taken.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

WARNING

Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Table of contents

1	Class: POptions.....	5
1.1	Public Subs.....	5
1.1.1	SetOption.....	5
1.1.2	RefreshOptions.....	5
1.1.3	ScanPipeClasses.....	5
1.1.4	ResetPreferredCDeviceForFunctionCode.....	6
1.2	Public Functions.....	6
1.2.1	GetOptionSystemDefault.....	6
1.2.2	CheckOptCode.....	6
1.2.3	GetConnectionTypeProperties.....	6
1.2.4	GetConnectionTypeProperties2.....	6
1.2.5	GetIndexOption.....	7
1.2.6	GetIndexOption2.....	7
1.2.7	GetIndexOptionWithSymbol.....	7
1.2.8	GetOptionFromProject.....	7
1.2.9	GetOption.....	8
1.2.10	GetHeaderClassTable.....	8
1.2.11	GetStdTableBySystemFullName.....	8
1.2.12	PipeClassRootCDevice.....	8
1.2.13	GetConnectionTypeTable.....	8
1.2.14	GetPrefixOrSeparatorForClassAndDetailClass.....	9
1.2.15	GetPrefixOrSeparatorByString.....	10
1.2.16	GetPrefixOrSeparatorByStringAndProject.....	12
1.2.17	GetOptions.....	13

Class: POptions

1.1 Public Subs

1.1.1 SetOption

```
SetOption ( ByVal optCode as Long, ByRef POption as Variant )
```

Sets a project option to a new value. Only applies to the current COMOS instance: when COMOS is closed and re-opened, the default value applies once again.

- Parameter `optCode`
The number of the project option to be changed
Example: "69" for "Automatic referencing of documents in @Namesystem".
Alternative: Call via constant, e.g. "PO_AutoDocRefNameSystem". The alternative is not available in the script, but only from DLLs with a reference to ivbcomos.dll.
- Parameter `POption`
The new value.
Please note: Depending on the POption, the type can be String, Integer, Long, etc., or also a COMOS object. COMOS objects can be identified by the prefix "IComos".

1.1.2 RefreshOptions

```
RefreshOptions ( ByVal WorkSet as IComosDWorkset )
```

Performs a complete update of all POptions for the current database.

- Parameter `WorkSet`
The current workset instance. Has the same meaning as the currently open database.

1.1.3 ScanPipeClasses

```
ScanPipeClasses ( Optional ByVal CDevPipeClass as , Optional ByVal Prefix as String = "", Optional ByVal Recursive as Boolean = False )
```

- Parameter `CDevPipeClass`
- Parameter `Prefix`
- Parameter `Recursive`

For internal use.

1.2 Public Functions

1.1.4 ResetPreferredCDeviceForFunctionCode

`ResetPreferredCDeviceForFunctionCode ()`

For internal use

1.2 Public Functions

1.2.1 GetOptionSystemDefault

`GetOptionSystemDefault (ByVal optCode as Long) as Variant`

Supplies the default value of the project option.

- Parameter `optCode`
The number of the project option whose default value is to be returned.

1.2.2 CheckOptCode

`CheckOptCode (ByVal optCode as Long) as Boolean`

Checks the validity of a project option. A POption is valid if its option code is between 1 and the total number of POptions. The number of POptions is recorded in `ivbcomos.dll`.

- Parameter `optCode`
The number of the project option to be checked.

1.2.3 GetConnectionTypeProperties

`GetConnectionTypeProperties (ByVal ConLayer as String, ByVal ConCategorie as String, Optional ByVal Planart as String = "") as TypConnectionType`

Calls `GetConnectionTypeProperties2`. Recommendation: Use `GetConnectionTypeProperties2`, see chapter `GetConnectionTypeProperties2` (Page 6).

- Parameter `ConLayer`
- Parameter `ConCategorie`
- Parameter `Planart`

1.2.4 GetConnectionTypeProperties2

`GetConnectionTypeProperties2 (ByVal ConLayer as String, ByVal ConCategorie as String, ByRef bStdValueFound as Boolean) as TypConnectionType`

- Parameter ConLayer
- Parameter ConCategorie
- Parameter bStdValueFound

For internal use.

1.2.5 GetIndexOption

GetIndexOption (ByVal optCode as Long, ByVal Index as Variant) as Variant

- Parameter optCode
- Parameter Index

Calls GetIndexOption2. Recommendation: Use GetIndexOption2, see chapter GetIndexOption2 (Page 7).

1.2.6 GetIndexOption2

GetIndexOption2 (ByVal optCode as Long, ByVal Index as String, ByRef bStdValueFound as Boolean) as TypConnectionType

- Parameter optCode
- Parameter Index
- Parameter bStdValueFound

For internal use.

1.2.7 GetIndexOptionWithSymbol

GetIndexOptionWithSymbol (ByVal optCode as Long, ByVal Index as Variant, Optional ByVal Planart as String = "") as Variant

- Parameter optCode
- Parameter Index
- Parameter Planart

Obsolete; only provided for compatibility reasons.

1.2.8 GetOptionFromProject

GetOptionFromProject (ByVal optCode as Long, ByVal otherProject as IComosDProject) as Variant

Returns the value of a POption from another project.

- Parameter `optCode`
The number of the project option whose value is to be returned.
- Parameter `otherProject`
The object of the COMOS project from which the POption is to be calculated.

1.2.9 GetOption

`GetOption (ByVal optCode as Long) as Variant`

Returns the value of a POption from the current project.

- Parameter `optCode`
The number of the project option whose value is to be returned.

1.2.10 GetHeaderClassTable

`GetHeaderClassTable (ByVal TableName as String) as IComosDStandardTable`

- Parameter `TableName`

For internal use.

1.2.11 GetStdTableBySystemFullName

`GetStdTableBySystemFullName (ByVal StdTableFullName as String) as IComosDStandardTable`

Returns a standard table from the current project. Alternatively, the `GetStandardTable` routine can be called from the `Lib.dll` with the same result.

- Parameter `StdTableFullName`
The `FullName` of the standard table to be returned.

1.2.12 PipeClassRootCDevice

`PipeClassRootCDevice () as IComosDCDevice`

For internal use.

1.2.13 GetConnectionTypeTable

`GetConnectionTypeTable (ByVal ConLayer as String) as IComosDStandardTable`

- Parameter `ConLayer`

For internal use.

1.2.14 GetPrefixOrSeparatorForClassAndDetailClass

```
GetPrefixOrSeparatorForClassAndDetailClass ( ByVal prefixOrSeparator  
as String, ByVal comosObject as IComosBaseObject ) as String
```

Returns the set value for the specified prefix type or delimiter type and for the COMOS object. Refers to the "Prefix and delimiter" mask in the project options.

For objects of the type `IComosDDocument`, only the class of the base object is evaluated.

The following applies to objects of the type `IComosDDDevice` or `IComosDCDevice`: first the detail class is evaluated. If this is empty, the class is evaluated.

- Parameter `prefixOrSeparator`
The prefix or delimiter. Permitted values:
 - "LabelPrefix": Prefix
 - "LabelSeparator2": Delimiter (same type)
 - "LabelSeparator1": Delimiter (different type)
- Parameter `comosObject`
The COMOS object from which the class or detail class is read out. Permitted types:
 - `IComosDDocument`
 - `IComosDDDevice`
 - `IComosDCDevice`

Example 1: Object with class and without detail class

Default in COMOS:

Prefix and delimiter	
Classes	Location
Prefix	+
Delimiter (different type)	+
Delimiter (same type)	.
<input checked="" type="checkbox"/> Objects of the 'General' class get prefixes and separators according to use as unit/location/document group	

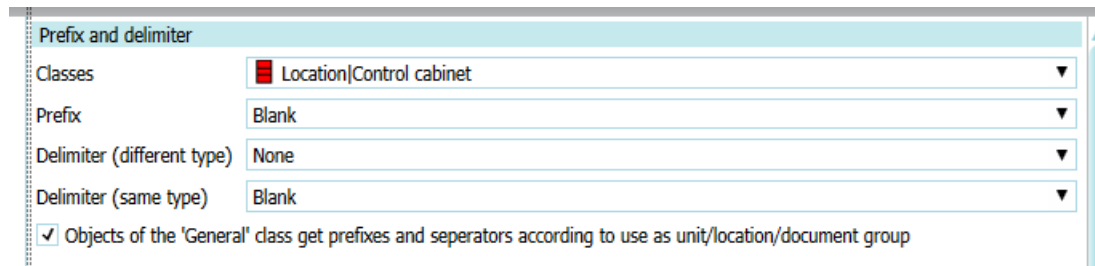
- Object to be evaluated: "A40 Location, two-level"
Object is derived from the "@30 > M00 > A20 > A10 > A40 > | A40" base object
Class of the object: B; Detail class: Empty

Evaluation takes place in the script. The engineering object to be evaluated was first assigned to the parameter "b".

- Evaluation for: Prefix
`GetPrefixOrSeparatorForClassAndDetailClass("LabelPrefix", b)`
 Result: "+"
- Evaluation for: Delimiter (different type)
`GetPrefixOrSeparatorForClassAndDetailClass("LabelSeparator1", b)`
 Result: "+"
- Evaluation for: Delimiter (same type)
`GetPrefixOrSeparatorForClassAndDetailClass("LabelSeparator2", b)`
 Result: "."

Example 2: Object with class and detail class

Default in COMOS:



- Object to be evaluated: "A10 Cabinet"
 Object is derived from the "@30 > M00 > A40 > A20 > A10 > A40 > A10 | A10" base object
 Class of the object: B; Detail class: N

Evaluation takes place in the script. The engineering object to be evaluated was first assigned to the parameter "b".

- Evaluation for: Prefix
`GetPrefixOrSeparatorForClassAndDetailClass("LabelPrefix", b)`
 Result: ()
- Evaluation for: Delimiter (different type)
`GetPrefixOrSeparatorForClassAndDetailClass("LabelSeparator1", b)`
 Result: (/)
- Evaluation for: Delimiter (same type)
`GetPrefixOrSeparatorForClassAndDetailClass("LabelSeparator2", b)`
 Result: ()

1.2.15 GetPrefixOrSeparatorByString

```
GetPrefixOrSeparatorByString ( ByVal prefixOrSeparatorAndClass as
String ) as String
```

Returns the set value for the specified prefix type or delimiter type. Refers to the "Prefix and delimiter" mask in the project options.

- Parameter `String`
The prefix type or delimiter type including input of the class or detail class in the following format:
 - <LabelPrefix/Separator1/Separator2>Class<ObjectClass>Or
 - <LabelPrefix/Separator1/Separator2>DetailClass<ObjectDetailClass>

Example 1: Object with class and without detail class

Default in COMOS:

Prefix and delimiter	
Classes	Location
Prefix	+
Delimiter (different type)	+
Delimiter (same type)	.
<input checked="" type="checkbox"/> Objects of the 'General' class get prefixes and separators according to use as unit/location/document group	

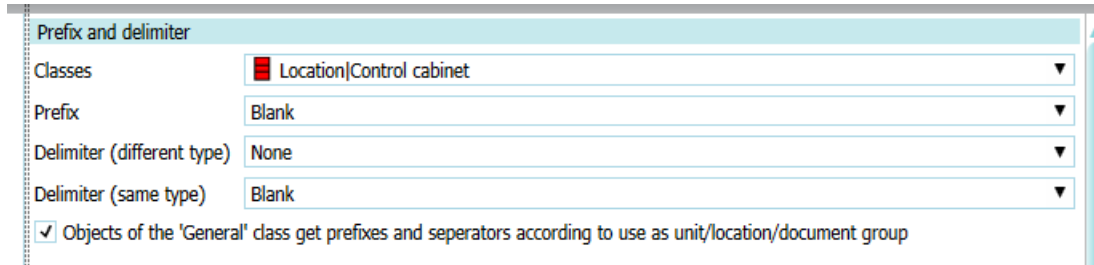
- Object to be evaluated: "A40 Location, two-level"
Object is derived from the "@30 > M00 > A20 > A10 > A40 > | A40" base object
Class of the object: B; Detail class: Empty

Evaluation takes place in the script.

- Evaluation for: Prefix
`GetPrefixOrSeparatorByString("LabelPrefixClassL")`
Result: "+"
- Evaluation for: Delimiter (different type)
`GetPrefixOrSeparatorByString("LabelSeparator1ClassL")`
Result: "+"
- Evaluation for: Delimiter (same type)
`GetPrefixOrSeparatorByString("LabelSeparator2ClassL")`
Result: "."

Example 2. Object with class and detail class

Default in COMOS:



- Object to be evaluated: "A10 Cabinet"
Object is derived from the "@30 > M00 > A40 > A20 > A10 > A40 > A10 | A10" base object
Class of the object: B; Detail class: N

Evaluation takes place in the script.

- Evaluation for: Prefix
`GetPrefixOrSeparatorByString("LabelPrefixDetailClassN")`
Result: ()
- Evaluation for: Delimiter (different type)
`GetPrefixOrSeparatorByString("LabelSeparator1DetailClassN")`
Result: (/)
- Evaluation for: Delimiter (same type)
`GetPrefixOrSeparatorByString("LabelSeparator2DetailClassN")`
Result: ()

See also

[GetPrefixOrSeparatorForClassAndDetailClass \(Page 9\)](#)

1.2.16 GetPrefixOrSeparatorByStringAndProject

```
GetPrefixOrSeparatorByStringAndProject ( ByVal  
prefixOrSeparatorAndClass as String, ByVal project as  
IComosDProject ) as String
```

Works in a similar way to `GetPrefixOrSeparatorForClassAndDetailClass`, but delivers values from the specified project. See chapter `GetPrefixOrSeparatorForClassAndDetailClass` (Page 9).

- Parameter `prefixOrSeparatorAndClass`
The prefix or delimiter. Permitted values:
 - "LabelPrefix": Prefix
 - "LabelSeparator2": Delimiter (same type)
 - "LabelSeparator1": Delimiter (different type)
- Parameter `project`
The project from which the values are to be read out.

1.2.17 **GetOptions**

`GetOptions (ByVal optCode as Long) as Variant`

Returns a set of related project options. Currently supported: Parameters from the PQM range.

- Parameter `optCode`
Currently supported: `PO_M65_DDM_GRP (217)`

