Unity SFC View
Version 2.0
User's manual

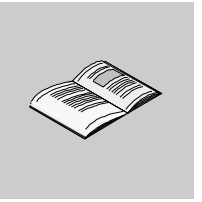
June 2004

a brand of Schneider Electric

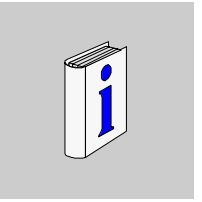
Telemecanique

33003015.00

Table of Contents



About the Book



At a Glance

Document Scope

This documentation contains a description of the Unity SFCView.

Validity Note

This document applies to Unity SFC View 2.0 with Unity Pro 2.0, OPC Factory Server 3.1, Microsoft Windows 2000 or Microsoft Windows XP Professional.

Product Related Warnings

The data and illustrations found in this document are not binding. We reserve the right to modify our products in line with our policy of continuous product development. The information in this document is subject to change without notice and should not be construed as a warranty by Schneider Electric.

Schneider Electric assumes no responsibility for any errors that may appear in this document. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us.

No part of this document may be reproduced in any form or by any means, electronic or mechanical, including photocopying, without express written permission of Schneider Electric.

All pertinent state, regional, and local safety regulations must be observed when installing and using this product. For reasons of safety and to ensure compliance with documented system data, only the manufacturer should perform repairs to components.

When controllers are used for applications with technical safety requirements, please follow the relevant instructions.

Failure to use Schneider Electric software or approved software with our hardware products may result in injury, harm, or improper operating results.

Failure to observe this product related warning can result in injury or equipment damage.

User Comments

We welcome your comments about this document. You can reach us by e-mail at TECHCOMM@modicon.com

Unity SFC View

I

At a Glance

Overview

This section contains information on the Unity SFCView software package, henceforth referred to as SFCView .

What's in this Part?

This part contains the following chapters:

General description

1

General description

Target group for documentation

The documentation is targeted towards configuration specialists who want to integrate SFC View in an HMI application and towards users of such HMI applications.

Brief description

Unity SFCView is made up of an ActiveX control and its SFCView function block library for Unity Pro.
The ActiveX control is configured on an operating station (HMI) and can then make real-time information available regarding the state of the sections in the controller.
It can:

- Display sections
- show the state of sections
- show the section diagnosis information
- navigate through sections
- control sections online
- automatically recognize changes in project data

Note: Several instances for a control can also be configured on one operating station. This makes it possible to access several chains at one time.

Three views

SFCView has three views:

- Overview
- Details
- Details Simple

The view: shows all the sections and is used to select the desired section.
The views: Details and Details Simple show information on the state of a selected section as well as diagnosis information. In these views it is also possible to navigate through the individual sections.

Programming environment

The programming environment is a HMI application that supports ActiveX container.

SFC View Demo-Application

A sample program written in Visual Basic is installed along with SFC View.
This **SFC View demo application** is a standalone program that can also be executed without Visual Basic.
Nearly all the properties, methods and events for SFC View were configured in this program and can be tested in conjunction with Unity Pro, the OPC Factory Server (OFS), and the Unity Pro PLC Simulator.
The user can experiment with this demo application and learn how SFC View is used and programmed.
(See *Starting and Using the SFC View Demo Application, p. 37*).

Requirements and restrictions

2

Introduction

Overview

This chapter contains information on requirements and restrictions which should be noted when using Unity SFC View.

What's in this Chapter?

This chapter contains the following topics:

System requirements

Operating System	One of the following operating systems is required: • Microsoft Windows 2000 • Microsoft Windows XP Professional
Unity Pro Version	Unity Pro V2.0 must be installed at least once on the system.
SFC View Library	The SFC View library must be installed (update type library).
OPC Factory Server	An OPC Factory Server (OFS) Version 3.1 must be installed on the system.
ActiveX Container	The programming environment is a HMI application that supports ActiveX container.
HMI Application	HMI applications tested for Unity SFC View are: • Monitor Pro 7.2, Schneider Electric • Vijeo Look 2.5, Schneider Electric • iFix 3.0, GE Fanuc • InTouch 7.2, Woderware Other applications tested are: • Internet Explorer 6.0, Microsoft • Visual Basic 6.0, Microsoft
PLCs supported	The following PLCs programmed with Unity Pro are supported: • Quantum • Premium • Atrium

System Architecture

General Mode of Operation	SFC View reads the structure of the sections and other data from the Unity Pro project (*.stu). For this Unity Pro must be installed. Via the OPC Factory Server (OFS), SFC View reads the online data from the SFC (sections status, variable status, diagnosis buffer etc.). SFC View recognizes project changes in the PLC and automatically updates the display.

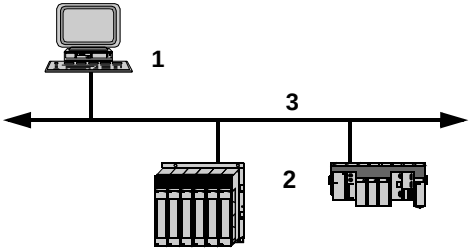
This guarantees that the sections display in the HMI application and the PLC program always remain consistent.
Function blocks from the SFC View library must be used for a particular diagnosis mode and for online controlling when configuring Unity Pro.
Apart from this no other programming needs to be carried out in the PLC.

System configuration

SFC View can be used in simple and in distributed system configurations.

Simple Configuration

A simple configuration is shown below:



1 Operating station

2 PLCs

3 Communication

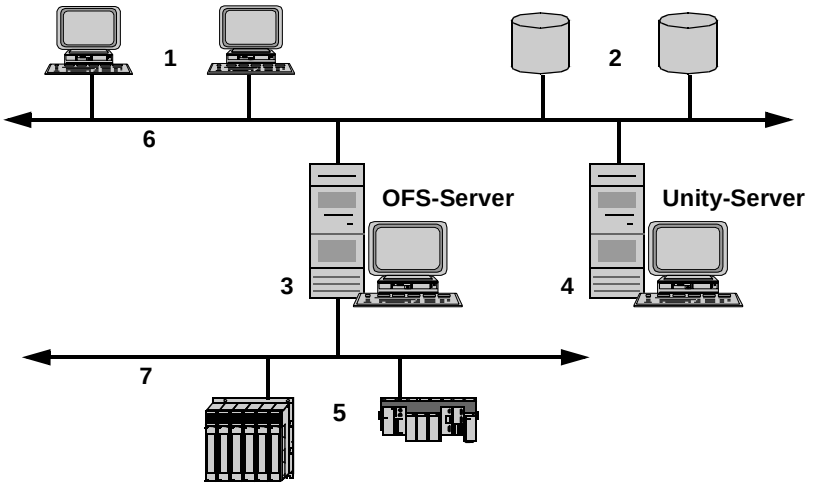
Operating station (1)
The following components must be on the operating station:
- HMI Application
- Unity Pro program
- Unity Pro projects
- OPC Factory Server (OFS) as a server installation
- SFC View

PLCs (2)
The Unity Pro projects run on the PLCs with the sections to be visualized.

Communication (3)
The communication between the operating station and the PLCs can be carried out via the following buses, as long as these are supported by the relevant PLCs:
- Modbus
- Modbus Plus
- Modbus TCP-IP
- Uni-Telway

Distributed Configuration

A distributed configuration is shown below:



1 Operating stations

2 additional PCs

3 OPC Factory Server

4 Unity Server

5 PLCs

6 Communication via Modbus TCP-IP

7 Communication via various busses e.g. Modbus Plus

Operating stations (1)
The following components must be on the operating stations:
- HMI Application
- OPC Factory Server (OFS) as a client installation
- SFC View

Additional PCs (2)
The Unity Pro projects are stored on additional PCs in the system and must be accessible by the operating stations and the OPC Factory Server.
OPC Factory Server (3)
The OPC Factory Server (OFS) is on this PC as a server installation.
Unity Server (4)
The Unity Pro program is installed on the Unity Server.

Note: The Unity Pro program must be installed **once** and only once on the entire system.

PLCs (5)
The Unity Pro projects run on the PLCs with the sections to be visualized.
Communication via Modbus TCP-IP (6)
The communication between the operating station and the PCs is carried out via the Modbus TCP-IP (DCOM).
Communication via various buses (7)
The communication between the OFS server and the PLCs can be carried out via the following buses, as long as they are supported by the relevant PLCs:
- Modbus
- Modbus Plus
- Modbus TCP-IP
- Uni-Telway

User requirements

Programming environment	The programming environment is a HMI application that supports ActiveX container.
Configuring SFC View	In order to be able to configure SFC View in a HMI application the user should have a basic understanding of the following areas: ● SFC programming in Unity Pro ● OFS parameterization ● Embedding ActiveX controls in HMI applications
Programming interface	The SFC View software package has a programming interface that makes it possible to set up SFC View in such a way that that the workflow and display is appropriate for a HMI application.
SFC View Demo-Application	A sample program written in Visual Basic is installed along with SFC View. This **SFC View demo application** is a standalone program that can also be executed without Visual Basic. Nearly all the properties, methods and events for SFC View were configured in this program and can be tested in conjunction with Unity Pro, the OPC Factory Server (OFS), and the Unity Pro PLC Simulator. The user can experiment with this demo application and learn how SFC View is used and programmed. (See *Starting and Using the SFC View Demo Application, p. 37*).

Prerequisites for online controlling

General

In order to be able to control sections from SFCView online, certain prerequisites must be fulfilled in the appropriate Unity Pro project.

Function Block SFCVIEW_CTRL

Online control is carried out using a special `SFCVIEW_CTRL`function block in the appropriate Unity Pro project.
For this reason for every Unity Pro project a function block of this type `SFCVIEW_CTRL` must be configured.

Note: Function blocks from the SFCView library may only be used in FBD sections in the Unity Pro projects.

SFC View Library

The `SFCVIEW_CTRL`function block is available for use after the SFCView library has been installed. See *Installation of the SFC View Library, p. 30*).
A description of the SFCView function blocks can be found in the *SFCView block library, p. 221* chapter.

Unity Pro variables

In Unity Pro a variable of type `SVCCMD` must be defined.
If the variable name chosen is `SVC_Command`, SFCView will find the variable automatically.
If another name is chosen, it must be assigned in SFCView using the `ChainControlVariableName` property.

Prerequisites for the diagnosis

General

There are two basic diagnosis modi:

- Diagnostics via the PLC diagnostics buffer
- Diagnostics via the SFC View function blocks in the Unity Pro Project

Note: Further information on diagnostics can be found in the *View: Details, p. 60* chapter.

SFC View Library

For the SFCView function block diagnostics, the SFCView library must be installed. (See *Installation of the SFC View Library, p. 30*).
A description of the SFCView function blocks can be found in the *SFCView block library, p. 221* chapter.

Settings in Unity Pro

To use the diagnostics in Unity Pro, certain settings must be made. (See *Unity Pro presettings, p. 34*).

Control diagnostic mode

The diagnostic mode is controlled via the control property `UsePLCDiagSystem`:

Property `UsePLCDiagSystem`	Diagnostics via the SPS diagnostics buffer	Diagnostics via the SFCView function blocks
`True`	X	-
`False`	-	X

Diagnostics via the PLC diagnostics buffer

If the `UsePLCDiagSystem` property is set to `True` the control reads the diagnosis information from the PLC diagnostic buffer.

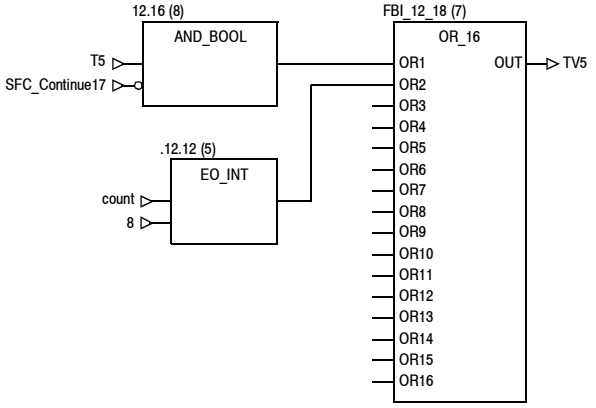
Diagnostics via the SFCView function blocks

If the property `UsePLCDiagSystem` is set to `False` , the diagnosis is done via special SFC View function blocks in the respective Unity Pro project.
For this reason, for every transition in the Unity Pro project, a SFC View function block of this type must be configured and its output variable used as a transition variable.
The following SFC View function blocks are available for use after the SFCView library has been installed:

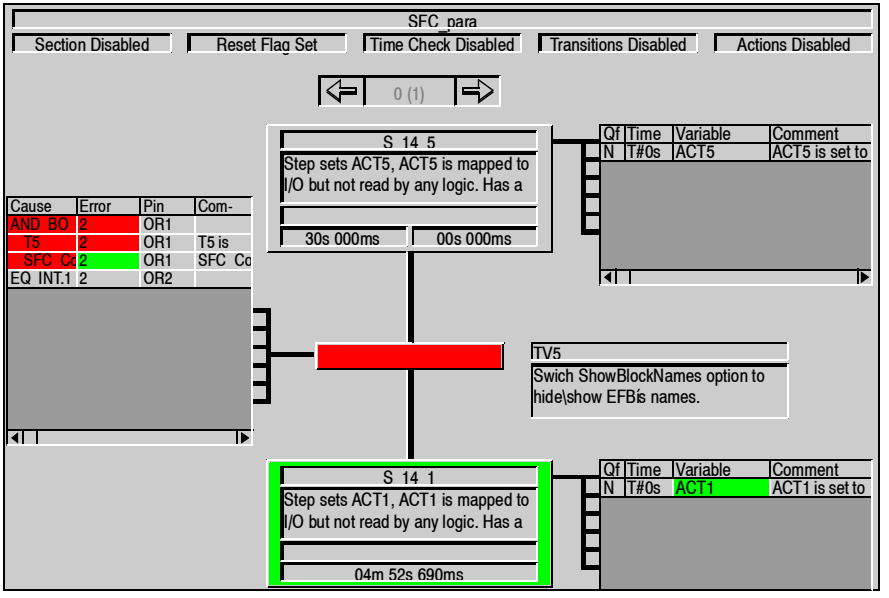
- `AND_16`
- `OR_16`
- `AND_OR_8`

Variables or a logic can be used as inputs for these function blocks. All inputs and the nested logic for these function blocks are analyzed by SFCView.
Every connection for an SFCView function block which is not an `AND_BOOL` or `OR_BOOL`function block is displayed in SFCView as an unknown connection (see `EQ_INT` function block graphic).
Inputs for SFCView function blocks which are not occupied will be ignored.

Example from a Unity Pro project



Representation in SFCView



Note: SFCView function blocks may only be used in FBD sections in the Unity Pro projects, since SFCView only looks for them there.

Differences between the diagnostic modes

The user should decide which diagnosis mode he wants to use for each section.
Only one diagnosis mode should be used for each section.
The following table can be used to make decisions.

Diagnostics via the SPS diagnostics buffer	Diagnostics via the SFCView function blocks
A transition logic must be configured in a transition section.	The transition logic must be configured using SFCView function blocks.
Only erroneous signals will be shown:	All its signals will be shown:
Errors will only be shown after the maximum supervision time has been exceeded, i.e. a maximum supervision time must be configured.	Transition logic and diagnosis results are always shown
The diagnostics information can also be read by other tools.	The diagnostics information can only be read by SFCView.
No more programming is required	The SFCView function blocks must be configured in an FBD section of their own.

Restrictions

General

SFCView supports the IEC-compliant functionalities for section programming in Unity Pro.
Other functionalities from Unity Pro are **not supported** or **supported to a limited extent** by this version of SFCView.
This must be noted when creating Unity Pro projects.

Multi-Token

The execution sequence **Multi-Token** is **not** supported by SFCView.

Macro steps

Macro sections are symbolically views in SFCView (double line above and below).
The lower steps in macro steps are **not** shown in SFCView.

Action section

For action sections in SFCView (in the details view) only the following is shown:

- Section name
- Type (S = Action **S**ection)

The logic contained in the action section is **not**shown in the SFCView.

Installation

3

Introduction

Overview

This chapter contains information on installations that are required to use Unity SFC View.

What's in this Chapter?

This chapter contains the following topics:

Installation sequence

Prescribed sequence

Adhere to the following installation sequence to make sure that SFC View is operating without any errors:

1. Unity Pro 2.0
2. OPC Factory Server (OFS) 3.1
3. SFC View 2.0
4. SFC View Library
5. HMI software

Software versions

Make sure that only software versions are installed which are guaranteed to work together perfectly . For information on this see the `readme.txt` file.

Installation and registration of Unity Pro and OFS

Installation

Install Unity Pro and the OPC (OFS) as described in their respective manuals.

Registration

Register Unity Pro and the OPC (OFS) as described in their respective manuals.

Installing SCF View

Changing and canceling

For the installation described here you can change options already selected by clicking on **Back** or cancel the installation by clicking on **Cancel**.

Installation directories

The installation directories for the software are preset as defaults. However they can be changed.
If this is necessary, click **Find...**to select a different installation directory.

Installing SFC View

The step-by-step instructions for installing the SFCView software are described here.

Step	Action
1	Insert the CD-ROM.
2	Execute the Setup.exe file.
3	Select the language for the installation wizard and confirm the SFCView installation by clicking on **OK**.
4	Confirm the start screen by clicking on**Next**.
5	Read the licensing conditions and accept them by clicking on **Yes**.
6	Check the configuration and then confirm it with **Next**.
7	Enter your name, the name of your company and the serial number and then confirm by clicking on **Next**.
8	Select the directory into which the program should be installed using **Find** and confirm using**Next**.
9	Select a program folder and confirm it using **Next**..
10	Check the current settings confirm them finally by clicking on **Next**.
11	Exit the installation by clicking on **Finish**to return to Windows. **Note**: If you have selected the option **Install the SFC View library in Unity** the installation program necessary for this starts automatically.

Installation of the SFC View Library

General

To use the SFC View function block diagnosis, the SFC View library must be installed.

Automatic start

The program **Types Library Update** starts either automatically at the end of the installation of SFC View or it can be started as described below.

Execute update

These instructions describe the steps that must be carried out to install the SFC View library using the tool **Type library update** .

Step	Action
1	Start the program via: **Start** → **Programs** → **Schneider Electric** → **Unity Pro** → **Types Library Update**
2	Select the `family.dsc` file. The file is on the CD in `...\Unity SFCView Lib\family.dsc`
3	Start the installation via **Install family**.
4	Click on the **Exit**button.

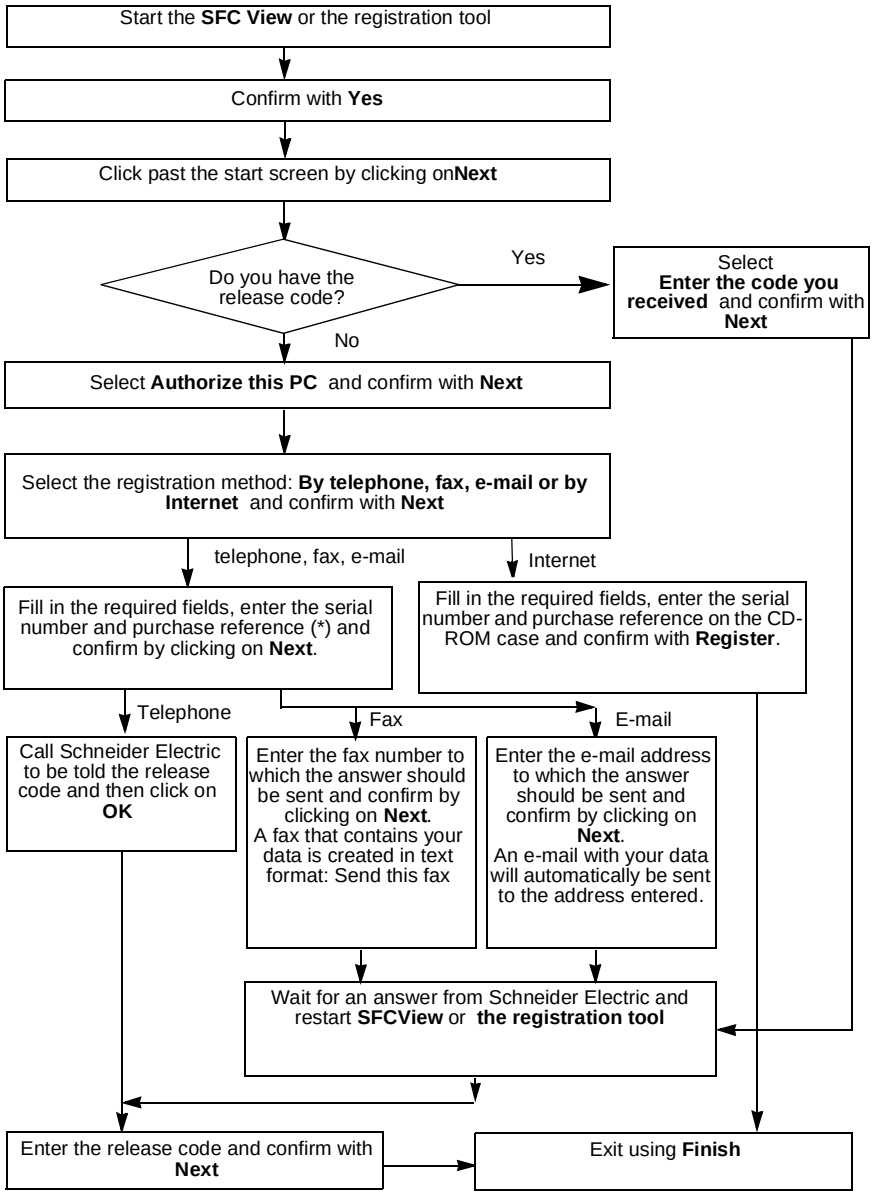
Registration with Schneider Electric

Permanent usage

To be able to use the software on a permanent basis, it must be registered with Schneider Electric.
The software must be registered within 21 days after installation.

Registering SFC View

These instructions give the individual steps for registering SFC View.



(*) These numbers are on the label stuck to the inside of the software CD-ROMs case.

Implementation of the SFC View ActiveX Controls

Registration on the PC

The SFC View ActiveX control is automatically registered on the PC when SFC View is installed.
This makes it available for the individual programming environments.

Implementation

ActiveX controls are implemented differently in every programming environment. The implementation procedure is described in the technical documentation for your programming environment.

Preliminary Settings

4

Introduction

Overview

This chapter contains information on the preliminary setting that are required to use Unity SFC View.

What's in this Chapter?

This chapter contains the following topics:

Unity Pro presettings

General

Before the SFC View can be used, Unity Pro must be configured in a certain way.

Note: These settings must be configured separately for every Unity Pro project.

Diagnostics

Make sure that under **Tools** → **Project settings...** → **Build** → **Diagnostics** the option **Application Diagnostics** is selected and the application level is set to **Local Diagnostics**.

Note: This setting is required if the SFC View diagnostics is carried out via the PLC diagnostics buffer.

Sequential Function Chart (SFC)

Make sure that under **Tools** → **Project settings...** → **Language extension** the **Allow multiple token** option is **not** selected.

Save project automatically

Make sure that under **Tools** → **Options** → **General** → **Save project automatically** the option **On Download** is selected.

OPC Factory Server Presettings

General

Before the SFC View can be used, certain settings must be made in the OFS configuration tool.

Starting OFS configuration tool

Execute the command **Start** → **Program** → **Schneider Electric** → **OFS** → **OFS configuration tool**.

Options

In the tab **Options** under **Features** the following option must be selected:

- Enable OPC Extensions

Note: In case the tab **Options** is not displayed, via the command button **Advanced...** all tabs can be shown.

Comm. settings

In the **Comm. tab** in **Options for devices without alias** the following option must be selected:

- Check Consistency

Note: If the **Comm.** tab is not shown, all of the tabs can be shown via the **Advanced...** tab.

Create alias

An alias is required to ensure that SFC View and OFS can work together. The following table describes the procedure to create a new alias.

Step	Action
1	Click the **Alias** tab.
2	Click the  (Create new alias) button.
3	Enter a new name.
4	Click on the **<Driver> field:<PLC addr>** and using the **Arrow down** open the page to enter the driver name.
5	From the ADDRESS tree structure on the left hand side select a network type.
6	Enter the network address and for the **PLC** select UNITY.
7	Confirm your entries using **OK**.
8	Click in the **Symbol table file** and **using the arrow pointing down** select the page to select the symbol file.
9	For the **file type select** `UnityPro project file(*.stu)` and the desired **file**.
10	Confirm your selection using **Open**.
11	Click on the button **Alias properties...** and check the option **Check Consistency** . Confirm the option is activated using **Apply**.
12	Now confirm all your entries for the new alias using **Apply**.

Loading the project

The following table describes the procedure to load a project.

Step	Action
1	Launch Unity Pro.
2	Open the desired `*.STU` file via **File** → **Open**.
3	Via **PLC** → **Connect...** create a connection to the PLC or PLC simulator controller.
4	Via **PLC** → **Transfer Project to PLC** load the program into the PLC or PLC simulator controller. **Note**: If you select the option **PLC RUN after transfer** , the PLC starts automatically after the program is transferred.

Test the connection between the OFS and the PLC

The following table describes the procedure for testing the connection between the OFS and the PLC.

Step	Action
1	Start the OFS client via **Start** → **Programs** → **Schneider Electric** → **OFS** → **OFS Test Client**. **Note:** The OFS Client program is only available if the appropriate option was selected for the OFS installation.
2	Click on **Schneider Aut.OFS** as server and confirm using **OK**. The diagnosis window for the OPC factory server is opened in the foreground.
3	Put the OFS Client window in the foreground again and create a new group via **Group** → **New Group...** and give it the **name:** e.g. GRP1.
4	Under **Notification** activate all the options and select the following settings: ● **Update rate:** 1000 ms ● **Dead banding:** 0.000000 [0.0,1.0] ● **OPC version:** Auto Confirm with **OK**.
5	Via **Item** → **New...** open the **AddItem**window.
6	In the tree structure in the left window select the desired project. The project variables are shown in the right hand window.
7	Click on one of the variables and confirm with **OK**.
8	If a green symbol appears to the left of the variable in the properties window which now appears, the connection between the OFS and the PLC is intact.

Diagnosis window for the OFS

Errors which might appear are shown in red in the OPC Factory Server diagnosis window.

Diagnosis window not visible

The installation of a HMI application (e.g. Vijeo Look) can make it so that the diagnosis window of the OPC Factory Server is not visible in the foreground and is displayed only as an icon in the footer line of the screen.
If this is the case, run the program `ChangeOFSSettings.exe` . This program is saved by default during the installation of SFC View under `...Installation directory of the SFCView...\` .
So that the change is effective, you must restart, via **Start** → **Programs** → **Schneider Electric** → **OFS** , the **OPC Factory Server** .

Starting and Using the SFC View Demo Application

5

Starting and using the SFC View Demo Application

General

When installing SFC View, a **SFC View Demo Application** and the corresponding source code (in Visual Basic) are installed.
The **SFC View Demo Application** is an independent program that can also be executed without HMI, SPS and Visual Basic.
Nearly all the properties, methods and events for SFC View were configured in this program, are available using menus, and can be tested in conjunction with Unity Pro, the OPC Factory Server (OFS), and the Unity Pro PLC Simulator.
The user can experiment with this demo application and learn how SFC View is used and programmed.

Source code

If the user's development environment is Visual Basic 6.0, the source code for the demo application can be viewed in it.
If a different development environment is used, the Visual Basic source code can be viewed in any editor.
The Visual Basic files can be found in the default directory: `...installation directory for SFCView...\SFCView\Example`.

Preliminary Settings

The demo application only works properly when all the *Preliminary Settings, p. 33*are correct.

Saving the Unity Pro test project

The Unity Pro test project that belongs to the demo application is saved during the installation of SFC View as an `*.XEF`-file by default under `...Installation directory of the SFCView...\TESTSFC`.

Settings for the conversion

Note: Before the Unity Pro test project can be converted, the SFC View library must be installed. Further information on this can be found in *Installation of the SFC View Library, p. 30*.

Converting the Unity Pro test project

The following table describes the procedure for converting the Unity Pro test project.

Step	Action
1	Launch Unity Pro.
2	Open the TESTSFCV.XEF file via **File** → **Open**.
3	Generate the project via **Build** → **Rebuild All Project**.
4	Save the project via **File** → **Save** using the name TESTSFCV.STU.

Loading the Unity Pro test project in the PLC simulator

The following table describes the procedure for loading the Unity Pro test project into the PLC simulator.

Step	Action
1	Launch Unity Pro.
2	Open the TESTSFCV.STU file via **File** → **Open**.
3	Via **PLC** → **Connect...** create a connection to the PLC simulator controller.
4	Via **PLC** → **Transfer Project to PLC** load the program into the PLC simulator controller. **Note**: If you select the **PLC RUN after sending** option, the PLC starts automatically after the program is sent.

OFS configuration tool

Before the SFC View can be used, certain settings must be made in the OFS configuration tool.
Execute the command **Start** → **Programs** → **Schneider Electric** → **OFS** → **OFS Configuration Tool** .

Comm. settings

In the **Comm. tab** in **Options for device without alias** the following option must be selected:

- Check Consistency

Note: If the **Comm.** tab is not shown, all of the tabs can be shown via the **Advanced...** tab.

Create Alias for demo application

An alias is required to ensure that SFC View and OFS can work together. The following table describes the procedure to create an alias for the demo application.

Step	Action
1	Click the **Alias** tab.
2	Click the  (Create new alias) button.
3	Enter the name e.g. TESTSFCV.
4	Click on the **<Driver> field:<PLC addr>** and using the **Arrow down** open the page to enter the driver name.
5	In the ADDRESS tree structure on the left click on **DIRECT** → **TCP IP**.
6	For the **TCP IP address** `enter 127.0.0.1` and for the **PLC, select** UNITY.
7	Confirm your entries using **OK**.
8	Click in the **Symbol table file** and **using the arrow pointing down** select the page to select the symbol file.
9	For the file type select **File type** `UnityPro project file (*.stu)` and for the file select **File** TESTSFCV.STU.
10	Confirm your selection using **Open**.
11	Click on the button **Alias properties...** and check the **Check Consistency** option. Confirm the option is activated using **Apply**.
12	Now confirm all your entries for the `TestSFCV` alias using **Apply**.

Test the connection between the OFS and the PLC

The following table describes the procedure for testing the connection between the OFS and the PLC.

Step	Action
1	Start the OFS client via **Start** → **Programs** → **Schneider Electric** → **OFS** → **OFS Test Client**. **Note:** The program **OFS Client** is only available if the appropriate option was selected for the OFS installation.
2	Click on **Schneider Aut.OFS** as server and confirm using **OK**.
3	Create a new group via **Group** → **New Group...** and give it the **name:** e.g. GRP1.
4	Under **Notification** activate all the options and select the following settings: • **Update rate:** 1000 ms • **Dead banding:** 0.000000 [0.0,1.0] • **OPC version:** Auto Confirm with **OK**.
5	Via **Item** → **New...** open the **AddItem**window.
6	In the tree structure in the left window of the project select **TestSFCV**. The project variables are shown in the right hand window.
7	Click on a variable e.g. `ACT1` and confirm with **OK**.
8	If a green symbol appears to the left of the variable in the properties window which now appears, the connection between the OFS and the PLC is intact.

Diagnosis window for the OFS

Errors which might appear are shown in red in the OPC Factory Server diagnosis window.

Starting the demo application

The following table describes the procedure for starting the demo application.

Step	Action
1	Execute the command **Start** → **Programs** → **Schneider Electric** → **Unity SFC View** → **SFC View Demo Application** .
2	Using the button select  the project `TestSFCV` so that it can be opened, and confirm it using **OK**.
3	The project data are loaded from the Unity Pro project.
4	Start the demo application via the demo application button  .
5	Select the appropriate view via one of the following command buttons: ●  Overview ●  Details ●  DetailsSimple

Using the demo application

Use the items **View** and **Run** in the main menu to activate and deactivate the individual properties, methods and events in SFC View and to observe the changes to the individual views.

SFCView appearance and behavior

6

Introduction

Overview

This chapter contains information on how to control the appearance and behavior of SFCView.
Further information on the control elements object, method, event, and constant can be found in the *Functional Reference, p. 87* chapter.

What's in this Chapter?

This chapter contains the following sections:

6.1 General controlling

Introduction

Overview

This chapter contains information on the general control of the appearance and behavior of SFCView.
Controlling is carried out using objects, properties, methods, events and constants.
All general controlling options are described in the following chapter.
All controlling options assigned to a particular view will be described in the appropriate chapter.

What's in this Section?

This section contains the following topics:

General description of the control elements

General

The appearance and behavior of the SFCView can be controlled using objects, properties, methods, events, and constants.

Object

An object, as used in object-oriented programming, is any data structure which has its own code.
An object could be a button that "knows" how it is pressed and what then happens.

Property

Properties are data that are assigned to an object. Properties are used to make information and settings available to an object.
Height and width, for example, can be properties of an info window.

Methods

A method is a subprogram that returns or does not return a value. In traditional programming a method is called a "procedure" or "function". In object oriented programming a method is assigned to a specific object and ensures access to the object data.
For example, a window for displaying error messages may have a method which is triggered by an error string. This opens the window and the error is shown.

Event

Events are triggered using input devices or when states or value change. E.g. events can be triggered by a mouse click or a signal from a timer.
In object oriented programming, events are used for communicating between objects, e.g. between a control and its container.

Constants

A constant is a parameter that cannot be changed. The value of a constant does not change while a program is running.
Constants can be used to define the value for a property. E.g. In a program for recording operating data, the number of manhours per shift might be defined. If this value is changed later it only needs to be changed in one place.

Objects for general controlling of SFCView

SFC View

View for information and controlling of step chains

Object	Description
`SFCView`	This ActiveX control enables the display of information about step chains, as well as the navigation through the chains and online control of the chains.

OFS info

Information on the OFS configuration

Object	Description
`OFSInfo`	Makes it possible to access the information on the OFS configuration using the `SFCView.GetOFSInfo` method.

OFS device collection

Information on the OFS device collection

Object	Description
`OFSDevices`	Makes it possible to access the information on the OFS device collection using the (`Count`, `Item`).

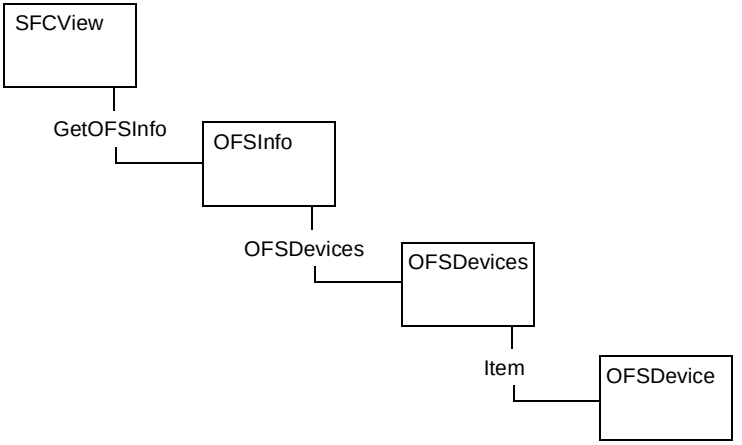
OFS devices

Information on the OFS devices

Object	Description
`OFSDevice`	Makes it possible to access the information on the OFS devices (`Alias`, `MaxChannel`, `Path`, `Symb`).

Object hierarchy

The object hierarchy is shown below:



Properties for general controlling of SFCView

View

The following property is available for the SFCView views.

Property	Description
`ViewMode`	Sets the SFCView view. The view that is shown is decided by the constants.(`SFCViewModes`constants , p. 53). ● 0 = Overview ● 1 = Details ● 2 = Details Simple

Project data

The following properties are available for loading project data.

Property	Description
`ProjectFile`	Sets the project file inlc. the project path.
`AutomaticProjectReload`	When this is set, the project data is automatically reloaded when changes are made in the PLC.
`UnityNetworkServer`	For a distributed configuration, this sets the PC where the Unity Pro-Program is located (PC name and IP address).

OFS communication

The following properties are available for communicating with the OPC Factory Server (OFS):

Property	Description
`UseOPCProject`	Specifies that the project file is automatically determined from the OFS path (`OPCAccessPath`). Alternatively the `ProjectFile` property can also be used.
`EnableOPCUpdates`	Specifies that changes in the PLC are reported from the OPC Factory Server to SFCView.
`OPCNetworkServer`	For a distributed configuration, this sets the PC where the OPC Factory Server is located (PC name and IP address).
`OPCAccessPath`	Sets the path for the OFS alias. The alias is defined in the OFS configuration tool (*OPC Factory Server Presettings, p. 34*).
`OPCConnect`	Creates the connection between the OPC factory server and the SFC View.
`OPCUpdateRate`	Sets the time delay for the OPC Factory Server to read data from the PLC [in msec].

OFS information

You can access the OFS information via the following properties

Property	Description
OPCNode	For a distributed configuration this sets the PC where the OPC Factory Server is located (PC name and IP address).
OFSDevices	Makes information regarding the OFS device collection available.

Device collection properties

Properties on the device collection defined in the OFS configuration tool

Property	Description
count	Is the number of devices in the OFS device collection.
Item	Is the index of a device in the OFS device collection.

Device properties

Properties for the devices defined in the OFS configuration tool

Property	Description
Symb	Is the name and path of a file with file ending information (e.g. *STU).
Alias	Is the alias name for a device in the OFS device collection.
MaxChannel	Is the maximum number of channels in a device.
Path	Is the device address.

Online control

The following property is available for online controlling.

Property	Description
ChainControlVariableName	Specifies the name of the variable configured as the input variable of the SFCVIEW_CTRL function block in the Unity Project .

Width

The following property specifies the width of the SFCView control.

Property	Description
Width	Specifies the width of the control

Font

The following property specifies the general fonts.

Property	Description
`Font`	Specifies the fonts for the text in the individual SFC View views. The following text fonts can be set seperately: ● Step name in the view: Details (`DetailsSimpleStepNameFont`) ● Step name in the view: Details Simple (`DetailsStepNameFont`)

Background color

The following property specifies the background color of the SFCView control.

Property	Description
`BackColor`	Background color in the views ● Details and ● Details Simple

Methods for general controlling of SFCView

Information on SFCView

Version information for SFCView.

Methods	Description
`About`	Shows the SFCView info window.

OFS configuration

Information on the OFS configuration

Methods	Description
`GetOFSInfo`	Makes it possible to access the information on the OFS configuration.

Loading project

Loading the Unity Pro project

Methods	Description
`ReloadProject`	Loads the Unity Pro project data into the SFCView.

Controlling the PLC

The following methods are available for controlling the PLC:

Methods	Description
`PLCDisableActions`	Activates/deactivates the actions for the selected string step.
`PLCDisableSection`	Activates/deactivates the selected string step.
`PLCDisableTimeCheck`	Activates/deactivates the time monitoring for the selected section.
`PLCDisableTransitions`	Activates/deactivates the transitions for the selected section.
`PLCGotoNextStep`	Activates the next step. It can be predefined as to whether the next transition must be carried out or not (`unconditional false/true`).
`PLCResetTimeErrors`	Resets the supervision time for the section.
`PLCSetInitializeFlag`	Sets the initialize flag or releases it.
`PLCClearChain`	Resets all active steps in the chain.

DANGER

Danger of unsafe, dangerous and destructive processes.

The "Initialize chain", "Reset chain", "Cancel analysis of transitions", "Cancel editing of actions and reset all actions of the chain", "Activate the next step independently of the transition condition" and "Activate the next step depending on the transition condition " functions should not be used to search for controller errors in machine tools, processes or material management systems, if they are running.

Failure to follow this precaution will result in death, serious injury, or equipment damage.

Events for general controlling of SFCView

Changing the display

Display change in SFCView.

Event	Description
`ViewModeChanged`	Is triggered when the SFCView view changes (Overview/ Details/ Details Simple).

Change in state of the chain

Change in state of the section.

Event	Description
`ChainStatusChanged`	Is triggered, if the step chain state changes. The `ChainStatusChanged` event can be made up of one or more `ChainStatusFlags`constants (`ChainStatusFlags`constants , p. 53constants).

Change in the PLC

Project change in the PLC.

Event	Description
`ProjectChanged`	Is triggered, if the project data changes in the PLC, e.g. after **Transfer Project to PLC**

Constants for general controlling of SFCView

ChainStatus-Flagsconstants

The ChainStatusChanged (*Change in state of the chain, p. 52*) event can be made up of one or more ChainStatusFlagsconstants.

Constants	Value	Description
CsfRunning	0	The chain is activated.
CsfSectionDisabled	1	The chain is deactivated.
CsfInitializeFlagSet	2	The chain initialize flag is set.
CsfTimeCheckDisabled	4	Time monitoring is deactivated.
CsfTransitionDisabled	8	The transitions are deactivated.
CsfActionsDisabled	16	The actions are deactivated.
CsfUnknown	4096	The chain state is unknown.

SFCViewModes-constants

The ViewMode (*View, p. 48*) property is set by the following constants:

Constants	Bit	Description
SfcOverview	0	The section is shown in the Overview view.
SfcDetails	1	The section is shown in the Details view.
SfcDetailsSimple	2	The section is shown in the Details Simple view.

6.2 View: Overview

Introduction

Overview

This chapter contains general information about the view: Overview, that displays the SFCView.

What's in this Section?

This section contains the following topics:

General description

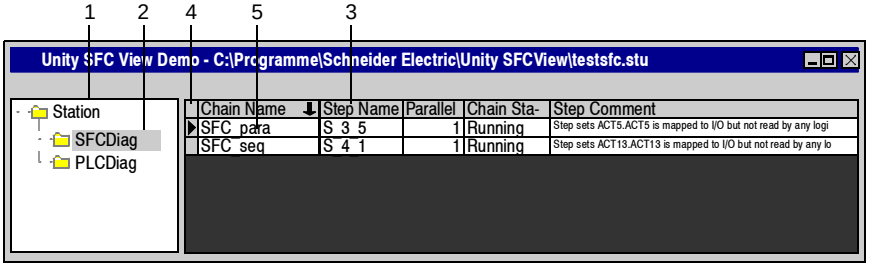
Brief description

The view: Overview, the SFCView offers the following options:

- Selecting a Unity Pro project
- Navigate through the project via the function view
- Listing all chains of a Unity Pro project
- Display of real-time information on the status of the step chains
- Selecting one or more step chains of the project
- Switching to a different view (Details or Details Simple) for a step chain

Display

View display: Overview.



View components Overview

Number	Components	Description
1	Functional view	After a project has been loaded directly or via the OFS, the functional view for the Unity Pro project is shown. It is possible to navigate through the project via this functional view.
2	Functional project units	If a functional unit was selected, all the step chains for it are shown in the table on the right. **Note:** All chains that are not assigned to a functional unit are shown in the main directory.
3	Information on the step chains	• Step chain name • Name of the active step for each chain • Number of parallel/alternative steps • Chain state • Step / chain comment
4A	Selecting a step chain	Click on the cell in the table on the right. In the first column, an arrow is shown.

Number	Components	Description
4B	Selecting more than one step chain	Click in a line in the right hand table and confirm by pressing the **spacebar**. Instead of an arrow in the first column, a cross is shown. Using this method you can select more than one step chain and for all chains at once, e.g. `PLCSetInitializeFlag` apply the method. **Note:** If more than one step chain is marked by a cross, the method to be implemented acts on them. A step chain that is marked with only one arrow is considered in this case to be **not** selected.
5	Switching to another view	You can switch to a different view (Details or Details Simple) by double-clicking on a step chain.

Controlling the controls

To control the behavior and the appearance of the SFCView, the person carrying out the configuration has various methods, events, constants and properties available for use.
All the controlling options which are assigned to the Overview display are described in the following chapter.
All general controlling options are described in their own chapter *General controlling, p. 45*.

Properties for controlling the view: Overview

Statistics window

Display of the statistics window:

Property	Description
`ShowStatistic`	This property can be used to open the statistics window.

View structure tree

`ShowChainGroups` property

Property	Description
`ShowChainGroups`	Specifies whether the function view is shown which can be used to navigate through the project. If the property is set to `false` the function view is not displayed and all the step chains are shown in a table.

Step or chain comment

`ShowStepComments`property

Property	Description
`ShowStepComments`	This sets whether step or chain comments are shown.

Step chain view

The `OnChainOpen` property specifies the view that is shown when opening a step chain.

Property	Description
`OnChainOpen`	Specifies the behavior when opening a step chain. The behavior is set using constants. (`OnChainOpen`constants , p. 59). ● 0 = NoAction ● 1 = ShowDetails ● 2 = ShowDetailsSimple

Representation of the chain information

Column headers

Property	Description
`ColumnHdrChainName`	ChainName column header
`ColumnHdrStepName`	StepName column header
`ColumnHdrParallelSteps`	ParallelSteps column header
`ColumnHdrChainStatus`	ChainStatus column header
`ColumnHdrChainComment`	ChainComment column header
`ColumnHdrStepComment`	StepComment column header The `ShowStepComments` property is used to decide if the step or the chain comments are shown.

Note: The SFCView texts are stored in English by default but can be changed by the person carrying out the configuration.

`ChainStatus` column contents

Tests in the `ChainStatus`column. These texts are also shown in the: Details Simple view.

Property	Description
`OverviewTextRunning`	Specifies the text in the `ChainStatus` column if the chain status flag is set to `CsfRunning`.
`OverviewTextSectionDisabled`	Specifies the text in the `ChainStatus` column if the chain status flag is set to `CsfSectionDisabled`.
`OverviewTextSetInitializeFlag`	Specifies the text in the `ChainStatus` column if the chain status flag is set to `CsfInitializeFlagSet`.
`OverviewTextDisableTimeCheck`	Specifies the text in the `ChainStatus` column if the chain status flag is set to `CsfTimeCheckDisabled`.
`OverviewTextDisableTransitions`	Specifies the text in the `ChainStatus` column if the chain status flag is set to `CsfTransitionDisabled`.
`OverviewTextDisableActions`	Specifies the text in the `ChainStatus` column if the chain status flag is set to `CsfActionsDisabled`.

Note: The SFCView texts are stored in English by default but can be changed by the person carrying out the configuration.

Events for controlling the view: Overview

Open step string

Opening a step string.

Event	Description
`ChainOpenEvent`	Triggered by: • double-clicking on a line in the step string table • by pressing the enter key when a step string has focus in the table

Select step string

Selecting a step string.

Event	Description
`ChainSelectEvent`	Is triggered when a new row in the table step string has been selected.

Constants for controlling the view: Overview

`OnChainOpen`-constants

The `OnChainOpen` (*Step chain view, p. 57*) property is set using the following constants:

Constants	Value	Description
`NoAction`	0	The input is ignored.
`ShowDetails`	1	The step string is shown in the details overview.
`ShowDetailSimple`	2	The step string is shown in the details simple overview.

6.3 View: Details

Introduction

Overview

This chapter contains general information about the view: Details, which displays the SFCView.

What's in this Section?

This section contains the following topics:

General description

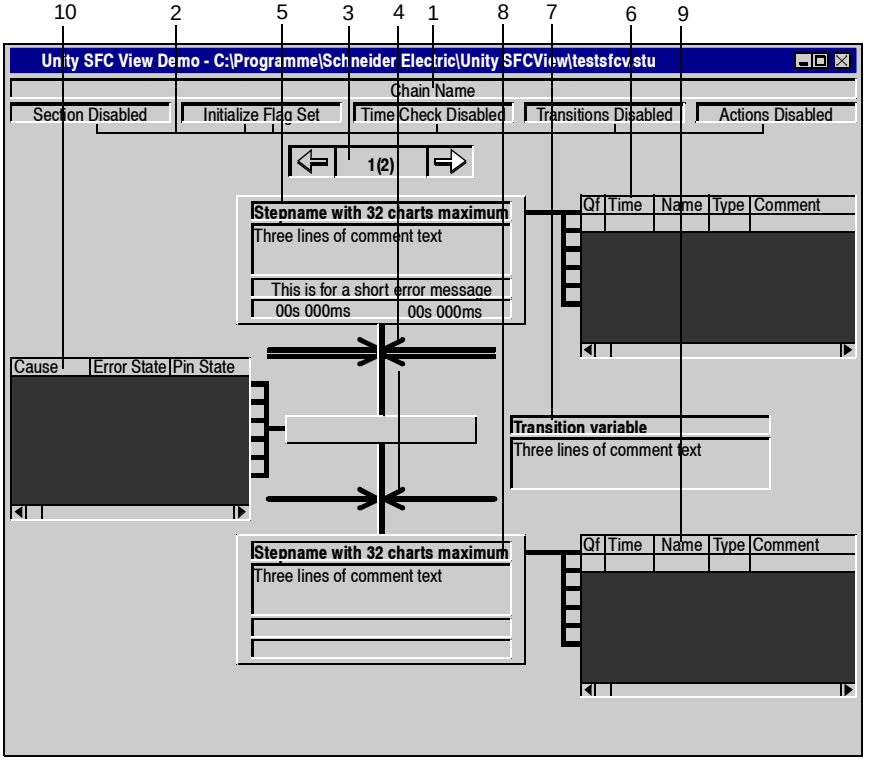
Brief description

The view: Details, the SFCView offers the following options:

- Display of real-time information on the status of a section
- Display of two steps at a time and the associated transition
- Shows the actions for the steps
- Shows the diagnostic information
- Shows state of variables, errors etc. using various colors
- Navigation between active steps for parallel branches
- Navigation between the sections via methods, e.g. `DisplayNextStep`

Display

View display: Details.



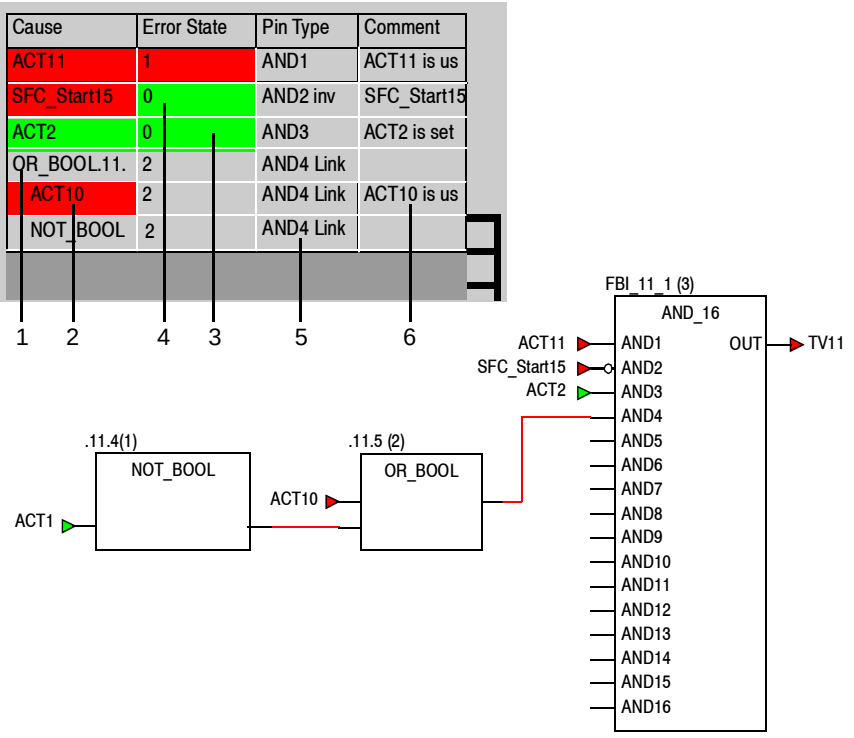
View components Details

Number	Components	Description
1	Section name	Name of the selected section

Number	Components	Description
2	Status flags	Shows the state of section. The state is represented using a color outline.
3	Number of active steps and navigation	Display of the number of active steps. The arrow keys can be used to navigate between the active parallel steps.
4	Parallel/ alternative branches	Display of parallel (double line) or. alternative branches (single line)
5	Information on the active steps	● Active step name ● Comment (three lines) ● Short Error message ● Delay Time ● Current dwell time The outline around this info field can have a different color depending on the state of the step. On the right hand side the actions for the step are displayed.
6	Information on the actions for the active step	● Identifier ● Time ● Name of variables or the section ● (**v**ariable or **s**ection type) ● Comment The background color for the variables can have a different color depending on the state of the variables.
7	Information on transitions	● Name of transition variables ● Comment (three lines) The rectangle representing the transition can have a different color depending on the state of the transition.
8	Information on the following step	● Next step name ● Comment (three lines) ● Short Error message: ● Delay Time ● Current dwell time On the right hand side the actions for the step are displayed.
9	Information on the actions for the next step	● Identifier ● Time ● Name of variables or the section ● (**v**ariable or **s**ection type) ● Comment
10	Diagnostic information	Display of error messages for the transition. The user can set individually which diagnostic information is shown.

Diagnostic information

Then, for example, a field with diagnostic information and its associated FBD section from a Unity Pro project are used to give an initial overview.
The appearance of the field with the diagnostic information can be controlled using, for example, the properties `ValueOffColor`, `ShowAllTransitionInputs`, `ShowBlockNames`, `ContentErrorStateColumn`, `StateErrorStateColumn` etc.
More detailed information on the properties, methods, events and constants which are available for controlling the diagnostic information display can be found in the following chapters.
Example: Diagnostic information and its FBD section



Description of the example

Number	Components	Description
1	Text in the column `Cause`	Listing of all the input variables and the nested logic of the AND_16 function block. Input variables for nested function blocks are indented, e.g. `ACT10` for function block `OR_BOOL`.

Number	Components	Description
2	Color in the column `Cause`	Green: Input has the value 1, e.g. `ACT2` Red: Input has the value 0, e.g. `ACT11` Gray: Input with an unknown value
3	Color in the column `Error State`	The output `TV11` has the value 0. The color in the `Error State` column shows if the input in question is the cause. Green: Not the reason for the value 0 at the output `TV11` e.g. `ACT2` **Note:** The`SFC_Start15` input may indeed have the value 0, but it is inverted and is therefore not the reason for the value 0 at the `TV11` output. For this reason it is shown in green. red: The reason for the value 0 on the output `TV11` e.g. `ACT11` gray: Input, whose logic cannot be analyzed e.g. the `OR_BOOL` function block as an input for `OR_BOOL` is connected to a `NOT_BOOL`. **Note:** SFCView cannot analyse function blocks which are not `AND_BOOL`or `OR_BOOL` function blocks.
4	Text in the column `Error State`	In this case the text matches the colors in the `Error State`column. 0: Not the cause 1: Cause 2: Not analyzable
5	Text in the `Pin Type` column	Examples: `AND1`: Variable at input `AND2 inv`: Variable at input, inverted `AND4 Link`: Nested logic at the input
6	Text in the `Comment` column	Comment on the variables from the Unity Pro project.

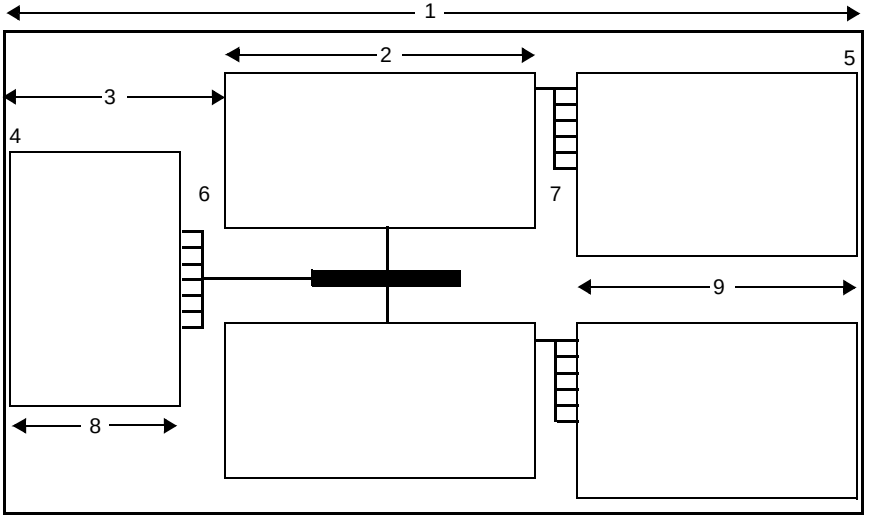
Controlling the controls

To control the behavior and the appearance of the SFCView, the person carrying out the configuration has various methods, events, constants and properties available for use.
All the controlling options which are assigned to the details display are described in the following chapter.
All general controlling options are described in their own chapter *General controlling, p. 45*.

Properties for controlling the view: Details

Info field sizes

Schematic display of the individual info fields



Control of the size of the fields:

No.	Property	Description
1	`Width`	Total width of the SFCView control
2	`DetailsStepsWidth`	The width of the step info fields for the active and next step
3	`DetailsStepsLeft`	The distance for the step info field from the left edge of the SFCView control
4	-	The left edge of the diagnostic info field is a fixed distance from the left edge of the control.
5	-	The right edge of the action info field is a fixed distance from the right edge of the control.
6	-	The distance between the diagnostic info field and the step info field is fixed.
7	-	The distance between the step info field and the action info field is fixed.
8	-	The width of the diagnostic info field is calculated from no. 3 and no. 6.
9	-	The width of the action info field is calculated from no. 1 ,no. 2, no. 3 and no. 7.

Direct view of a chain

Direct view of a chain in the views: Details and Details Simple.

Property	Description
`ChainName`	Via this property a chain can be displayed directly in the Details or Details Simple views (without going through the Overview view).

Always show active steps

Active step display

Property	Description
`ShowActiveStep`	If this property is set to `True` the active step is always shown in the Details and Details Simple views. In this case navigation is only possible between parallel **active** steps. If this property is set to `False` various methods, e.g. `DisplayNextStep` can be used to navigate through the section.

Representation of the status flags

Text display for the status flags

Property	Description
`DetailsTextSectionDisabled`	Text that is displayed for the status flag if the section is deactivated.
`DetailsTextSetInitializeFlag`	Text that is displayed for the status flag if the initialize flag is set.
`DetailsTextDisableTimeCheck`	Text that is displayed for the status flag if the time check is deactivated.
`DetailsTextDisabledTransitions`	Text that is displayed for the status flag if the transitions are deactivated.
`DetailsTextDisabledActions`	Text that is displayed for the status flag if the actions are deactivated.

Note: The SFCView texts are stored in English by default but can be changed by the person carrying out the configuration.

Color display for the status flags

Property	Description
`ChainFlagsOffBackColor`	Color of background display for status flag, if the status flags is deactivated.
`ChainFlagsOffForeColor`	Color of text for status flag if the status flag is deactivated.
`ChainFlagsOnBackColor`	Color of background for status flag if the status flag is activated.
`ChainFlagsOnForeColor`	Color of text for status flag if the status flag is activated.

Variables display

Color display for variables (actions, transitions and diagnostic variables)

Property	Description
`ValueOffBackColor`	The background color for a variable in its off state.
`ValueOffForeColor`	The text color for a variable in its off state.
`ValueOnBackColor`	The background color for a variable in its on state.
`ValueOnForeColor`	The text color for a variable in its on state.

Lines display

Connection lines color

Property	Description
`DetailsViewLinesColor`	Color of the lines between step information and action information or between transitions and diagnostic information.

General settings for diagnostics

Diagnostic mode

Property	Description
`UsePLCDiagSystem`	If this property is `true` the diagnosis is carried out via the PLC diagnosis buffer. If this property is `false` the diagnosis is done via special SFCView diagnosis function blocks in the Unity Pro project.

Note: Further information on diagnostics can be found in the *Prerequisites for the diagnosis, p. 22*chapter.

Reloading the project when the diagnostic mode is changed.

Property	Description
`UseEasyModeSwitch`	If this property is set to `false` (default), if the `UsePLCDiagSystem` property is changed, the project data is reloaded into SFCView. If this property is set to `true`, if the `UsePLCDiagSystem` property is changed, the project data is **not** reloaded into SFCView. **Note**: Setting this property to `true` makes switching between diagnostic modes quicker but slows down the loading of the project and uses more memory. **Note**: Therefore, for applications in which only one diagnostic mode is used, it is advisable to set this property to `false`.

Settings for diagnostics via the PLC diagnostics buffer

Automatic update of error display

Property	Description
`DiagAutoRetrigger`	If this property is set to `true` the PLC diagnostic buffer reanalyzes the cause of a transition error again in a cyclical manner. The cycle time is set via the `DiagAutoRetriggerInterval` property. If this property is set to `false` the causes of the errors are only shown for the time at which the monitoring time was exceeded and are not updated afterwards. This property is only available if `UsePLCDiagSystem` is set to `true`. **Note:** If the diagnostic is configured via SFCView function blocks, the error display is updated automatically for any change.
`DiagAutoRetriggerInterval`	Sets the cycle time (1 to 65535 msec), for cyclically reanalyzing the causes of errors (presetting 1000 msec).

Diagnosis information (via the PLC diagnostics buffer)

Display errors for all active parallel steps.

Property	Description
`ShowAllDiagErrors`	Specifies that the errors for all active parallel steps are displayed. If this property is set to `false`only the errors for the step currently being displayed are shown. This property is only available if `UsePLCDiagSystem` is set to `true`.

Diagnosis information (via the SFCView function blocks)

Display all inputs for the transition.

Property	Description
`ShowAllTransitionsInputs`	Specifies that all inputs for all active transitions are displayed. If this property is set to `false` , only the inputs for the transition currently being displayed are shown. This property is only available if `UsePLCDiagSystem` is set to `false`.

Show the names of the function blocks.

Property	Description
`ShowBlockNames`	Specifies that the names of all the function blocks which are nested in the SFCView blocks are also shown. If this property is set to `false` , only the variable names are shown. This property is only available if `UsePLCDiagSystem` is set to `false`.

Define the `ErrorState` column contents.

Property	Description
`ContentErrorStateColumn`	Defines the contents of the `ErrorState` column where diagnostic information for the transition is being displayed. The content to be displayed is decided by the constants.(`ContentErrorStateColumn`constants , p. 75). ● 0 = CecEmpty ● 1 = CecErrorState ● 2 = CecVariableName ● 3 = CecPinType

Define the colors in the `ErrorState` column.

Property	Description
`StateErrorStateColumn`	Defines the colors in the `ErrorState` column in the field where diagnostic information for the transition is displayed. The color is decided by the constants (`StateErrorStateColumn`constants , p. 75). ● 0 = secsNoState ● 1 = secsErrorState ● 2 = secsPinState ● 3 = secsVariableState

Representation of the diagnostics information

Number of lines

Property	Description
`NumberErrorGridLines`	Number of lines in the diagnostics table

Column widths

Property	Description
`WidthErrorStepNameColumn`	Width of the ErrorStep column
`WidthErrorVariableColumn`	Width of the ErrorVariable column
`WidthErrorPinTypeColumn`	Width of the ErrorPinType column (only via SFCView blocks for diagnostics)
`WidthErrorStateColumn`	Width of the ErrorState column (only via SFCView blocks for diagnostics)
`WidthErrorCommentColumn`	Width of the ErrorComment column

If `UsePLCDiagSystem` is set to `true` the ErrorState and ErrorPinType widths are 0 (Zero).
If `ShowAllTransitionInputs` is set to `false` the ErrorStep column width is 0 (Zero).

Note: To make a column invisible the width must be set to 0 (Zero).

Texts

Property	Description
ColumnHdrErrorStep	ErrorStep column header
ColumnHdrErrorVariable	ErrorVariable column header
ColumnHdrErrorPinType	Header for the ErrorPinType column (only via SFCView blocks for diagnostics)
ColumnHdrErrorState	Header for the ErrorState column (only via SFCView blocks for diagnostics)
ColumnHdrErrorComment	ErrorComment column header

Note: The SFCView texts are stored in English by default but can be changed by the person carrying out the configuration.

Representation of the action information

Column widths

Property	Description
WidthActionQualifierColumn	Width of the ActionQualifier column
WidthActionTimeColumn	Width of the ActionTime column
WidthActionNameColumn	Width of the ActionName column
WidthActionTypeColumn	Width of the ActionType column
WidthActionCommentColumn	Width of the ActionComment column

Note: To make a column invisible the width must be set to 0 (zero).

Texts

Property	Description
ColumnHdrActionQualifier	ActionQualifier column header
ColumnHdrActionTime	ActionTime column header
ColumnHdrActionName	ActionName column header
ColumnHdrActionType	ActionType column header
ColumnHdrActionComment	ActionComment column header

Note: The SFCView texts are stored in English by default but can be changed by the person carrying out the configuration.

Representation of the step information

Font for the step name:

Property	Description
DetailsStepNameFont	Font for the step name:

Note: The font that is set using the DetailsStepNameFont property is also used for chain names and the transition names. The font for all the other texts is set using the Font property.

Texts

Property	Description
StepMaxTimeErr	Text that is shown in the step information field, if the maximum monitoring time is exceeded.
StepMinTimeErr	Text that is shown in the step information field, if the minimum monitoring time has not been reached.

Note: The SFCView texts are stored in English by default but can be changed by the person carrying out the configuration.

Unit and color for the monitoring time field

Property	Description
ShowTimeInms	If this property is set to true, the monitoring time resolution is milliseconds. If it is false , the resolution is in seconds.
StepMaxTimeErrBackColor	Background color of the step information field that indicates that the maximum monitoring time has been exceeded.
StepMaxTimeErrForeColor	Text color of the step information field that indicates that the maximum monitoring time has been exceeded.
StepMinTimeErrBackColor	Background color of the step information field that indicates that the minimum monitoring time has not been reached.
StepMinTimeErrForeColor	Text color of the step information field that indicates that the minimum monitoring time has not been reached.

Color for special steps

Property	Description
DetailsActiveStepBackColor	Color for the edge of the info field for the active step
DetailsInactiveStepBackColor	Color for the edge of the info field for the inactive step
DetailsInitialStepBackColor	Color indicator for the initial step
DetailsMacroStepBackColor	Color indicator for a macro step.

Methods for controlling the view: Details

Functions for diagnostics via the PLC diagnostics buffer

The following methods are available for controlling the PLC diagnostics functions:

Methods	Description
DiagResetErrorBuffer	Resets all diagnosis errors in the PLC.
DiagRetrigger	Triggers another analysis of the diagnosis error in the section which is currently being shown in the details view.

Note: Further information on diagnostics*Prerequisites for the diagnosis, p. 22*.

Control of the steps display and transitions

The following methods are available for controlling the steps and transitions display:

Methods	Description
DisplayInitialStep	Shows the first step of the step chain.
DisplayNextActiveStep	Shows the next active step.
DisplayNextAltTran	Shows the next alternative transition.
DisplayNextParStep	Shows the next parallel step
DisplayNextStep	Displays the next step.
DisplayPreviousStep	Displays the previous step.
DisplayPrevActiveStep	Shows the previous active step
DisplayPrevAltTran	Shows the previous alternative transition.
DisplayPrevParStep	Shows the previous parallel step

Events for controlling the view: Details

Action variable

Information on action variables

Event	Description
`ActionVarSelect`	Is triggered by double-clicking on a line in the window where an action for the steps is being displayed. The name of the action variable and the name of the step chain are given as parameters.

Error variable

Information on error variables

Event	Description
`DiagVarSelect`	Is triggered by double-clicking on a line in the window where diagnostic information for the transition is being displayed. The name of the error variable and the name of the step chain are given as parameters.

Constants for controlling the view: Details

ContentErrorStateColumn constants

The property `ContentErrorStateColumn` (*Diagnosis information (via the SFCView function blocks), p. 69*) is defined by the following constants:

Constants	Value	Description
`CecEmpty`	0	The `Error State`column is empty.
`CecErrorState`	1	The error status for the input variables is shown.
`CecVariableName`	2	The name for the input variables is shown.
`CecPinType`	3	The Pin type is shown (`AND` or `OR`)

StateErrorStateColumn-constants

The property `StateErrorStateColumn` (*Diagnosis information (via the SFCView function blocks), p. 69*) is defined by the following constants:

Constants	Value	Description
`sescNoState`	0	The background is gray.
`sescErrorState`	1	The background color shows if the input variable is erroneous (red/green).
`sescPinState`	2	The background color shows the state of the pin which is directly connected to the variable. If the pin is **not** inverted, the pin state is the same as the variable state. If the pin is inverted, the pin state is the different from the variable state.
`secsVariableState`	3	The background color shows the state of the variables.

Note: The `StateErrorStateColumn` property is only available if the diagnosis is made via a special SFCView function block.

6.4 View: Details Simple

Introduction

Overview

This chapter contains general information about the view: Details Simple, which shows the SFC View.

What's in this Section?

This section contains the following topics:

General description

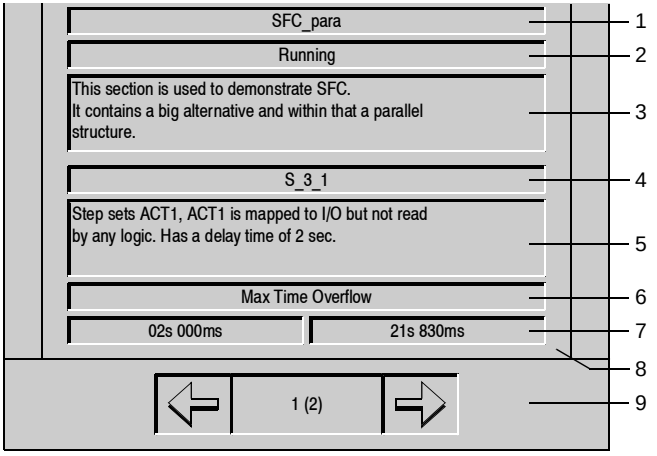
Brief description

The view: Details Simple of the SFCView offers the following options:

- Display of real-time information on the status of the active or selected step in a step chain
- Navigation between active steps for parallel branches
- Navigation between the step chains via methods, e.g. `DisplayNextStep`

Display

View display: Details Simple.



View components Details Simple

Number	Components	Description
1	Chain name	Name of the selected chain
2	Chain state	Shows the state of the step chain. The state is represented by a color outline.
3	Chain comment	Three line comment text for step chain
4	Step name	Name of step shown
5	Step comment	Three line comment text for shown step
6	Short Error message	Single line error message for the step shown
7	Time monitoring	View of monitoring and delay time
8	Frames	The outline around this info field can have a different color depending on the state of the step.
9	Number of active steps and navigation	Display of the number of active steps. The arrow keys can be used to navigate between the active parallel steps.

Controlling the controls

To control the behavior and the appearance of the SFCView, the person carrying out the configuration has various methods, events, constants and properties available for use.
All the controlling options which are assigned to the Details Simple display are described in the following chapter.
All general controlling options are described in their own chapter *General controlling, p. 45*.

Properties for controlling the view: Details Simple

Direct view of a chain

Direct view of a chain in the views: Details and Details Simple.

Property	Description
`ChainName`	Via this property a chain can be displayed directly in the Details or Details Simple views (without going through the Overview view).

Always show active steps

Active step display

Property	Description
`ShowActiveStep`	If this property is set to `True` the active step is always shown in the Details and Details Simple views. In this case navigation is only possible between parallel **active** steps. If this property is set to `False` various methods, e.g. `DisplayNextStep` can be used to navigate through the step chain.

View display: Details Simple

Summary of the view: Details Simple

Property	Description
`DetailsSimpleShow ChainName`	Specifies that the line is shown with the chain name.
`DetailsSimpleShow ChainStatus`	Specifies that the line is shown with the chain state.
`DetailsSimpleShow ChainComment`	Specifies that the line is shown with the chain comment.
`DetailsSimpleShow StepComment`	Specifies that the line is shown with the step comment.
`DetailsSimpleShow StepErrorLabel`	Specifies that the line is shown with the error message.

Property	Description
DetailsSimpleShow InitialStepIndicator	Specifies that a bar on the left and right of the field indicates that the field is an initial step.
DetailsSimpleShow Navigation	When this is set the buttons for navigating between the active steps for parallel branches are displayed.

Note: If the line with the state of the chain or the line with the error message is not shown, then the state of the chain or an error in the chain is shown by a color outline around the line with the chain name. The color outline is set in the ChainFlagsOffBackColor/ChainFlagsOnBackColor or in the StepMaxTimeErrBackColor/StepMinTimeErrBackColorproperties.

Step name font in the view: Details Simple

Property	Description
DetailsSimpleStepNameFont	Font for the step name:

Note: The font set using the DetailsSimpleStepNameFont property is also used for chain names. The font for all the other texts is set using the Font property.

Text for displaying the step chain state. These texts are also shown in the Overview view in the column ChainStatus.

Property	Description
OverviewTextRunning	Sets the text shown if the chain status flag is set to CsfRunning.
OverviewTextSectionDisabled	Sets the text shown if the chain status flag is set to CsfSectionDisabled.
OverviewTextSetInitializeFlag	Sets the text shown if the chain status flag is set to CsfInitializeFlagSet.
OverviewTextDisableTimeCheck	Sets the text shown if the chain status flag is set to CsfTimeCheckDisabled.
OverviewTextDisableTransitions	Sets the text shown if the chain status flag is set to CsfTransitionDisabled.
OverviewTextDisableActions	Sets the text shown if the chain status flag is set to CsfActionsDisabled.

Texts for the monitoring time

Property	Description
`StepMaxTimeErr`	Text shown in the step information field, if the maximum monitoring time is exceeded.
`StepMinTimeErr`	Text that is shown in the step information field, if the minimum monitoring time has not been reached.

Note: The SFCView texts are stored in English by default but can be changed by the person carrying out the configuration.

Unit and color for the monitoring time field

Property	Description
`ShowTimeInms`	If this property is set to `true`, the monitoring time resolution is milliseconds. If it is `false` , the resolution is in seconds.
`StepMaxTimeErrBackColor`	Background color of the step information field that indicates that the maximum monitoring time has been exceeded.
`StepMaxTimeErrForeColor`	Text color of the step information field that indicates that the maximum monitoring time has been exceeded.
`StepMinTimeErrBackColor`	Background color of the step information field that indicates that the minimum monitoring time has not been reached.
`StepMinTimeErrForeColor`	Text color of the step information field that indicates that the minimum monitoring time has not been reached.

Color for special steps

Property	Description
`DetailsActiveStepBackColor`	Color for the edge of the info field for the active step
`DetailsInactiveStepBackColor`	Color for the edge of the info field for the inactive step
`DetailsInitialStepBackColor`	Color indicator for the initial step
`DetailsWaitingStepBackColor`	Color indicator for a step in the waiting state **Note:** Any step in the waiting state is either an initial step or the last step displayed, if the SetInitialize flag is set.

Methods for controlling the view: Details Simple

Controlling the display for the steps

The following methods are available for controlling the display of steps:

Methods	Description
`DisplayInitialStep`	Shows the first step of the step chain.
`DisplayNextActiveStep`	Shows the next active step (for parallel branches).
`DisplayNextStep`	Displays the next step.
`DisplayPreviousStep`	Displays the previous step.
`DisplayPrevActiveStep`	Shows the previous active step (for parallel branches).

Events for controlling the view: Details Simple

Double-click

Double-click in the view: Details Simple.

Event	Description
`DetailsSimpleDblClick`	The event is deleted by double-clicking in the view: Details Simple.

Tips and Tricks

7

Introduction

Overview

This chapter contains tips for improving the performance of your system and tricks that have arisen from previous versions of SFCView.

What's in this Chapter?

This chapter contains the following topics:

Reading data and instantiating groups

Reading the Unity Pro project	When SFCView is first started, all of the SFC sections in the Unity Pro project are read. If the SFC View internal diagnostic was selected via the diagnosis block, all of the FBD sections will be read also.
Instantiating groups	OPC groups are instantiated during the first call. In the display: Overview, all the groups in the active function view are instantiated. In the display: Details, only the groups that belong to the respective step chain are instantiated.

System performance

Loading Unity Pro projects

Preference should be given to the following procedures in order to speed up the loading time for Unity Pro projects:

- Set the `ViewMode` property (using the `OPCAccessPath` property) before you load the project.
- If you are using the view: Details or Details Simple, set the `ChainName` property (using the `OPCAccessPath` property) before you load the project.
- Only load the project once the SFCView window has opened.
- Use the `UseEasySwitchMode` property only if you really need it, otherwise SFCView always has to load the data for both diagnostics modes.

Communication in the system

Preference should be given to the following procedures in order to speed up the communications in the system:

- In the OFS configuration tool, increase the **MaxChannels** option. Use the maximum value allowed by the application.
- Set the `ViewMode` property before creating a link to the PLC. SFC View only instantiates the OPC groups that are required in the desired view.
- If you are using the view: Details or Details Simple, set the `ChainName` property before you create a link to the PLC. SFC View only instantiates the OPC groups that are required for the desired section.
- Set the `EnableOPCUpdates` property, for the time in which the SFCView window remains in the background, to `false`. This will reduce the data traffic for this duration.

Resetting SFCView

To revert to the SFC View Controls presettings, you must delete it from the ActiveX container and then insert it again.

mfc42.dll

SFC View was developed with Microsoft Visual Studio 6.0 (SP5). Therefore, the control requires mfc42.dll, version 6.0 or higher.
If the ActiveX container used by you has a lower DLL version, it can create conflicts between SFC View and the ActiveX container.
Therefore you must ensure that your ActiveX container works correctly with mfc42.dll, version 6.0 or higher.

Functional Reference

II

At a glance

Introduction

In this part you will find information about objects, properties, methods, events and constants.

What's in this Part?

This part contains the following chapters:

Objects

8

At a glance

Introduction

In this chapter you will find information about the objects.

What's in this Chapter?

This chapter contains the following topics:

OFSDevice Object

Brief description An OFSDevice object contains OFS' configuration data for a device.

Syntax The OFSDevice object syntax has these parts:

Part	Description
`object`	An object expression that evaluates to an OFSInfo object.
`index`	Either an integer or string that uniquely identifies a member of an OFSDevices collection.

Remarks The unique string to access an element of the OFSDevices collection is the value of the OFSDevice object's Alias property.
See the Visual Basic documentation for more information about collections.

OFSDevices Collection

Brief description An OFSDevices collection is a collection of OFSDevice objects.

Syntax

```
object.OFSDevices
object.OFSDevices (index)
```

The syntax lines above refer to the collection and to individual elements in the collection, respectively, according to the standard collection syntax.
The OFSDevices collection syntax has these parts:

Part	Description
`object`	An object expression that evaluates to an OFSInfo object.
`index`	Either an integer or string that uniquely identifies a member of an OFSDevices collection.

Remarks The unique string to access an element of the OFSDevices collection is the value of the OFSDevice object's Alias property.
See the Visual Basic documentation for more information about collections.

OFSInfo Object

Brief description Makes data of the currently configured OFS devices available.

Syntax `OFSInfo`

Remarks When the object is created by a call to the GetOFSInfo method of the SFC View control, the OFS configuration data currently available in the system registry is read.

SFC View Control

Brief description A SFC View control is a graphical control to display the contents and the current state of all SFC sections of a Concept project in a PLC.

Syntax `SFCView`

Remarks See the chapter Overview for an overview of the main concepts of the control or look for the details in the other chapters of this reference.

Properties

9

At a glance

Introduction

In this chapter you will find information about the properties.

What's in this Chapter?

This chapter contains the following sections:

9.1 ChainXxx Properties

At a glance

Introduction

In this section you will find an overview of the ChainXxx properties.

What's in this Section?

This section contains the following topics:

ChainName Property

Brief description Returns/Sets string value, which represents chain name displayed in detail view.

Syntax `object.ChainName [= string]`

The ChainName property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`string`	A string expression representing chain name, which is displayed in detail view.

Remarks If ViewMode property is set to sfcDetails, ChainName property allows to display directly detail view of particular chain without going back to Overview.

ChainControlVariableName Property

Brief description Returns/Sets string value, which represents variable name used as input variable of the `SFCVIEW_CTRL` in the Unity Pro project.

Syntax `object.ChainControlVariableName [= string]`

The ChainControlVariableName property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`string`	A string expression representing variable name used as input variable of the `SFCVIEW_CTRL` in the Unity Pro project. Default is `SVC_Command`.

ChainFlagsOffBackColor Property

Brief description Returns/Sets the background color to display the 'Off' state of the chain status flags.

Syntax `object.ChainFlagsOffBackColor [= color]`

The ChainFlagsOffBackColor property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`color`	A value or constant that determines the background or foreground colors to display the `On' and `Off' states of the chain flags.

Remarks See the Visual Basic documentation about possible color values or color constants.

ChainFlagsOffForeColor Property

Brief description Returns/sets the foreground color to display the 'Off' state of the chain status flags.

Syntax `object.ChainFlagsOffForeColor [= color]`

The ChainFlagsOffForeColor property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`color`	A value or constant that determines the background or foreground colors to display the `On' and `Off' states of the chain flags.

Remarks See the Visual Basic documentation about possible color values or color constants.

ChainFlagsOnBackColor Property

Brief description Returns/Sets the background color to display the 'On' state of the chain status flags.

Syntax `object.ChainFlagsOnBackColor [= color]`

The ChainFlagsOnBackColor property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`color`	A value or constant that determines the background or foreground colors to display the `On' and `Off' states of the chain flags.

Remarks See the Visual Basic documentation about possible color values or color constants.

ChainFlagsOnForeColor Property

Brief description Returns/Sets the foreground color to display the 'On' state of the chain status flags.

Syntax `object.ChainFlagsOnForeColor [= color]`

The ChainFlagsOnForeColor property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`color`	A value or constant that determines the background or foreground colors to display the `On' and `Off' states of the chain flags.

Remarks See the Visual Basic documentation about possible color values or color constants.

9.2 ColumnHdrActionXxx Properties

At a glance

Introduction

In this section you will find an overview of the ColumnHdrActionXxx properties.

What's in this Section?

This section contains the following topics:

ColumnHdrActionComment Property

Brief description Returns/Sets the header text of the **Comment** column for step actions.

Syntax `object.ColumnHdrActionComment [string]`

The ColumnHdrActionComment property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`string`	A string expression that evaluates to the text displayed as the header of the columns for the step action grids in the detail view.

ColumnHdrActionName Property

Brief description Returns/sets the header text of the **Name** column for step actions.

Syntax `object.ColumnHdrActionName [string]`

The ColumnHdrActionName property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`string`	A string expression that evaluates to the text displayed as the header of the columns for the step action grids in the detail view.

ColumnHdrActionQualifier Property

Brief description Returns/Sets the header text of the **Qualifier** column for step actions.

Syntax `object.ColumnHdrActionQualifier [string]`

The ColumnHdrActionQualifier property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`string`	A string expression that evaluates to the text displayed as the header of the columns for the step action grids in the detail view.

ColumnHdrActionTime Property

Brief description Returns/Sets the header text of the **Time** column for step actions.

Syntax `object.`**`ColumnHdrActionTime [`**`string`**`]`**

The ColumnHdrActionTime property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`string`	A string expression that evaluates to the text displayed as the header of the columns for the step action grids in the detail view.

ColumnHdrActionType Property

Brief description Returns/sets the header text of the **Type** column for step actions.

Syntax `object.`**`ColumnHdrActionType [`**`string`**`]`**

The ColumnHdrActionType property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`string`	A string expression that evaluates to the text displayed as the header of the columns for the step action grids in the detail view.

9.3 ColumnHdrChainXxx Properties

At a glance

Introduction

In this section you will find an overview of the ColumnHdrChainXxx properties.

What's in this Section?

This section contains the following topics:

ColumnHdrChainComment Property

Brief description Returns/Sets the header text of the **Chain Comment** column.

Syntax `object.ColumnHdrChainComment [string]`

The ColumnHdrChainComment property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`string`	A string expression that evaluates to the text displayed as the header of the columns for the chain grid in the overview.

ColumnHdrChainName Property

Brief description Returns/Sets the header text of the **Chain Name** column.

Syntax `object.ColumnHdrChainName [string]`

The ColumnHdrChainName property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`string`	A string expression that evaluates to the text displayed as the header of the columns for the chain grid in the overview.

ColumnHdrChainStatus Property

Brief description Returns/sets the header text of the **Chain Status** column.

Syntax

```
object.ColumnHdrChainStatus [string]
```

The ColumnHdrChainStatus property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`string`	A string expression that evaluates to the text displayed as the header of the columns for the chain grid in the overview.

9.4 ChainHdrErrorXxx Properties

At a glance

Introduction

In this section you will find an overview of the ChainHdrErrorXxx properties.

What's in this Section?

This section contains the following topics:

ColumnHdrErrorComment Property

Brief description Returns/sets the header text of the **Comment** column for transition errors.

Syntax `object.ColumnHdrErrorComment [= string]`

The ColumnHdrErrorComment property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`string`	A string expression that evaluates to the text displayed as the header of the columns for the diagnostic error grid in the detail view.

ColumnHdrErrorStep Property

Brief description Returns/Sets the header text of the **Step** column for transition errors.

Syntax `object.ColumnHdrErrorStep [= string]`

The ColumnHdrErrorStep property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`string`	A string expression that evaluates to the text displayed as the header of the columns for the diagnostic error grid in the detail view.

ColumnHdrErrorVariable Property

Brief description Returns/sets the header text of the **Variable** column for transition errors.

Syntax `object.ColumnHdrErrorVariable [= string]`

The ColumnHdrErrorVariable property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`string`	A string expression that evaluates to the text displayed as the header of the columns for the diagnostic error grid in the detail view.

ColumnHdrErrorPinType Property

Brief description Returns/sets the header text of the **Pin Type** column for transition errors.

Syntax `object.ColumnHdrErrorPinType [= string]`

The ColumnHdrErrorPinType property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`string`	A string expression that evaluates to the text displayed as the header of the columns for the diagnostic error grid in the detail view.

Remarks Column **ErrorPin** and **ErrorState** occur only when UsePLCDiagSystem property is set to false.

ColumnHdrErrorState Property

Brief description

Returns/sets the header text of the **Comment** column for transition errors.

Syntax

```
object.ColumnHdrErrorState [= string]
```

The ColumnHdrErrorState property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`string`	A string expression that evaluates to the text displayed as the header of the columns for the diagnostic error grid in the detail view.

Remarks

Column **ErrorPin** and **ErrorState** occur only when UsePLCDiagSystem property is set to false.

9.5 ColumnHdrStepXxx Properties

At a glance

Introduction

In this section you will find an overview of the ColumnHdrStepXxx properties.

What's in this Section?

This section contains the following topics:

ColumnHdrParallelSteps Property

Brief description Returns/sets the header text of the **Parallel Steps** column.

Syntax

```
object.ColumnHdrParallelSteps [= string]
```

The ColumnHdrParallelSteps property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`string`	A string expression that evaluates to the text displayed as the header of the columns for the chain grid in the overview.

ColumnHdrStepComment Property

Brief description Returns/sets the header text of the **Step Comment** column.

Syntax

```
object.ColumnHdrStepComment [= string]
```

The ColumnHdrStepComment property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`string`	A string expression that evaluates to the text displayed as the header of the columns for the diagnostic error grid in the detail view.

ColumnHdrStepName Property

Brief description Returns/Sets the header text of the **Step Name** column.

Syntax

```
object.ColumnHdrStepName [= string]
```

The ColumnHdrStepName property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`string`	A string expression that evaluates to the text displayed as the header of the columns for the diagnostic error grid in the detail view.

9.6 DetailsXxx Properties

At a glance

Introduction

In this section you will find an overview of the DetailsXxx properties.

What's in this Section?

This section contains the following topics:

DetailsActiveStepBackColor Property

Brief description Returns/Sets the background color to display a step in active-state in the detail view.

Syntax `object.DetailsActiveStepBackColor [= color]`

The DetailsActiveStepBackColor property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`color`	A value or constant that determines the background colors to display the `Active', Inactive' and `Initial' states of the steps in the detail view.

Remarks See the Visual Basic documentation about possible color values or color constants.

DetailsInactiveStepBackColor Property

Brief description Returns/Sets the background color to display a step in inactive-state in the detail view.

Syntax `object.DetailsInactiveStepBackColor [= color]`

The DetailsInactiveStepBackColor property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`color`	A value or constant that determines the background colors to display the `Active', Inactive' and `Initial' states of the steps in the detail view.

Remarks See the Visual Basic documentation about possible color values or color constants.

DetailsInitialStepBackColor Property

Brief description

Returns/Sets the background color to display initial step indicator in the details and details simple view. In details simple view dependently on property. DetailsSimpleShowInitStepIndicator the Initial step indicator or step name label will be displayed in this color when currently displayed step is initial.

Syntax

```
object.DetailsInitialStepBackColor [= color]
```

The DetailsInitialStepBackColor property syntax has these parts:

Part	Description
object	An object expression that evaluates to a SFC View control.
color	A value or constant that determines the background colors to display the `Active', Inactive' and `Initial' states of the steps in the detail view.

Remarks

See the Visual Basic documentation about possible color values or color constants.

DetailsWaitingStepBackColor Property

Brief description

Returns/Sets the background color to display a step in waiting-state in the details simple view. Waiting-state means that displayed step is initial step or reset flag is set for the chain.

Syntax

```
object.DetailsWaitingStepBackColor [= color]
```

The DetailsWaitingStepBackColor property syntax has these parts:

Part	Description
object	An object expression that evaluates to a SFC View control.
color	A value or constant that determines the background colors to display the `Active', Inactive' and `Initial' states of the steps in the detail view.

Remarks

See the Visual Basic documentation about possible color values or color constants.

DetailsViewLinesColor Property

Brief description Returns/sets lines color in details view.

Syntax

```
object.DetailsViewLinesColor [= color]
```

The DetailsViewLinesColor property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`color`	A value or constant that determines the lines colors in details view.

Remarks See the Visual Basic documentation about possible color values or color constants.

9.7 DetailsSimpleXxx Properties

At a glance

Introduction

In this section you will find an overview of the DetailsSimpleXxx properties.

What's in this Section?

This section contains the following topics:

DetailsSimpleShowChainName Property

Brief description Returns/Sets whether the chain name is displayed in details simple view.

Syntax `object.DetailsSimpleShowChainName [= boolean]`

The DetailsSimpleShowChainName property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`boolean`	A boolean expression that specifies whether the chain name (chain status, chain comment or step comment) is displayed in details simple view.

Settings The settins for `boolean` are:

Setting	Description
True	(Default) Chain name (chain status, chain comment or step comment) is displayed.
False	Chain name (chain status, chain comment or step comment) is not displayed.

DetailsSimpleShowChainStatus Property

Brief description

Returns/Sets whether the chain status is displayed in details simple view. When set to false the chain status will be indicated by back groundcolor of step name label.

Syntax

`object.`**`DetailsSimpleShowChainStatus [`**`= boolean`**`]`**

The DetailsSimpleShowChainStatus property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`boolean`	A boolean expression that specifies whether the chain name (chain status, chain comment or step comment) is displayed in details simple view.

Settings

The settins for `boolean` are:

Setting	Description
True	(Default) Chain name (chain status, chain comment or step comment) is displayed.
False	Chain name (chain status, chain comment or step comment) is not displayed.

DetailsSimpleShowChainComment Property

Brief description Returns/Sets whether the chain comment is displayed in details simple view.

Syntax `object.DetailsSimpleShowChainComment [= boolean]`

The DetailsSimpleShowChainComment property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`boolean`	A boolean expression that specifies whether the chain name (chain status, chain comment or step comment) is displayed in details simple view.

Settings The settins for `boolean` are:

Setting	Description
True	(Default) Chain name (chain status, chain comment or step comment) is displayed.
False	Chain name (chain status, chain comment or step comment) is not displayed.

DetailsSimpleShowStepErrorLabel Property

Brief description

Returns/Sets whether step error state is displayed in the details simple view. When set to false the error state will be indicated by background color of step name label.

Syntax

`object.`**`DetailsSimpleShowStepErrorLabel [`**`= boolean`**`]`**

The DetailsSimpleShowStepErrorLabel property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`boolean`	A boolean expression that specifies whether the chain name (chain status, chain comment or step comment) is displayed in details simple view.

Settings

The settins for `boolean` are:

Setting	Description
True	(Default) Chain name (chain status, chain comment or step comment) is displayed.
False	Chain name (chain status, chain comment or step comment) is not displayed.

DetailsSimpleShowStepComment Property

Brief description Returns/Sets whether the step comment is displayed in details simple view.

Syntax

```
object.DetailsSimpleShowStepComment [= boolean]
```

The DetailsSimpleShowStepComment property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`boolean`	A boolean expression that specifies whether the chain name (chain status, chain comment or step comment) is displayed in details simple view.

Settings The settins for `boolean` are:

Setting	Description
True	(Default) Chain name (chain status, chain comment or step comment) is displayed.
False	Chain name (chain status, chain comment or step comment) is not displayed.

DetailsSimpleShowInitStepIndicator Property

Brief description

Returns/Sets whether indicator for initial step is displayed in the details simple view. When set to false the initial step will be indicated by background color of step name label.

Syntax

`object.`**`DetailsSimpleShowInitStepIndicator [`**`= boolean`**`]`**

The DetailsSimpleShowInitStepIndicator property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`boolean`	A boolean expression that specifies whether the chain name (chain status, chain comment or step comment) is displayed in details simple view.

Settings

The settins for `boolean` are:

Setting	Description
True	(Default) Chain name (chain status, chain comment or step comment) is displayed.
False	Chain name (chain status, chain comment or step comment) is not displayed.

DetailsSimpleShowNavigation Property

Brief description Returns/Sets whether navigation buttons are displayed in details simple view.

Syntax

```
object.DetailsSimpleShowNavigation [= boolean]
```

The DetailsSimpleShowNavigation property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`boolean`	A boolean expression that specifies whether navigation buttons are displayed in details simple view.

Settings The settins for `boolean` are:

Setting	Description
True	(Default) Navigation buttons are displayed in details simple view.
False	Navigation buttons are not displayed in details simple view.

Remarks Navigation buttons can be visible only when DetailsSimpleShowNavigation property is true.

DetailsSimpleStepNameFont Property

Brief description

Returns/Sets the font to display the step name in the details simple view.

Syntax

[Step] `object.`**`DetailsSimpleStepNameFont [`**`= font_object`**`]`**

The DetailsSimpleStepNameFont property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`font_object`	An object expression that evaluates to a font object that determines the font name, size and other attributes to display the step names in the details simple view.

Remarks

The same font is used for the chain name in the detail simple view, too. All other texts in the detail view and in overview are displayed with the font specified with the standard Font property.

9.8 DetailsStepXxx Properties

At a glance

Introduction

In this section you will find an overview of the DetailsStepXxx properties.

What's in this Section?

This section contains the following topics:

DetailsStepNameFont Property

Brief description Returns/Sets the font to display the step name in the details view.

Syntax [Set] `object.`**`DetailsStepNameFont [`**`= font_object`**`]`**

The DetailsStepNameFont property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`font_object`	An object expression that evaluates to a font object that determines the font name, size and other attributes to display the step names in the details view.

Remarks The same font is used for the chain name and the transition name in the detail view, too. All other texts in the detail view and in overview are displayed with the font specified with the standard Font property.

DetailsStepsLeft Property

Brief description Returns/Sets the left positions to display the steps in the detail view.

Syntax `object.`**`DetailsStepsLeft [`**`= value`**`]`**

The DetailsStepsLeft property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`value`	A numeric expression specifying a distance or dimension.

Remarks The DetailsStepsLeft property determines the distance between the inner left edge of the SFC View control and the left edge of the rectangles in which the steps are displayed in the detail view.
This property is measured in Twips. See the Visual Basic documentation for more information about this measurement unit.

DetailsStepsWidth Property

Brief description Returns/Sets the width to display the steps in the detail view.

Syntax

```
object.DetailsStepsWidth [= value]
```

The DetailsStepsWidth property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`value`	A numeric expression specifying a distance or dimension.

Remarks The DetailsStepsWidth property determines the width of the rectangles in which the steps are displayed in the detail view.
This property is measured in Twips. See the Visual Basic documentation for more information about this measurement unit.

9.9 DetailsTextXxx Properties

At a glance

Introduction

In this section you will find an overview of the DetailsTextXxx properties.

What's in this Section?

This section contains the following topics:

DetailsTextDisableActions Property

Brief description Returns/Sets the text to display the chain state 'Actions Disabled' in the detail view.

Syntax

```
object.DetailsTextDisableActions [= string]
```

The DetailsTextDisableActions property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`string`	A string expression that evaluates to the text displayed for the chain states in the detail view.

DetailsTextDisableTimeCheck Property

Brief description Returns/Sets the text to display the chain state 'Time Check Disabled' in the detail view.

Syntax

```
object.DetailsTextDisableTimeCheck [= string]
```

The DetailsTextDisableTimeCheck property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`string`	A string expression that evaluates to the text displayed for the chain states in the detail view.

DetailsTextDisableTransitions Property

Brief description Returns/Sets the text to display the chain state 'Transitions Disabled' in the detail view.

Syntax `object.DetailsTextDisableTransitions [= string]`

The DetailsTextDisableTransitions property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`string`	A string expression that evaluates to the text displayed for the chain states in the detail view.

DetailsTextSectionDisabled Property

Brief description Returns/Sets the text to display the chain state 'Section Disabled' in the detail view.

Syntax `object.DetailsTextSectionDisabled [= string]`

The DetailsTextSectionDisabled property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`string`	A string expression that evaluates to the text displayed for the chain states in the detail view.

DetailsTextSetResetFlag Property

Brief description Returns/Sets the text to display the chain state 'Reset Flag Set' in the detail view.

Syntax `object.DetailsTextSetResetFlag [= string]`

The DetailsTextSetResetFlag property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`string`	A string expression that evaluates to the text displayed for the chain states in the detail view.

9.10 DiagXxx Properties

At a glance

Introduction

In this section you will find an overview of the DiagXxx properties.

What's in this Section?

This section contains the following topics:

DiagAutoRetrigger Property

Brief description

Returns/Sets whether diag errors are automatically re-triggered for analysis or not.

Syntax

```
object.DiagAutoRetrigger [= boolean]
```

The DiagAutoRetrigger property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`boolean`	A boolean expression that specifies whether diag error analysis is automatically re-triggered or not.

Settings

The settings for `boolean` are:

Setting	Description
True	(Default) Enables automatic re-triggering of the error analysis.
False	Disables automatic re-triggering of the error analysis.

Remarks

If the property is set to false, the client application can re-trigger the analysis by calling the method DiagRetrigger.

DiagAutoRetriggerInterval Property

Brief description

Returns/Sets the interval (1 to 65535ms) at which diag entries will be retriggered. Default is 1000ms.

Syntax

```
object.DiagAutoRetriggerInterval [= milliseconds]
```

The DiagAutoRetriggerInterval property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`milliseconds`	A numeric expression specifying the number of milliseconds (1 to 65535) between two retrigger actions on all active SFC errors in actually displayed section in the details view. Default is 1000.

Remarks

Setting the value too low might have a negative influence on the system performance.

9.11 OPCXxx Properties

At a glance

Introduction

In this section you will find an overview of the OPCXxx properties.

What's in this Section?

This section contains the following topics:

OPCNetworkServer Property

Brief description Returns/Sets information about the location of the OPC server.

Syntax

```
object.OPCNetworkServer [= string]
```

The OPCNetworkServer property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`string`	A string expression that determines where the OPC server is located.

Remarks Default value of the property is " ".
That means the OPC server is located on the local machine.
In case of distributed configuration please use this property to define the location (PC name or IP address) of the OPC server.

OPCAccessPath Property

Brief description Returns/Sets information how the OPC server should get the data.

Syntax

```
object.OPCAccessPath [= string]
```

The OPCAccessPath property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`string`	A string expression that determines how the OPC server should get the data.

Remarks The string has to be one of the alias names defined with the OFS configuration tool, which determines the Concept project and the PLC to be used by OFS.
When the property is set with the control's property pages, one can select an alias name from the current OFS configuration data. This configuration data can also be retrieved with the GetOFSInfo method.

OPCConnect Property

Brief description

Returns/Sets whether to connect to or disconnect from the OPC server.

Syntax

```
object.OPCConnect [= boolean]
```

The OPCConnect property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`boolean`	A boolean expression that specifies whether to connect to or disconnect from the OPC server.

Settings

The settings for `boolean` are:

Setting	Description
True	(Default) Starts the connection with the OPC server at runtime.
False	Closes all connections with the OPC server.

Remarks

The OPCConnect property is ignored at designtime. At runtime, the connection with the OPC server will be started only after the project data was read by the control and when the OPCConnect property is set to true.

OPCUpdateRate Property

Brief description Returns/Sets the fastest rate at which the OPC server should deliver changed data.

Syntax

```
object.OPCUpdateRate [= milliseconds]
```

The OPCUpdateRate property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`milliseconds`	A numeric expression specifying the minimum number of milliseconds between two transmissions of changed data by the OPC server. Default is 1000.

Remarks The OPCUpdateRate property determines how often the OPC server should send changed data to the control. Setting the value too low might have a negative influence on the system performance.

9.12 OverviewTextXxx Properties

At a glance

Introduction

In this section you will find an overview of the OverviewTextXxx properties.

What's in this Section?

This section contains the following topics:

OverviewTextDisableActions Property

Brief description Returns/Sets the text to display the chain state 'Actions Disabled' in the overview and details simple view.

Syntax

```
object.OverviewTextDisableActions [= string]
```

The OverviewTextDisableActions property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`string`	A string expression that evaluates to the text displayed for the chain states the overview and details simple view.

OverviewTextDisableTimeCheck Property

Brief description Returns/Sets the text to display the chain state 'Time Check Disabled' in the overview and details simple view.

Syntax

```
object.OverviewTextDisableTimeCheck [= string]
```

The OverviewTextDisableTimeCheck property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`string`	A string expression that evaluates to the text displayed for the chain states the overview and details simple view.

OverviewTextDisableTransitions Property

Brief description Returns/Sets the text to display the chain state 'Transitions Disabled' in the overview and details simple view.

Syntax `object.OverviewTextDisableTransitions [= string]`

The OverviewTextDisableTransitions property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`string`	A string expression that evaluates to the text displayed for the chain states the overview and details simple view.

OverviewTextRunning Property

Brief description Returns/Sets the text to display the chain state 'Running' in the overview.

Syntax `object.OverviewTextRunning [= string]`

The OverviewTextRunning property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`string`	A string expression that evaluates to the text displayed for the chain states the overview and details simple view.

OverviewTextSectionDisabled Property

Brief description Returns/Sets the text to display the chain state 'Section Disabled' in the overview and details simple view.

Syntax `object.OverviewTextSectionDisabled [= string]`

The OverviewTextSectionDisabled property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`string`	A string expression that evaluates to the text displayed for the chain states the overview and details simple view.

OverviewTextSetResetFlag Property

Brief description Returns/Sets the text to display the chain state 'Reset Flag Set' in the overview and details simple view.

Syntax `object.OverviewTextSetResetFlag [= string]`

The OverviewTextSetResetFlag property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`string`	A string expression that evaluates to the text displayed for the chain states the overview and details simple view.

9.13 ShowXxx Properties

At a glance

Introduction

In this section you will find an overview of the ShowXxx properties.

What's in this Section?

This section contains the following topics:

ShowActiveStep Property

Brief description Returns/Sets whether the details and details simple views displays always the active step.

Syntax `object.`**`ShowActiveStep [`**`= boolean`**`]`**

The ShowActiveStep property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`boolean`	A boolean expression that specifies whether the details and details simple view displays always the active step.

Settings The settings for `boolean` are:

Setting	Description
True	(Default) The detail and details simple view displays always the active step.
False	The detail view does not follow the active step.

ShowAllDiagErrors Property

Brief description Returns/Sets whether all diag errors are shown or not.

Syntax `object.`**`ShowAllDiagErrors [`**`= boolean`**`]`**

The ShowAllDiagErrors property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`boolean`	A boolean expression that specifies whether all diag errors of the chain are shown in the detail view or not.

Settings The settings for `boolean` are:

Setting	Description
True	(Default) The detail view displays all diag errors for the current chain.
False	The detail view shows only the errors for the displayed step.

Remarks The property is valid only when UsePLCDiagSystem property is set to true.

ShowAllTransitionsInput Property

Brief description Returns/Sets whether diagnostic information is shown for all active steps or only for actually selected step.

Syntax `object.`**`ShowAllTransitionsInput [`**`= boolean`**`]`**

The ShowAllTransitionsInput property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`boolean`	A boolean expression that determines whether diagnostic information in detail view is shown for all active steps or only for actually selected step.

Settings The settings for `boolean` are:

Setting	Description
True	(Default) Diagnostic information in detail view is displayed for all active steps.
False	Diagnostic information in detail view is displayed only for actually selected step.

Remarks The property is valid only when UsePLCDiagSystem property is set to false.

ShowChainGroups Property

Brief description

Returns/Sets whether the overview displays the hierarchy of the chain groups.

Syntax

```
object.ShowChainGroups [= boolean]
```

The ShowChainGroups property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`boolean`	A boolean expression that specifies whether the overview displays the hierarchy of the chain groups or not.

Settings

The settings for `boolean` are:

Setting	Description
True	(Default) The overview displays the hierarchy of the chain groups as configured in the project and only the chains in the selected group are shown.
False	The overview displays all chains of the project in one table.

Remarks

It's recommended to configure chain groups (functional moduls) in the Unity/ Concept project and to set this property to True, if there are many SFC sections in the project. Since the control needs real-time data for all steps in all chains displayed in the overview at the same time, it might have a negative influence on the system performance when there are too much chains displayed in the overview.

ShowStatistics Property

Brief description Displays the statistics panel.

Syntax `object.ShowStatistics [= boolean]`

The ShowStatistics property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`boolean`	A boolean expression that specifies whether to display the statistics panel.

Settings The settings for `boolean` are:

Setting	Description
True	The statistics panel is shown.
False	(Default) The statistics panel is hidden.

ShowStepComments Property

Brief description Returns/Sets whether step comments or chain comments are displayed in the overview grid.

Syntax `object.ShowStepComments [= boolean]`

The ShowStepComments property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`boolean`	A boolean expression that specifies whether to display step comments or chain comments in the overview grid.

Settings The settings for `boolean` are:

Setting	Description
True	(Default) The comments of the active steps are shown in the overview grid.
False	The chain comments are shown in the overview grid.

ShowTimeInms Property

Brief description

Returns/Sets whether in the details and details simple views, step delay time and current time are displayed in milliseconds.

Syntax

```
object.ShowTimeInms [= boolean]
```

The ShowTimeInms property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`boolean`	A boolean expression that specifies whether in the details and details simple views, step delay time and current time are displayed in milliseconds.

Settings

The settings for `boolean` are:

Setting	Description
True	(Default) Step delay time and current time are displayed in milliseconds.
False	Step delay time and current time are displayed in seconds.

ShowBlockNames Property

Brief description

Returns/Sets whether to display structure of logic assigned to transition in the grid for transition errors.

Syntax

```
object.ShowBlockNames [= boolean]
```

The ShowBlockNames property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`boolean`	A boolean expression that specifies whether to display logic structure or not.

Settings

The settings for `boolean` are:

Setting	Description
True	(Default) Input variables assigned to the transition are shown together with logic structure (AND, OR blocs etc.).
False	Input variables assigned to the transition are shown without logic structure.

Remarks

The property is valid only when UsePLCDiagSystem property is set to false.

9.14 StepMaxTimeErrXxx Properties

At a glance

Introduction

In this section you will find an overview of the StepMaxTimeErrXxx properties.

What's in this Section?

This section contains the following topics:

StepMaxTimeErrBackColor Property

Brief description Returns/Sets the background color to display that the maximum step time is exceeded.

Syntax `object.StepMaxTimeErrBackColor [= color]`

The StepMaxTimeErrBackColor property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`color`	A value or constant that determines the background and foreground colors to display that the maximum step time is exceeded.

Remarks See the Visual Basic documentation about possible color values or color constants.

StepMaxTimeErrForeColor Property

Brief description Returns/Sets the foreground color to display that the maximum step time is exceeded.

Syntax `object.StepMaxTimeErrForeColor [= color]`

The StepMaxTimeErrForeColor property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`color`	A value or constant that determines the background and foreground colors to display that the maximum step time is exceeded.

Remarks See the Visual Basic documentation about possible color values or color constants.

StepMaxTimeErrText Property

Brief description

Returns/Sets the text to display that the maximum step time is exceeded (detail view).

Syntax

```
object.StepMaxTimeErrText [= string]
```

The StepMaxTimeErrText property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`string`	A string expression that determines the text to display in the detail view when the maximum step time is exceeded.

9.15 StepMinTimeErrXxx Properties

At a glance

Introduction

In this section you will find an overview of the StepMinTimeErrXxx properties.

What's in this Section?

This section contains the following topics:

StepMinTimeErrBackColor Property

Brief description Returns/Sets the background color to display that the minimum step time was not reached.

Syntax `object.StepMinTimeErrBackColor [= color]`

The StepMinTimeErrBackColor property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`color`	A value or constant that determines the background and foreground colors to display that the minimum step time was not reached.

Remarks See the Visual Basic documentation about possible color values or color constants.

StepMinTimeErrForeColor Property

Brief description Returns/Sets the foreground color to display that the minimum step time was not reached.

Syntax `object.StepMinTimeErrForeColor [= color]`

The StepMinTimeErrForeColor property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`color`	A value or constant that determines the background and foreground colors to display that the minimum step time was not reached.

Remarks See the Visual Basic documentation about possible color values or color constants.

StepMinTimeErrText Property

Brief description

Returns/Sets the text to display that the minimum step time was not reached (detail view).

Syntax

```
object.StepMinTimeErrText [= string]
```

The StepMinTimeErrText property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`string`	A string expression that determines the text to display in the detail view when the minimum step time was not reached.

9.16 UseXxx Properties

At a glance

Introduction

In this section you will find an overview of the UseXxx properties.

What's in this Section?

This section contains the following topics:

UseOPCProject Property

Brief description

Returns/Sets whether the project file will be determined automatically from the OPC access path.

Syntax

```
object.UseOPCProject [= boolean]
```

The UseOPCProject property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`boolean`	A boolean expression that specifies whether the project file will be determined automatically from the OPC access path.

Settings

The settings for `boolean` are:

Setting	Description
True	(Default) The project file will be determined automatically from the control's OPCAccessPath property.
False	The project file has to be set with the ProjectFile property.

UsePLCDiagSystem Property (Unity Pro)

Brief description

Returns/Sets whether the PLC's diagnostic system is used to get the causes of transition errors. If set to false, the causes are determined internally. If changed at runtime, it reloads the current project or not depending on the UseEasySwitchMode property.

Syntax

```
object.UsePLCDiagSystem [= boolean]
```

The UsePLCDiagSystem property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`boolean`	A boolean value that determines used diagnostic mode.

Settings

The settings for `boolean` are:

Setting	Description
True	(Default) Control reads error buffer in PLC, to get diagnostic information.
False	Control does not use PLC diagnoses. SFC error causes are determined internally based on the logic connected to transitions. To be able to get diagnostic data, each transition has to be assigned to the variable, which is an output of special EFB (SFC View library: AND_16, OR_16, AND_OR_8).

UsePLCDiagSystem Property

Brief description

Returns/Sets whether the PLC's diagnostic system is used to get the causes of transition errors. If set to false, the causes are determined internally. If changed at runtime, it reloads the current project.

Syntax

```
object.UsePLCDiagSystem [= boolean]
```

The UsePLCDiagSystem property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`boolean`	A boolean value that determines used diagnostic mode.

Settings

The settings for `boolean` are:

Setting	Description
True	Control reads error buffer in PLC, to get diagnostic information. Concept project has to contain ERR2HMI block. Properties VarDiagIn and VarDiagOut, which define input and output variables of the block have to be also defined.
False	(Default) Control does not use PLC diagnoses. SFC error causes are determined internally based on the logic connected to transitions. To be able to get diagnostic data, each transition has to be assigned to the variable, which is an output of special EFB (Concept SFC_VIEW library: AND_16, OR_16, AND_OR_8).

UseEasyModeSwitch Property

Brief description

Returns/Sets whether reload or not current project when UsePLCDiagSystem property has changed.

Syntax

```
object.UseEasyModeSwitch [= boolean]
```

The UseEasyModeSwitch property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`boolean`	A boolean value that determines used diagnostic mode.

Settings

The settings for `boolean` are:

Setting	Description
True	Control will not reload project data when UsePLCDiagSystem property has changed. Setting this property to true, make switching between modes faster but it slows down loading the project and causes bigger memory consumption. Set the property to true only when you really need it.
False	(Default) Reload project data when UsePLCDiagSystem property has changed.

Remarks

Changing this property in UserMode causes project reload.

9.17 ValueXxx Properties

At a glance

Introduction

In this section you will find an overview of the ValueXxx properties.

What's in this Section?

This section contains the following topics:

ValueOffBackColor Property

Brief description Returns/Sets the background color to display a variable value in `Off' state.

Syntax `object.ValueOffBackColor [= color]`

The ValueOffBackColor property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`color`	A value or constant that determines the background or foreground colors to display the `On' and `Off' states of the variable values.

Remarks See the Visual Basic documentation about possible color values or color constants.

ValueOffForeColor Property

Brief description Returns/Sets the foreground color to display a variable value in `Off' state.

Syntax `object.ValueOffForeColor [= color]`

The ValueOffForeColor property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`color`	A value or constant that determines the background or foreground colors to display the `On' and `Off' states of the variable values.

Remarks See the Visual Basic documentation about possible color values or color constants.

ValueOnBackColor Property

Brief description Returns/Sets the background color to display a variable value in `On' state.

Syntax `object.ValueOnBackColor [= color]`

The ValueOnBackColor property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`color`	A value or constant that determines the background or foreground colors to display the `On' and `Off' states of the variable values.

Remarks See the Visual Basic documentation about possible color values or color constants.

ValueOnForeColor Property

Brief description Returns/Sets the foreground color to display a variable value in `On' state.

Syntax `object.ValueOnForeColor [= color]`

The ValueOnForeColor property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`color`	A value or constant that determines the background or foreground colors to display the `On' and `Off' states of the variable values.

Remarks See the Visual Basic documentation about possible color values or color constants.

9.18 WidthActionXxx Properties

At a glance

Introduction

In this section you will find an overview of the WidthActionXxx properties.

What's in this Section?

This section contains the following topics:

WidthActionQualifierColumn Property

Brief description Returns/Sets width of **Action Qualifier** column in the grid for step (actual and next) actions.

Syntax

```
object.Width[ActionQualifierColumn [= long]]
```

The WidthActionQualifierColumn property syntax has these parts:

Part	Description
object	An object expression that evaluates to a SFC View control.
long	A long type expression that specifies width of column in the grid for step (actual and next) actions.

Remarks To hide a column you have to set Width property of the column to 0.

WidthActionTimeColumn Property

Brief description Returns/Sets width of **Error Variable** column in the grid for step (actual and next) actions.

Syntax

```
object.Width[ActionTimeColumn [= long]]
```

The WidthActionTimeColumn property syntax has these parts:

Part	Description
object	An object expression that evaluates to a SFC View control.
long	A long type expression that specifies width of column in the grid for step (actual and next) actions.

Remarks To hide a column you have to set Width property of the column to 0.

WidthActionVariableColumn Property

Brief description Returns/Sets width of **Pin Type** column in the grid for step (actual and next) actions.

Syntax

```
object.Width ActionVariableColumn [= long]
```

The WidthActionVariableColumn property syntax has these parts:

Part	Description
object	An object expression that evaluates to a SFC View control.
long	A long type expression that specifies width of column in the grid for step (actual and next) actions.

Remarks To hide a column you have to set Width property of the column to 0.

WidthActionCommentColumn Property

Brief description Returns/Sets width of **Error State** column in the grid for step (actual and next) actions.

Syntax

```
object.Width ActionCommentColumn [= long]
```

The WidthActionCommentColumn property syntax has these parts:

Part	Description
object	An object expression that evaluates to a SFC View control.
long	A long type expression that specifies width of column in the grid for step (actual and next) actions.

Remarks To hide a column you have to set Width property of the column to 0.

9.19 WidthErrorXxx Properties

At a glance

Introduction

In this section you will find an overview of the WidthErrorXxx properties.

What's in this Section?

This section contains the following topics:

WidthErrorStepNameColumn Property

Brief description Returns/Sets width of **Step Name** column in the grid for transition errors.

Remarks When UsePLCDiagSystem property is true, width of **Error State** and **Pin Type** columns is always 0. When ShowAllTransitionsInput property is set to false, **Step Name** column width is also 0.

> **Note:** To hide a column you have to set Width property of the column to 0.

WidthErrorVariableColumn Property

Brief description Returns/Sets width of **Error Variable** column in the grid for transition errors.

Syntax

```
object.WidthErrorVariableColumn [= long]
```

The WidthErrorVariableColumn property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`long`	A long type expression that specifies width of column in the grid for transition errors.

Remarks When UsePLCDiagSystem property is true, width of **Error State** and **Pin Type** columns is always 0. When ShowAllTransitionsInput property is set to false, **Step Name** column width is also 0.

> **Note:** To hide a column you have to set Width property of the column to 0.

WidthErrorPinTypeColumn Property

Brief description Returns/Sets width of **Pin Type** column in the grid for transition errors.

Syntax `object.Width`**`ErrorPinTypeColumn [`**`= long`**`]`**

The WidthErrorPinTypeColumn property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`long`	A long type expression that specifies width of column in the grid for transition errors.

Remarks When UsePLCDiagSystem property is true, width of **Error State** and **Pin Type** columns is always 0. When ShowAllTransitionsInput property is set to false, **Step Name** column width is also 0.

Note: To hide a column you have to set Width property of the column to 0.

WidthErrorStateColumn Property

Brief description Returns/Sets width of **Error State** column in the grid for transition errors.

Syntax `object.Width`**`ErrorStateColumn [`**`= long`**`]`**

The WidthErrorStateColumn property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`long`	A long type expression that specifies width of column in the grid for transition errors.

Remarks When UsePLCDiagSystem property is true, width of **Error State** and **Pin Type** columns is always 0. When ShowAllTransitionsInput property is set to false, **Step Name** column width is also 0.

Note: To hide a column you have to set Width property of the column to 0.

WidthErrorCommentColumn Property

Brief description Returns/Sets width of **Error Comment** column in the grid for transition errors.

Syntax `object.Width`**`ErrorCommentColumn [`**`= long`**`]`**

The WidthErrorCommentColumn property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`long`	A long type expression that specifies width of column in the grid for transition errors.

Remarks When UsePLCDiagSystem property is true, width of **Error State** and **Pin Type** columns is always 0. When ShowAllTransitionsInput property is set to false, **Step Name** column width is also 0.

Note: To hide a column you have to set Width property of the column to 0.

9.20 Other Properties

At a glance

Introduction

In this section you will find an overview of the other properties.

What's in this Section?

This section contains the following topics:

Alias Property

Brief description

Returns a string specifying the alias name of a device configured for use with OFS.

Syntax

`object.`**`Alias`**

The Alias property syntax has this part:

Part	Description
`object`	An object expression that evaluates to an OFSDevice object.

Remarks

This is a read-only property.
The value of this property for a given device should be entered in the SFC View control's OPCAccessPath property to specify which Unity/Concept project to use and from which PLC OFS should read the real-time data.

AutomaticProjectReload Property

Brief description

Returns/Sets whether to reload the project automatically after a change of the project within the PLC was recognized.

Syntax

```
object.AutomaticProjectReload [= boolean]
```

The AutomaticProjectReload property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`boolean`	A boolean expression that specifies whether to reload the project automatically after a change of the project within the PLC was recognized.

Settings

The settings for `boolean` are:

Setting	Description
True	(Default) Allows reloading the project automatically.
False	Prevents automatic reloading of the project.

Remarks

If the property is set to false, the client application can implement its own strategy for reloading the project by responding to the ProjectChanged event with calling the method ReloadProject only under special circumstances.

BackColor Property

Brief description Returns/Sets the control background color in details and details simple view.

Syntax `object.BackColor [= color]`

The BackColor property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`color`	A value or constant that determines the control background color in details and details simple view.

Remarks See the Visual Basic documentation about possible color values or color constants.

Count Property

Brief description Returns the number of OFSDevice objects in the OFSDevices collection.

Syntax `object.Count`

The Count property syntax has this part:

Part	Description
`object`	An object expression that evaluates to an OFSDevices object.

Remarks This is a read-only property.

ContentErrorStateColumn Property

Brief description

Returns/Sets what to display in the **Error State** column in the grid for transition errors.

Syntax

`object.`**`ContentErrorStateColumn [`**`= value`**`]`**

The ContentErrorStateColumn property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`value`	Numeric expression, which determines what, is displayed in **Error State** column.

Settings

The settings for `value` are:

Setting	Description
0 - cecEmpty	**Error State** column is empty.
1 - cecErrorState	(Default) **Error State** column contains error state of input variable.
2 - cecVariableName	**Error State** column contains variable name.
3 - cecPinType	**Error State** column contains pin type.

Remarks

The property is valid only when UsePLCDiagSystem property is false.

EnableOPCUpdates Property

Brief description

Returns/Sets whether to start or stop notifications about changed data by the OPC server.

Syntax

`object.`**`EnableOPCUpdates [`**`= boolean`**`]`**

The EnableOPCUpdates property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`boolean`	A boolean expression that specifies whether to start or stop notifications by the OPC server.

Settings

The settings for `boolean` are:

Setting	Description
True	(Default) Enables notifications by the OPC server.
False	Disables notifications by the OPC server.

Remarks

By setting the property to false, the client application can temporarily stop notifications about changed data (from the OPC server to the SFC View control), e.g. when the window with the control is currently in the background.

Font Property

Brief description

Returns/Sets the font to display chain status, chain comment, step comment etc in the overview details and details simple view.

Syntax

[Set] `object.`**`Font [`**`= font_object`**`]`**

The Font property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`font_object`	An object expression that evaluates to a font object that determines the font name, size and other attributes to display chain status, chain comment, step comment etc in the overview details and details simple view.

Item Property

Brief description Returns a specific member of the OFSDevices collection either by position or by key.

Syntax

```
object.Item (index)
object(index)
```

It is not necessary to specify the name of this property as shown in the second syntax line above, since the Item property is the default property of the OFSDevices object.
The Item property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to an OFSDevices object.
`index`	An expression that specifies the position of a member of the collection. If a numeric expression, index must be a number from 1 to the value of the collection's Count property. If a string expression, index must correspond to the Alias property of the OFSDevice member object referred to.

Remarks This is a read-only property.

MaxChannel Property

Brief description Returns a number specifying the maximum number of channels of a device configured for use with OFS.

Syntax

```
object.MaxChannel
```

The MaxChannel property syntax has this part:

Part	Description
`object`	An object expression that evaluates to an OFSDevices object.

Remarks This is a read-only property.

NumberErrorGridLines Property

Brief description Returns/Sets value, which indicates number of lines displayed in error grid.

Syntax

```
object.NumberErrorGridLines [= Integer]
```

The NumberErrorGridLines property syntax has these parts:

Part	Description
object	An object expression that evaluates to a SFC View control.
value	Integer value representing number of lines displayed in error grid. Default value 16.

OFSDevices Property

Brief description Returns the current OFSDevices collection of an OFSInfo object.

Syntax

```
object.OFSDevices
```

The OFSDevices property syntax has this part:

Part	Description
object	An object expression that evaluates to an OFSInfo object.

Remarks This is a read-only property.

OnChainOpen Property

Brief description Returns/Sets the action to be performed when the user double clicks a row in the grid while in overview mode.

Syntax `object.OnChainOpen [= value]`

The OnChainOpen property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`value`	A numeric expression that specifies what to do when the user double clicks a chain in the grid while in overview mode.

Settings For the settings for value see *OnChainOpenActions Constants, p. 218*.

Path Property

Brief description Returns a string specifying the address of a device configured for use with OFS.

Syntax `object.Path`

The Path property syntax has this part:

Part	Description
`object`	An object expression that evaluates to an OFSDevices object.

Remarks This is a read-only property.

ProjectFile Property

Brief description Returns/Sets the name of the file with the project data.

Syntax

```
object.ProjectFile [= string]
```

The ProjectFile property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`string`	A string expression that determines from which file to read the project data.

Remarks The string should contain the complete path to the Concept/Unity project (.STU for Unity; .PRJ for Concept) file.
This property will be set automatically by setting the OPCAccessPath property when the property UseOPCProject is set to true at the same time.
The ProjectFile property is ignored at designtime. At runtime, the control starts reading the project data when the property is set. If there was a connection with the OPC server with any previous project, the connection will be closed and started again for the new project data.

Refresh Property

Brief description Returns a number specifying whether automatic refresh of a device configured for use with OFS is enabled.

Syntax

```
object.Refresh
```

The Refresh property syntax has this part:

Part	Description
`object`	An object expression that evaluates to an OFSDevices object.

Remarks This is a read-only property.
If the value of this property is zero, automatic refresh is not configured for this device.

StateErrorStateColumn Property

Brief description

Returns/Sets what indicate the background color in the **Error State** column in the grid for transition errors.

Syntax

```
object.StateErrorStateColumn [= value]
```

The StateErrorStateColumn property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`value`	Numeric expression, which determines what indicate the background color in the **Error State** column.

Settings

The settings for `value` are:

Setting	Description
0 - sescNoState	The background color is gray.
1 - sescErrorState	The background color indicate, if the variable is an error or not.
2 - sescPinState	(Default) The background color indicate pin state, to which the variable is directly connected. (If the pin is inverted the state differs from variable state. If the pin is not inverted the pin state is equal to the variable state.)
3 - sescVariableState	The background color indicate variable state.

Remarks

The property is valid only when UsePLCDiagSystem property is false.

Symb Property

Brief description

Returns a string specifying the full path to the file with information about the symbols of a device configured for use with OFS.

Syntax

```
object.Symb
```

The Symb property syntax has this part:

Part	Description
`object`	An object expression that evaluates to an OFSDevices object.

Remarks

This is a read-only property.
If the value of the object's Alias property is entered in the SFC View control's OPCAccessPath property and the control's property UseOPCProject is set to true, the value of this property has to be the file name of the Unity program (.STU file).

UnityNetworkServer Property

Brief description

Returns/Sets information about the location of the Unity Pro program.

Syntax

```
object.UnityNetworkServer [= string]
```

The UnityNetworkServer property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`string`	A string expression that determines where the Unity Pro program is located.

Remarks

Default value of the property is " ".
That means the Unity Pro program is located on the local machine.
In case of distributed configuration please use this property to define the location (PC name or IP address) of the Unity Pro program.

ViewMode Property

Brief description

Returns/Sets the kind of information to be displayed.

Syntax

```
object.ViewMode [= value]
```

The ViewMode property syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`value`	A numeric expression that specifies whether to show the overview, details or details simple view.

Settings

For the settings for value see *SFCViewModes Constants, p. 219*.

Methods

10

At a glance

Introduction

In this chapter you will find information about the methods.

What's in this Chapter?

This chapter contains the following sections:

10.1 DiagXxx Methods

At a glance

Introduction

In this section you will find an overview of the DiagXxx methods.

What's in this Section?

This section contains the following topics:

DiagResetErrorBuffer Method

Brief description Resets all diagnostic errors within the PLC.

Syntax `object.DiagResetErrorBuffer`

The DiagResetErrorBuffer method syntax has this part:

Part	Description
`object`	An object expression that evaluates to a SFC View control.

Return Value None

DiagRetrigger Method

Brief description Retriggers the analysis of the diagnostic errors of the chain, which is currently displayed in the detail view.

Syntax `object.DiagRetrigger`

The DiagRetrigger method syntax has this part:

Part	Description
`object`	An object expression that evaluates to a SFC View control.

Return Value None

10.2 DisplayXxx Methods

At a glance

Introduction

In this section you will find an overview of the DisplayXxx methods.

What's in this Section?

This section contains the following topics:

DisplayInitialStep Method

Brief description Navigate to and display the initial step. This method works in details and details simple view mode.

Syntax `object.DisplayInitialStep`

The DisplayInitialStep method syntax has this part:

Part	Description
`object`	An object expression that evaluates to a SFC View control.

Return Value A boolean value indicating the success (true) or failure (false) of the operation.

DisplayNextActiveStep Method

Brief description Navigate to and display the next active step. This method works in details and details simple view mode.

Syntax `object.DisplayNextActiveStep`

The DisplayNextActiveStep method syntax has this part:

Part	Description
`object`	An object expression that evaluates to a SFC View control.

Return Value A boolean value indicating the success (true) or failure (false) of the operation.

DisplayNextAltTran Method

Brief description Navigate to and display the next alternate transition. This method works only in details mode.

Syntax `object.DisplayNextAltTran`

The DisplayNextAltTran method syntax has this part:

Part	Description
`object`	An object expression that evaluates to a SFC View control.

Return Value A boolean value indicating the success (true) or failure (false) of the operation.

DisplayNextParStep Method

Brief description Navigate to and display the next parallel step. This method works only in details mode.

Syntax `object.DisplayNextParStep`

The DisplayNextParStep method syntax has this part:

Part	Description
`object`	An object expression that evaluates to a SFC View control.

Return Value A boolean value indicating the success (true) or failure (false) of the operation.

DisplayNextStep Method

Brief description Navigate to and display the next step. This method work in details and details simple mode.

Syntax `object.DisplayNextStep`

The DisplayNextStep method syntax has this part:

Part	Description
`object`	An object expression that evaluates to a SFC View control.

Return Value A boolean value indicating the success (true) or failure (false) of the operation.

DisplayPreviousStep Method

Brief description Navigate to and display the previous step. This method work in details and details simple mode.

Syntax `object.DisplayPreviousStep`

The DisplayPreviousStep method syntax has this part:

Part	Description
`object`	An object expression that evaluates to a SFC View control.

Return Value A boolean value indicating the success (true) or failure (false) of the operation.

DisplayPrevActiveStep Method

Brief description Navigate to and display the previous active step. This method works in details and details simple mode.

Syntax `object.DisplayPrevActiveStep`

The DisplayPrevActiveStep method syntax has this part:

Part	Description
`object`	An object expression that evaluates to a SFC View control.

Return Value A boolean value indicating the success (true) or failure (false) of the operation.

DisplayPrevAltTran Method

Brief description Navigate to and display the previous alternate transition. This method works only in details mode.

Syntax `object.DisplayPrevAltTran`

The DisplayPrevAltTran method syntax has this part:

Part	Description
`object`	An object expression that evaluates to a SFC View control.

Return Value A boolean value indicating the success (true) or failure (false) of the operation.

DisplayPrevParStep Method

Brief description

Navigate to and display the previous parallel step. This method works only in details mode.

Syntax

`object.`**`DisplayPrevParStep`**

The DisplayPrevParStep method syntax has this part:

Part	Description
`object`	An object expression that evaluates to a SFC View control.

Return Value

A boolean value indicating the success (true) or failure (false) of the operation.

10.3 PLCXxx Methods

At a glance

Introduction

In this section you will find an overview of the PLCXxx methods.

What's in this Section?

This section contains the following topics:

PLCClearChain Method

Brief description

Resets all active steps in the sequence.

WARNING

Danger of unsafe, dangerous and destructive operations for tools or processes.

`PLCClearChain` should not be used for trouble shooting while operating machine tools, processes or material administration systems while they are running. This can lead to unsafe, dangerous and destructive operations by the tools or processes linked to the controller.

Failure to follow this precaution can result in death, serious injury, or equipment damage.

Syntax

`object.`**`PLCClearChain(`**`value`**`)`**

The PLCClearChain method syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`value`	A boolean expression that indicates to reset all active steps in the sequence.

Return Value

A boolean value indicating the success (true) or failure (false) of the operation.

Remarks

The sequence can only be started again by the 1 -> 0 edge of the PLCSetInitializeFlag.

PLCDisableActions Method

Brief description

Enables or disables the processing of actions for the selected SFC section.

WARNING

Danger from unsafe, dangerous and destructive operations in tools or processes.

`PLCDisableActions` should not be used for finding errors with controllers of machine tools, processes or material maintenance systems, when they are running. Since no logic processing is taking place any longer, the control panel ignores all input information. This can cause unsafe, dangerous, and destructive operations of tools or processes connected to the control.

Failure to follow this precaution can result in death, serious injury, or equipment damage.

Syntax

```
object.PLCDisableActions(disable)
```

The PLCDisableActions method syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`disable`	A boolean expression that indicates whether to disable the actions (true) or not (false).

Return Value

A boolean value indicating the success (true) or failure (false) of the operation.

PLCDisableSection Method

Brief description

Enables or disables the processing of the selected SFC section.

WARNING

Danger from unsafe, dangerous and destructive operations in tools or processes.

`PLCDisableSection` should not be used for controllers of machine tools, processes or material maintenance systems, when they are running. Since no logic processing is taking place any longer, the control panel ignores all input information. This can cause unsafe, dangerous, and destructive operations of tools or processes connected to the control.

Failure to follow this precaution can result in death, serious injury, or equipment damage.

Syntax

```
object.PLCDisableSection(disable)
```

The PLCDisableSection method syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`disable`	A boolean expression that indicates whether to disable the processing (true) or not (false).

Return Value

A boolean value indicating the success (true) or failure (false) of the operation.

PLCDisableTimeCheck Method

Brief description Enables or disables the time check.

Syntax `object.PLCDisableTimeCheck(disable)`

The PLCDisableTimeCheck method syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`disable`	A boolean expression that indicates whether to disable the time check (true) or not (false).

Return Value A boolean value indicating the success (true) or failure (false) of the operation.

Remarks If DisableTimeCheck is activated, time supervision of the steps is no longer performed. However, delay time is still active.

PLCDisableTransitions Method

Brief description

Enables or disables transitions.

WARNING

Danger from unsafe, dangerous and destructive operations in tools or processes.

`PLCDisableTransitions` should not be used for controllers of machine tools, processes or material maintenance systems, when they are running. Since no logic processing is taking place any longer, the control panel ignores all input information. This can cause unsafe, dangerous, and destructive operations of tools or processes connected to the control.

Failure to follow this precaution can result in death, serious injury, or equipment damage.

Syntax

`object.`**`PLCDisableTransitions(`**`disable`**`)`**

The PLCDisableTransitions method syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`disable`	A boolean expression that indicates whether to disable the transitions (true) or not (false).

Return Value

A boolean value indicating the success (true) or failure (false) of the operation.

Remarks

If DisableTransitions is activated, the states of the transitions will no longer be analyzed. The sequence remains in its current state, regardless of the signals on the transitions. Operation of the sequence is only possible with the control commands SetResetFlag, StepUnconditional, and StepTransDependent.

PLCGotoNextStep Method

Brief description

Activates the next step.

WARNING

Danger from unsafe, dangerous and destructive operations in tools or processes.

Using the method `PLCGotoNextStep` it is possible to activate the next step, even when the transition is not satisfied. `PLCGotoNextStep` should therefore not be used for finding errors with controllers of machine tools, processes or material maintenance systems, when they are running. This can cause unsafe, dangerous, and destructive operations of tools or processes connected to the control.

Failure to follow this precaution can result in death, serious injury, or equipment damage.

Syntax

`object.`**`PLCGotoNextStep(`**`unconditional`**`)`**

The PLCGotoNextStep method syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`unconditional`	A boolean expression that indicates whether to activate the next step regardless of the state of the transition (true) or dependent from the state of the transition (false).

Return Value

A boolean value indicating the success (true) or failure (false) of the operation.

Remarks

If unconditional is true, the next step will be activated independent of the state of the transition, but not until the delay time of the active step has expired.
In parallel branches, a step in each branch will be activated, while in alternative branches it always activates the left branch.

PLCResetTimeErrors Method

Brief description Resets supervision time errors for the selected SFC section.

Syntax `object.`**`PLCResetTimeErrors`**

The PLCResetTimeErrors method syntax has this part:

Part	Description
`object`	An object expression that evaluates to a SFC View control.

Return Value A boolean value indicating the success (true) or failure (false) of the operation.

PLCSetInitializeFlag Method

Brief description

Sets or clears the initialize flag.

WARNING

Danger of unsafe, dangerous and destructive operations for tools or processes.

`PLCSetInitializeFlag` should not be used for trouble shooting while operating machine tools, processes or material administration systems while they are running. This can lead to unsafe, dangerous and destructive operations by the tools or processes linked to the controller.

Failure to follow this precaution can result in death, serious injury, or equipment damage.

Syntax

```
object.PLCSetInitializeFlag(value)
```

The PLCSetInitializeFlag method syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`value`	A boolean expression that indicates whether to set the initialize flag (true) or to clear it (false).

Return Value

A boolean value indicating the success (true) or failure (false) of the operation.

Remarks

Use the initialize flag to initialize the sequence for a standardized start.
To initialize the sequence, activate the initialize flag. This will stop the sequence and all actions will be initialized. Operator interventions are not possible.
Standardized sequence start:
To start the sequence, the initialize flag must first be activated and then deactivated. The 1 -> 0 edge will initialize the sequence, i.e. the initial step is activated.

10.4 Other Methods

At a glance

Introduction

In this section you will find an overview of the other methods.

What's in this Section?

This section contains the following topics:

About Method

Brief description Displays the **About** box for the control.

Syntax `object.About`

The About method syntax has this part:

Part	Description
`object`	An object expression that evaluates to a SFC View control.

Return Value None

Remarks This is the same as clicking **About** in the **Properties** window.

GetOFSInfo Method

Brief description Get an object with OFS configuration data.

Syntax `object.GetOFSInfo`

The GetOFSInfo method syntax has this part:

Part	Description
`object`	An object expression that evaluates to a SFC View control.

Return Value An object of type OFSInfo, which allows to read OFS configuration data.

ReloadProject Method

Brief description Reloads the current project.

Syntax `object.`**`ReloadProject`**

The ReloadProject method syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.

Return Value A boolean value indicating the success (true) or failure (false) of the operation.

Remarks If there was a connection with the OPC server previously, the connection will be closed and started again.

Events

11

At a glance

Introduction

In this chapter you will find information about the events.

What's in this Chapter?

This chapter contains the following sections:

11.1 ChainXxx Events

At a glance

Introduction

In this section you will find an overview of the ChainXxx events.

What's in this Section?

This section contains the following topics:

ChainOpen Event

Brief description Raised in overview mode when the user double clicks on a chain in the grid or presses the **Enter** key while the grid has the focus.

Syntax Sub `object_`**`ChainOpen`**(ByVal `ChainName` As String**)**

The ChainOpen event syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`ChainName`	The name of the chain to open.

ChainSelect Event

Brief description Raised in overview mode when the user selects a new row in the grid.

Syntax Sub `object_`**`ChainSelect`**(ByVal `ChainName` As String**)**

The ChainSelect event syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`ChainName`	The name of the selected chain.

ChainStatusChanged Event

Brief description Raised when a chain's status has changed.

Syntax Sub `object_`**`ChainStatusChanged`**(ByVal `ChainName` As String, ByVal `NewStatus` As SFCViewCtrl.ChainStatusFlags**)**

The ChainStatusChanged event syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`ChainName`	The name of the chain.
`NewStatus`	The new status of the chain represented by a combination of one or more of the ChainStatusFlags constants.

11.2 Other Events

At a glance

Introduction

In this section you will find an overview of the other events.

What's in this Section?

This section contains the following topics:

ActionVarSelect Event

Brief description Raised in detail mode when the user double clicks on the grid with the current or next step's actions in it. The variable clicked on is passed with the chain name.

Syntax Sub `object_`**`ActionVarSelect`**(ByVal `ActionVar` As String, ByVal `ChainName` As String**)**

The ActionVarSelect event syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`ActionVar`	The name of the selected action variable.
`ChainName`	The name of the current chain.

DetailsSimpleDblClick Event

Brief description Raised in details simple mode when the user double clicks on the step name label.

Syntax Sub `object.`**`DetailsSimpleDblClick ()`**

The DetailsSimpleDblClick event syntax has this part:

Part	Description
`object`	An object expression that evaluates to a SFC View control.

DiagVarSelect Event

Brief description Raised in detail mode when the user double clicks on the grid with the diagnostics in it. The variable clicked on is passed with the chain name.

Syntax Sub `object_`**`DiagVarSelect`**(ByVal `DiagVar` As String, ByVal `ChainName` As String**)**

The DiagVarSelect event syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`DiagVar`	The name of the selected variable in the diagnostics grid.
`ChainName`	The name of the chain.

ProjectChanged Event

Brief description Raised when a change of the project within the PLC was recognized.

Syntax Sub `object_`**`ProjectChanged()`**

The ProjectChanged event syntax has this part:

Part	Description
`object`	An object expression that evaluates to a SFC View control.

ViewModeChanged Event

Brief description Raised when the view mode has changed.

Syntax Sub `object_`**`ViewModeChanged`**(ByVal `NewViewMode` As SFCViewCtrl.SFCViewModes**)**

The ViewModeChanged event syntax has these parts:

Part	Description
`object`	An object expression that evaluates to a SFC View control.
`NewViewMode`	The new view mode of the control represented by one of the SFCViewModes constants.

Constants

12

At a glance

Introduction

In this chapter you will find information about the constants.

What's in this Chapter?

This chapter contains the following topics:

ChainStatusFlags Constants

Constants

The chain's status will be delivered by the ChainStatusChanged event as a combination of one or more of the following values.

Constants:

Constant	Value	Description
CsfRunning	0	The chain is in running mode.
CsfSectionDisabled	1	The chain is disabled.
CsfResetFlagSet CsfInitializeFlagSet	2	The chain's reset flag is set (Concept). The chain's initialize flag is set (Unity Pro).
CsfTimeCheckDisabled	4	The chain's time check is disabled.
CsfTransitionsDisabled	8	The chain's transitions are disabled.
CsfActionsDisabled	16	The chain's actions are disabled.
CsfUnknown	4096	The chain's status is unknown.

OnChainOpenActions Constants

Constants

The following values can be written to or returned from the control's property OnChainOpen, which allows to get or set the action to be performed when the user double clicks a row in the grid while in overview mode.

Constants:

Constant	Value	Description
NoAction	0	The user's input will be ignored.
ShowDetails	1	The chain will be shown in detail view.
ShowDetailsSimple	2	The chain will be shown in details simple view.

SFCViewModes Constants

Constants

The following values can be written to or returned from the control's property ViewMode, which allows getting or setting the control's current display mode. Additionally, the control's ViewModeChanged event delivers one of these values when the view mode has changed.

Constants:

Constant	Value	Description
SfcOverview	0	The chain is/will be shown in the overview.
SfcDetails	1	The chain is/will be shown in the detail view.
SfcDetailsSimple	2	The chain is/will be shown in the details simple view.

ContentsErrorStateColumn Constants

Constants

The following values can be written to or returned from the control's property ContentErrorStateColumn, which allows getting or setting the content of **Error State** column in error grid.

Constants:

Constant	Value	Description
CecEmpty	0	Error State column is empty.
CecErrorState	1	Error state column contains error state of input variable.
CecVariableName	2	Error state column contains variable name.
CecPinType	3	Error state column contains Pin Type.

StatesErrorStateColumn Constants

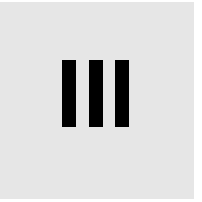
Constants

The following values can be written to or returned from the control's property StateErrorStateColumn, which allows to get or set what indicate the background color in the **Error State** column.

Constants:

Constant	Value	Description
sescNoState	0	The background color is gray.
sescErrorState	1	The background color indicate, if the variable is an error or not.
sescPinState	2	The background color indicate pin state, to which the variable is directly connected.
sescVariableState	3	The background color indicate variable state.

SFCView block library



Introduction

Overview

The SFC View block library contains three diagnostic EFBs and an EFB for controlling step chains by SFC View, which are not included in the standard Unity Pro shipping.

What's in this Part?

This part contains the following chapters:

Block types and their applications

13

Introduction

Overview

This chapter describes the different block types and their applications.

What's in this Chapter?

This chapter contains the following topics:

Block types

Block types

Different block types are used in Unity Pro. The general term for all block types is FFB.
There are the following types of block:
- Elementary Function (EF)
- Elementary Function Block (EFB)
- Derived Function Block (DFB)
- Procedure

Elementary Function

Elementary functions (EF) have no internal status.. If the input values are the same, the value at the output is the same for all executions of the function, e.g. the addition of two values gives the same result at every execution.
An elementary function is represented in the graphical languages (FDB and LD) as a block frame with inputs and an output. The inputs are always represented on the left and the outputs always on the right of the frame The name of the function, i.e. the function type, is shown in the center of the frame.
The number of inputs can be increased with some elementary functions.

Elementary function block

Elementary function blocks (EFB) have an internal status. If the inputs have the same values, the value on the output can have another value during the individual executions. For example, with a counter, the value on the output is incremented.
An elementary function block is represented in the graphical languages (FDB and LD) as a block frame with inputs and outputs. The inputs are always represented on the left and the outputs always on the right of the frame The name of the function block, i.e. the function block type, is shown in the center of the frame. The instance name is displayed above the frame.

Derived function block

Derived function blocks (DFBs) have the same properties as elementary function blocks. They are created by the user in the programming languages FBD, LD, IL and/or ST.

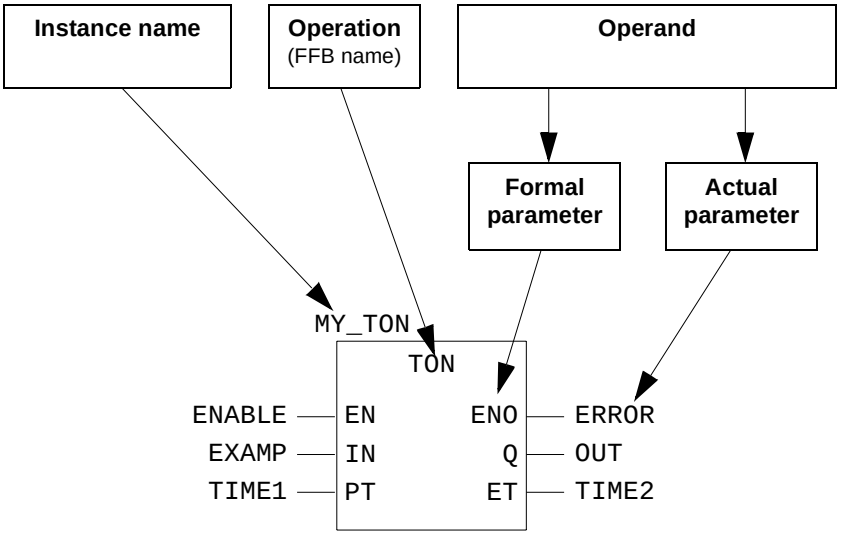
Procedure

Procedures are technical functions.
The only difference from elementary functions is that procedures can have more than one output and they support variables of the `VAR_IN_OUT` data type.
Procedures do not return a value.
Procedures are a supplement to IEC 61131-3 and must be enabled explicitly.
There is no visual difference between procedures and elementary functions.

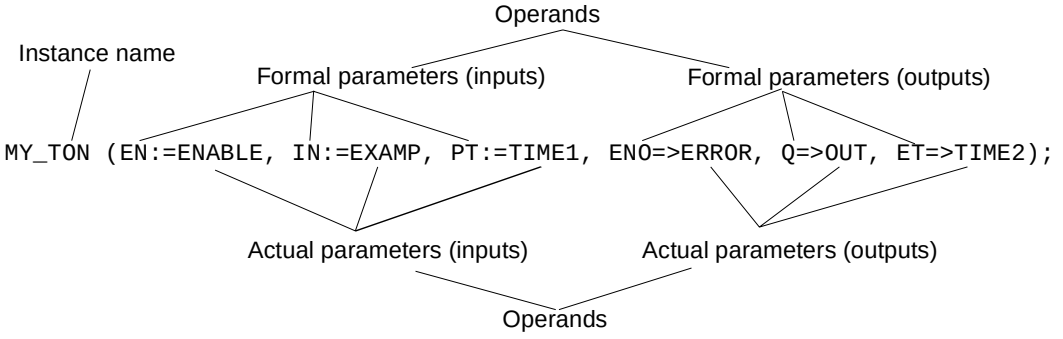
FFB Structure

Structure

Each FFB is made up of an operation (name of the FFB), the operands required for the operation (formal and actual parameters) and an instance name for elementary/ derived function blocks.
Call of a function block in the FBD programming language:



Formal call of a function block in the ST programming language:



Operation

The operation determines which function is to be executed with the FFB, e.g. shift register, conversion operations.

Operand

The operand specifies what the operation is to be executed with. With FFBs, this consists of formal and actual parameters.

Formal/actual parameters

Inputs and outputs are required to give values to the FFB or to take values from the FFB. These are called formal parameters.
Objects are connected to the formal parameters which contain the current process states. These are called actual parameters.
During program runtime, the actual parameters are used to pass the process values to the FFB and output them after processing.
The data type of the actual parameters must match the data type of the input/output (formal parameters). The only exceptions are generic inputs/outputs, for which the data types are determined by the actual parameters. If all actual parameters are literals, the correct data type for the function block will be selected.

FFB Call in IL/ST

In text languages IL and ST, FFBs can be called in formal and in informal form. Details can be found in the *Reference manual*.
Example of a formal function call:

```
out:=LIMIT (MN:=0, IN:=var1, MX:=5) ;
```

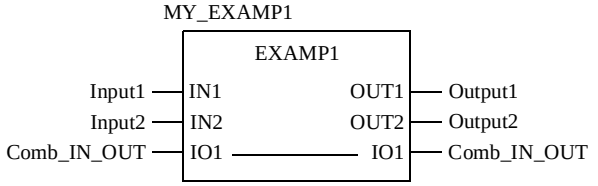
Example of an informal function call:

```
out:=LIMIT (0, var1, 5) ;
```

Note: Take note that the use of `EN` and `ENO` is only possible for formal calls.

VAR_IN_OUT variable

FFBs are often used to read a variable on an input (input variables), to process them and output the changed value of the **same** variable again (output variables).
This is special case for an input/output variable and is also called VAR_IN_OUT variable.
The input and output variable are linked in the graphic languages (FBD and LD) using a line showing that they belong together.
Function block with VAR_IN_OUT variable in FBD:



Function block with VAR_IN_OUT variable in ST:

```
MY_EXAMP1 (IN1:=Input1, IN2:=Input2, IO1:=Comb_IN_OUT,
           OUT1=>Output1, OUT2=>Output2) ;
```

The following points must be considered when using FFBs with VAR_IN_OUT variables:

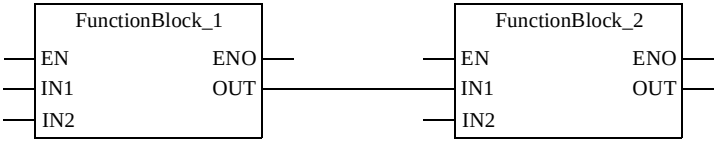
- VAR_IN_OUT variables absolutely must be assigned to as variable.
- The same variable/variable components must be assigned to the VAR_IN_OUT input and the VAR_IN_OUT output.
- In the graphic languages (FBD and LD), graphic connections cannot be made on VAR_IN_OUT inputs/outputs.
- Literals or constants cannot be assigned to VAR_IN_OUT inputs/outputs.
- In the graphic languages (FBD and LD), negations cannot be used on VAR_IN_OUT inputs/outputs.

EN and ENO

Description

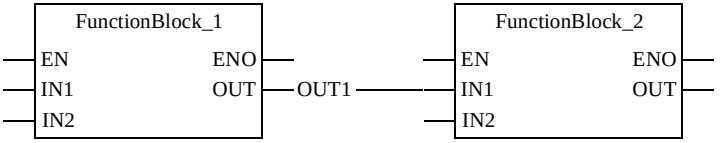
An EN input and an ENO output can be configured for all FFBs.
If the value of EN is "0" when the FFB is called up, the algorithms defined by the FFB are not executed and ENO is set to "0".
If the value of EN is "1" when the FFB is called up, the algorithms defined by the FFB are executed. After the algorithms have been executed successfully, the value of ENO is set to "1". If an error occurs when executing these algorithms, ENO is set to "0".
If ENO is set to "0" (caused by EN=0 or an error during execution):

- Function blocks
  - EN/ENO-handling with function blocks that (only) have one connection as output parameter:



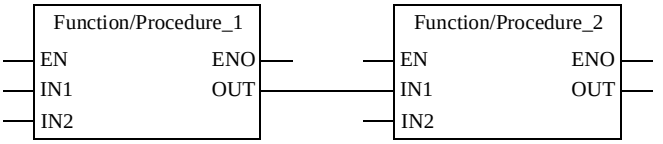
If EN from FunctionBlock_1 is set to "0", the output connection OUT from FunctionBlock_1 retains the status it had in the last correctly executed cycle.

  - EN/ENO-handling with function blocks that have one variable and one connection as output parameters:

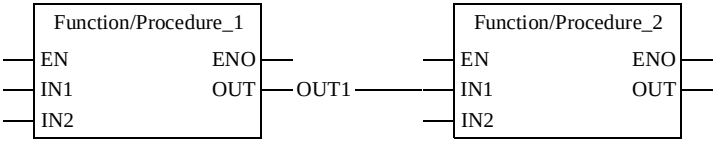


If EN from FunctionBlock_1 is set to "0", the output connection OUT from FunctionBlock_1 retains the status it had in the last correctly executed cycle. The variable OUT1 on the same pin, either retains its previous status or can be changed externally without influencing the connection. The variable and the connection are saved independently from one another.

- Functions/Procedures
  As defined in IEC61131-3, the outputs from deactivated functions (EN-input set to "0") is undefined. (The same applies for procedures.)
  Nevertheless here is an explanation of the output statuses for this case:
  - EN/ENO-handling with function/procedure blocks that (only) have one connection as output parameter:



    If EN from Function/Procedure_1 is set to "0", the output connection OUT from Function/Procedure_1 is also set to "0".
  - EN/ENO-handling with function/procedure blocks that have one variable and one connection as output parameters:



    If EN from Function/Procedure_1 is set to "0", the output connection OUT from Function/Procedure_1 is also set to "0", however the variable OUT1 on the same pin retains its previous value. In this way it is possible that the variable and the connection have different values.

The output behavior of the FFBs does not depend on whether the FFBs are called up without EN/ENO or with EN=1.

Conditional/ Unconditional FFB Call

"Unconditional" or "conditional" calls are possible with each FFB. The condition is realized by pre-linking the input EN.

- EN connected
  conditional calls (the FFB is only processed if EN = 1)
- EN not used or set to TRUE
  unconditional calls (FFB is always processed)

Note for FBD

If the EN input is used, it must be connected to logic (conditional call) or permanently set to TRUE (unconditional call) because otherwise the FFB will never be processed.

Note for LD

In LD, each FFB must be connected with the left power rail using a Boolean input. Normally, the EN input is used for this purpose.
If the EN input is not connected to the left power rail, it cannot be used or it must be permanently set to TRUE because otherwise the FFB will never be processed.

Note for IL and ST

The use of EN and ENO is only possible in the text languages for a formal FFB call, e.g.

```
MY_BLOCK (EN:=enable, IN1:=var1, IN2:=var2,
ENO=>error, OUT1=>result1, OUT2=>result2);
```

Assigning the variables to ENO must be done with the operator =>.
With an informal call, EN and ENO cannot be used.

AND_16: boolean AND with 16 predefined inputs

14

Description

Function description

This function block is used to implement a boolean AND operation with up to 16 inputs. For ascertaining the results of the operation only used, i.e. connected inputs are used.

Note: If **no** inputs are connected the block returns the value 'TRUE'.

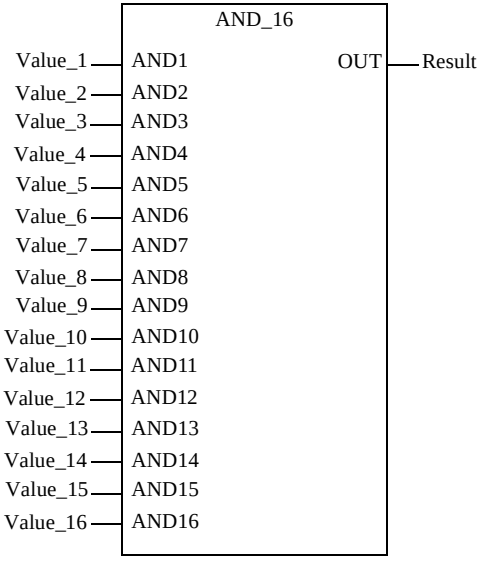
EN and ENO can be configured as additional parameters.

Formula

OUT = AND1 & AND2 & ANDn

Representation in FBD

Representation:



Representation in LD

The function block can only be used in FBD sections.

Representation in IL

The function block can only be used in FBD sections.

Representation in ST

The function block can only be used in FBD sections.

Parameter description

Description of the input parameters:

Parameters	Data type	Meaning
AND1	BOOL	Input 1
AND2	BOOL	Input 2
:	:	:
AND16	BOOL	Input 16

Description of the output parameters:

Parameters	Data type	Meaning
OUT	BOOL	Boolean AND connection for the connected inputs

AND_OR_8: Combined boolean AND-OR with 8 predefined inputs

15

Description

Function description

This block is used to implement a boolean AND connection with up to 8 inputs and a Boolean OR-connection with up to 8 inputs.
The result of the AND_OR_8-block is an AND-connection of the above connections.

> **Note:** Non-connected AND-inputs have by default the value 'TRUE'. Non-connected OR-inputs have by default the value 'FALSE'. Thus, if there is not at least **one** OR-input connected, the block delivers the value 'FALSE'.

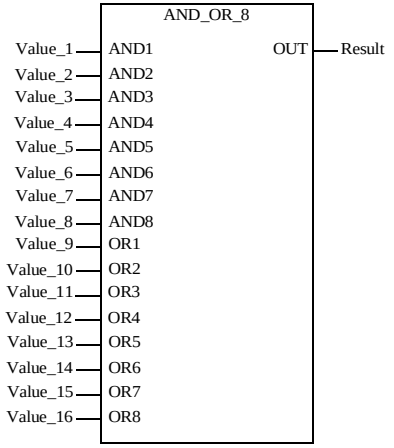
EN and ENO can be configured as additional parameters.

Formula

OUT = (AND1 & AND2 & ANDn) AND (OR1 OR OR2 OR ORn)

Representation in FBD

Representation:



Representation in LD

The function block can only be used in FBD sections.

Representation in IL

The function block can only be used in FBD sections.

Representation in ST

The function block can only be used in FBD sections.

Parameter description

Description of the input parameters:

Parameters	Data type	Meaning
AND1	BOOL	AND input 1
AND2	BOOL	AND input 2
:	:	:
AND8	BOOL	AND input 8
OR1	BOOL	OR input 1
OR2	BOOL	OR input 2
:	:	:
OR8	BOOL	OR input 8

Description of the output parameters:

Parameters	Data type	Meaning
OUT	BOOL	Boolean connection for the connected inputs

OR_16: boolean OR with 16 predefined inputs

16

Description

Function description

This block is used to implement a boolean OR connection with up to 16 inputs. For ascertaining the results of the operation only used, i.e. connected inputs are used.

Note: If **no** inputs are connected the block returns the value 'FALSE'.

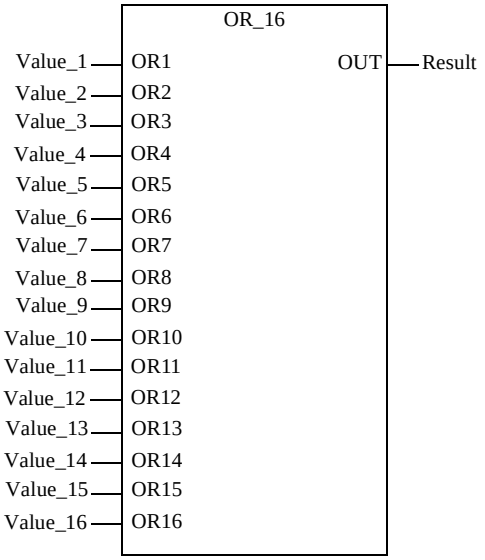
EN and ENO can be configured as additional parameters.

Formula

OUT = OR1 OR OR2 OR ORn

Representation in FBD

Representation:



Representation in LD

The function block can only be used in FBD sections.

Representation in IL

The function block can only be used in FBD sections.

Representation in ST

The function block can only be used in FBD sections.

Parameter description

Description of the input parameters:

Parameters	Data type	Meaning
OR1	BOOL	Input 1
OR2	BOOL	Input 2
:	:	:
OR16	BOOL	Input 16

Description of the output parameters:

Parameters	Data type	Meaning
OUT	BOOL	Boolean OR connection for the connected inputs

SFCVIEW_CTRL: Step chain control via SFCView

17

Description

Function description

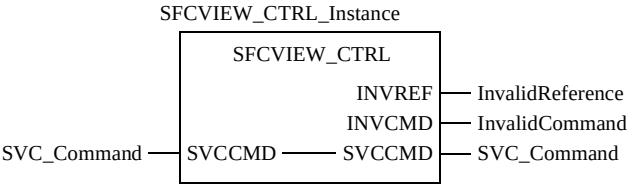
The function block is used to control the execution chain sequences via the `ChainControlVariableName` property in SFCView.
For example, you can go through step by step, processing transition conditions can be turned on or off or the chain can be reset to its initialization state.

	**DANGER**
	Danger of unsafe, dangerous and destructive processes. The "Initialize chain", "Reset chain", "Cancel analysis of transitions", "Cancel editing of actions and reset all actions of the chain", "Activate the next step independently of the transition condition" and "Activate the next step depending on the transition condition " functions should not be used to search for controller errors in machine tools, processes or material management systems, if they are running. **Failure to follow this precaution will result in death, serious injury, or equipment damage.**

`EN` and `ENO` can be configured as additional parameters.

Representation in FBD

Representation:



Representation in LD

The function block can only be used in FBD sections.

Representation in IL

The function block can only be used in FBD sections.

Representation in ST

The function block can only be used in FBD sections.

Parameter description

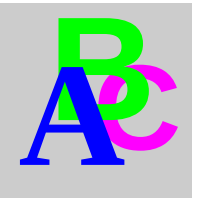
Description of the input/output parameters:

Parameters	Data type	Meaning
SVCCMD	SVCCMD	To control execution chain sequences via the `ChainControlVariableName` property in SFCView, assign to this input a variable with the name `SVC_Command`.

Description of the output parameters:

Parameters	Data type	Meaning
`INVREF`	`BOOL`	1: The reference sent to the function block is valid. 0: The reference sent to the function block is invalid. (Results in an entry into the error buffer).
`INVCMD`	`BOOL`	1: The command sent to the function block is valid. 0: The command sent to the function block is invalid. (Results in an entry into the error buffer).

Index



D

E

F

G

I

L

M

N

O

P

R

S

T

U

V

W