Unity Dif 2.1
a SoCollaborative software
User Manual

05/2009

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 help ensure compliance with documented system data, only the manufacturer should perform repairs to components.
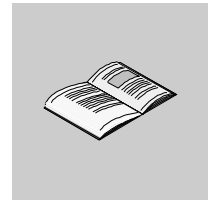
When devices are used for applications with technical safety requirements, the relevant instructions must be followed.

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 information can result in injury or equipment damage.

Table of Contents

Safety Information

Important Information

NOTICE

Read these instructions carefully, and look at the equipment to become familiar with the device before trying to install, operate, or maintain it. The following special messages may appear throughout this documentation or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.

The addition of this symbol to a Danger or Warning safety label indicates that an electrical hazard exists, which will result in personal injury if the instructions are not followed.

This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

⚠ **DANGER**

DANGER indicates an imminently hazardous situation which, if not avoided, **will result in** death or serious injury.

⚠ **WARNING**

WARNING indicates a potentially hazardous situation which, if not avoided, **can result in** death or serious injury.

> ### ⚠ CAUTION
>
> **CAUTION** indicates a potentially hazardous situation which, if not avoided, **can result in** minor or moderate injury.

> ### CAUTION
>
> **CAUTION**, used without the safety alert symbol, indicates a potentially hazardous situation which, if not avoided, **can result in** equipment damage.

PLEASE NOTE

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material.

A qualified person is one who has skills and knowledge related to the construction and operation of electrical equipment and the installation, and has received safety training to recognize and avoid the hazards involved.

About the Book

At a Glance

Document Scope

This manual describes the installation and the exploitation of the Unity Dif software.

Validity Note

This documentation is valid for Unity Dif version 2.1.

User Comments

We welcome your comments about this document. You can reach us by e-mail at techcomm@schneider-electric.com.

Overview of Unity Dif

I

Subject of this Part

This part introduces the main points of Unity Dif.

What's in this Part?

This part contains the following chapters:

Overview of Unity Dif

1

Presentation of Unity Dif

This chapter presents the main characteristics of Unity Dif.

What's in this Chapter?

This chapter contains the following topics:

Topic	Page
General Principles of Unity Dif	12
Requirements	13
Compatibility Files	15
The General Table of the Compared Elements	16

General Principles of Unity Dif

Function

Unity Dif is an optional tool to Unity Pro. This software is used to compare *(see page 33)* two projects (*STU* or *XEF* files) and provides an exhaustive list of all differences between:

- Hardware configuration for Modicon Quantum, Modicon Premium or Modicon M340.
- Network configuration.
- Project structure (Tasks, Functional Modules, etc).
- Program, regardless of the language used.
- All variables.
- User Function Blocs (DFB type).
- Strutured data type (DDT types).

The result *(see page 40)* of the comparison is displayed and can be printed or saved in *.TXT* format.

NOTE: In the manual, the term "Unity Pro files" will be used for PLC projects.

Requirements

Unity Dif Requirements

This software version of Unity Dif requires different configuration.

Computer configuration

Definition:
- The configuration type determines the level of expected performance.
- Performance of Unity Dif software can be affected by simultaneously opening several windows, accessing large quantities of animated data, or using a large application.

Unity Dif software requires one of the following operating systems to be present on the terminal:
- Windows Vista SP1,
- Windows XP SP2.

Microsoft Internet Explorer version $\geq$ 5.01 is required. Version 5.5 is recommended.

The following table provides the minimum characteristics necessary to install Unity Dif software:

Nodes	Characteristics	
System	Pentium 500 MHz or higher recommanded	
RAM	512 Mb	
Drives	Hard disk	1 GB available
	CD-ROM	
Monitor	1024*768	

NOTE: For the best use of Unity Dif, full administrator rights on the system are required.
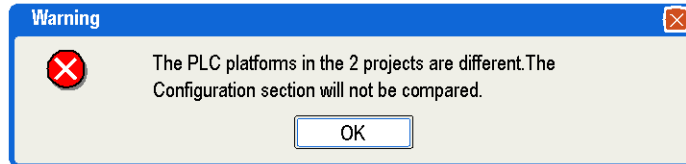
Software Requirements

The following list describes the software requirements:
- Unity Pro must be installed.
- .NET Framework v2.0 must be installed, if the installed version is too old, the required version is then installed automatically according to the Windows system.

Unity Dif use

The following list describes the requirements for a right use of Unity Dif:

● Supports installation and the version installed are available in English and French only.
● The *STA* files are not supported.
● Offline comparison is supported (2 project files located on PC or on remote PC).
● Comparison with a loaded PLC project is not supported.
● Comparison with 2 different PLC families is possible but the PLC-Conf comparison is not performed and the following message appears.

Warning

The PLC platforms in the 2 projects are different.The Configuration section will not be compared.

OK

● The generated report file can not be reopened in Unity Dif.

Compatibility Files

Compatibility Files of Unity Dif

All the Unity Pro files are not compatible in Unity Dif.

Table of Compatibilities

The following table describes the comparison possibles between 2 files.

Unity Pro Version installed		With the following file					
	First file	STU n-1	STU n	STA n-1	STA n	XEF n-1	XEF n
Vn-1	STU n-1	Yes	No	No	No	Yes	No
Vn	STU n	No	Yes	No	No	Yes	Yes
Vn-1	STA n-1	No	No	No	No	No	No
Vn	STA n	No	No	No	No	No	No
Vn-1	XEF n-1	Yes	No	No	No	Yes	No
Vn	XEF n	No	Yes	No	No	Yes	Yes
Legend:							
Application of level n	It is an application that is developed using Unity Pro.Vn.						
Unity Pro Vn	The Unity Pro that support applications of level <= n.						

Example:

If the version of installed Unity Pro is 2.3 then n=2.3 and the last version will be 2.2 (n-1=2.2).

If the version of installed Unity Pro is 3.0 then n=3.0 and the last version will be 2.3 (n-1=2.3).

The General Table of the Compared Elements

Compared Elements by Unity Dif

All the elements are not compared by Unity Dif.

The following table describes the compared general elements.

Elements	Compared Elements	Comments
Configuration		
	CPU	
	Cartridge	
	Configuration Rack	* Not all parameters are compared
	Platform Configuration	* Not all parameters are compared
Derivate Data Types	Name Version Comment	
Derivate FB Types		
	Name	
	Version	
	Comment	
	Attribute	
	Input Parameters	
	InOut Parameters	
	Public Local Variables	
	Privates Local Variables	
	FB Program	* Not all parameters are compared
Variables & FB Instances	Name Comment Address Type Name Value	
Motion	Axis	

Elements	Compared Elements	Comments
Communication		
	Modbus	
	Ethernet	The complete table *(see page 90)* is described in the appendix.
	Fipway	For Premium PLC only.
	Routing tables	For Fipway network only.
Program		
	Task	* Not all parameters are compared
	Execution order	
	FBD section/SR section	
	ST section/SR section	
	IL section/SR section	
	SFC section/SR section	* Not all parameters are compared
	LD section/SR section	
Animation Tables		
	Name	
	Version	
	Comment	
	Element Description	
	Attributes	* Not all parameters are compared
Operator Screen	Display Screen Number Full Screen Variable Display Variable Control	
Project Setting	Editors Language Extension Build	

NOTE: The not compared parameters are described in the appendix *(see page 88)*.

Installation and Registration

2

Subject of this Chapter

This chapter describes the steps for installation and registration of Unity Dif.

What's in this Chapter?

This chapter contains the following topics:

Installing Unity Dif

Installation of Unity Dif

Unity Dif setup checks the presence and version of Unity Pro. If the version is not compatible with Unity Dif, installation of the correct version is requested and setup aborts.

How to Install Unity Dif

This procedure describes how to install Unity Dif.

Step	Action
1	Insert the Unity Dif CD-ROM. **Result:** The **Select language** of installation screen appears. Note: ● If the Unity Pro version is not available with the Unity Dif version, the installation exits with a dialog box. ● If the Operating System is incorrect, the installation aborts.
2	Select the language installation and click **Ok**. **Result:** InstallShield Wizard starts and the installation startup screen appears.
3	Click **Next ...**.
4	Deselect the **ReadMe** file and the **Release Notes** if you won't read them, and click **Next ...**.
5	After reading the license agreement, check **I accept the terms of the license agreement** and click **Next ...**. **Note:** You can print the license agreement with the **Print** button.
6	Enter the following information and click **Next ...**: ● first name, ● last name, ● company, ● part number (mandatory). For the part numbers, please refer to the box label. **Note:** Check **Only for Me** if you do not want to install Unity Dif for other users.
7	Select the folder where setup will install files and click **Next ...**. **Note:** The default directory is *C:\Program Files\Schneider Electric\UnityDif*.
8	Select the features for the setup to install Click **Next ...**.
9	Select the language you want to install and click **Next ...**

Step	Action
10	Click **Install** to configure and launch the installation. **Result:** When the installation is completed, the exit screen appears.
11	You can select **Create a shortcut on the desktop** and **I would like to launch the Registration Wizard**. **Note:** The first time you install Unity Dif, you are asked to register *(see page 22)*.

NOTE: For Installation and Uninstallation, the program uses a log file to register the setup main operations. This name is *UnityDif.log* and it is located (by default) at *C:\Program Files\Common Files\Schneider Electric Shared*.

Installation of Unity Dif Over an Existing Previous Version

If a previous version of Unity Dif is already installed, Unity Dif setup removes the previous installation (including documentation, samples, tools, etc) before installing the new version.

Unity Dif Registration

At a Glance

When the installation of Unity Dif is completed, you can register Unity Dif with Schneider Electric.

How to Register Unity Dif

The following procedure describes how to register Unity Dif.

Step	Action
1	Launch the registration. **Note:** You may launch the registration: ● at the end of the installation, ● when the reminder appears, or ● by clicking on *UnityDif.exe* in **Help → About Unity Dif... → License Agreement**.
2	In the welcome screen, click **Next...**.
3	In the **Select Task** screen, if you do not have an authorization code, tick off **Ask for an authorization code** and click **Next...**. (If you have the code, skip to step 5).
4	Select the method that suits you best and follow the procedure given in the bottom pane. **Result:** You will obtain your authorization code and return to the select task screen.
5	In the select task screen, check **Enter received authorization code** and click **Next...**.
6	Enter the received authorization code and click **Next...**.
7	Click **Finished**. **Result:** Your version of Unity Dif is registered.

NOTE:

● The registration help is accessible to *C:\Program Files\Schneider Electric\Unity Dif\Help*
● For the Registration, the software use a log file for register the main operations of the registration. This name is *RegBrick.log* and it is located (by default) at *C:\Program Files\Schneider Electric\Unity Dif*.

Launch

3

Subject of this Chapter

This chapter describes the main methods to launch Unity Dif.

What's in this Chapter?

This chapter contains the following topics:

Launching Unity Dif from Unity Pro

Caracteristics of this Mode

The following list gives the possibilities for launching Unity Dif from Unity Pro:

- From the Tools menu by **Tools → Project Comparison ...**.
- Compare the open project file with a second project file that can be selected.

NOTE:

- If a project is not open in Unity Pro, the menu entry does not appears.
- The functionalities are the same in Standalone mode and in this case.

Standalone Mode

Caracteristics of this Mode

Unity Dif can be used without Unity Pro in the Standalone mode.

To launch the tool, use the command: **Start** → **All Programs** → **Unity Dif** → **Unity Dif**.

Its characteristics are as follows:

- The Standalone functionalities work with the different Unity Pro files (*XEF* and *STU*).
- It compares 2 *STU* files from the installed version of Unity Pro and 2 *XEF* files from differents installed versions of Unity Pro. Table of Compatibilities *(see page 15)*.
- This mode does not support project files from a different version of Unity Pro.
- The software can display the graphicals differences by opening 2 Unity Pro windows.

NOTE: To display the graphical differences you can click rigth on the section and select **Open in Unity Pro**.

The PC performance can decrease according to the available memory (RAM).

Tool

<div style="text-align: right; font-size: 3em; font-weight: bold;">4</div>
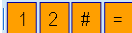
Unity Dif User Interface

Subject of this Chapter

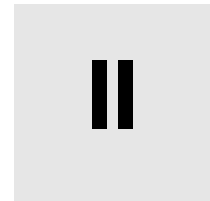This chapter describes the main view of the tool.

GUI presentation

This following image shows the Unity Dif window.

The following table describes the previously screen.

Zone	Description
1	Menu Bar. Description of the menus *(see page 80)*.
2	Toolbar: ● : **Compare** button. ● : **Print** button. ● : **Previous** and **Next** buttons. Allows to see previous and next difference. ● : Allow to select your project browser view (**Project 1 only**, **Project 2 only**, **Differences** and **Identical Parts**). ● : **Generate** button. ● : **Help** button.
3	General Information Browser. Indicates the characteristics of the projects *(see page 41)* or the browser view *(see page 43)*.
4	Information display panel. ● When the General Information Browser is selected, all the differences and the project details *(see page 41)* are indicated in this zone. ● The details of the differences are displayed with a color code (blue for the Project1 and red for the Project2) in the browser view *(see page 43)*.
5	The Status bar: ● In the first part, the status of the application is displayed (Ready, Comparing, Comparison Completed, etc). ● At the bottom, the status bar displays the color and the marker code of both projects: [Prj1] [Prj2] .

Using Unity Dif

II

Subject of this Part

This part describes Unity Dif use.

What's in this Part?

This part contains the following chapters:

Comparing Two Files

5

How to Compare Two Files

Subject of this Chapter

This chapter describes the procedure to compare two files.

Steps for Comparison

The following table describes how to compare 2 Unity Pro files.
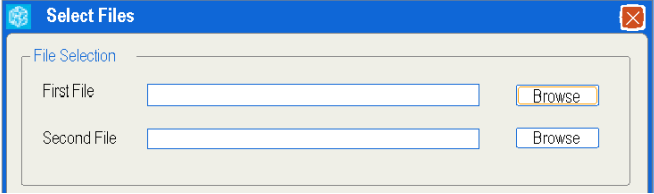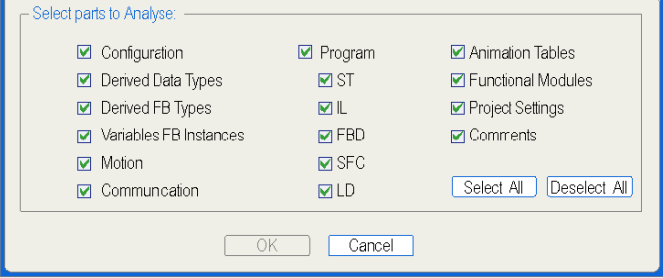
Steps	Actions
1	Click on [icon], or select **File → Compare**.
2	Select the **First file** and the **Second file** with the buttons **Browse ....** Note: • If you compare 2 *STU* files, the comparison time is longer because the software converts the files. • If one of the 2 compared *STU* files is saved on a server, make sure that the server must be accessible in Read/Write mode to be able to perform the comparison.

Steps	Actions
3	Choose the elements you want to compare parts to analyse *(see page 38)*. **Note:** When one program section (ST, IL, FBD, SFC , or LD) is checked, the program structure is compared.
4	Click **OK ...**. **Result:** The progress bar appears. When it is complete, the result appears on the right screen. **Note:** ● If the security editor is on for Unity Pro, the Unity Dif will ask for the username and password of the relevant profile. The user shall be prompted for this information for each *STU* file compared. ● The comparison time depends on the file size. ● You can stop the comparison by using any of the following: ● **Cancel** button, ● ESC button, ● ALT + F4 combination, ● The Close Window icon .
5	Choose your result view: General Information Browser *(see page 41)* or Project Browser *(see page 43)*.
Optional after the comparison	

Steps	Actions
6	If you want access to the program section, select the command with a right-click
7	If you want extract the result, select the command Generate a report *(see page 49)*.

NOTE: During runtime, the software registers the information on the errors and warnings in a log file. The log file is named *UnityDifError.log* and is located at *C:\.*

If the compared STU file has been created with the security editor enabled, the following popup appears before the comparison.

If the compared STU file has been created with password protection, the following popup appears before the comparison.

Description of Unity Dif's Elements

6

Subject of this Chapter

This chapter presents the important elements of Unity Dif.

What's in this Chapter?

This chapter contains the following sections:

Section	Topic	Page
6.1	Information Selected	38
6.2	Results	40
6.3	Reports	48

6.1 Information Selected

Description of the Information

Subject of this Section

This section describes the information that the user wants compare.

Elements to Analyse

The following screen allows to select the files to compare and the parts to analyse

With **Select parts to Analyse**, included in the **Select Files** view, it is possible to select the types of information that you want compare.

The following table describes these types of information.

Information	Description
Configuration	Compare the differents configurations: PLC, FIPIO, RIO Bus and CANopen.
Derived Data Types	Compare the **DDT** type.
Derived FB Types	Compare the **DFB** type.
Variables FB Instances	Compare the **Variables and FB Instances**.
Motion	Compare the **Motion** function block.

Information	Description
Communication	Compare the **Networks** Configuration and **Routing Table**. **Note:** **Routing Table** comparison is used only for Prenium PLC.
Program	If it is selected, all the checkboxes under it are selected. However, you can uncheck specific program sections.
ST	Compare all **ST** sections.
IL	Compare all **IL** sections.
FBD	Compare all **FBD** sections.
SFC	Compare all **SFC** sections.
LD	Compare all **LD** sections.
Animation tables	Compare the **Animation Tables** parts.
Functional Modules	Compare the **Functional Modules** parts.
Project Settings	Compare the **Project Settings** parts.
Comments	Compare the **Comments** of the element properties (Variables and DFB/DDT) parts.

6.2 Results

Subject of this Section

This section describes the results obtained after the comparison and explains the system coordonate of the programming languages.

What's in this Section?

This section contains the following topics:

Topic	Page
General Information Browser	41
Project Browser View	43
Unity Pro and Unity Dif System Coordonate	47

General Information Browser

Description of the General Information View

The following screen represents the first view of the results. It is the **General Information Browser** and is represented by this icon

Zone		Description
1	Project1 and Project2	- The file path, - the date and time of the registration, - the file version, - the product version.
2a and 2b	First and Second file	- The file path, - the project name, - the size, date, time, version of the file, - the product version, - the company name, - the PLC address.
3	Compared Part(s)	Indicates all selected informations and the numbers of elements which are: **Modified**, **Added**, **Deleted** and **Moved**.
4	Non Compared Part(s)	Indicates the elements not compared.

Project Browser View

Description of the Project Browser View

The following screen represents the second view of the results and is represented by this icon 🔖 . It is the **Project Browser**.

Project Browser View

Description of the Project Browser View

The following screen represents the second view of the results and is represented by this icon 🔖 . It is the **Project Browser**.

The following table describes this screen.

Zone	Description
1	It has the same organization as Unity Pro Browser. The following icons are presented in the zone: ● ![1]: for information from Project 1 only, ● ![2]: for information from Project 2 only, ● ![#→] : for information present in the projects but different. This icon is present at the folder level only, ● ![#]: this icon is present at the item level and it is used for information present in both projects but different, ● ![X]: for information not compared, ● ![lock] for DFB information and section information protected, ● ![pin] if the same name is used for 2 different programming languages, this icon appears and the comparison is not performed. **Note:** Right-click on the program section and an option appears to view the section within Unity Pro.

Zone	Description
2	This zone gives the details of all differences It is presented like this: ● the elements of the first project are in blue, ● the elements of the second project are in red, ● the syntax is the same as *Unity Pro*, ● the system coordonate *(see page 47)* is the same in *UnityDif* and *Unity Pro*. Presentation: Description: ● The file path is always on the top of the sheet. ● The compared element state is used only for elements of project 2. ● The possible states are: **ADDED**, **MODIFIED**, **DELETED**, **MOVED**. ● When the state is **ADDED**, only the element of project2 is posted.

Zone	Description
	Note: • After an export of the application downloaded onto the PLC, the format of the variable's initialize value is changed. This value is the same on the application uploaded and the application not uploaded but Unity Dif displays a difference. • For Variables Instance and Function Block Instance, Unity Dif use the same XML tag. There are no diffenrences between the Variable instance and the FBI when the **Section Name** is compared. • When a property of a sub-element is modified the name of the sub-element is shown in the result.

NOTE: If in project 2 a new task is inserted, the new sections created underneath this task can not be compared and hence can not be opened using Using Pro.

Specifications

See the graphical differences of the section.

Step	Action
1	Rigth-click on the program section in the first zone.
2	Select 🔷 Open In Unity Pro
3	2 Unity Pro windows display. In the first, there is the graphical program section of project1. The second window is used for the graphical program section of project2.

Unity Pro and Unity Dif System Coordonate

FDB Section Size

In *Unity Pro*, FBD sections have a grid behind them. A grid unit consists of 10 columns and 10 rows. A grid unit is the smallest possible space between 2 objects in an FBD section.

The FBD programming language is not cell oriented but the objects are still aligned with the grid coordinates.

An FBD section has 300 horizontal grid coordinates (= 30 grid units) and 230 vertical coordinates (= 23 grid units).

In *UnityDif*, the deleted, added, and modified objects are shown as deleted, added, or modified at appropriate COL and ROW which is similar to *Unity Pro*.

LD Section Size

In *Unity Pro*, LD sections have a background grid that divides the section into columns and rows. The LD programming language is cell oriented, only one object can be placed in each cell. LD section can be 11 to 64 columns and 17 to 2 000 lines in size.

In *UnityDif* LD sections, the comparison is done on each row. Each row is saved with a tag "TypeLine" in XEF file. If a new row is added, deleted, or modified than *UnityDif* will display as TypeLine added, deleted, or modified. All the cells of each row are compared.

In *UnityDif*, if the object is of types function blocks and if it is modified, then the difference is given as modified in particular COL and ROW.

SFC Section Size

In *Unity Pro*, SFC sections have a background grid that divides the section into columns and rows. The SFC programming language is cell oriented, only one object can be placed in each cell. SFC sections are 32 columns and 200 rows.

In *UnityDif*, the deleted, added, and modified objects are shown as deleted, added, or modified at appropriate COL and ROW which is similar to *Unity Pro*.

6.3 Reports

Subject of this Section

This section describes reports.

What's in this Section?

This section contains the following topics:

Topic	Page
A Report	49
A Report	50
Validate a Report	52

A Report

What Is a Report?

A report is a text file which contains all informations of the comparison. A report is composed of data obtained with the General Information Browser *(see page 41)* and the Project Browser *(see page 43)*.

Generating a report

Once the comparison completed, a report can be generated by two different ways:
- Using command **Report → Generate**.

- Clicking on [icon].

Once the generation completed, you can enter the name of the *TXT* report file.

NOTE: The report can be printed whithout having generated it. The software prints result in the report format.

A Report

Description of the Report

The first part of the report contains data from the General Information Browser, while the second part of the report contains Project Browser data.

```
[Prj1] <Filename>
[Prj2] <Filename>

The date and time of report generation.
The machine name.
The Windows user name.

First file: <Filename>
Comments, details.
...
Second file: <Filename>
Comments, details.
...
Compared part(s):
...
Non compared part(s):
...
Report in tree views:
...
Compared files:
...
Checksum
```

The following table describes the report above.

Part	Description	References
1	Contains the parameters of project file 1.	These parts are the same as the General Information Browser *(see page 41)*.
2	Contains the parameters of project file 2.	
3	Indicates the compared parts and the numbers of elements which are **MODIFIED**, **ADDED**, **DELETED** and **MOVED**.	
4	Indicates the no compared parts.	
5	Indicates the compared elements on the structured format.	It is the same organisation of Unity Pro Browser.
6	Indicates the compared information with more details. It is the most important part of the document.	This part is the same as the Project Browser *(see page 43)*.
7	The checksum makes it possible to validate the report. It is used like a serial number.	Validate a report *(see page 52)*.

Validate a Report

How to Validate a Report

By validating a report, you can detect if it has been modified by an external tool.

NOTE: To see if the report file has been modified, select **Report** → **Check Status** and choose your file.

Result: If the report file has not been modified, the following popup screen appears with the message **IDENTICAL**. If the report file has been modified, the message is **DIFFERENT**.

NOTE: During runtime, the software registers the information on errors and warnings in the log file. The name as a log file is *UnityDifError.log* and it is located in ****.

Comparing in Other Modes

7

Subject of this Chapter

This chapter describes two other methods (Batch Mode and Server Mode) to launch Unity Dif.

What's in this Chapter?

This chapter contains the following sections:

7.1 Batch Mode

Description of Batch Mode

In batch mode, you can launch the software without an interactive window. The user does not see the program.

It is used for a specific application in different programming languages (VB, VB.Net, C#, etc).

What's in this Section?

This section contains the following topics:

Topic	Page
Batch Mode	55
Command `-c`	56
Command `-cr`	58
Command `-v`	59
Command `-crs`	60

Batch Mode

Launching Batch Mode from the Keyboard

This table describes the procedure to execute the software in batch mode from the keyboard:

Step	Action
1	Click on **Start** → **Run...**.
2	Enter **cmd** in the **Open** label.
3	In the new screen, enter the *UnityDif.exe* file path.

NOTE: By default, *UnityDif.exe* is located under *C:\Program Files\Schneider Electric\Unity Dif*.

Functions

The comparison done by batch mode includes the following list of functionalities:

- Launch the compare function *(see page 56)* to display the results in the application user interface with the **General Information Browser** and **Project Browser**.
- Launch the compare function to generate a report *(see page 58)*.
- Launch the checksum verification *(see page 59)*.
- Launch the compare function for safety applications and generate a report *(see page 60)*.

NOTE: To see the output of the batch mode, please redirect the output of the batch mode execution to a text file of your choice. For example, you can do the following to redirect output to the text file *outputfile.txt*:

```
"UnityDif.exe file path"\UnityDif.exe -c
"XEF path1\file1.XEF" "XEF path2\file2.XEF" >>
"C:\outputfile.txt"
```

In this case, you will see the output of all commands in the file *outputfile.txt*.

Command -c

Generality

This command allows to compare *XEF* or *STU* files and bring up the result in the application user interface.

Description

Syntax:

```
"UnityDif.exe file path"\UnityDif.exe -c
"File path1\file1.XEF" "File path2\file2.XEF"
```

● The following table describes the inputs of the command.

Input	Description
UnityDif.exe file path	Path of the UnityDif.exe folder.
File path1 and File path2	Paths of the *XEF* or *STU* files that you want to compare.
file1.XEF and file2.XEF	Name of the *XEF* or *STU* files that you want to compare.

● Example:

● Output:

After the comparison, the result is displayed in the application user interface with the **General Information Browser** and the **Project Browser**.

NOTE: If the compred STU file has been created with the security editor enabled, the following popup appears before the comparison.

Command **-cr**

Generality

This command allows to compare *XEF* or *STU* files and generate a report in the specified *TXT* file.

Description

Syntax:

```
"UnityDif.exe file path"\UnityDif.exe -cr
"File path1\file1.XEF" "File path2\file2.XEF"
"Report path\report.txt username password"
```

● The following table describes the inputs of the command.

Input	Description
`UnityDif.exe file path`	Path of the UnityDif.exe folder.
`File path1` and `File path2`	Path of the *XEF* or *STU* files that you want to compare.
`file1.XEF` and `file2.XEF`	Name of the *XEF* or *STU* files that you want to compare.
`Report path`	Choose the path of the report file.
`report.txt`	Choose the name of the report file.
`username`	Username of the profile that has access to Unity Pro.
`password`	Password of the profile that has access to Unity Pro.

NOTE: Username and password can be omitted in case they are not required.

● Example without username and password.:

● Output:

After the comparison, the report is generated in the specified *TXT* file.

Command **−v**

Generality

This command allows to validate a report file.

Description

Syntax:

```
"UnityDif.exe file path"\UnityDif.exe -v
"Report path\report.TXT" >> Output path\output.txt
```

● The following table describes the inputs of the command.

Input	Description
UnityDif.exe file path	Path of the UnityDif.exe folder.
Report path	Path of the report file that you want to control.
report.TXT	Name of the report file that you want to control.
Output path	Path of the output file in which the comparison result will be saved.
output.TXT	Name of the output file.

● Example:

● Output:

After the checksum verification, a message box appears with the result: **IDENTICAL**, if the report is validated or **DIFFERENT** in the other case.

Command **-crs**

Generality

This command allows to compare safety applications and generate a report in the specified TXT file.

You can also use this command for comparing any non-safety application (XEF or STU) files.

Description

Syntax:

```
"UnityDif.exe file path"\UnityDif.exe -crs
"File path1\file1.stu" "File path2\file2.stu"
"Report path\report.txt" "username" "userpwd" app1_pwd
app2_pwd
```

● The following table describes the inputs of the command.

Input	Description
UnityDif.exe file path	Path of the *UnityDif.exe* folder.
File path1 and File path2	Paths of the XEF or STU files that you want to compare.
file1.stu and file2.stu	Name of the XEF or STU files that you want to compare.
Report path	Choose the path of the report file.
report.txt	Choose the name of the report file.
username	Username of the profile that has access to Unity Pro.
userpwd	Password of the profile that has access to Unity Pro.
app1_pwd	Password of safety application 1.
app2_pwd	Password of safety application 2.

NOTE: Username and userpwd can be empty in case they are not required.

App1_pwd can be empty if first application is not SAFE / not password protected.

`App2_pwd` can be empty if second application is not SAFE / not password protected.

● Example without username and password:

● Output:

After the comparison, the report is generated in the specified TXT file.

7.2   Server Mode

Description of Server Mode

With Server Mode, it is possible to develop an application that uses the function-alities of Unity Dif.

What's in this Section?

This section contains the following topics:

Server Mode

Functions

The functionalities are the same as used in batch mode. The following list describes them:

- Launch the compare function *(see page 65)* to display the results in the application user interface with the **General Information Browser** and **Project Browser**.
- Launch the compare function to generate a report *(see page 67)*.
- Launch the checksum verification *(see page 69)*.
- Launch the compare function for safety applications *(see page 71)* and generate a report.

The functionalities are the same between the batch mode and the server mode, but a different syntax is used.

Example

The following example, included in the Unity Dif CD-ROM, illustrates the function-alities previously described.

The user interface is shown as below.

Following is a description of the screen.

Button	Description
Browse	Select the *XEF* projects files.
Validate Report	Select the *TXT* file that you want to control.

Button	Description
Compare With UnityDif GUI	Compare both project files. The result will display in Unity Dif.
CompareFilesWithReport	Compare both project files and generate a report in a specified *TXT* file.
Exit	Close the application.

NOTE: It is critical in the Server mode to copy the executable in the same folder as UnityDif.exe (by default, *C:\Program Files\Schneider Electric\Unity Dif*). The *UnityDifClient.exe* should be copied to the same folder as the *UnityDif.exe* for the sample to function correctly.

The server samples that are installed with the software are configured to copy the executable in the same folder as *UnityDif.exe*. However, in the VB and VB.net samples, the location of the executable has to be specified at compile-time. Please sepcify the location as the location of Unity Dif when prompted for the path during compilation.

NOTE: In order to use the included samples in the package, you must use at least the version 6 of Microsoft Visual C++.

When you develop a client, the server resources (*.tlb*) provides a large variety of objects that you can use. If they are not already included in the software you are using, you have to add those resources *(see page 73)*.

Compare with GUI

Generality

This command allows to compare *XEF* or *STU* files and bring up the result in the application user interface.

Description

Syntax:

```
int CompareFilesWithGUI(string strFileName1, string
strFileName2)
```

● The following table describes the inputs of the command.

Input	Description
`strFileName1` and `strFileName2`	*XEF* or *STU* files compared.

● Code return:

This table describes all the error code return by this function.

Integer value	Error code	Description
1	UDS_E_SUCCESS_NO_DIFF	No differences between the 2 files.
2	UDS_E_SUCCESS_WITH_DIFF	Differences between the 2 files.
3	UDS_E_PRODUCT_TRIALPERIOD_OVER	Product trial version over.
4	UDS_E_CONVERT_STU_TO_XEF_FAILED	Conversion between .*STU* and .*XEF* failed.
5	UDS_E_INVALID_FILE	Invalid file.
6	UDS_E_COMPARE_FAILED	Comparison failed.
7	UDS_E_USER_PASSWORD_NOT_VALID	User Name or Password are not valid.

Output example:

After the comparison, the result is displayed in the application user interface with the **General Information Browser** and the **Project Browser**.

NOTE: If the compred STU file has been created with the security editor enabled, the following popup appears before the comparison.

Compare files with Report

Generality

This command allows to compare *XEF* or *STU* files and generate a report in the specified *TXT* file.

Description

Syntax:

```
int CompareFilesWithReport(string strFileName1, string
strFileName2, string strReportFile, string UserName, string
Password)
```

- The following table describes the inputs of the command.

Input	Description
strFileName1 and strFileName2	*XEF* or *STU* files compared.
strReportFile	Name of the report file.
UserName	Username of the profile that has access to Unity Pro.
Password	Password of the profile that has access to Unity Pro.

NOTE: By default the report file is generated under *C:\.*

- Code return:

This table describes all the error code return by this function.

Integer value	Error code	Description
1	UDS_E_SUCCESS_NO_DIFF	No differences between the 2 files.
2	UDS_E_SUCCESS_WITH_DIFF	Differences between the 2 files.
3	UDS_E_PRODUCT_TRIALPERIOD_OVER	Product trial version over.
4	UDS_E_CONVERT_STU_TO_XEF_FAILED	Conversion between *.STU* and *.XEF* failed.
5	UDS_E_INVALID_FILE	Invalid file.
6	UDS_E_COMPARE_FAILED	Comparison failed.
7	UDS_E_USER_PASSWORD_NOT_VALID	User Name or Password are not valid.

NOTE: Username and password should be passed as empty in case they are not required.

The example included in the cd assumes that the security editor is not enabled for Unity Pro. If the sample has to be used with security editor, then the username and the password have to be added in the code.

Output example:

After the comparison,the report is generated in the specified *TXT* file.

Validate a report

Generality

This command allows to validate a report file.

Description

Syntax:

```
int ValidateReport(string strFileName)
```

● The following table describes the inputs of the command.

Input	Description
strFileName	Report file that you want to control.

● Code return:

This table describes all the error code return by this function.

Integer value	Error code	Description
8	UDS_E_FILE_NOT_FOUND	Report file not found.
9	UDS_E_FILE_DIFFERENT	Report file modified.
10	UDS_E_FILE_IDENTICAL	Report file identical.

Output example:

After the checksum verification, a message box appears with the result.

If a report file has not changed, the following message box is displayed.

If a report file has changed or if the file is not a report, the following message box is displayed.

Compare Files with Report for Safety

Generality

This command allows to compare 2 safety applications and to generate a report in the specified TXT file.

Description

Syntax:

```
int CompareFilesWithReportForSafety(string strFileName1,
string strFileName2, string strReportFile, string UserName,
string UserPwd, string AppPwd1, string AppPwd2)
```

● The following table describes the inputs of the command.

Input	Description
strFileName1 and strFileName2	XEF or STU files compared
strReportFile	name of the report file
UserName	Username of the profile that has access to Unity Pro.
UserPwd	Password of the profile that has access to Unity Pro.
AppPwd1	Password of the safety application1.
AppPwd2	Password of the safety application2.

NOTE: By default the report file is generated under *C:*.

● Code return:

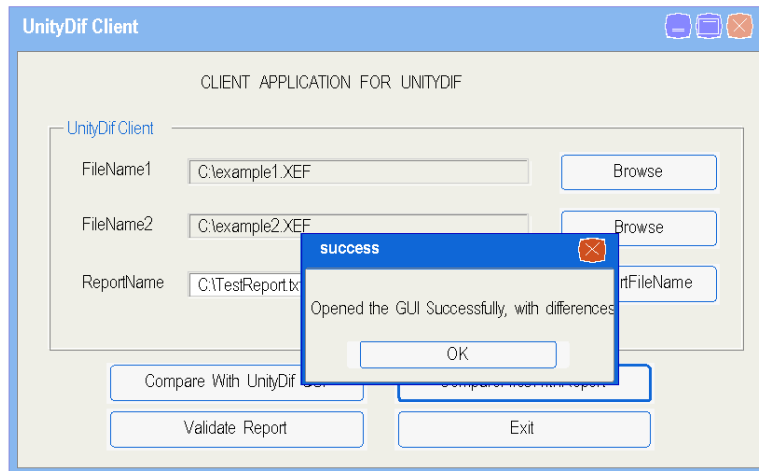This table describes all the error code return by this function.

Integer value	Error code	Description
1	UDS_E_SUCCESS_NO_DIFF	no differences between the 2 files
2	UDS_E_SUCCESS_WITH_DIFF	differences between the 2 files
3	UDS_E_PRODUCT_TRIALPERIOD_OVER	product trial version over
4	UDS_E_CONVERT_STU_TO_XEF_FAILED	conversion between .STU and .XEF failed
5	UDS_E_INVALID_FILE	invalid file
6	UDS_E_COMPARE_FAILED	comparison failed
7	UDS_E_USER_PASSWORD_NOT_VALID	username or password not valid
12	UDS_E_APP_PWD_NOT_VALID	application password not valid

NOTE: `UserName` and `UserPwd` can be empty in case they are not required.

`App1_pwd` can be empty if first application is not SAFE / not password protected.

`App2_pwd` can be empty if second application is not SAFE / not password protected.

Add Reference

How to add reference in a VB project

This procedure describes how to add reference in a VB Project in Excel.

Step	Action
1	In the Visual Basic Editor, click **Tools** → **References...**.
2	Tick off all the UnityDif reference and click **OK**. The same item may be listed twice if an *.exe*, a *.dll* and/or a *.tlb* have the same name. Select the exe (or the *.dll*).
3	If no UnityDif reference are listed, click **Browse...**.
4	Select the *.tlb* files in *C:\Program Files\Common Files\Schneider Electric\Unity Dif* and click **Open**.
5	In the **References** window, tick off the Schneider Electric Unity references and click **OK**.

Note

Depending on the software you use to develop client applications, you have other menus to access the **References** window.

Appendices

Presentation of the Appendices

The appendices contain information about errors, view (the menus and the User Interface) and the selected elements.

What's in this Appendix?

The appendix contains the following chapters:

Errors

A

Management

Subject of this Chapter

This chapter describes the main errors met during a comparison.

Common Use

When you compare a *XEF* file generated with Unity Pro v2.1, Unity Dif does not display the version of the *XEF* file.

Through Graphical Interface

The following table describes the possible errors when using the software in Standalone mode or when it is launched from Unity Pro.

An error message will be displayed in the following cases.

Possible error	Why	When	Possible solution
Installation has not finished.	● PC does not have the required Operating System. ● The version of Unity Pro is lower than 2.1.	During installation *(see page 20)* of Unity Dif.	● Install a more recent version of Unity Pro (>v2.1). ● Change the PC's Operating System.
Problem with a selected element.	The *STU* or the *XEF* file version is incompatible.	When you click on **OK** to launch the comparison.	● The version of the *STU* files is not the same as the installed version of Unity Pro. ● The version of the *XEF* files is not good.

Possible error	Why	When	Possible solution
The report is not generated	● The disk space is limited while generating the project. ● Incorrect format used for a report validation.	During generation.	● Free up enough space on your hard disk. ● The report file's format must be *.TXT*.
The project 1 name does not appear in a first **File Selection**.	The project is not saved	For a new comparison after a comparison from Unity Pro	Save the project before launch the first comparison

In Batch Mode or Server Mode

The following table describes the possible errors when you use the software in Batch Mode or Server Mode.

An error message or an error code will be generated in the following cases.

Mode	Possible errors	Why	Possible solution
Batch Mode	Output is not visible at the command prompt.	Output is not shown at the command prompt but is redirected to the console.	Redirect output to a file.
	After using -C option, the Unity Dif user interface is not shown.	There could be an error in the input files.	Redirect output to a file and check the output file for errors indicated.
	After using -CR option, the report is not created.	There could be an error in the input files.	Redirect output to a file and check the output file for errors indicated.
Server Mode.	Runtime Error while executing client.	The client is not in the Unity Dif root folder (e.g. *C:\Program Files\Schneider Electric\Unity Dif*)	Copy the client executable and related files if any to the Unity Dif root folder.
	Runtime Error while executing client even if files are in the Unity Dif root folder.	The Unity Dif server is not registered properly.	Reinstall the software.

View

B

Subject of this Chapter

This chapter presents some views of menus and the description of the About Box.

What's in this Chapter?

This chapter contains the following topics:

Topic	Page
Menus	80
Presentation of the About Box	82

Menus

Description of the Menus

This part describes all the menus of Unity Dif.

The **File** menu:

1 To compare 2 files.

2 When the comparison is done, it can be printed in report format.

3 To close the application.

The **View** menu:

1 Activate the display of the Tool Bar, the Status Bar or the Browser.

2 Select the view in the project browser.

3 Locate to the next or previous difference in the project browser.

The **Report** menu:

1 To generate a report.

2 To validate a report.

The **Help** menu:

1 Open the index page of the online help file.

2 Search an element with key words.

3 Open the online help file on the content topics.

4 Open the **About** box of Unity Dif.

Presentation of the About Box

Description

About Box contains information about Unity Dif. The following screenshots show the About Box.

NOTE: The About Box is accessible via the command **Help** → **About Unity Dif...**.

The **General** tab:

1 Information about the Unity Dif version, copyright, etc.

The **License** tab:

1 Names of the user and the company.

2 Part number of Unity Dif.

3 Information about the installed options.

4 Information about your license.

NOTE: Unity Dif has no Serial Number.

The **License Agreement** tab:

1 Information about the license. With the buttons, you can define the Page Setup, Print or see a Preview of the License Agreement.

2 Information about the country-specific details of the License. With the buttons, you can define the Page Setup, Print or see a Preview on this informations.

3 If you are not registered, you may do so with this button.

The **Technical Info** tab:

1 Presentation of the DLLs and other technical information about the software.

2 Information on the system.

Elements Compared

C

Subject of this Chapter

This chapter describes the not compared elements and the compared elements of the communication ethernet.

What's in this Chapter?

This chapter contains the following topics:

The General Table of the Not Compared Elements

Not Compared Elements by Unity Dif

The following table describes the not compared elements.

Elements	Compared Elements	Not Compared Elements
Configuration	CPU Cardridge Configuration Rack Quantum Configuration	*Auxiliary Rack Premium* is not compared. *Extension Desc* is not compared.
Derivate Data Types	Name Version Comment	
Derivate FB Types	Name Version Comment Attribute Input Parameters InOut Parameters Public Local Variables Privates Local Variables FB Program	*Version*, *Date Time*, *Activation Control*, and *Logic Control* are not compared.
Variables & FB Instances	Name Comment Address Type Name Value	
Motion	Axis	
Communication	Modbus Ethernet Fipway Routing tables	The complete table *(see page 90)* is in the next table.

Elements	Compared Elements	Not Compared Elements
Program	Task Execution order FBD section/SR section ST section/SR section IL section/SR section LD section/SR section SFC section/SR section	*Functional Module Id* is not compared. *Area Number* and *Operator Control* are not compared.
Animation Tables	Name Version Comment Element Description Attributes	*Extended String Animation*, *Extended String Animation Length*, *Functional Module Id*, and *Date Time* are not compared.
Operator Screen	Display Screen Number Full Screen Variable Display Variable Control	
Project Setting	Editors Language Extension Build	

Compatibity of Ethernet Communication Parameters

Table of compatiilities

The following table describes the compared Ethernet Communication Elements.

Used CPU	Module	Compared Elements
For all CPUs		Name Attributes
Premium	Ethway	IP Configuration Messaging CommonWords SNMP
	ETY Intranet and Intranet CoproP	IP Configuration Messaging IO Scanning Global Data SNMP Adress Server Bandwith

Used CPU	Module	Compared Elements
Quantum	TCPIP10 and TCPIP10100 Factory Cast	IP Configuration
	TCPIP10100	IP Configuration IO Scanning
	TCPIP10100 Regular Connection and TCPIP10100 Extended Connection	IP Configuration Messaging IO Scanning Global Data SNMP Adress Server Bandwith
Modicon M340	*IP Config Messaging IO Scanning Global Data SNMP Adress Server Bandwith*	All parameters are compared. All parameters are compared.

Glossary

A

ASCII

ASCII is the abbreviation of American Standard Code for Information Interchange.

This is an American code (but which has become an international standard) that uses 7 bits to define every alphanumerical character used in English, punctuation symbols, certain graphic characters and other miscellaneous commands.

B

BCD

The Binary Coded Decimal (BCD) format is used to represent decimal numbers between 0 and 9 using a group of four bits (half-byte).

In this format, the four bits used to code the decimal numbers have a range of unused combinations.

Example of BCD coding:
- the number `2450`
- is coded: `0010 0100 0101 0000`

BIT

This is a binary unit for a quantity of information which can represent two distinct values (or statuses): 0 or 1.

BOOL

BOOL is the abbreviation of Boolean type. This is the elementary data item in computing. A BOOL type variable has a value of either: 0 (FALSE) or 1 (TRUE).

A BOOL type word extract bit, for example: %MW10.4.

BYTE

When 8 bits are put together, this is callad a BYTE. A BYTE is either entered in binary, or in base 8.

The BYTE type is coded in an 8 bit format, which, in hexadecimal, ranges from 16#00 to 16#FF

C

CPU

Is the abbreviation of Control Processing Unit.

This is the microprocessor. It is made up of the control unit combined with the arithmetic unit. The aim of the control unit is to extract the instruction to be executed and the data needed to execute this instruction from the central memory, to establish electrical connections in the arithmetic unit and logic, and to run the processing of this data in this unit. We can sometimes find ROM or RAM memories included in the same chip, or even I/O interfaces or buffers.

D

DBCD

Representation of a Double BCD-format double integer.

The Binary Coded Decimal (BCD) format is used to represent decimal numbers between 0 and 9 using a group of four bits.

In this format, the four bits used to code the decimal numbers have a range of unused combinations.

Example of DBCD coding:
- the number 78993016
- is coded: 0111 1000 1001 1001 0011 0000 0001 0110

DDT

DDT is the abbreviation of Derived Data Type.

A derived data type is a set of elements of the same type (ARRAY) or of various types (structure)

DFB

DFB is the abbrevation of Derived Function Block.

DFB types are function blocks that can be programmed by the user ST, IL, LD or FBD.

By using DFB types in an application, it is possible to:

- simplify the design and input of the program,
- increase the legibility of the program,
- facilitate the debugging of the program,
- reduce the volume of the generated code.

DFB instance

A DFB type instance occurs when an instance is called from a language editor.

The instance possesses a name, input/output interfaces, the public and private variables are duplicated (one duplication per instance, the code is not duplicated).

A DFB type can have several instances.

DINT

DINT is the abbrevation of Double Integer format (coded on 32 bits).

The lower and upper limits are as follows: -(2 to the power of 31) to (2 to the power of 31) - 1.

Example:

-2147483648, 2147483647, 16#FFFFFFFF.

Documentation

Contains all the information of the project. The documentation is printed once compiled and used for maintenance purposes.

The information contained in the documentation cover:

● the hardware and software configuration,

● the program,

● the DFB types,

● the variables and animation tables,

● the cross-references.

● ...

When building the documentation file, you can include all or some of these items.

DWORD

DWORD is the abbreviation of Double Word.

The DWORD type is coded in 32 bit format.

This table shows the lower/upper limits of the bases which can be used:

Base	Lower limit	Upper limit
Hexadecimal	16#0	16#FFFFFFFF
Octal	8#0	8#37777777777
Binary	2#0	2#11111111111111111111111111111111

Representation examples:

Data content	Representation in one of the bases
00000000000010101101110011011110	16#ADCDE
00000000000000010000000000000000	8#200000
00000000000010101011110011011110	2#10101011110011011110

E

EBOOL

EBOOL is the abbrevation of Extended Boolean type. A EBOOL type variable brings a value (0 (FALSE) or 1 (TRUE) but also rising or falling edges and forcing capabilities.

An EBOOL type variable takes up one byte of memory.

The byte split up into:

- one bit for the value,
- one bit for the history bit (each time the state's object changes, the value is copied inside the history bit),
- one bit for the forcing bit (equals to 0 if the object isn't forced, equal to 1 if the bit is forced.

The default type value of each bit is 0 (FALSE).

EDT

EDT is the abbreviation of Elementary Data Type.

These types are as follows:
- BOOL,
- EBOOL,
- WORD,
- DWORD,
- INT,
- DINT,
- UINT,
- UDINT,
- REAL,
- DATE,
- TOD,
- DT.

EF

Is the abbreviation of Elemantary Function.

This is a block which is used in a program, and which performs a predefined software function.

A function has no internal status information. Multiple invocations of the same function using the same input parameters always supply the same output values. Details of the graphic form of the function invocation can be found in the "[Functional block (instance)] ". In contrast to the invocation of the function blocks, function invocations only have a single unnamed output, whose name is the same as the function. In FBD each invocation is denoted by a unique [number] via the graphic block, this number is automatically generated and can not be altered.

You position and set up these functions in your program in order to carry out your application.

You can also develop other functions using the SDKC development kit.

EFB

Is the abbreviation for Elementary Function Block.

This is a block which is used in a program, and which performs a predefined software function.

EFBs have internal statuses and parameters. Even where the inputs are identical, the output values may be different. For example, a counter has an output which indicates that the preselection value has been reached. This output is set to 1 when the current value is equal to the preselection value.

Elemantary Function

see EF

F

FBD

FBD is the abbreviation of Function Block Diagram.

FBD is a graphic programming language that operates as a logic diagram. In addition to the simple logic blocks (AND, OR, etc.), each function or function block of the program is represented using this graphic form. For each block, the inputs are located to the left and the outputs to the right. The outputs of the blocks can be linked to the inputs of other blocks to form complex expressions.

FFB

Collective term for EF (Elementary Function), EFB (Elementary Function Block) and DFB (Derived Function block)

Flash Eprom

PCMCIA memory card containing the program and constants of the application.

Function block

see EFB

Functional Module

A functional module is a group of program elements (sections, sub-programs, macro steps, animation tables, runtime screen, etc.) whose purpose is to perform an automation device function.

A functional module may itself be separated into lower-level functional modules, which perform one or more sub-functions of the main function of the automation device.

H

Hyperlink

The hyperlink function enables links to be created between your project and external documents. You can create hyperlinks in all the elements of the project directory, in the variables, in the processing screen objects, etc.

The external documents can be web pages, files (xls, pdf, wav, mp3, jpg, gif, etc.).

I

I/O Object

An I/O object is an implicit or explicit language object for an expert function module or a I/O device on a fieldbus. They are of the following types: %Ch, %I, %IW, %ID, %IF, %Q, %QW, % QD, QF, %KW, %KD, %KF, %MW, %MD, and %MF.

The objects' topological address depends on the module's position on the rack or the device's position on the bus.

For Premium/Atrium PLCs double-type instances of located data (`%MD<i>`, `%KD<i>`) or floating (`%MF<i>`, `%KF<i>`) should be located by an integer type (`%MW<i>`, `%KW<i>`). Only I/O objects make it possible to locate type instances (`%MD<i>`, `%KD<i>`, `%QD`, `%ID`, `%MF<i>`, `%KF<i>`, `%QF`, `%IF`) by using their topological address (for example `%MD0.6.0.11`, `%MF0.6.0.31`).

For Modicon M340 PLCs, double-type instances of located data (`%MD<i>`, `%KD<i>`) or floating (`%MF<i>`, `%KF<i>`) are not available.

IL

IL is the abbreviation of Instruction List.

This language is a series of basic instructions.

This language is very close to the assembly language used to program processors.

Each instruction is composed of an instruction code and an operand.

INT

`INT` is the abbreviation of single integer format (coded on 16 bits).

The lower and upper limits are as follows: -(2 to the power of 31) to (2 to the power of 31) - 1.

Example:

`-32768, 32767, 2#1111110001001001, 16#9FA4`.

IODDT

IODDT is the abbreviation of Input/Output Derived Data Type.

The term IODDT designates a structured data type representing a module or a channel of a PLC module. Each application expert module possesses its own IODDTs.

K

Keyword

A keyword is a unique combination of characters used as a syntactical programming language element (See annex B definition of the IEC standard 61131-3. All the key words used in Unity Proand of this standard are listed in annex C of the IEC standard 61131-3. These keywords cannot be used as identifiers in your program (names of variables, sections, DFB types, etc.)).

L

LD

LD is the abbreviation of Ladder Diagram.

LD is a programming language, representing the instructions to be carried out in the form of graphic diagrams very close to a schematic electrical diagram (contacts, coils, etc.).

N

Network

Mainly used in communication, a network is a group of stations which communicate among one another. The term network is also used to define a group of interconnected graphic elements. This group forms then a part of a program which may be composed of a group of networks.

O

Operator screen

This is an editor that is integrated into Unity Pro, which is used to facilitate the operation of an automated process. The user regulates and monitors the operation of the installation, and, in the event of any problems, can act quickly and simply.

P

Protection

Option preventing the contents of a program element to be read (read protected), or to write or modify the contents of a program element (read/write protected).

The protection is confirmed by a password.

R

RUN

Function enabling the startup of the application program of the PLC.

RUN Auto

Function enabling the execution of the PLC application program to be started automatically in the case of a cold start.

S

Section

Program module belonging to a task which can be written in the language chosen by the programmer (FBD, LD, ST, IL, or SFC).

A task can be composed of several sections, the order of execution of the sections corresponding to the order in which they are created, and being modifiable.

SFC

SFC is the abbreviation of Sequential Function Chart.

SFC enables the operation of a sequential automation device to be represented graphically and in a structured manner. This graphic description of the sequential behavior of an automation device, and the various situations which result from it, is performed using simple graphic symbols.

SFC objects

An SFC object is a data structure representing the status properties of an action or transition of a sequential chart.

ST

ST is the abbreviation of Structured Text language.

Structured Text language is an elaborated language close to computer programming languages. It enables you to structure series of instructions.

STRING

A variable of the type STRING is an ASCII standard character string. A character string has a maximum length of 65534 characters.

Structure

View in the project navigator with represents the project structure.

Subroutine

Program module belonging to a task (Mast, Fast, Aux) which can be written in the language chosen by the programmer (FBD, LD, ST, or IL).

A subroutine may only be called by a section or by another subroutine belonging to the task in which it is declared.

T

Task

A group of sections and subroutines, executed cyclically or periodically for the MAST task, or periodically for the FAST task.

A task possesses a level of priority and is linked to inputs and outputs of the PLC. These I/O are refreshed in consequence.

TIME

The type TIME expresses a duration in milliseconds. Coded in 32 bits, this type makes it possible to obtain periods from 0 to (2 to the power of 32)-1 milliseconds.

Time literals

The units of type TIME are the following: the days (d), the hours (h), the minutes (m), the seconds (s) and the milliseconds (ms). A literal value of the type TIME is represented by a combination of previous types preceded by T#, t#, TIME# or time#.

Examples: T#25h15m, t#14.7S, TIME#5d10h23m45s3ms

Time Out

In communication projects, The Time out is a delay after which the communication is stopped if there is no answer of the target device.

U

UDINT

UDINT is the abbreviation of Unsigned Double Integer format (coded on 32 bits) unsigned. The lower and upper limits are as follows: 0 to (2 to the power of 32) - 1.

Example:

`0,4294967295,2#11111111111111111111111111111111,8#37777777777, 16#FFFFFFFF.`

UINT

UINT is the abbreviation of Unsigned integer format (coded on 16 bits). The lower and upper limits are as follows: 0 to (2 to the power of 16) - 1.

Example:

`0, 65535, 2#1111111111111111, 8#177777, 16#FFFF.`

Unlocated variable

An unlocated variable is a variable for which it is impossible to know its position in the PLC memory. A variable which have no address assigned is said to be unlocated.

V

Variable

Memory entity of the type BOOL, WORD, DWORD, etc., whose contents can be modified by the program during execution.

Visualization window

This window, also called a watch window, displays the variables that cannot be animated in the language editors. Only those variables that are visible at a given time in the editor are displayed.

W

WORD

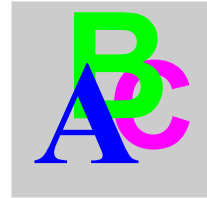The `WORD` type is coded in 16 bit format and is used to carry out processing on bit strings.

This table shows the lower/upper limits of the bases which can be used:

Base	Lower limit	Upper limit
Hexadecimal	16#0	16#FFFF
Octal	8#0	8#177777
Binary	2#0	2#1111111111111111

Representation examples

Data content	Representation in one of the bases
0000000011010011	16#D3
1010101010101010	8#125252
0000000011010011	2#11010011

Index

T

Tool
   User interface, *27*

V

View
   About Box, *82*
   Menus, *80*