Applicable Product:

- **Range-free Multi-controller FA-M3**
  - Model Name: SF630-MCW
  - FA-M3 Programming Tool WideField3

The document number and document model code for this manual are given below. Refer to the document number in all communications; also refer to the document number or the document model code when purchasing additional copies of this manual.

- Document No.: IM 34M06Q16-02E
- Document Model Code: DOCIM
Important

About This Manual

- This Manual should be passed on to the end user.
- Before using the controller, read this manual thoroughly to have a clear understanding of the controller.
- This manual explains the functions of this product, but there is no guarantee that they will suit the particular purpose of the user.
- Under absolutely no circumstances may the contents of this manual be transcribed or copied, in part or in whole, without permission.
- The contents of this manual are subject to change without prior notice.
- Every effort has been made to ensure accuracy in the preparation of this manual. However, should any errors or omissions come to the attention of the user, please contact the nearest Yokogawa Electric representative or sales office.

Symbols Related to Safety

Danger. This symbol on the product indicates that the operator must follow the instructions laid out in this user's manual to avoid the risk of personnel injuries, fatalities, or damage to the instrument. Where indicated by this symbol, the manual describes what special care the operator must exercise to prevent electrical shock or other dangers that may result in injury or the loss of life.

Protective Ground Terminal. Before using the instrument, be sure to ground this terminal.

Function Ground Terminal. Before using the instrument, be sure to ground this terminal.

Alternating current. Indicates alternating current.

Direct current. Indicates direct current.
The following symbols are used only in the user's manual.

**WARNING**

Indicates a “Warning”.
Draws attention to information essential to prevent hardware damage, software damage or system failure.

**CAUTION**

Indicates a “Caution”.
Draws attention to information essential to the understanding of operation and functions.

**TIP**

Indicates a “TIP”.
Gives information that complements the present topic.

**SEE ALSO**

Indicates a “SEE ALSO” reference.
Identifies a source to which to refer.

### Safety Precautions when Using/Maintaining the Product

- For the protection and safe use of the product and the system controlled by it, be sure to follow the instructions and precautions on safety stated in this manual whenever handling the product. Take special note that if you handle the product in a manner other than prescribed in these instructions, the protection feature of the product may be damaged or impaired. In such cases, Yokogawa cannot guarantee the quality, performance, function and safety of the product.

- When installing protection and/or safety circuits such as lightning protection devices and equipment for the product and control system as well as designing or installing separate protection and/or safety circuits for fool-proof design and fail-safe design of processes and lines using the product and the system controlled by it, the user should implement it using devices and equipment, additional to this product.

- If component parts or consumable are to be replaced, be sure to use parts specified by the company.

- This product is not designed or manufactured to be used in critical applications which directly affect or threaten human lives and safety — such as nuclear power equipment, devices using radioactivity, railway facilities, aviation equipment, shipboard equipment, aviation facilities or medical equipment. If so used, it is the user’s responsibility to include in the system additional equipment and devices that ensure personnel safety.

- Do not attempt to modify the product.

- In order to prevent electrical shock, turn off all the power sources before connecting wires, etc.

- This product is classified as Class A for use in industrial environments. If used in a residential environment, it may cause electromagnetic interference (EMI). In such situations, it is the user’s responsibility to adopt the necessary measures against EMI.
Exemption from Responsibility

- Yokogawa Electric Corporation (hereinafter simply referred to as Yokogawa Electric) makes no warranties regarding the product except those stated in the WARRANTY that is provided separately.
- Yokogawa Electric assumes no liability to any party for any loss or damage, direct or indirect, caused by the use or any unpredictable defect of the product.

Software Supplied by the Company

- Yokogawa Electric makes no other warranties expressed or implied except as provided in its warranty clause for software supplied by the company.
- Use the software with one computer only.
- You must purchase another copy of the software for use with each additional computer.
- Copying the software for any purposes other than backup is strictly prohibited.
- Store the original media that contain the software in a safe place.
- Reverse engineering, such as decompiling of the software, is strictly prohibited.
- Under absolutely no circumstances may the software supplied by Yokogawa Electric be transferred, exchanged, or sublet or leased, in part or as a whole, for use by any third party without prior permission by Yokogawa Electric.
General Requirements for Using the FA-M3 Controller

- **Set the product in a location that fulfills the following requirements:**
  - Where the product will not be exposed to direct sunlight, and where the operating surrounding air temperature is from 0°C to 55°C (32°F to 131°F).
  - Where the relative humidity is from 10 to 90%.
  - For use in Pollution Degree 2 Environment.
  - Where there are no corrosive or flammable gases.
  - Where there will not be exposed to mechanical vibration or shock that exceed specifications.
  - Where there is no chance the product may be exposed to radioactivity.

- **Use the correct types of wire for external wiring:**
  - USE COPPER CONDUCTORS ONLY.
  - Use conductors with temperature ratings greater than 75°C.

- **Securely tighten screws:**
  - Securely tighten module mounting screws and terminal screws to avoid problems such as faulty operation.
  - Tighten terminal block screws with the correct tightening torque. Refer to the hardware user’s manual or the applicable user's manual for the appropriate tightening torque.

- **Securely lock connecting cables:**
  - Securely lock the connectors of cables, and check them thoroughly before turning on the power.

- **Interlock with emergency-stop circuitry using external relays:**
  - Equipment incorporating the FA-M3 controller must be furnished with emergency-stop circuitry that uses external relays. This circuitry should be set up to interlock correctly with controller status (stop/run).

- **Ground for low impedance:**
  - For safety reasons, connect the [FG] grounding terminal to a Japanese Industrial Standards (JIS) Class D (earlier called Class 3) Ground. For compliance to CE Marking, use braided or other wires that can ensure low impedance even at high frequencies for grounding.

*1 Japanese Industrial Standard (JIS) Class D Ground means grounding resistance of 100 Ω max.
Configure and route cables with noise control considerations:
- Perform installation and wiring that segregates system parts that may likely become noise sources and system parts that are susceptible to noise. Segregation can be achieved by measures such as segregating by distance, installing a filter or segregating the grounding system.

Configure for CE Marking Conformance:
- For compliance with CE Marking, perform installation and cable routing according to the description on compliance to CE Marking in the “Hardware Manual”.

We recommend that you stock up on maintenance parts:
- We recommend that you stock up on maintenance parts, including spare modules, in advance.
- Preventive maintenance (replacement of the module) is required for using the module beyond 10 years.

Discharge static electricity before touching the system:
- Because static charge can accumulate in dry conditions, first touch grounded metal to discharge any static electricity before touching the system.

Wipe off dirt with a soft cloth:
- Gently wipe off dirt on the product’s surfaces with a soft cloth.
- If you soak the cloth in water or a neutral detergent, tightly wring it out before wiping the product.
  Letting water enter the module interior can cause malfunctions.
- Do not use volatile solvents such as benzine or paint thinner or chemicals for cleaning, as they may cause deformity, discoloration, or malfunctioning.

Avoid storing the FA-M3 controller in places with high temperature or humidity:
- Since the CPU module has a built-in battery, avoid storage in places with high temperature or humidity.
- Since the service life of the battery is drastically reduced by exposure to high temperatures, take special care (storage surrounding air temperature should be from –20°C to 75°C).
- There is a built-in lithium battery in a CPU module which serves as backup power supply for programs, device information and configuration information. The service life of this battery is more than 10 years in standby mode at room temperature. Take note that the service life of the battery may be shortened when installed or stored at locations of extreme low or high temperatures. Therefore, we recommend that modules with built-in batteries be stored at room temperature.

Always turn off the power before installing or removing modules:
- Failing to turn off the power supply when installing or removing modules, may result in damage.
● Do not touch components in the module:
  - In some modules you can remove the right-side cover and install ROM packs or change switch settings. While doing this, do not touch any components on the printed-circuit board, otherwise components may be damaged and modules may fail to work.

● Do not use unused terminals:
  - Do not connect wires to unused terminals on a terminal block or in a connector. Doing so may adversely affect the functions of the module.

● Use the following power source:
  - Use only power supply module F3PU□□-□□ in FA-M3 Controller for supplying power input for control circuit connection.
  - If using this product as a UL-approved product, for the external power supply, use a limited voltage / current circuit power source or a Class 2 power source.

● Refer to the user's manual before connecting wires:
  - Refer to the hardware user's manual or the applicable user's manual for the external wiring drawing.
  - Refer to “A3.6.5 Connecting Output Devices” in the hardware user’s manual before connecting the wiring for the output signal.
  - Refer to “A3.5.4 Grounding Procedure” in the hardware user’s manual for attaching the grounding wiring.
Waste Electrical and Electronic Equipment

Waste Electrical and Electronic Equipment (WEEE), Directive 2002/96/EC
(This directive is only valid in the EU.)

This product complies with the WEEE Directive (2002/96/EC) marking requirement. The following marking indicates that you must not discard this electrical/electronic product in domestic household waste.

Product Category
With reference to the equipment types in the WEEE directive Annex 1, this product is classified as a “Monitoring and Control instrumentation” product.
Do not dispose in domestic household waste.
When disposing products in the EU, contact your local Yokogawa Europe B. V. office.

How to Discard Batteries

The following description on DIRECTIVE 2006/66/EC (hereinafter referred to as the EU new directive on batteries) is valid only in the European Union.

Some models of this product contain batteries that cannot be removed by the user. Make sure to dispose of the batteries along with the product.

Do not dispose in domestic household waste.
When disposing products in the EU, contact your local Yokogawa Europe B. V. office.

Battery type: Lithium battery

Note: The symbol above means that the battery must be collected separately as specified in Annex II of the EU new directive on batteries.
Introduction

About This Manual

The WideField3 manual set consists of the following four volumes.

Table 1 Manual List

<table>
<thead>
<tr>
<th>Volume</th>
<th>Document No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction and Troubleshooting</td>
<td>IM 34M06Q16-01E</td>
</tr>
<tr>
<td>Offline</td>
<td>IM 34M06Q16-02E</td>
</tr>
<tr>
<td>Online</td>
<td>IM 34M06Q16-03E</td>
</tr>
<tr>
<td>Script</td>
<td>IM 34M06Q16-04E</td>
</tr>
</tbody>
</table>

This manual is the operation manual, Offline, for the Range-free Multi-controller FA-M3 Programming Tool (known as WideField3 in this manual).

Overview of This Manual

This manual describes operations of WideField3 when its connection status is offline.

For enquiries, please contact the store where you purchased the product or the nearest Yokogawa sales office listed at the back of this manual.

We recommend using this manual together with the operation manuals of your computer or printer, as required.

Structure of the Manual

This manual consists of 4 parts: D, E, F and G.

Part D describes various settings required to use WideField3, as well as how to build and manage projects and programs that are created.

Part E describes how to edit ladder programs using WideField3.

Part F describes how to program utilizing object ladders in WideField3.

PART D Setting Manual

D1. Initial Setup of WideField3

Describes the initial setup of WideField3.

D2. Overview of Projects

Gives an overview of a project and describes how to create a project as well as how to open executable programs created in older formats such as CADM3 and WideField2.

D3. Building and Managing a Project

Describes how to build and manage a project.

D4. Creating and Managing Blocks and Macros

Describes how to handle block and macro files that store ladder programs.
PART E  Ladder Program Editing Manual

E1. Editing Programs
Describes how to edit ladder programs.

E2. Tag Name Definition
Describes the function to assign addresses and I/O comments to tag names.

E3. Constant Definition
Describes the function to assign constant values and comments to constant names.

E4. Find and Replace
Describes find and replace functions.

E5. Device List
Describes the device list function.

E6. Cross Reference
Describes the cross reference function.

E7. Comparing Files
Describes the functions for comparing project data for checking, and to display, compare and edit programs.

E8. Printing
Describes how to print programs created with WideField3 and information displayed in windows.

PART F  Object Ladder Editing Manual

F1. Using Macros
Describes how to create and use macros.

F2. Using Local Devices
Describes how to set up and use local devices and how to program with them.

F3. Using Group Tag Names
Describes the functions to group tag names into a single group tag name and use it like a data structure.

F4. Structures
Describes WideField3 structures.
How to Read This Manual

Be sure to read the “Introduction” as well as “How to read this manual” before using WideField3.

Part D of the manual describes how to set up WideField3.

Part E describes the function to edit ladder programs.

Part F describes how to program using object ladders.

This manual is structured so that each chapter or section can be read independently for details on the basic specifications and individual functions of the application.

We have tried to make the user interface, operations and editing functions of the WideField3 application as similar as possible to other generally available Windows software. This manual does not contain information on general Windows editing operations, which are not specific to WideField3.

Notation

- Notation for Windows Screens and Operation
  - Items in initial caps denote symbols, names and window names.
    Example: WideField3, Program Monitor dialog
  - Bracketed items denote menu bar items, dialog box fields, commands, and buttons.
    Example: Select [File]–[New] from the menu bar.
    Click [OK].

- Representations in WideField3 Figures and Screens
  Screen examples given in this manual assumes that the application is running under Windows XP operating system environment. Under Windows 2000, Windows Vista and Windows 7 operating systems, you may observe slight differences such as differences in icon names or application names.
  Some figures in this manual may, for reasons of convenience, be emphasized or simplified, or parts of it may be omitted. Some screen images in this manual may differ from actual screens due to differences in the operating machine environment.

- Notation for Procedures
  Procedure pages are laid out with the procedure steps on the left and the corresponding screen images on the right.
  Procedure :User actions are displayed in bold.
  Description of the results of user actions is provided after the ⇒ mark.
  Screens :The procedure step(s) corresponding to a screen image is indicated by step numbers below the screen.

- Function Keys and Shortcut Keys
  In addition to using a mouse, you can operate WideField3 menus using function keys and shortcut keys.
  In general, this manual describes operations using a mouse, and does not include equivalent operations using function keys or shortcut keys.
Other User's Manuals

You should read the following user's manuals.

- FA-M3 Programming Tool WideField3 Read Me First (IM 34M06Q16-11E)
- FA-M3 Programming Tool WideField3 Introduction and Troubleshooting (IM 34M06Q16-01E)
- FA-M3 Programming Tool WideField3 Online (IM 34M06Q16-03E)
- FA-M3 Programming Tool WideField3 Script (IM 34M06Q16-04E)

For individual sequence CPU modules, please refer to the relevant user's manuals.

- **F3SP71, 76**
  - Sequence CPU Instruction Manual – Functions (for F3SP71-4N/4S, F3SP76-7N/7S) (IM 34M06P15-01E)
  - Sequence CPU – Network Functions (for F3SP71-4N/4S, F3SP76-7N/7S) (IM 34M06P15-02E)
  - Sequence CPU Instruction Manual – Instructions (IM 34M06P12-03E)

- **F3SP66, 67**
  - Sequence CPU – Functions (for F3SP66-4S, F3SP67-6S) (IM 34M06P14-01E)
  - Sequence CPU – Network Functions (for F3SP66-4S, F3SP67-6S) (IM 34M06P14-02E)
  - Sequence CPU Instruction Manual – Instructions (IM 34M06P12-03E)

- **F3SP22, 28, 38, 53, 58, 59**
  - Sequence CPU Instruction Manual – Functions (for F3SP22-0S, F3SP28-3N/3S, F3SP38-6N/6S, F3SP53-4H/4S, F3SP58-6H/6S, F3SP59-7S) (IM 34M06P13-01E)
  - Sequence CPU Instruction Manual – Instructions (IM 34M06P12-03E)

- **F3SP05, 08, 21, 25, 35**
  - Sequence CPU – Functions (for F3SP21, F3SP25 and F3SP35) (IM 34M06P12-02E)
  - Sequence CPU Instruction Manual – Instructions (IM 34M06P12-03E)
Refer to the following manuals as required.

- **Specifications and Layout**\(^1\) of the FA-M3, Mounting and Wiring, Testing, Maintenance and Inspection, and System-wide Restrictions for Mounting Modules
  
  \(^1\): See specific manuals for products other than the power module, base module, I/O module, cables, and terminal block units.
  
  - Hardware Manual (IM 34M06C11-01E)

- **Fiber-optic FA-Bus Functions**
  
  - Fiber-optic FA-bus Module and Fiber-optic FA-bus Type 2 Module, FA-bus Type 2 Module (IM 34M06H45-01E)

- **FA Link Functions**
  
  - FA Link H Module, Fiber-optic FA Link H Module (IM 34M06H43-01E)
Copyrights and Trademarks

■ Copyrights

Copyrights of the programs and online manual included in this CD-ROM belong to Yokogawa Electric Corporation.

This online manual may be printed but PDF security settings have been made to prevent alteration of its contents.

This online manual may only be printed and used for the sole purpose of operating this product. When using a printed copy of the online manual, pay attention to possible inconsistencies with the latest version of the online manual. Ensure that the edition agrees with the latest CD-ROM version.

Copying, passing, selling or distribution (including transferring over computer networks) of the contents of the online manual, in part or in whole, to any third party, is strictly prohibited. Registering or recording onto videotapes and other media is also prohibited without expressed permission of Yokogawa Electric Corporation.

■ Trademarks

- The trade and company names that are referred to in this document are either trademarks or registered trademarks of their respective companies.
CONTENTS

Applicable Product: ................................................................. i
Important ........................................................................... ii
Introduction ......................................................................... iii
Copyrights and Trademarks .............................................. xiv

PART-D Setting Manual

D1. Initial Setup of WideField3 .................................................. D1-1
  D1.1 WideField3 System Files and Generated Data ................. D1-2
  D1.2 Environment Setup ..................................................... D1-3
    D1.2.1 Basic Operations with the Setup Environment Dialog Box.. D1-4
    D1.2.2 Folder Setup ....................................................... D1-5
    D1.2.3 Communication Setup .......................................... D1-9
    D1.2.4 Circuit Display/Input Setup ................................... D1-15
    D1.2.5 Program Monitor Setup ........................................ D1-20
    D1.2.6 Program Syntax Check Setup ............................... D1-23
    D1.2.7 Toolbar Setup ................................................... D1-25
    D1.2.8 Email Setup ....................................................... D1-30
    D1.2.9 File Comparison Setup ......................................... D1-32
    D1.2.10 Restoring the Default Settings of All Items in the  
      Environment Setup .................................................. D1-33
    D1.2.11 Exporting the Environment Setup ......................... D1-34
    D1.2.12 Importing the Environment Setup ....................... D1-35

D1.3 Language Setup ............................................................. D1-36

D1.4 Key Customization .......................................................... D1-37
  D1.4.1 Key Customization Setup .................................... D1-37
  D1.4.2 Default Setup ...................................................... D1-38
  D1.4.3 Clear All .............................................................. D1-38
  D1.4.4 Exporting Key Customizations ............................... D1-38
  D1.4.5 Importing Key Customizations .............................. D1-39

D2. Overview of Projects ............................................................ D2-1
  D2.1 What Is a Project? ..................................................... D2-3
    D2.1.1 Relationship between Projects and CPUs ................ D2-3
    D2.1.2 Configuration of a Project .................................. D2-3
    D2.1.3 Managing Projects in WideField3 ......................... D2-5
    D2.1.4 Limitations When a Project is Closed ................. D2-6
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>D2.2</td>
<td>Creating a Project</td>
<td>D2-7</td>
</tr>
<tr>
<td>D2.2.1</td>
<td>Creating a New Project</td>
<td>D2-7</td>
</tr>
<tr>
<td>D2.2.2</td>
<td>Opening a Project</td>
<td>D2-10</td>
</tr>
<tr>
<td>D2.2.3</td>
<td>Project Edit History</td>
<td>D2-13</td>
</tr>
<tr>
<td>D2.2.4</td>
<td>Closing a Project</td>
<td>D2-14</td>
</tr>
<tr>
<td>D2.2.5</td>
<td>Saving a Project</td>
<td>D2-15</td>
</tr>
<tr>
<td>D2.2.6</td>
<td>Saving a Project with a Different Name</td>
<td>D2-16</td>
</tr>
<tr>
<td>D2.2.7</td>
<td>Opening Projects in Other Formats</td>
<td>D2-17</td>
</tr>
<tr>
<td>D2.2.8</td>
<td>Saving Projects in Other Formats</td>
<td>D2-22</td>
</tr>
<tr>
<td>D2.2.9</td>
<td>Saving the Current Screen States</td>
<td>D2-30</td>
</tr>
<tr>
<td>D2.2.10</td>
<td>Opening a Project (and Restoring the Screen States)</td>
<td>D2-30</td>
</tr>
<tr>
<td>D2.2.11</td>
<td>Redisplaying the Current Project with Saved Screen States</td>
<td>D2-31</td>
</tr>
<tr>
<td>D3.1</td>
<td>Building a Project</td>
<td>D3-3</td>
</tr>
<tr>
<td>D3.1.1</td>
<td>Project Settings</td>
<td>D3-5</td>
</tr>
<tr>
<td>D3.1.2</td>
<td>CPU Type Settings</td>
<td>D3-7</td>
</tr>
<tr>
<td>D3.1.3</td>
<td>Executable Program Setup</td>
<td>D3-9</td>
</tr>
<tr>
<td>D3.1.4</td>
<td>CPU Properties</td>
<td>D3-29</td>
</tr>
<tr>
<td>D3.1.5</td>
<td>User Log Message</td>
<td>D3-30</td>
</tr>
<tr>
<td>D3.1.6</td>
<td>Run Operation Setup</td>
<td>D3-31</td>
</tr>
<tr>
<td>D3.1.7</td>
<td>Input/Output Setup</td>
<td>D3-32</td>
</tr>
<tr>
<td>D3.1.8</td>
<td>Device Setup</td>
<td>D3-35</td>
</tr>
<tr>
<td>D3.1.9</td>
<td>Interrupt Setup</td>
<td>D3-38</td>
</tr>
<tr>
<td>D3.1.10</td>
<td>Built-in Functions Setup</td>
<td>D3-39</td>
</tr>
<tr>
<td>D3.1.11</td>
<td>Error Handling Setup</td>
<td>D3-41</td>
</tr>
<tr>
<td>D3.1.12</td>
<td>Initial Data Setup</td>
<td>D3-42</td>
</tr>
<tr>
<td>D3.1.13</td>
<td>Inter-CPU Shared Memory Setup</td>
<td>D3-44</td>
</tr>
<tr>
<td>D3.1.14</td>
<td>FA Link Setup</td>
<td>D3-49</td>
</tr>
<tr>
<td>D3.1.15</td>
<td>Sampling Trace Setup</td>
<td>D3-55</td>
</tr>
<tr>
<td>D3.1.16</td>
<td>Component Block Setup</td>
<td>D3-57</td>
</tr>
<tr>
<td>D3.2</td>
<td>Managing a Project</td>
<td>D3-61</td>
</tr>
<tr>
<td>D3.2.1</td>
<td>Checking Programs</td>
<td>D3-62</td>
</tr>
<tr>
<td>D3.2.2</td>
<td>Finding in Project</td>
<td>D3-66</td>
</tr>
<tr>
<td>D3.2.3</td>
<td>Finding Instruction in Project</td>
<td>D3-66</td>
</tr>
<tr>
<td>D3.2.4</td>
<td>Replacing in Project</td>
<td>D3-66</td>
</tr>
<tr>
<td>D3.2.5</td>
<td>Changing an Address Allocated to a Tag Name over an Entire Project</td>
<td>D3-67</td>
</tr>
<tr>
<td>D3.2.6</td>
<td>Changing a Tag Name Assigned to an Address over an Entire Project</td>
<td>D3-68</td>
</tr>
<tr>
<td>D3.2.7</td>
<td>Changing I/O Installation Position</td>
<td>D3-69</td>
</tr>
<tr>
<td>D3.2.8</td>
<td>Displaying a Device List</td>
<td>D3-72</td>
</tr>
<tr>
<td>D3.2.9</td>
<td>Obsolete Device List</td>
<td>D3-72</td>
</tr>
<tr>
<td>D3.2.10</td>
<td>Finding a Circuit Comment-out in a Project</td>
<td>D3-72</td>
</tr>
</tbody>
</table>
PART-E  Ladder Program Editing Manual

E1.  Editing Programs ................................................................. E1-1

E1.1  Cautions for Editing Ladder Program ................................. E1-2

E1.1.1  Limitations on Editing Ladder Program ............................ E1-2

E1.1.2  Circuits with Conversion Errors ....................................... E1-2

E1.1.3  Circuits that Cannot be Amended Using Ladder Programming .................................................. E1-7

E1.2  Editing Ladder Programs ..................................................... E1-8

E1.2.1  Ladder Program Edit Screen Layout ................................. E1-8

E1.2.2  Set up Circuit Display and Input ....................................... E1-12

E1.2.3  Distinguishing Devices ................................................... E1-12

E1.2.4  Basic Operations .............................................................. E1-13

E1.2.5  Input Using Alphanumeric Keys ....................................... E1-15

E1.2.6  Entering Instructions Using Input Completion Function ....... E1-17

E1.2.7  Entering Instruction Parameters Using Input Completion Function .................................................. E1-18

E1.2.8  Entering Structures Using Input Completion Function ......... E1-19

E1.2.9  Entering Basic Instructions ............................................... E1-21

E1.2.10  Entering Application Instructions ................................. E1-23

E1.2.11  Entering Timer Instruction .............................................. E1-29

E1.2.12  Entering Counter Instructions ......................................... E1-30

E1.2.13  Entering Scripts ............................................................... E1-32

E1.2.14  Deleting Ladder Program Elements ............................... E1-33

E1.2.15  Entering Constants and Constant Names ....................... E1-33

E1.2.16  Entering Character Strings ............................................. E1-35

E1.2.17  Entering Floating Points ................................................ E1-36

E1.2.18  Entering Escape Sequences, Block Names, and Label Names .................................................. E1-37

E1.2.19  Input and Display of Indexed Devices ............................. E1-38

E1.2.20  Input and Display of Devices with a Long-Word Index Modification .................................................. E1-39

E1.2.21  Input and Display of Indirect Specification Devices .......... E1-40

E1.2.22  Input and Display of Tag Names, Structure Names and Structure Member Names ...................... E1-42

E1.2.23  Inserting Connection Lines .............................................. E1-46

E1.2.24  Entering Continuation Circuit .......................................... E1-49

E1.2.25  Entering Labels ............................................................... E1-51

E1.2.26  Inserting Empty Lines ................................................... E1-52

E1.2.27  Selecting Circuits ........................................................... E1-53

E1.2.28  Deleting Circuits ............................................................. E1-56

E1.2.29  Temporarily Deleting Lines and Restoring Temporarily Deleted Lines in Circuits ............................ E1-58

E1.2.30  Copying and Moving Circuits ........................................... E1-59

E1.2.31  Copying and Moving Circuits Including Tag Name Definitions .................................................. E1-62
E1.2.32 Copying Circuit Images ......................................................... E1-65
E1.2.33 Canceling an Operation ...................................................... E1-65
E1.2.34 Overwrite Mode and Insert Mode ....................................... E1-68
E1.2.35 Entering and Deleting Circuit Comments ......................... E1-70
E1.2.36 Creating Hierarchized Circuit Comments (To Be Supported) E1-71
E1.2.37 Entering and Deleting Subcomments .................................. E1-72
E1.2.38 Entering and Deleting I/O Comments ................................. E1-74
E1.2.39 Creating and Deleting Balloon Comments ......................... E1-77
E1.2.40 Index View ....................................................................... E1-85
E1.2.41 Inserting and Deleting Page Breaks ................................. E1-88
E1.2.42 Changing Page Display .................................................. E1-89
E1.2.43 Mnemonic Editing Function .............................................. E1-91
E1.2.44 Function for Commenting out Circuits ............................ E1-93

E1.3 Entering Tag Names and Addresses ...................................... E1-94
  E1.3.1 Enter Tag Name Without Address ..................................... E1-94
  E1.3.2 Enter Tag Name With Address .......................................... E1-94

E1.4 Circuit and Tag Name Definition Reference .......................... E1-95

E1.5 Switching Views .................................................................... E1-96
  E1.5.1 Switching Views for I/O Comments, Tag Names and Addresses, and Instruction Numbers .................................................... E1-96
  E1.5.2 Expanded Display for Instruction Parameter ....................... E1-98
  E1.5.3 Display Instruction Range ............................................... E1-100
  E1.5.4 Setting Multiple Display Modes ...................................... E1-101
  E1.5.5 Zooming Up and Down .................................................. E1-102

E1.6 Converting Circuits and Displaying Errors .......................... E1-103
  E1.6.1 Emphasized Display of Modified Line ............................... E1-103

E1.7 Importing/Exporting Circuit Comments and Subcomments .... E1-104
  E1.7.1 Exporting Circuit Comments and Subcomments .............. E1-104
  E1.7.2 Importing Circuits Comments and Subcomments ............ E1-105

E2. Tag Name Definition ................................................................ E2-1
  E2.1 Fundamentals of Tag Name Definition ................................ E2-2
    E2.1.1 Tag Name Definition Window Layout ................................ E2-2
    E2.1.2 Common Tag Name Definition and Block Tag Name Definition ................................................................. E2-3
    E2.1.3 Restrictions on Tag Name Definitions ............................. E2-5

  E2.2 Editing Tag Name Definition ............................................. E2-6
    E2.2.1 Basic Operations for Editing Tag Name Definition .......... E2-6
    E2.2.2 Save Function for Tag Name Definitions ....................... E2-8
    E2.2.3 Error Handling .......................................................... E2-9
    E2.2.4 Copying, Cutting, Pasting and Deleting ....................... E2-10
    E2.2.5 Sort ........................................................................... E2-12
    E2.2.6 Read Circuits Function ................................................ E2-14
    E2.2.7 Switching Display ..................................................... E2-18
    E2.2.8 Structure Object Definition ........................................ E2-20
E2.2.9 Reading Templates of Common Tag Name Definitions ..........E2-23

E2.3 Merging Tag Name Definitions..........................................................E2-24
E2.3.1 Merging Other Block’s Tag Name Definitions .........................E2-26
E2.3.2 Merging Common Tag Name Definitions..............................E2-28

E2.4 Deleting Unused Tag Names .............................................................E2-29

E2.5 Group Templates .............................................................................E2-31
E2.5.1 Creating Group Templates.....................................................E2-31
E2.5.2 Using Group Templates .........................................................E2-32

E2.6 Changing Maximum Number of Common Tag Name Definitions and Storing to CPU ..............................................E2-33

E2.7 Importing and Exporting Tag Name Definitions .........................................E2-34
E2.7.1 Exporting Tag Name Definitions.............................................E2-34
E2.7.2 Importing Tag Name Definitions.............................................E2-35

E2.8 Multiple I/O Comments ......................................................................E2-36
E2.8.1 Enabling to Display Multiple I/O Comments .........................E2-36
E2.8.2 Assigning Multiple I/O Comments ..........................................E2-38
E2.8.3 Selecting I/O Comments to Be Displayed .............................E2-39
E2.8.4 Saving the I/O Comment Display Settings as Default ..........E2-40

E3. Constant Definition .............................................................................E3-1

E3.1 Fundamentals of Constant Definition ..................................................E3-3
E3.1.1 Constant Definition Window Layout.....................................E3-3
E3.1.2 Restrictions on Constant Definition .....................................E3-4

E3.2 Editing Constant Definition .............................................................E3-5
E3.2.1 Basic Operations for Editing Constant Definition ..................E3-5
E3.2.2 Specification of Elements of Constant Definition .....................E3-7
E3.2.3 Error Handling .........................................................................E3-8
E3.2.4 Copying, Cutting, Pasting and Deleting ...................................E3-9
E3.2.5 Sort ....................................................................................E3-9
E3.2.6 Read Circuits Function ...........................................................E3-10

E3.3 Deleting Unused Constant Names .....................................................E3-12

E3.4 Displaying Number of Constant Devices Used ................................E3-13

E4. Find and Replace ..................................................................................E4-1

E4.1 Find ..............................................................................................E4-2
E4.1.1 Finding Devices/Comments ......................................................E4-2
E4.1.2 Finding Instruction .................................................................E4-12
E4.1.3 Finding the Next Candidate ......................................................E4-18
E4.1.4 Finding Tag Name Definitions ..................................................E4-19
E4.1.5 Finding Constant Definitions ..................................................E4-20
E4.1.6 Jumps ....................................................................................E4-21
E4.1.7 Jump to Tag Name Definition ..................................................E4-23
E4.1.8 Jump to Circuit Comment (To Be Supported) .........................E4-23
E4.1.9 Obsolete Device List ...............................................................E4-24
E4.1.10 Find Circuit Comment-out ......................................................E4-27
E4.2 Replace

E4.2.1 Replacing within the Open Block

E4.2.2 Replacing over an Entire Project

E4.3 Finding and Replacing in Structures during Program Editing

E4.3.1 Finding in Structures

E4.3.2 Replacing in Structures

E4.4 Finding and Replacing in Structures in the Project

E4.4.1 Overview

E4.4.2 Procedure for Finding in Structures in the Project

E4.4.3 Procedure for Replacing in Structures in the Project

E4.5 Simple Find

E4.5.1 Find Device

E4.5.2 Find in Project

E4.5.3 Changing Search Conditions

E5. Device List

E5.1 Fundamentals of Device List

E5.1.1 Display and Layout of the Device List Dialog Box

E5.1.2 Finding a Device in a Device List

E5.1.3 Jumping to Used or Unused Devices

E5.1.4 Restrictions on the Device List Function

E5.2 Creating Cross References in the Device List Dialog Box

E5.3 Exporting a Device List

E5.3.1 Exporting from the Currently Displayed Tab

E5.3.2 Exporting from All Tabs

E6. Cross Reference

E6.1 Fundamentals of Cross References

E6.1.1 Display and Layout of Windows for Cross References

E6.1.2 Creating Cross References

E6.1.3 Restrictions on the Cross Reference Function

E6.2 Creating Cross References Automatically

E6.3 Jumps

E6.4 Sorting Cross References

E6.5 Exporting Cross References

E7. Comparing Files

E7.1 Basic File Comparison Operations

E7.1.1 File Comparison Bar and Shortcut Keys

E7.1.2 Optional File Comparison Settings

E7.2 Compare Project

E7.2.1 Compare Project

E7.2.2 Project Comparison Results List

E7.2.3 Operations on the Project Comparison Results List

E7.3 Compare Window

E7.3.1 Compare Window
E7.3.2 Display Details of the Window Comparison Results..............E7-18
E7.3.3 Operations in the Comparison Results Screen......................E7-22

E8. Printing........................................................................................ E8-1

E8.1 Printer Setup ..................................................................................E8-4
E8.2 Print and Print Preview .................................................................E8-5
  E8.2.1 Print Setup Dialog Box.............................................................E8-5
  E8.2.2 Printing a Screen ....................................................................E8-7
  E8.2.3 Printing a Project.................................................................E8-8
  E8.2.4 Procedure for Previewing Print Image....................................E8-15

E8.3 Page Format Settings .................................................................E8-16
  E8.3.1 Configuring Paper Settings and Margins.............................E8-17
  E8.3.2 Editing a Cover.......................................................................E8-19
  E8.3.3 Editing Headers and Footers .................................................E8-21
  E8.3.4 Configuring Program Output Settings.................................E8-23

E8.4 Print Layout Settings .................................................................E8-25
  E8.4.1 Project Information Settings.................................................E8-26
  E8.4.2 Project Settings/Configuration.............................................E8-27
  E8.4.3 Block/Macro Settings ............................................................E8-30
  E8.4.4 Device List Settings ..............................................................E8-41

E8.5 Print Layout..................................................................................E8-44
  E8.5.1 Print Layout for Project Information.....................................E8-44
  E8.5.2 Print Layout for Project Settings/Configuration.....................E8-45
  E8.5.3 Print Layout for Circuits.......................................................E8-46
  E8.5.4 Print Layout for System Log..................................................E8-47
  E8.5.5 Print Layout for Device List...................................................E8-48
PART-F Object Ladder Editing Manual

F1. Using Macros

F1.1 About Macros

F1.1.1 What are Macros?

F1.1.2 The Purpose of Macros

F1.1.3 Precautions When Using Macros

F1.1.4 Output Macro Call and Input Macro Call Instructions

F1.1.5 Macro Devices and Macro Local Devices

F1.1.6 About Structure Macros

F1.2 Developing Macros

F1.2.1 Initial Setup of Macros

F1.2.2 Creating Macros

F1.2.3 Using Macros

F1.2.4 Registering Macros

F2. Using Local Devices

F2.1 What are Local Devices?

F2.2 Setting Local Devices

F2.2.1 Block Setup for Local Devices

F2.2.2 Project Setup for Local Devices

F2.3 Programming with Local Devices

F2.4 Reusing Programs Containing Local Devices

F3. Using Group Tag Names

F3.1 Defining Group Tag Names

F3.2 Programming Using Group Tag Names

F4. Structures

F4.1 Overview of Structures

F4.1.1 Using Structures in WideField3

F4.1.2 Structure-related Limitations

F4.2 Defining Structure Types

F4.2.1 What Is a Structure Type Definition?

F4.2.2 Creating a New Structure Type Definition

F4.2.3 Changing Structure Type Definitions

F4.2.4 Exiting the Structure Type Definition Window and Saving Changes to File

F4.2.5 Editing Structure Type Definitions

F4.3 Defining Structure Objects

F4.3.1 Overview of Structure Object Definition

F4.3.2 Structure Object Definition Dialog Box

F4.3.3 Editing Structure Object Definitions

F4.3.4 Devices Allocated to Structures

F4.3.5 Tag Name Definition Window Operations
F4.3.6 Checking Structure Devices for Overlaps ............................................. F4-21

F4.4 Programming Using Structures .......................................................... F4-22

F4.4.1 Display of Structure Parameters in Program Editing Windows ................................................. F4-22
F4.4.2 Entering Structures in a Program Editing Windows ............................................. F4-23
F4.4.3 Instructions and Instruction Parameters .................................................. F4-24

F4.5 Structure Macros .................................................................................. F4-25

F4.5.1 Differences Between Structure Macros and Normal Macros .................................................. F4-26
F4.5.2 Setting Argument Types of A Structure Macro ........................................ F4-27
F4.5.3 Structure Macro Programming ........................................................................ F4-28

F4.6 Structure-related Functions for Projects .............................................. F4-30

F4.6.1 Checking for Overlapping Structures .................................................. F4-30
F4.6.2 Finding in Project and Replacing in Project ............................................. F4-31
F4.6.3 Changing I/O Installation Position ....................................................... F4-34
F4.6.4 Structure-related Syntax Check Errors .................................................. F4-35
F4.6.5 Used Devices List/Device List .................................................................... F4-36

Index ........................................................................................................... Index-1

Revision Information ..................................................................................... i
This manual describes initial setup of WideField3, overview of a project, how to build and manage projects, as well as how to create and manage blocks and macros.
D1. Initial Setup of WideField3

This chapter describes the initial setup of WideField3. While you can use WideField3 with its default settings, you may want to register often used icons, change color and input settings, or otherwise customize WideField3 to your specific needs and preferences.

- **Function Limitations for Each CPU Type**

<table>
<thead>
<tr>
<th>CPU Type</th>
<th>Limitations</th>
<th>SEE ALSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>F3SP22-0S, F3SP28-3S, F3SP38-6S, F3SP53-4S, F3SP58-6S, F3SP59-7S</td>
<td>N/A</td>
<td>-</td>
</tr>
<tr>
<td>F3SP66-4S, F3SP67-6S</td>
<td>N/A</td>
<td>-</td>
</tr>
<tr>
<td>F3SP71-4N, F3SP76-7N</td>
<td>N/A</td>
<td>-</td>
</tr>
<tr>
<td>F3SP71-4S, F3SP76-7S</td>
<td>N/A</td>
<td>-</td>
</tr>
</tbody>
</table>

**TIP**

This section does not describe hardware-dependant limitations for each CPU type, such as the number of available devices.

**SEE ALSO**

For details on limitations for each CPU type, refer to the user's manual for each type.
D1.1 WideField3 System Files and Generated Data

The figure below shows the sub-folders and files that are generated in the WideField3 folder during installation. These files generated by WideField3 will be repeatedly deleted and generated automatically during operation.

![Folders Generated by WideField3](Figure_D1.1_Folders_Generated_by_WideField3)

**CAUTION**

Always exit from WideField3 before deleting any folder or file from the WideField3 folder. Do not delete files in the WideField3 main folder, or files in system folders. Otherwise WideField3 may fail to start.

**TIP**

- The help folder for storing various manuals is created as "\Program Files\Common Files\yokogawa\FAM3\HELP".
- The driver folder is created as "\Program Files\Common Files\yokogawa\Driver".
D1.2 Environment Setup

To set up the operating environment of WideField3, use the Set up Environment dialog box.

You can customize the following environment setup. Different setup takes effect at different times after setup is performed on the Set up Environment dialog window.

Table D1.2 Environment Setup

<table>
<thead>
<tr>
<th>Settings</th>
<th>Description</th>
<th>Takes effect when:</th>
<th>Caution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Folder setup</td>
<td>Defines the development environment such as locations for storing programs.</td>
<td>Basically when a new screen is opened.</td>
<td>Close all screens (revert to initial state of WideField3) before changing folder settings.</td>
</tr>
<tr>
<td>(Folder Setup tab)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Communication setup</td>
<td>Defines communication destinations and conditions.</td>
<td>When you next connect to FA-M3.</td>
<td>Disconnect from FA-M3 before changing communication settings.</td>
</tr>
<tr>
<td>(Communications Setup tab)</td>
<td>Defines communications conditions for the device manager tool, sampling</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>trace tool, and FA link setup tool.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Circuit display/input setup</td>
<td>Defines use of colors for circuit components, grid lines and various</td>
<td>Immediately on all displayed screens.</td>
<td></td>
</tr>
<tr>
<td>(Circuit Display/Input tab)</td>
<td>program edit window elements.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Program monitor setup</td>
<td>Defines operation restrictions of the ladder program editor and program</td>
<td>When you next connect to FA-M3.</td>
<td>Disconnect from FA-M3 before changing program monitor settings.</td>
</tr>
<tr>
<td>(Program Monitor Setup tab)</td>
<td>monitor, whether to perform monitoring during online editing, and references for various comments and tag name definitions during online editing.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Program syntax check setup</td>
<td>Defines the severity of errors detected during program syntax checking.</td>
<td>When you next perform a program syntax check.</td>
<td></td>
</tr>
<tr>
<td>(Program Syntax Check tab)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Toolbar setup</td>
<td>Registers icons to the toolbar.</td>
<td>Immediately</td>
<td></td>
</tr>
<tr>
<td>(Toolbar Setup tab)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Email setup</td>
<td>Defines the Email environment.</td>
<td>Immediately</td>
<td></td>
</tr>
<tr>
<td>(Email Setup tab)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>File comparison setup</td>
<td>Defines comparison method, comparison targets, and display color for the</td>
<td>When you next perform a comparison.</td>
<td></td>
</tr>
<tr>
<td>(File Comparison tab)</td>
<td>file comparison function.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
D1.2.1 Basic Operations with the Set up Environment Dialog Box

This section describes how to open the Set up Environment dialog box and how to revert to default environment settings. First, open the Set up Environment dialog box. Next, click [Default] to revert to default environment settings. The relevant procedure is given below.

◆ Procedure ◆

(1) Select [Tools]–[Set up Environment] from the menu bar.
⇒ The Set up Environment dialog box opens.

TIP
To select an individual Set up Environment dialog window, click the appropriate tab such as Folder Setup or Communications Setup.

SEE ALSO
For details on each tab, see Sections D1.2.2, "Folder Setup;“
D1.2.3, "Communication Setup;“
D1.2.4, "Circuit Display/Input Setup;“
D1.2.5, "Program Monitor Setup;“
D1.2.6, "Program Syntax Check Setup;“
D1.2.7, "Toolbar Setup;“
D1.2.8, "Email Setup“ and
D1.2.9, "File Comparison Setup“

(2) Click [Default].
⇒ A confirmation dialog box opens.

(3) Click [Yes].
⇒ You are returned to the Set up Environment dialog box displaying factory default settings or the settings in effect when the Set up Environment dialog box was opened.

(4) Click [OK].
⇒ The settings are saved.

TIP
Clicking [Cancel] discards all changes made to the settings.
D1.2.2 Folder Setup

Use the Folder Setup tab to specify folders for storing data.

Figure D1.2 Folder Setup Tab
**Folders to be Defined**

There are five folders to be set up. The table below lists the purpose, default setting of each folder, as well as, the operation when the [Default] button is clicked.

<table>
<thead>
<tr>
<th>Folder</th>
<th>Purpose</th>
<th>System Default</th>
<th>Value When [Default] Is Clicked</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Location</td>
<td>The root folder for creating a project. A folder with the same name as the project file will be created under this project location.</td>
<td>\Fam3pjt</td>
<td>Value current when the Set up Environment dialog box was opened.</td>
</tr>
<tr>
<td>Common Folder</td>
<td>Folder for storing the system log files, user log files, group template files and structure type definition files.</td>
<td>\Fam3com</td>
<td>Value current when the Set up Environment dialog box was opened.</td>
</tr>
<tr>
<td>Common Library Folder</td>
<td>Folder for storing circuit blocks that can be used for general purpose without depending on the project.</td>
<td>\Fam3lib</td>
<td>Value current when the Set up Environment dialog box was opened.</td>
</tr>
<tr>
<td>Macro Folder</td>
<td>Master registry for instruction macros.</td>
<td>\Fam3mac</td>
<td>Value current when the Set up Environment dialog box was opened.</td>
</tr>
<tr>
<td>Work Folder</td>
<td>Folder for storing temporary work files created during editing.</td>
<td>\Fam3tmp</td>
<td>Value current when the Set up Environment dialog box was opened.</td>
</tr>
<tr>
<td>Names of files on your computer and block/macro names are displayed together.</td>
<td>Allows you to specify whether to show the block/macro file names together with the block/macro names in the project window.</td>
<td>On</td>
<td>On</td>
</tr>
<tr>
<td>The names of files on your computer are displayed in ToolTip.</td>
<td>Allows you to specify whether to show the block/macro file name as a tooltip when the mouse pointer points to a block name or macro name in the project window.</td>
<td>On</td>
<td>On</td>
</tr>
</tbody>
</table>
How to Set Up the Folders

There are two ways to define the required folders.

- **Entering a Folder Name Using Keyboard**
  To enter a folder name from the keyboard, click the text box for the folder to be defined, delete all characters in the text box, and enter a new folder name from the keyboard.

  ![Figure D1.3 Entering a Folder Name (for [Project Location]) from Keyboard](D0102_04.VSD)

- **Selecting a Folder Name Using the [Browse] Button**
  To select a folder name using the [Browse] button, click the [Browse] button beside the text box to open the Folder Selection dialog box. Then, click the desired folder, and click [OK].

  ![Figure D1.4 Folder Selection Dialog Box](D0102_05.VSD)
Display Setup for the Project Window

You can specify the method for showing the block and macro file names on the PC together with the display names for the blocks and macros stored in FA-M3.

TIP

A "file name on the PC" means a block, macro, or project file name that is used on the PC. A block name or macro name is a name of a block or macro stored in the CPU module and can be up to 8 characters long.

WideField3 R2 or earlier does not support the display setup for the project window.
D1.2.3 Communication Setup

WideField3 communicates with FA-M3 through FA-M3 Communication Server. Use the Communications Setup tab to specify a communication port and set up the communication method of the port.

**Figure D1.7 Communications Setup Tab**

- **Communications Setup Tab Settings**
  - To connect WideField3 to FA-M3 online, click [Connect].
  - To specify a communication port to be used for connection, enter the port number of the communication port. To perform the detailed communication setup for each communication port, click [Setup] to open the Communications Setup dialog box.
  - To use previous connection information from the connection history, click [Recent Connections].
  - You can set up RS-232C communication so that the communication speed is automatically set to the highest speed when the connection starts. To allow this automatic switching, select the [Automatically Set Communication Speed to the Highest Speed] option button.

**TIP**

- The connection history information can be used only for Ethernet and FL-net communication.
- You can run multiple instances of WideField3 simultaneously and connect each instance to a different FA-M3 through a different communication port.

**SEE ALSO**

- For details on FA-M3 Communication Server, see Chapter H12, "FA-M3 Communication Server" (Online).
- For details on multiple connections, see Section H1.1.4, "Making Multiple Connections" (Online).
Communications Setup Dialog Box

Use the Communications Setup dialog box to specify the settings for a communication port.

The port number of the communication port is displayed on the title bar of the Communications Setup dialog box.

Figure D1.8 Communications Setup Dialog Box

TIP

You cannot specify the settings that overlap with those of other communication ports.
**Communication Media**

Select the communication medium by turning on one of the following option buttons: [USB], [RS-232C], [RS-232C via Modem], [Ethernet] and [FL-net].

**TIP**

Modified settings are reflected at the next connection to the FA-M3.

---

**CAUTION**

Connection to FL-net is available for FL-net (OPCN-2) Interface Module revision 01:00 or later.

---

- **[USB]**
  
  If [USB] is selected as communication medium, no other setting is required on this dialog.

- **[RS-232C]**
  
  If [RS-232C] is selected as communication medium, perform the following procedure:
  
  - If you want to match the communications mode of the CPU, turn on the [Automatic Recognition] option button.
  
  - Otherwise, turn on the [Fixed] option button.

If [RS-232C via Modem] is selected as communication medium, ignore the [Connection Method] setting.

The default connection method is [Automatic Recognition].

**TIP**

- If [Automatic Recognition] is selected, connection to FA-M3 takes some time.

- If you select [Fixed], you have to select a communications mode from the communications mode drop-down list. Select [Fixed] only if you know the current communications mode of the CPU.

- Communication Timeout, Number of Retries, and COM Port Number

  If the communications medium is [RS-232C], set up [Communication Timeout], [Number of Retries], and [COM Port Number]. Normally, the default settings should be used.

  If communications medium is [RS-232C via Modem], set up the [Communication Timeout] and [Number of Retries] settings for ENQ-ACK communication.

  By default, [Communication Timeout] is 1 second, [Number of Retries] is 2, and [COM Port Number] is 1.

  You can set up RS-232C communication so that the communication speed is automatically set to the highest speed when the connection starts. To allow this automatic switching, turn on the [Automatically Set Communication Speed to the Highest Speed] option button. The default setting is off; when the option button is off, communication will be performed at the speed established when the connection is made.
If the communication medium is set to [Ethernet], specify [Destination IP Address]. You can define the [Destination IP Address] as either a host name or an IP address. You can select an IP address from the drop-down list containing previously defined IP addresses (up to 1024 addresses).

**TIP**
When the [Refer to Host Name] button is clicked, you can view and set up host names for WideField3 specified in the hosts file in Windows.

For Ethernet communication, specify [Connection Timeout]. The default setting is 20 seconds.
● [RS-232C via Modem]

If the communication medium is [RS-232C via Modem], click the [Select Dial-up] button to open the Select Dial-up dialog box, and select the connection destination.

![Select Dial-up Dialog Box](D0102_11.VSD)

**Figure D1.9  Select Dial-up Dialog Box**

**CAUTION**

If you use a modem, define communications speed and other communications settings by selecting [Control Panel]–[Phone and Modem Options], followed by the Modems tab in Windows.

**SEE ALSO**

For details on modem settings, see the Windows manual, Windows help, or the modem manual provided by the manufacturer.
[FL-net]

When using FL-net as communication medium, you can set up the PC in the same way as specifying the destination when using Ethernet, or you can also set up the PC in another way.

Setting up the PC involves specifying the IP address and the node number of the PC.

(1) Specifying the IP address

If multiple network cards are installed in the PC, you can specify which network card is to be used by specifying the IP address of the network card (but not by specifying the host name). If only one network card is installed or the network card with the highest priority is to be used, you need not specify the IP address.

(2) Specifying the node number

You can specify a node number (1 to 254), independent of the IP address of the PC. If no node number is specified, the least significant byte of the network card is used as the node number. When performing a temporary connection of the PC to the FL-net, you can specify a node number, independent of the IP address. You need to set different node numbers for the PC and FA-M3 so you should specify a number that is usually not used, such as a number close to 254.

Click the [Set up Local Node] button, and specify the IP address (numbers only) of the PC and the node number (1-254) on the displayed Set up Local Node dialog box. By default, the [Not Specified] option button is selected for both the PC network card and PC node number sections on the Set up Local Node dialog box.

![Set up Local Node Dialog Box](image)

**Figure D1.10  Set up Local Node Dialog Box**

- In FL-net, the least significant byte of the destination IP address is taken as the node number of the destination (1-254) so you should specify a value that is different from the node number of the PC.

**CAUTION**

If multiple network cards are installed in a PC and used as FL-net nodes, operation will not work properly.
D1.2.4 Circuit Display/Input Setup

Use the Circuit Display/Input tab to specify colors and character sizes for circuit components, as well as the program input mode and display mode.

Figure D1.11 Circuit Display/Input Tab
Circuit Display/Input Tab

Set up Circuit Components
You can define the color and character size for each circuit component. What items can be defined depends on individual circuit components as shown below.

Table D1.4 Settings

<table>
<thead>
<tr>
<th>Components</th>
<th>Color Setting</th>
<th>Other settings (default values are given in parentheses)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Circuit</td>
<td>Customizable</td>
<td>Column width (53)</td>
</tr>
<tr>
<td>Instruction Parameter</td>
<td>Customizable</td>
<td>Character size (11)</td>
</tr>
<tr>
<td>Circuit Comment</td>
<td>Customizable</td>
<td>Character size (11), Line height (20), Automatic adjustment of line height (On)</td>
</tr>
<tr>
<td>Subcomment</td>
<td>Customizable</td>
<td>Character size (10)</td>
</tr>
<tr>
<td>I/O Comment</td>
<td>Customizable</td>
<td>Character size (10), Number of lines displayed (2)</td>
</tr>
<tr>
<td>ON Indicator</td>
<td>Customizable</td>
<td></td>
</tr>
<tr>
<td>Increasing Data Display Color</td>
<td>Customizable</td>
<td></td>
</tr>
<tr>
<td>Decreasing Data Display Color</td>
<td>Customizable</td>
<td></td>
</tr>
<tr>
<td>Local Device</td>
<td>Customizable</td>
<td></td>
</tr>
<tr>
<td>Undefined Tag Name</td>
<td>Customizable</td>
<td></td>
</tr>
<tr>
<td>Label</td>
<td>Customizable</td>
<td></td>
</tr>
<tr>
<td>Line No.</td>
<td>Customizable</td>
<td></td>
</tr>
<tr>
<td>Instruction No.</td>
<td>Customizable</td>
<td></td>
</tr>
<tr>
<td>Modified Line (line number)</td>
<td>Customizable</td>
<td></td>
</tr>
<tr>
<td>Modified Line (circuit)</td>
<td>Customizable</td>
<td></td>
</tr>
<tr>
<td>Modification Prohibited Circuit (online)</td>
<td>Customizable</td>
<td></td>
</tr>
<tr>
<td>IL-ILC Instruction Range</td>
<td>Customizable</td>
<td></td>
</tr>
<tr>
<td>SUB-RET Instruction Range</td>
<td>Customizable</td>
<td></td>
</tr>
<tr>
<td>INTP-IRET Instruction Range</td>
<td>Customizable</td>
<td></td>
</tr>
<tr>
<td>FOR-NEXT Instruction Range</td>
<td>Customizable</td>
<td></td>
</tr>
<tr>
<td>SCRIPT Instruction Range</td>
<td>Customizable</td>
<td></td>
</tr>
</tbody>
</table>

TIP

- You can use circuit comments to add function names, etc. to a ladder program. You can enter up to 72 characters.
- You can use subcomments to insert comments in the right side margin of a ladder program. You can enter up to 24 characters.
- You can use I/O comments to add comments to devices. You can enter up to 32 characters.
- Increasing/Decreasing Data Display Color are display functions for recognizing changes in device values shown on the program monitor.
- The instruction ranges for IL-ILC through FOR-NEXT specify the start and end positions of instructions used in pairs. The SCRIPT instruction range specifies the range of ladder programs where scripts are expanded.

SEE ALSO

For details on circuit comments, subcomments and I/O comments, see Chapter A3, "Basic Specifications" (Introduction and Troubleshooting).
- **Color setting**

  To change a color setting, select a component from the drop down list, click the [Set Color] button to display the Color dialog box, select a color, and click [OK].

  ![Color Dialog Box](image1)

  **Figure D1.12 Color Dialog Box**

- To change character size or number of lines displayed, use the [Character Size] or [Display Lines] spin button respectively.

### Set up Background Colors

You can specify the background color of ladder program edit windows. Individual settings apply to different windows, as shown in the table below. The set up procedure is similar to that for circuit component colors.

<table>
<thead>
<tr>
<th>Settings</th>
<th>Applicable Windows</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offline</td>
<td>Edit Block/Edit Macro windows</td>
</tr>
<tr>
<td>Online</td>
<td>Monitor Block/Monitor Macro windows</td>
</tr>
<tr>
<td>Grid Color</td>
<td>All ladder windows</td>
</tr>
<tr>
<td>Cursor</td>
<td>All ladder windows</td>
</tr>
<tr>
<td>Index View</td>
<td>All ladder windows</td>
</tr>
<tr>
<td>On</td>
<td>Edit Block/Edit Macro windows</td>
</tr>
<tr>
<td>Not Executed</td>
<td>Edit Block/Edit Macro windows</td>
</tr>
<tr>
<td>Page Separator</td>
<td>None</td>
</tr>
<tr>
<td>Online Edit</td>
<td>Online edit screens of the Monitor Block/Monitor Macro windows</td>
</tr>
</tbody>
</table>

**TIP**

WideField3 R2 or earlier does not support page separators.

### Set up Parameter Input

You can enable or disable the input of tag name definition information at the time of parameter input. The behavior of parameter input depends on the settings as shown below.

<table>
<thead>
<tr>
<th>Settings</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enter tag name with address and I/O comment</td>
<td>The Address Assignment dialog box for address or I/O comment assignment opens when you enter a tag name or an address.</td>
</tr>
<tr>
<td>Enter tag name without address and I/O comment</td>
<td>The Address Assignment dialog box does not open when you enter a tag name. However, it is displayed when you enter a new tag name in online editing.</td>
</tr>
</tbody>
</table>
● Numeric Parameters Display
- If you turn on the [Use Digit Grouping] checkbox, a comma "," is used as a separator between every three digits in a number displayed in the program screen.
- In the spin box, you can specify the minimum number of digits that is used to determine whether a number should be displayed in exponential notation. The valid range of the number of digits is from 8 to 21.

● Window Display Position
If you turn on the [Display on top left of display area] checkbox, windows are placed at the top left of the display when opened.

● Set up TipHelp Display
You can specify how TipHelp is to be displayed. Details are given in the table below.

<table>
<thead>
<tr>
<th>Settings</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display All Items</td>
<td>Corresponding tag name, address, and I/O comment are displayed.</td>
</tr>
<tr>
<td>Tag Name or Address</td>
<td>Addresses are also displayed when tag names are displayed and vice versa.</td>
</tr>
<tr>
<td>I/O Comment Only</td>
<td>Only I/O comments are displayed.</td>
</tr>
<tr>
<td>Do not Display</td>
<td>No TipHelp is displayed.</td>
</tr>
</tbody>
</table>

**TIP**
TipHelp is tag name definition, constant definition, instruction and other information text displayed when you place the mouse cursor over a circuit component for a certain time.

**SEE ALSO**
For details on TipHelp, see Section E1.2.1, "Ladder Program Edit Screen Layout."

● Instruction/Instruction Parameter Completion
This is an input completion (auto completion) function for assisting the user when entering instructions/instruction parameters while editing a program. It can be set to automatically display candidates when two or more characters have been entered.

**SEE ALSO**
For details about the input completion function, see Section E1.2.6, "Entering Instructions Using Input Completion Function" and Section E1.2.7, "Entering Instruction Parameters Using Input Completion Function."

● Emphasize Modified Lines (Identification of Unconverted Circuits)
You can set whether to identify circuits being edited in a program, as well as circuits that are incomplete due to conversion error.

● Parameter Direct Change Input Settings
You can specify whether to directly modify the parameters in a program edit. You need to select either [Command change has priority] or [Parameter change has priority].
Display of Timer/Counter Current Values

You can specify whether to display current values of timers (T) and counters (C) as count-up values or count-down values on monitor screens. When changing the current value of a timer or counter, you can also enter the new value according to the selected display mode (count-up or count-down).

<table>
<thead>
<tr>
<th>Settings</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count-down</td>
<td>When a timer (T) or counter (C) is started, its current value changes from its preset value to 0.</td>
</tr>
<tr>
<td>Count-up</td>
<td>When a timer (T) or counter (C) is started, its current value changes from 0 to its preset value.</td>
</tr>
</tbody>
</table>

CAUTION

All timers (T) and counters (C) of FA-M3 operate internally as count-down timers or counters, regardless of its display setup.

Set up Display Font

You can set the display font on the program edit/program monitor screens. You can set only the type of font. You cannot change the style. To change the font size, specify the size in [Set up Circuit Components].

CAUTION

Comments might not be displayed properly in the language mode selected on WideField3 depending on the font type selected.

Set Number of Local Devices

When using local devices in a program, this function automatically sets the number of local devices included in the block properties. In addition, you can specify a reserve area automatically reserved for this number. For example, if you set the reserve area to 20% and the largest device number used in the program is /D00100, reservations are ensured for /D00001 ~ /D00120. However, since the number of reservations ensured is for the number of local devices that can be set, you should set the reserve area only as a guide.

By specifying a reserve area, you can maintain a sufficient area to continue your edit operation when you add a local device during online editing.

TIP

If you select [Set Automatically] for [Set Number of Local Devices], the automatic setting is applied when you modify a block that uses local devices.
D1.2.5 Program Monitor Setup

Use the Program Monitor Setup tab to specify operation restrictions for ladder program editing and program monitoring. You can also specify the reference source for comments and the reference source for tag name definitions when connected online.

![Program Monitor Setup Tab](D0102_15.VSD)

**Figure D1.13 Program Monitor Setup Tab**

The following settings are available.

**Table D1.9 Setup Items**

<table>
<thead>
<tr>
<th>Item Group</th>
<th>Item Subgroup</th>
<th>Item</th>
<th>Default Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online Editing Monitor Function Setup</td>
<td></td>
<td>Activate/deactivate program monitoring during online editing.</td>
<td>On</td>
</tr>
<tr>
<td>Program Editing/Monitoring</td>
<td></td>
<td>Disallow concurrent ladder program editing and program monitoring for the same block.</td>
<td>On</td>
</tr>
<tr>
<td>Online Editing Setup</td>
<td></td>
<td>Program online edit results are always applied to offline files.</td>
<td>Off</td>
</tr>
<tr>
<td>Online Editing of Tag Name Definition/Comment Reference Setup</td>
<td>Tag Name Definition Reference Setup</td>
<td>Tag Name Definitions at the Time of the Last Download</td>
<td>Latest Tag Name Definitions</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Latest Comments (only if concurrent execution is disabled)</td>
<td>Latest Comments at the Time of the Last Download</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Comments at the Time of the Last Download</td>
<td></td>
</tr>
</tbody>
</table>
Online Editing Monitor Function Setup

Turning on the [Perform program monitoring during online editing] checkbox activates program monitoring display during online editing.

TIP

Turning off the checkbox deactivates program monitoring display. The debugging function is also disabled.

Program Editing/Monitoring

Turning on the [Disallow concurrent ladder program editing and program monitoring for the same block] checkbox disallows concurrent execution; turning it off allows concurrent execution. When disallowed, you cannot open the common tag name definition window under online and offline conditions concurrently.

TIP

This setting also applies to a block which is opened from the [Results of Program Syntax Checking], [Results of Project Search], [Results of Find Instruction in Project Operation], [Alarm Monitor], [Results of Comparison] (for File and CPU comparison, as well as CPU and ROM comparison) and other windows.

If the [Disallow concurrent ladder program editing and program monitoring for the same block] checkbox is turned on, any attempt to open the same block for ladder program editing and program monitoring at the same time would fail with the following messages.

Table D1.10 Error Messages for Illegal Concurrent Startup Operations

<table>
<thead>
<tr>
<th>Action</th>
<th>Error Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opening for ladder program editing</td>
<td>You cannot open an online screen and an offline screen concurrently.</td>
</tr>
<tr>
<td>Opening using program monitor</td>
<td>You cannot open an online screen and an offline screen concurrently.</td>
</tr>
</tbody>
</table>

Online Editing Setup

You can specify how to reflect the results of online editing.

Table D1.11 Online Editing Settings

<table>
<thead>
<tr>
<th>Settings</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program online edit results are always applied to offline files</td>
<td>If the checkbox is off, when you close a block edited in online editing, a message box is displayed to confirm that you want to reflect the edit result to an offline file. If the checkbox is on, the confirmation message is not displayed and the edit result is automatically reflected to the offline file.</td>
</tr>
<tr>
<td>Tag name definitions added/changed using online editing are always applied to offline files</td>
<td>If the checkbox is off, when you close a block after you edit a tag name definition in online editing, a message box is displayed to confirm that you want to reflect the edit result to an offline file. If the checkbox is on, the confirmation message is not displayed and the edit result is automatically reflected to the offline file.</td>
</tr>
</tbody>
</table>

TIP

If the project opened offline is different from the one opened online, the setting of [Program online edit results are always applied to offline files] is disabled and a message is displayed to confirm that you want to reflect the edit result to the offline file. If the block to be reflected is a reference block (i.e., a block in the common library or in another project), a relevant block is created in the offline project.
● Reference for Tag Name Definitions When Online

You can specify the tag name definition to be referenced during circuit monitoring or online editing when connected online.

Table D1.12  Reference for Tag Name Definitions When Online

<table>
<thead>
<tr>
<th>Settings</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latest Tag Name Definitions</td>
<td>Application refers to the tag name definitions within the project. Additions of new tag name definitions and editing of I/O comments are allowed. If the [Program online edit results are always applied to offline files] checkbox is on, the result of online editing is always reflected to an offline file.</td>
</tr>
<tr>
<td>Tag Name Definitions at the Time of the Last Download</td>
<td>Application refers to the tag name definitions current when downloading was last performed. Additions of new tag name definitions and editing of I/O comments are not allowed.</td>
</tr>
</tbody>
</table>

TIP

If the [Disallow concurrent ladder program editing and program monitoring for the same block] checkbox is turned off, screen controls in this group are disabled, and the [Tag Name Definitions at the Time of the Last Download] option is selected by default.

● Reference for Circuit Comments and Subcomments When Online

You can specify the reference source for circuit comments and subcomments for circuit monitoring or online editing when connected online.

Table D1.13  Reference for Circuit Comments and Subcomments When Online

<table>
<thead>
<tr>
<th>Settings</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latest Comments</td>
<td>Application refers to comments within the project.</td>
</tr>
<tr>
<td>Comments at the Time of the Last Download</td>
<td>Application refers to the comments current when downloading was last performed.</td>
</tr>
</tbody>
</table>

TIP

If the [Disallow concurrent ladder program editing and program monitoring for the same block] checkbox is turned off, all screen controls in this group are disabled, and the [Comments at the Time of the Last Download] option is selected by default.

CAUTION

If circuit comments and subcomments are downloaded to the CPU, then the application will refer to the downloaded comments regardless of the settings under [Reference for Circuit Comments and Subcomments When Online].
D1.2.6 **Program Syntax Check Setup**

Use the Program Syntax Check tab to change the severity of an error or disable/enable syntax check.

![Program Syntax Check Tab](D0102_16.VSD)

**Figure D1.14  Program Syntax Check Tab**

---

**CAUTION**

F3SP71-4N, F3SP76-7N, F3SP71-4S and F3SP76-7S do not support the partial download function.
Syntax Check Settings

The following syntax check settings are available.

Table D1.14  Syntax Check Settings

<table>
<thead>
<tr>
<th>Item</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duplicate use of coil</td>
<td>Warning</td>
</tr>
<tr>
<td>Duplicate use of SET/RST</td>
<td>Warning</td>
</tr>
<tr>
<td>Duplicate use of timer/counter</td>
<td>Error</td>
</tr>
<tr>
<td>Duplicate use of label</td>
<td>Error</td>
</tr>
<tr>
<td>Duplicate use of interrupt I/O address</td>
<td>Error</td>
</tr>
<tr>
<td>IL-ILC (interlock start - end) mismatch</td>
<td>Error</td>
</tr>
<tr>
<td>SUB-RET (subroutine start - end) mismatch</td>
<td>Error</td>
</tr>
<tr>
<td>INTP-IRET (interrupt routine start - end) mismatch</td>
<td>Error</td>
</tr>
<tr>
<td>FOR-NEXT (loop start - end) mismatch</td>
<td>Error</td>
</tr>
<tr>
<td>CALL-SUB (subroutine source - destination) mismatch</td>
<td>Error</td>
</tr>
<tr>
<td>Jump destination not found.</td>
<td>Error</td>
</tr>
<tr>
<td>Too many high-speed READ/WRITE instructions.</td>
<td>Error</td>
</tr>
<tr>
<td>Duplicate block name for ACT/INACT instructions</td>
<td>Error</td>
</tr>
<tr>
<td>Global device overlaps local device area.</td>
<td>Error</td>
</tr>
<tr>
<td>Tag name converted to local/macro device.</td>
<td>Error</td>
</tr>
</tbody>
</table>

Enabling/Disabling Syntax Check

To enable syntax check for an item, turn on the [Error] or [Warning] option button for that item. To disable syntax check for an item, turn on the [Ignore] option button for that item.

Syntax Check during Partial Downloading

Turning on the [Perform syntax check during partial download] checkbox performs syntax check on project basis; turning it off performs syntax check on block basis.

The [Perform syntax check during partial download] setting is applicable only to partial downloading.

SEE ALSO

For details on syntax check, see Section D3.2.1, "Checking Programs."
D1.2.7 Toolbar Setup

Use the Toolbar Setup tab to specify instructions to be added to the toolbar.

Figure D1.15  Toolbar Setup Tab
### Menu Instructions and Icons

The table below lists menu commands (commands which can be selected and executed from the menu bar) and their icons which you can add to the toolbar.

<table>
<thead>
<tr>
<th>Menu Bar Item</th>
<th>Menu Command</th>
<th>Icons</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>File</strong></td>
<td>New Project</td>
<td>![File-Icon]</td>
</tr>
<tr>
<td></td>
<td>Open Project</td>
<td>![File-Icon]</td>
</tr>
<tr>
<td></td>
<td>Close Project</td>
<td>![File-Icon]</td>
</tr>
<tr>
<td></td>
<td>Save Project</td>
<td>![File-Icon]</td>
</tr>
<tr>
<td></td>
<td>New</td>
<td>![File-Icon]</td>
</tr>
<tr>
<td></td>
<td>Open – Block/Macro</td>
<td>![File-Icon]</td>
</tr>
<tr>
<td></td>
<td>Save</td>
<td>![File-Icon]</td>
</tr>
<tr>
<td></td>
<td>Print Setup</td>
<td>![File-Icon]</td>
</tr>
<tr>
<td></td>
<td>Print</td>
<td>![File-Icon]</td>
</tr>
<tr>
<td></td>
<td>Compare Project</td>
<td>![File-Icon]</td>
</tr>
<tr>
<td></td>
<td>Compare Window</td>
<td>![File-Icon]</td>
</tr>
<tr>
<td><strong>Edit</strong></td>
<td>Undo</td>
<td>![Edit-Icon]</td>
</tr>
<tr>
<td></td>
<td>Convert</td>
<td>![Edit-Icon]</td>
</tr>
<tr>
<td></td>
<td>Cut</td>
<td>![Edit-Icon]</td>
</tr>
<tr>
<td></td>
<td>Copy</td>
<td>![Edit-Icon]</td>
</tr>
<tr>
<td></td>
<td>Paste</td>
<td>![Edit-Icon]</td>
</tr>
<tr>
<td></td>
<td>Delete</td>
<td>![Edit-Icon]</td>
</tr>
<tr>
<td></td>
<td>Image Copy</td>
<td>![Edit-Icon]</td>
</tr>
<tr>
<td></td>
<td>Insert Line</td>
<td>![Edit-Icon]</td>
</tr>
<tr>
<td></td>
<td>Delete Lines</td>
<td>![Edit-Icon]</td>
</tr>
<tr>
<td></td>
<td>Temporary Delete</td>
<td>![Edit-Icon]</td>
</tr>
<tr>
<td></td>
<td>Undo Temporary Delete</td>
<td>![Edit-Icon]</td>
</tr>
<tr>
<td></td>
<td>Page Break</td>
<td>![Edit-Icon]</td>
</tr>
<tr>
<td></td>
<td>Hide/Show Circuits – Hide Circuits</td>
<td>![Edit-Icon]</td>
</tr>
<tr>
<td></td>
<td>Hide/Show Circuits – Show Circuits</td>
<td>![Edit-Icon]</td>
</tr>
<tr>
<td></td>
<td>Block Tag Name Definition</td>
<td>![Edit-Icon]</td>
</tr>
<tr>
<td></td>
<td>Local Device/Properties</td>
<td>![Edit-Icon]</td>
</tr>
<tr>
<td>Menu bar</td>
<td>Menu instructions</td>
<td>Icons</td>
</tr>
<tr>
<td>----------</td>
<td>-------------------</td>
<td>-------</td>
</tr>
<tr>
<td>Find</td>
<td>Find</td>
<td>![icon]</td>
</tr>
<tr>
<td></td>
<td>Find Instruction</td>
<td>![icon]</td>
</tr>
<tr>
<td></td>
<td>Find Next – Find Next</td>
<td>![icon]</td>
</tr>
<tr>
<td></td>
<td>Jump – Top</td>
<td>![icon]</td>
</tr>
<tr>
<td></td>
<td>Jump – Bottom</td>
<td>![icon]</td>
</tr>
<tr>
<td></td>
<td>Jump – Line</td>
<td>![icon]</td>
</tr>
<tr>
<td></td>
<td>Jump – Instruction Number</td>
<td>![icon]</td>
</tr>
<tr>
<td></td>
<td>Replace</td>
<td>![icon]</td>
</tr>
<tr>
<td></td>
<td>Obsolete Device List</td>
<td>![icon]</td>
</tr>
<tr>
<td></td>
<td>Action Monitor</td>
<td>![icon]</td>
</tr>
<tr>
<td>View</td>
<td>Project Window</td>
<td>![icon]</td>
</tr>
<tr>
<td></td>
<td>Specify Display Mode – Display I/O Comment</td>
<td>![icon]</td>
</tr>
<tr>
<td></td>
<td>Specify Display Mode – Display Instruction Number</td>
<td>![icon]</td>
</tr>
<tr>
<td></td>
<td>Specify Display Mode – Display Address</td>
<td>![icon]</td>
</tr>
<tr>
<td></td>
<td>Redraw</td>
<td>![icon]</td>
</tr>
<tr>
<td></td>
<td>Zoom – Zoom Up</td>
<td>![icon]</td>
</tr>
<tr>
<td></td>
<td>Zoom – Zoom Down</td>
<td>![icon]</td>
</tr>
<tr>
<td></td>
<td>Specify Display Mode – Detail</td>
<td>![icon]</td>
</tr>
<tr>
<td></td>
<td>Specify Display Mode – Display Format – Decimal</td>
<td>![icon]</td>
</tr>
<tr>
<td></td>
<td>Specify Display Mode – Display Format – Hexadecimal</td>
<td>![icon]</td>
</tr>
<tr>
<td></td>
<td>Specify Display Mode – Display Format – Character String</td>
<td>![icon]</td>
</tr>
<tr>
<td></td>
<td>Specify Display Mode – Display Format – Floating Point</td>
<td>![icon]</td>
</tr>
<tr>
<td></td>
<td>Specify Display Mode – Display Format – Binary</td>
<td>![icon]</td>
</tr>
<tr>
<td></td>
<td>Suspend Monitoring</td>
<td>![icon]</td>
</tr>
<tr>
<td></td>
<td>Resume Monitoring</td>
<td>![icon]</td>
</tr>
<tr>
<td></td>
<td>Check Program</td>
<td>![icon]</td>
</tr>
<tr>
<td>Project</td>
<td>Find in Project</td>
<td>![icon]</td>
</tr>
<tr>
<td></td>
<td>Find Instruction in Project</td>
<td>![icon]</td>
</tr>
<tr>
<td></td>
<td>Replace in Project</td>
<td>![icon]</td>
</tr>
<tr>
<td></td>
<td>Change I/O Installation Position</td>
<td>![icon]</td>
</tr>
<tr>
<td></td>
<td>Obsolete Device List</td>
<td>![icon]</td>
</tr>
<tr>
<td>Menu bar</td>
<td>Menu instructions</td>
<td>Icons</td>
</tr>
<tr>
<td>-------------</td>
<td>----------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td><strong>Online</strong></td>
<td>Connect</td>
<td><img src="image" alt="Connect" /></td>
</tr>
<tr>
<td></td>
<td>Disconnect</td>
<td><img src="image" alt="Disconnect" /></td>
</tr>
<tr>
<td></td>
<td>Program Monitor</td>
<td><img src="image" alt="Program Monitor" /></td>
</tr>
<tr>
<td></td>
<td>Configuration</td>
<td><img src="image" alt="Configuration" /></td>
</tr>
<tr>
<td></td>
<td>Operating Mode – Run</td>
<td><img src="image" alt="Operating Mode – Run" /></td>
</tr>
<tr>
<td></td>
<td>Operating Mode – Stop</td>
<td><img src="image" alt="Operating Mode – Stop" /></td>
</tr>
<tr>
<td></td>
<td>Operating Mode – Debug</td>
<td><img src="image" alt="Operating Mode – Debug" /></td>
</tr>
<tr>
<td></td>
<td>Download – Project</td>
<td><img src="image" alt="Download – Project" /></td>
</tr>
<tr>
<td></td>
<td>Upload – Project</td>
<td><img src="image" alt="Upload – Project" /></td>
</tr>
<tr>
<td></td>
<td>Compare File and CPU – Project</td>
<td><img src="image" alt="Compare File and CPU – Project" /></td>
</tr>
<tr>
<td><strong>Debug/ Maintenance</strong></td>
<td>Forced Set</td>
<td><img src="image" alt="Forced Set" /></td>
</tr>
<tr>
<td></td>
<td>Forced Reset</td>
<td><img src="image" alt="Forced Reset" /></td>
</tr>
<tr>
<td></td>
<td>Cancel Forced Set/Reset</td>
<td><img src="image" alt="Cancel Forced Set/Reset" /></td>
</tr>
<tr>
<td></td>
<td>Cancel All Forced Set/Reset</td>
<td><img src="image" alt="Cancel All Forced Set/Reset" /></td>
</tr>
<tr>
<td></td>
<td>Change Word Data</td>
<td><img src="image" alt="Change Word Data" /></td>
</tr>
<tr>
<td></td>
<td>Change Long Word Data</td>
<td><img src="image" alt="Change Long Word Data" /></td>
</tr>
<tr>
<td></td>
<td>Change Current Value of Timer/Counter</td>
<td><img src="image" alt="Change Current Value of Timer/Counter" /></td>
</tr>
<tr>
<td></td>
<td>Change Preset Value of Timer/Counter</td>
<td><img src="image" alt="Change Preset Value of Timer/Counter" /></td>
</tr>
<tr>
<td></td>
<td>Start Online Editing</td>
<td><img src="image" alt="Start Online Editing" /></td>
</tr>
<tr>
<td></td>
<td>Start Block</td>
<td><img src="image" alt="Start Block" /></td>
</tr>
<tr>
<td></td>
<td>Stop Block</td>
<td><img src="image" alt="Stop Block" /></td>
</tr>
<tr>
<td></td>
<td>Stop Refreshing – Stop Input</td>
<td><img src="image" alt="Stop Refreshing – Stop Input" /></td>
</tr>
<tr>
<td></td>
<td>Stop Refreshing – Stop Output</td>
<td><img src="image" alt="Stop Refreshing – Stop Output" /></td>
</tr>
<tr>
<td></td>
<td>Stop Refreshing – Stop Shared</td>
<td><img src="image" alt="Stop Refreshing – Stop Shared" /></td>
</tr>
<tr>
<td></td>
<td>Stop Refreshing – Stop Link</td>
<td><img src="image" alt="Stop Refreshing – Stop Link" /></td>
</tr>
<tr>
<td></td>
<td>Restart Refreshing</td>
<td><img src="image" alt="Restart Refreshing" /></td>
</tr>
<tr>
<td></td>
<td>Display Alarm</td>
<td><img src="image" alt="Display Alarm" /></td>
</tr>
<tr>
<td></td>
<td>Cancel Alarm</td>
<td><img src="image" alt="Cancel Alarm" /></td>
</tr>
<tr>
<td></td>
<td>Display CPU Log – Display Operation Log</td>
<td><img src="image" alt="Display CPU Log – Display Operation Log" /></td>
</tr>
<tr>
<td></td>
<td>Display CPU Log – Display System Log</td>
<td><img src="image" alt="Display CPU Log – Display System Log" /></td>
</tr>
<tr>
<td></td>
<td>Display CPU Log – Display User Log</td>
<td><img src="image" alt="Display CPU Log – Display User Log" /></td>
</tr>
<tr>
<td></td>
<td>Display CPU Log – Clear Log</td>
<td><img src="image" alt="Display CPU Log – Clear Log" /></td>
</tr>
<tr>
<td><strong>Tool</strong></td>
<td>Set up Environment</td>
<td><img src="image" alt="Set up Environment" /></td>
</tr>
<tr>
<td><strong>Help</strong></td>
<td>WideField3 Help</td>
<td><img src="image" alt="WideField3 Help" /></td>
</tr>
</tbody>
</table>
Registering Icons

To register an icon to the toolbar, turn on the checkbox for the corresponding menu command. To remove or deregister an icon, turn off the checkbox for the corresponding menu command.
D1.2.8 Email Setup

Use the Email Setup tab to set up the remote OME Email environment.

Figure D1.16 Email Setup Tab
■ Recipient Information Settings

Set the Email recipient information: CPU mail address, CPU number, module password, and program password.

■ Sender Information Settings

Specify the mail address of WideField3, the server address, and dialup settings.

SEE ALSO

For details on the Email Setup tab, see Section H10.4.1, "Email Environment Setup" (Online).

⚠️ CAUTION

The remote OME Email function provides standard Email send/receive service based on SMTP/POP. Thus, you may also send Emails to or receive mails from systems other than the FA-M3. However, such usage is not guaranteed.
D1.2.9 File Comparison Setup

Use the File Comparison tab to set up comparison methods, comparison targets, and window components and display colors for the file comparison function.

- **Comparison Method**
  
  Set the comparison method used when comparing ladder programs. If you select [Address Base], programs that use tag names are converted to an address base before comparison is performed.
  
  If you select [No Change], programs are not converted to an address base; comparison is performed using the parameter conditions that have been set.

- **Comparison Target**
  
  Turn on the [Comment] checkbox to include circuit, sub, and I/O comment and label comparisons.

- **Comparison Source Location**
  
  Specify [Left/Up] or [Right/Down] as the display area for the target comparison source in the comparison results display screen.

- **Color**
  
  Specify the background color for areas with differences and for areas with corrected lines in the comparison results display screen. Specify background colors for Differences and Corrected Line.
D1.2.10 Restoring the Default Settings of All Items in the Environment Setup

You can restore all items in the environment setup to the WideField3 default settings. The relevant procedure is given below.

◆ Procedure ◆

(1) Select [Tools] - [Return to Default Environment Settings] from the menu bar.
⇒ A confirmation dialog box is displayed.
(2) Click [Yes].
⇒ All items in the environment setup are restored to the default settings.

⚠️ CAUTION

The above operation does not change the folder locations specified in the Folder Setup tab and the settings in the Email Setup tab.

TIP

Some items of the environment setup do not immediately reflect changes.

SEE ALSO

For details on the timing when the environment setup changes are reflected, see Section D1.2, “Environment Setup.”
D1.2.11 Exporting the Environment Setup

You can export the current environment setup in a folder created. The exported environment setup can be imported later.

The relevant procedure is given below.

◆ Procedure ◆

(1) Select [Tools] - [Export Environment Settings] from the menu bar.

⇒ A dialog box for selecting an export destination folder is displayed.

(2) Select a destination folder and click [OK].

⇒ A confirmation dialog box is displayed. In the selected folder, the "ExportedEnv" folder is created and the exported environment setup data is stored in the "ExportedEnv" folder.

CAUTION

- While exporting the environment setup, you cannot operate WideField3.
- Do not modify the files in the folder that stores the exported environment setup.
- The communication settings for each communication port are shared with other FA-M3 applications and cannot be exported.
D1.2.12 Importing the Environment Setup

You can import the exported environment setup into WideField3. The relevant procedure is given below.

◆ Procedure ◆

(1) Select [Tools] - [Import Environment Settings] from the menu bar.
⇒ A dialog box is displayed to prompt you to specify a folder that contains data to be imported.
(2) Select an environment setup folder to import, and click [OK].
⇒ The environment setup is imported.

TIP
- When the environment setup is imported, only the items that have a non-default setting in the import source environment setup are reflected to the current environment setup. Note however that the email setup and the various folder paths specified in the folder setup are imported regardless of whether the settings are default or not.
- When the environment setup is imported, if the major version number (i.e., "XX" in "WideFieldXXRYY.ZZ") of WideField3 is different between the import source and destination, the various folder paths specified in the folder setup are not imported.

CAUTION
- While importing the environment setup, you cannot operate WideField3.
- The environment setup exported from the older version of WideField3 cannot be imported into the newer version of WideField3.
- The communication settings for each communication port are shared with other FA-M3 applications and cannot be imported.
- The imported email setup is enabled after WideField3 is restarted.
D1.3 Language Setup

You can switch the language displayed in software menus and dialogs in WideField3. After switching the language displayed, you must restart WideField3 for the change to take effect.

The relevant procedure is given below.

◆ Procedure ◆

(1) Select [Tools]-[Language Change] from the menu bar.
⇒ The Change Languages dialog box opens.

TIP
The appearance of this dialog box might be different depending on the language currently being used.

(2) Select the language you want to change to from the [Languages] combo box.

(3) Click [OK].
⇒ The setting is saved, and the language you selected will take effect when WideField3 is restarted.
D1.4 Key Customization

You can assign the shortcut keys used in WideField3 to any key you wish. In addition, you can save key assignments to a file.

D1.4.1 Key Customization Setup

This subsection describes how to perform key customization. The relevant procedure is given below.

◆ Procedure ◆

(1) Select [Tools]-[Customize Keys] from the menu bar.
⇒ The Customize Keys dialog box opens.

(2) Select the applicable operation/function category from the Key Group.

(3) Move the cursor onto the command you wish to customize, and then press the keys you wish to assign to this command.
⇒ The keys are assigned to the applicable command.

**CAUTION**
If multiple assignments are made to the same shortcut key, an error is displayed and the assignments cannot be confirmed. Correct the settings so there are no multiple assignments for the same shortcut key.
D1.4.2 Default Setup
The procedure for returning WideField3 to its default conditions for the key group currently displayed is given below.

◆ Procedure ◆
(1) Open the Customize Keys dialog box.
(2) Click the [Return to Default] button.
⇒ The key settings of the key group currently displayed are returned to their default conditions.

D1.4.3 Clear All
The procedure for clearing all key assignments for the key group currently displayed is given below.

◆ Procedure ◆
(1) Open the Customize Keys dialog box.
(2) Click the [Clear All] button.
⇒ The key settings of the key group currently displayed are all cleared.

D1.4.4 Exporting Key Customizations
The procedure for saving key assignments that have been set to a file is given below.

◆ Procedure ◆
(1) Open the Customize Keys dialog box.
(2) Click the [Export] button.
⇒ The Save As dialog box is displayed.

(3) Enter a file name, and click the [Save] button.
⇒ The key assignment settings are saved to a file.
D1.4.5 Importing Key Customizations

The procedure for loading key assignments that are saved in a file into WideField3 is given below.

◆ Procedure ◆

(1) Open the Customize Keys dialog box.

(2) Click the [Import] button.
⇒ The Open File dialog box opens.

(3) Specify the file to load, and click the [Open] button.
⇒ The key assignment settings are loaded.

TIP
If the saved file contains an error or is corrupt, an error will be displayed and the file will not be loaded.
D2. Overview of Projects

This section gives an overview of a project and describes how to create a project as well as how to open FA-M3 project data saved in other formats. Each project has two types of names as follows:

- Project file name
  A project file name is used when the project is handled on a PC. This is the name assigned to the project file (.YPJT) or project folder when a new project is created.

- Project name (executable program name)
  A project name is used when the project is stored in a CPU module. This name is a string of up to eight characters long that begins with an alphabetical character.

For a project, its project file name and project name are the same each other, and usually you do not need to distinguish one from another.

**TIP**

The functions of the Configuration window in WideField2 are now included in the Project Settings/Configuration window in WideField3. The following table shows the relationship between the previous Configuration window and the Project Settings/Configuration window.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Project Settings/Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device Capacities</td>
<td>[Configuration] - [Device Setup] - [Device Area Setup]</td>
</tr>
<tr>
<td></td>
<td>[Configuration] - [Inter-CPU Shared Memory Setup]</td>
</tr>
<tr>
<td></td>
<td>[Configuration] - [FA Link Setup]</td>
</tr>
<tr>
<td>Power Failure/Local</td>
<td>[Configuration] - [Device Setup] - [Latch Range Setup at Power Failure]</td>
</tr>
<tr>
<td>Operation Control</td>
<td>[Configuration] - [Error Handling Setup]</td>
</tr>
<tr>
<td></td>
<td>[Configuration] - [Run Operation Setup]</td>
</tr>
<tr>
<td>Initial Data</td>
<td>[Configuration] - [Initial Data Setup]</td>
</tr>
<tr>
<td>DIO Setup</td>
<td>[Configuration] - [Input/Output Setup]</td>
</tr>
<tr>
<td>FA Link</td>
<td>[Configuration] - [FA Link Setup]</td>
</tr>
<tr>
<td>Sampling Trace</td>
<td>[Configuration] - [Sampling Trace Setup]</td>
</tr>
<tr>
<td>Communications Setup</td>
<td>[Configuration] - [Built-in Functions Setup]</td>
</tr>
<tr>
<td>ROM Setup</td>
<td>[Configuration] - [Built-in Functions Setup]</td>
</tr>
<tr>
<td>Interrupt Setup</td>
<td>[Configuration] - [Interrupt Setup]</td>
</tr>
<tr>
<td>Shared Refreshing</td>
<td>[Configuration] - [Inter-CPU Shared Memory Setup]</td>
</tr>
<tr>
<td>FL-net Refreshing</td>
<td>[Configuration] - [FA Link Setup]</td>
</tr>
<tr>
<td>Function Removal</td>
<td>[Configuration] - [Built-in Functions Setup]</td>
</tr>
</tbody>
</table>
### Function Limitations for Each CPU Type

#### Table D2.2 Function Limitations for Each CPU Type

<table>
<thead>
<tr>
<th>CPU Modules</th>
<th>Limitations</th>
<th>SEE ALSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>F3SP22-0S, F3SP28-3S, F3SP38-6S, F3SP53-4S, F3SP58-6S, F3SP59-7S</td>
<td>N/A</td>
<td>-</td>
</tr>
<tr>
<td>F3SP66-4S, F3SP67-6S</td>
<td>N/A</td>
<td>-</td>
</tr>
<tr>
<td>F3SP71-4N, F3SP76-7N</td>
<td>N/A</td>
<td>-</td>
</tr>
<tr>
<td>F3SP71-4S, F3SP76-7S</td>
<td>N/A</td>
<td>-</td>
</tr>
</tbody>
</table>

**TIP**

This section does not describe hardware-dependant limitations for each CPU type, such as the number of available devices.

**SEE ALSO**

For details on limitations for each CPU type, refer to the user's manual for each type.
D2.1 What Is a Project?
A project refers to a development environment for creating programs which runs on the FA-M3.

D2.1.1 Relationship between Projects and CPUs
One project is created for each CPU and consists of programs that run on the CPU.

D2.1.2 Configuration of a Project
A project is equivalent to a folder on a PC. A project stores programs (blocks, macros) created within the project, a configuration file, a user log message file, a common tag name definition file, a constant definition file and an executable program file.
It also stores a project file containing management information for the entire project which is used to manage all these data.
### Table D2.3 Files Created in a Project

<table>
<thead>
<tr>
<th>Type</th>
<th>Extension</th>
<th>Description</th>
<th>Tip</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project file</td>
<td>.ypjt</td>
<td>Stores project information, such as the project title and CPU type.</td>
<td></td>
</tr>
<tr>
<td>Executable program file</td>
<td>.yprg</td>
<td>Stores information defining the elements that composes the executable program, such as component blocks.</td>
<td>Only one file having the same name as the project can be generated.</td>
</tr>
<tr>
<td>Project setup file</td>
<td>.yc◯ ¹</td>
<td>Stores the environment for running programs, such as device capacities.</td>
<td>Only one file having the same name as the project can be generated.</td>
</tr>
<tr>
<td>User log message file</td>
<td>.yums</td>
<td>Sets and stores messages stored in the user log.</td>
<td>Only one file having the same name as the project can be generated.</td>
</tr>
<tr>
<td>Common tag name definition file</td>
<td>.ycmn</td>
<td>Stores tag name definition information shared by multiple blocks.</td>
<td>Only one file having the same name as the project can be created.</td>
</tr>
<tr>
<td>Constant definition file</td>
<td>.ycdf</td>
<td>Stores constant definitions.</td>
<td>Only one file having the same name as the project can be created.</td>
</tr>
<tr>
<td>Block files</td>
<td>.yblk</td>
<td>Stores circuits.</td>
<td></td>
</tr>
<tr>
<td>Block tag name definition files</td>
<td>.ysig</td>
<td>Stores tag names used in a block, together with the allocated addresses and I/O comments for the tag names.</td>
<td></td>
</tr>
<tr>
<td>Instruction macro files</td>
<td>.ymcr</td>
<td>Stores macro circuits.</td>
<td></td>
</tr>
<tr>
<td>Macro tag name definition files</td>
<td>.ymcs</td>
<td>Stores tag names used in a macro, together with the allocated addresses and I/O comments for the tag names.</td>
<td></td>
</tr>
<tr>
<td>CPU Property file</td>
<td>.yprp</td>
<td>Download this file to set up the CPU.</td>
<td>This file is generated when an applicable CPU type is selected.</td>
</tr>
<tr>
<td>Long and short names file</td>
<td>.txt</td>
<td>Stores the long file names of project files and block files, and also the corresponding short names of those files used when the files are stored in a CPU module.</td>
<td></td>
</tr>
<tr>
<td>Screen resume file</td>
<td>.yusr</td>
<td>Stores screen states</td>
<td></td>
</tr>
</tbody>
</table>

¹: CPU denotes the numeric portion of a CPU type name. For example, the file extension for the F3SP56-6H CPU module is ".yc58."  

**TIP**  
On the FA-M3, you can split a ladder program into smaller program units by function in your program design. These program units are known as blocks. You can perform program operations such as editing a program, starting (running) or stopping a program on individual blocks.

**SEE ALSO**  
For details on blocks, see "Sequence CPU – Functions."
D2.1.3 Managing Projects in WideField3

Projects can be managed using WideField3 as follows:

Root folder of project

Figure D2.3 Managing Projects Using WideField3

Individual projects may use macros created to be shared among projects, and use templates for grouping tag names.

In addition, a program created in a project can be registered in the common library and the registered programs can be used in another project. Reusing sections of data from projects created previously can also be achieved with ease.

SEE ALSO

- For details on incorporating files from other projects, see Section D3.3, “Managing Files.”
- For details on the use of macros, see Section F1, “Using Macros.”
- For details on the use of group templates, see Section F3, “Using Group Tag Names.”
## D2.1.4 Limitations When a Project is Closed

Some WideField3 functions can be used even when a project is closed whilst others cannot. Basically, offline editing of project data (block/macro edit, Project Settings/Configuration) and other functions cannot be done unless a project is open. Online functions can be used even when a project is closed, but there are some limitations; for instance, tag names cannot be displayed. The following table lists the limitations.

**Table D2.4 Project Limitations**

<table>
<thead>
<tr>
<th>Category</th>
<th>Function</th>
<th>When Project is Open</th>
<th>When Project is Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Initial Setup</strong></td>
<td>Environment setup</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Create/read</td>
<td>Create block (including block tag name definitions)</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>Create instruction macro (including macro tag name definitions)</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>Define program components</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>Configuration</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>Create user log message</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>Edit common tag name definition</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>Edit constant definition</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>Create group template</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>Create structure type definition</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Read system log</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Read user log</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>CPU properties</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Library</strong></td>
<td>Add to Library</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>Change Library Block Name</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>Delete Library Block</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td><strong>Print</strong></td>
<td>Print project</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>Print screen</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Compare</strong></td>
<td>Compare project</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td><strong>Help</strong></td>
<td>Help</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Transfer</strong></td>
<td>Download</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>Upload</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Compare file and CPU</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td><strong>Online control</strong></td>
<td>Connect and Disconnect</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Program monitor</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Macro monitor</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Device monitor</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Registered device monitor</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Configuration</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>ROM management</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Change operating mode</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Extended functions</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Debug/Maintenance</strong></td>
<td>Forced set</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Forced reset</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Change current value</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Change preset value</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Online editing</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Stop refreshing</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Display alarms</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Display system log</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Display user log</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Display operation log</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Extended functions</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Extended functions</strong></td>
<td>Device manager</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Sampling trace</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>E-Mail</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>FA-M3 Defender</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Screen</strong></td>
<td>Save current display status</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>Restore display status</td>
<td>✓</td>
<td>×</td>
</tr>
</tbody>
</table>
D2.2 Creating a Project

This section explains the operations required for creating and managing a project.
- Creating a new project
- Opening a project
- Project edit history
- Closing a project
- Saving a project
- Saving a project with a different name
- Opening projects in other formats
- Saving projects in other formats
- Saving the current screen states
- Opening a project (and restoring the screen states)
- Redisplaying the current project with saved screen states

D2.2.1 Creating a New Project

When you create a new project, a folder with the same name as the project is created
and the files that make up the project are automatically generated. The table below lists
the files that are generated.

Table D2.5 Files in a Project

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project file</td>
<td>A file containing the CPU type and project title is generated. A generated folder with the project file missing will not be recognized as a project.</td>
<td>—</td>
</tr>
<tr>
<td>Executable program file</td>
<td>A default executable program file is generated.</td>
<td>Configuration: Setup User log message: No Component blocks: None</td>
</tr>
<tr>
<td>Project setup file</td>
<td>A project setup file for the specified CPU type is generated when a new project is created.</td>
<td>Default values for each CPU type</td>
</tr>
<tr>
<td>User log message file</td>
<td>A default user log message file is generated.</td>
<td>All user log messages are empty.</td>
</tr>
<tr>
<td>Common tag name definition file</td>
<td>A common tag name definition file containing empty I/O comments is generated.</td>
<td>When a project is created, if the [Read default common tag name definition] checkbox is turned on, a file containing I/O comments for special relays (M) and special registers (Z) is generated. If the checkbox is off, an empty file is created.</td>
</tr>
<tr>
<td>Constant definition file</td>
<td>A file containing default constant definitions is generated.</td>
<td>An empty constant definition file is generated.</td>
</tr>
<tr>
<td>CPU Property file</td>
<td>A file containing default property values is generated. (only for applicable CPUs)</td>
<td>—</td>
</tr>
<tr>
<td>Long and short names file</td>
<td>Stores the long file names of project files and block files, and also the corresponding short names of those files used when the files are stored in a CPU module.</td>
<td>—</td>
</tr>
<tr>
<td>Screen resume file</td>
<td>Stores screen states.</td>
<td>—</td>
</tr>
</tbody>
</table>
Creating a New Project

The procedure for creating a new project is given below.

◆ Procedure ◆

(1) Select [File]–[New Project] from the menu.
⇒ The New Project dialog box is displayed.

(2) Enter the project name in the [Project Name] text box.

TIP
When you enter a project name, the entry is automatically reflected to the project file name.

(3) Select a CPU type from the [CPU Type] drop-down list.

(4) Enter the project title into the [Project Title] text box.

TIP
Entering a project title is optional.

(5) If you want to use default common tag name definitions, turn on the [Read default common tag name definition] checkbox.

(6) Click [New].
⇒ A new project is created and the project window is displayed.
TIP

- The project window displays the configuration of a project. Double-clicking an icon in the project window opens a corresponding edit window or dialog box.

**Figure D2.4  Project Window**

- When default common tag name definitions are loaded in a project, tag name definitions for special relays (M) and special registers (Z) are registered.

CAUTION

- A project created using WideField3 cannot be opened using WideField or WideField2. To open a WideField3 project in WideField or WideField2, you need to first convert the project to WideField or WideField2 project format. To do so, open the project file in WideField3 and select [File]–[Save Project in Other Formats]–[Downgrade and Save] from the menu bar.
D2.2.2 Opening a Project

Opening a Project

Opening a project and performing any of the following operations will update the date of the project.

- Changing the CPU type, project title, or project file name.
- Changing the maximum number of common tag name definitions allowed.
- Setting or removing password protection for a project.
- Changing the setting for storing common tag name definition to the CPU.
- Changing the executable program component definition.
- Changing the project configuration.
- Changing a user log message.
- Changing the common tag definition data
- Changing constant definition data
- Changing a block/macro (including changing a block/macro tag name definition)
- Running the Registered Device Monitor or performing printing

Performing any of the above operations after opening a WideField or WideField2 project file in WideField3 will also have the effect that the same project file can no longer be opened in WideField or WideField2. Even if you open a WideField or WideField2 project file in WideField3, if you do not perform any of the above operations, you can still open the file in WideField or WideField2.

The procedure for opening an existing project is given below.

◆ Procedure ◆

(1) Select [File]–[Open Project] from the menu.
⇒ The Open Project dialog box is displayed.
(2) Double-click the folder with the same name as the project.
⇒ The project files are displayed.
(3) Click on the project file to be opened.
⇒ Information in the project file is displayed on the right.
TIP
A project file contains the following information.
- Date the project file was last updated
- CPU type
- Project title

Only the first 16 characters of the project title are displayed.

(4) Click [Open].
⇒ The project is opened and the project window is displayed.

CAUTION
- If the executable program in a project is protected, the Password Confirmation dialog is displayed.
  Enter your password and click [OK]. The project is opened.

![Password Confirmation Dialog](D2012_05_VSD)

Figure D2.5 Password Confirmation Dialog

- You cannot open a project while a macro/block monitor window is open.
- With multiple copies of WideField3 running concurrently, you can open different projects concurrently, but you cannot open the same project in more than one WideField3 application.
- A project data file that is read-only cannot be opened. Beware of this point especially when handling project data saved on CD-R/RW and other memory media.
### Opening a Project Created in WideField, WideField2, or WideField3R1

In WideField3, you can open a project created in WideField or WideField2. When the project is opened, the software automatically converts the file to a format usable in WideField3, taking care of version differences.

#### Opening a Project Created in WideField

The following table summarizes the conversion performed for different user file types.

<table>
<thead>
<tr>
<th>File Type</th>
<th>File Extension</th>
<th>Overview of Conversion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common tag name definition file</td>
<td>.ycmn</td>
<td>The format is changed.</td>
</tr>
<tr>
<td>Block file</td>
<td>.yblk</td>
<td>Data is converted to tag name format. All period (&quot;.&quot; characters are changed to underscore (&quot;_&quot;) characters.</td>
</tr>
<tr>
<td>Block tag name definition file</td>
<td>.ysig</td>
<td>The format is changed. All period (&quot;.&quot; characters in tag names are changed to underscore (&quot;_&quot;) characters.</td>
</tr>
</tbody>
</table>

If a block was previously saved with errors, the block file is not converted but the block tag name definition file is converted.

Even if a block file is not converted, you can open and edit the file in WideField3. An unconverted block file remains as a block file in address format created using WideField, and is displayed in address format in the WideField3 Edit Block window.

**TIP**

To display parameters in tag name format in the Edit Block window, fix all errors in a block saved with errors, then select [Project]–[Convert Data]–[Tag Name Format] to convert the block file to tag name format.

Saving a block with errors refers to saving a block without correcting program errors in the block.

#### Opening a Project Created in WideField2

In WideField3, you can open a project created in WideField2.

The following table shows the files that are added when you open the project.

<table>
<thead>
<tr>
<th>File Type</th>
<th>File Extension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Screen resume file</td>
<td>.YUSR</td>
<td>Stores screen states.</td>
</tr>
<tr>
<td>Long and short names file</td>
<td>.TXT</td>
<td>Stores the long file names and also the corresponding short file names used when the files of a project are stored in a CPU module.</td>
</tr>
</tbody>
</table>

#### Opening a Project Created in WideField3R1

The script function is newly added to WideField3R2. As WideField3R1 cannot open script programs, version numbers of project files differ between WideField3R2 and WideField3R1. In WideField3R2, you can open a project created in WideField3R1, but in WideField3R1, you cannot open a project created in WideField3R2. To open a WideField3R2 project in WideField3R1, you can use WideField3R2 to convert the project into a downgraded format usable by WideField3R1.
D2.2.3  Project Edit History

You can open a project from the project edit history (projects opened recently) displayed on the [File] menu. The relevant procedure is given below.

◆ Procedure ◆

(1) Select [File] from the menu.
⇒ A list of projects recently opened is displayed.

(2) Click the project file to be opened.
⇒ The project file is opened.

TIP

The five files most recently opened are displayed.
D2.2.4 Closing a Project

The procedure for closing a project is given below.

◆ Procedure ◆

(1) Select [File]–[Close Project] from the menu

⇒ If you are editing a block, an instruction macro, a common tag name definition or constant definition, the respective confirmation dialog box is displayed.

TIP

The figure shows an example of the message displayed when editing the "MAIN.YBLK" block.

(2) Click [Yes] on the confirmation dialog box. To save all the files being edited in the subsequent screens, turn on the [Select [Yes] to save in the subsequent screens.] checkbox before clicking the [Yes] button.

⇒ The project is closed and the project window becomes empty.

TIP

- Clicking [Yes] saves the changes and closes the project.
- Clicking [No] closes the project without saving the changes.
- Clicking [Cancel] cancels the close operation.

TIP

Turning on [Select [Yes] to save in the subsequent screens.] will save files for blocks, instruction macros, common tag name definitions, constant definitions, project settings, structure type definitions, and group templates being edited.
D2.2.5  Saving a Project
The procedure for overwriting and saving a project is given below.

◆ Procedure ◆

(1) Select [File]–[Save Project] from the menu
⇒ All blocks, instruction macros, common tag name definition and constant definition being edited are saved.

TIP
You can continue editing after saving a project.
D2.2.6 Saving a Project with a Different Name

You can save all displayed project contents using a new project name (creating a new project). The procedure is given below:

◆ Procedure ◆

(1) Select [File]–[Save Project As] from the menu.
⇒ The Save Project as dialog box is displayed.

(2) Enter the new project name into the [Project Name] text box and click [Save].
⇒ The project is created and saved with the new name.

(3) A confirmation dialog is displayed. To continue your edit operation on the saved project (named in step 2), click [Yes]. To edit the currently opened project, click [No].
⇒ Continue your edit operation on the specified project.

CAUTION

- If errors are detected while saving each block, confirmation dialog boxes may be displayed to confirm whether to save the file with conversion errors or whether to delete tag name definition lines containing errors.

  Clicking [Yes] saves the project after all open blocks are saved. Clicking [No] cancels the save operation.
D2.2.7 Opening Projects in Other Formats

You can convert FA-M3 project data created and saved (or compressed) outside WideField3 into a format usable by WideField3 and open it as a project. The table below lists the kinds of project data that can be converted and opened using WideField3.

Table D2.8 Convertible Project Data

<table>
<thead>
<tr>
<th>Project Type</th>
<th>Overview</th>
<th>File Extension</th>
</tr>
</thead>
<tbody>
<tr>
<td>Card load format</td>
<td>Opens project data backed up to SD card from a CPU module and saved in card load format using WideField3.</td>
<td>.ypjc</td>
</tr>
<tr>
<td>Compressed and segmented project</td>
<td>Opens project data compressed and saved using WideField3. Restores and opens a compressed project that was previously segmented and saved onto multiple floppy disks.</td>
<td>.yp01, .yp02, …</td>
</tr>
<tr>
<td>CADM3 format</td>
<td>Opens an executable program created by CADM3.</td>
<td>.prg</td>
</tr>
</tbody>
</table>

■ Opening a Project in Card Load Format

The procedure for converting into WideField3 format project data saved in card load format from a CPU module via an SD card and then opening the project is given below.

◆ Procedure ◆

(1) If a project is open, close it.
(2) Select [File]–[Open Project in Other Formats]–[Open Project in Card Load Format] from the menu.
   ⇒ The Open File dialog box is displayed.
(3) Move to the folder containing the project data to be imported and select the file.
(4) Click [Open].
   ⇒ Conversion of project data begins. If conversion is successful, the newly created project is opened.

TIP

The project file name of the converted project is based on the project file name used when the project data was saved in card load format. To use a different project file name, specify a new project file name in the project selection dialog.
Opening a Compressed Segmented Project

A compressed project file can be restored.

If a project was previously split and saved onto multiple floppy disks, standby all the floppy disks before performing restoration. The procedure is given below.

⚠️ CAUTION ⚠️

A project compressed or segmented in WideField3 cannot be opened by programs other than WideField3.

◆ Procedure ◆

1. If a project is open, close it.
2. Select [File]—[Open Project in Other Formats]—[Open Compressed Segmented Project] from the menu.
   ⇒ The Select Project to Restore dialog box is displayed.
3. Select the file to be restored.
4. Click [Open].
   ⇒ A confirmation dialog box is displayed.

(5) Click [OK].
   ⇒ Project decompression begins. The Restoration Status dialog box is displayed during project restoration.

TIP

If the project was previously split and saved onto multiple floppy disks, a dialog box is displayed during restoration to confirm whether each floppy disk has been replaced in the drive. Follow the screen instruction, replace the floppy disk and click [OK].

⇒ When project restoration is completed, a completion dialog box is displayed.
Opening a CADM3 Executable Program

To use an executable program of Ladder Diagram Support Program M3 (SF510-J3N, SF510-J3P) format in WideField3, the program must be loaded into WideField3. Only executable programs that do not generate errors during program syntax check can be opened.

SEE ALSO
For details on how to open individual block files, see Chapter D4, "Creating and Managing Blocks and Macros."

CAUTION
Always perform CADM3 program syntax check before using an executable program created using CADM3. Otherwise, defined tag name definitions will be disabled after syntax check.

Generated Files

When an executable program of Ladder Diagram Support Program M3 format is opened in WideField3, it will be automatically converted into a WideField3 format file. The following table lists the files that are generated during conversion.

<table>
<thead>
<tr>
<th>Category</th>
<th>File</th>
<th>Extension</th>
<th>Overview</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project-related files</td>
<td>Project file</td>
<td>.ypjt</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Executable program file</td>
<td>.yprg</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Common tag name definition file</td>
<td>.ycmn</td>
<td>All block tag name definitions of registered blocks are combined into a common tag name definition file.</td>
</tr>
<tr>
<td></td>
<td>Project setup file</td>
<td>.yc□□*1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Long and short names file</td>
<td>.txt</td>
<td>This file stores the file names used on the PC and also the corresponding file names used when the files are stored in the CPU module.</td>
</tr>
<tr>
<td></td>
<td>User log message file</td>
<td>.yums</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Screen resume file</td>
<td>.yusr</td>
<td>This file stores screen states.</td>
</tr>
<tr>
<td>Block-related files</td>
<td>Block file</td>
<td>.yblk</td>
<td>File is generated in tag name format. The reference tag name definition is set to common tag name definition.</td>
</tr>
<tr>
<td></td>
<td>Block tag name definition file</td>
<td>.ysig</td>
<td>An empty file is created.</td>
</tr>
</tbody>
</table>

*1: □□ denotes the numeric portion of a CPU type name. For example, the file extension for the F3SP58-6H CPU module is "yc58."
Opening an Executable Program in CADM3 Format

The procedure for opening an executable program of Ladder Diagram Support Program M3 format is given below.

◆ Procedure ◆

(1) If a project is open, close it.
(2) Select [File]—[Open File in Other Formats]—[Open CADM3 Executable Program] from the menu.
   ⇒ The Open CADM3 Executable Program dialog box is displayed.
(3) Move to the folder containing the CADM3 executable program and select the file.
(4) Click [Open].
   ⇒ The New Project dialog box is displayed.
(5) Enter the name of the new project in the [Project Name] text box.
(6) Select the CPU type from the CPU Type drop-down list.
   TIP
   If the CPU type is changed, a confirmation dialog box is displayed. Clicking [Yes] returns the display to the New Project dialog box.
(7) Click [New].
   ⇒ The CADM3 format executable program, as well as its component blocks and tag name definitions are converted into WideField3 format. The conversion status is displayed in the Open CADM3 Executable Program dialog box. When conversion and reading are completed, a confirmation dialog box is displayed.
(8) Click [OK].
⇒ The confirmation dialog box closes.

(9) Click [Close].
⇒ The Open CADM3 Executable Program dialog box closes.
If conversion ends successfully, the newly created project is opened.

CAUTION

- Configuration contains elements for configuring an executable program. The CPU type is one of these elements. Configuration data is inherited only if the CPU Type is changed to another CPU type within the same model group. Otherwise, you need to create a new configuration and modify it. If the CPU type is changed to a CPU type belonging to a different model group, the configuration data either changes to the previous configuration of the new CPU type, or reverts to default. The table below lists the CPU types in each model group.

<table>
<thead>
<tr>
<th>Model Group Name</th>
<th>CPU Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group 1</td>
<td>F3SP05, F3SP08, F3SP21</td>
</tr>
<tr>
<td>Group 2</td>
<td>F3SP25, F3SP35, F3FP36</td>
</tr>
<tr>
<td>Group 3</td>
<td>F3SP20, F3SP22, F3SP28, F3SP38, F3SP53, F3SP58, F3SP59, F3SP66, F3SP67, F3SP71, F3SP76</td>
</tr>
</tbody>
</table>

- If you change the CPU type to a CPU type belonging to a different model group, a confirmation dialog box is displayed.

- Even if configuration data is inherited, the allowable value range for each setup item may differ between CPU types. Hence, errors may be reported during program syntax checking.

- When the CPU type is changed, WideField3 creates a configuration with default values for the new CPU type. You should set and adjust the configuration setup for the new CPU type.

- The CPU type can be changed even after conversion. Print the configuration information of the old CPU type from WideField3. Select [Project]–[Project Settings]–[CPU Type Settings] and change the CPU type. Compare the configuration setup of the new CPU type with the printout and edit it accordingly.

- Ladder blocks saved by mnemonic editing cannot be converted. Save and amend blocks using the CADM3 ladder program edit function before performing conversion.

- The result of the conversion is saved in text format in a log file (HENKAN.log) in the project folder.

- You cannot open a CADM3 format executable program while a block monitor window or macro monitor window is open.

- You cannot open a CADM3 format executable program while an offline project is open.
D2.2.8 Saving Projects in Other Formats

You can save project data created using WideField3 into a format usable outside of WideField3.
The table below lists the kinds of project data that can be converted and saved.

<table>
<thead>
<tr>
<th>Project Type</th>
<th>Overview</th>
<th>File Extension</th>
</tr>
</thead>
<tbody>
<tr>
<td>Downgraded format</td>
<td>Converts and saves project data in a format usable by WideField or WideField2, the predecessor of WideField3.</td>
<td>.ypjt, etc.</td>
</tr>
<tr>
<td>Card load format</td>
<td>Converts and saves project data to an SD card in a data format usable by the CPU module.</td>
<td>.ypjc</td>
</tr>
<tr>
<td>Compressed format</td>
<td>Compresses and saves project data</td>
<td>.yp01</td>
</tr>
<tr>
<td>Compressed and Segmented to Floppy Disks</td>
<td>Compresses project data, and then segments and saves the data on floppy disks.</td>
<td>.yp01, .yp02, ...</td>
</tr>
</tbody>
</table>

**Downgrading and Saving a Project**

The Downgrade and Save Project function converts and saves a WideField3 project in a data format usable in WideField or WideField2, the predecessor of WideField3. The procedure is given below.

![Screenshot](D0202_21.VSD)

**CAUTION**

In WideField3R1, you cannot open a project created in WideField3R2. To use a WideField3R2 project in WideField3R1, select [Convert to WideField3R1 format] to convert and save the project in a downgraded format usable by WideField3R1.

**Procedure**

1. **Open the project and select**
   [File]–[Save Project in Other Formats]–[Downgrade and Save] from the menu.
   
   ⇒ Downgrading and saving operates on saved files. If you initiate downgrading with an open ladder program edit window (tag name definition window) after having made changes in the window, a dialog box is displayed to confirm whether to first save the edited file in WideField3 format.

2. **To save in WideField3 format before downgrading, click [Yes]; to perform downgrading without first saving the changes, click [No].**
   
   ⇒ The Downgrade and Save Project dialog box is displayed.
(3) Select the destination folder for saving the project after conversion, as well as the target conversion format.
Click [Convert].

**TIP**
Clicking [Browse] displays a dialog box for folder selection.

---

**CAUTION**

The destination folder cannot be the same as the location of the project being edited.
Always specify a different folder name.

---

**TIP**
If the CPU Type of the project does not support the selected downgraded version, an error message is displayed and conversion is aborted.

⇒ The Program Check dialog box is displayed and program syntax checking begins.

**TIP**
Conversion is aborted if any error is detected.
Conversion continues if no error is detected or if only warnings are detected.

⇒ Next, the Status of Version Conversion dialog box is displayed, and conversion begins.

**TIP**
If an error is detected, the error is displayed and the conversion is aborted.

⇒ When conversion is completed, a completion dialog box is displayed.

---

**CAUTION**

If an error is detected during downgrade conversion, the error file name is displayed in a dialog box. Error information is displayed as and when errors are detected so if there are multiple errors within a project, they are not all displayed at the same time.
Saving a Project in Card Load Format

You can convert and save project data on a SD card data format usable by the CPU module.

CAUTION

- A project saved by WideField3 in card load format cannot be opened by programs other than WideField3.
- To save a WideField3 project in WideField2 compatible card load format, downgrade and save the project using WideField3 in WideField2 format, and then open the project in WideField2 and save it in card load format.
- To save comments or tag name definitions in card load format, you must specify to store comments and tag name definitions in advance.
- If a script uses any tag name, saving project data in card load format without tag name definitions leaves tag names in the script as they are, instead of being converted to the corresponding addresses. This causes a syntax check error of the project in card load format. Therefore, store the tag name definitions when a script uses any tag name.

TIP

For details on storing comments and tag name definitions, see Chapter H9, "Storing Comments and Tag Name Definitions" (Online).

◆ Procedure ◆

1. Open the project to be saved in WideField3.
2. Select [File]–[Save Project in Other Formats]–[Save Project in Card Load Format] from the menu.
   ⇒ The Save Project in Card Load Format dialog box is displayed.

3. Turn on the checkboxes of the items to be saved in card load format, and click [OK].
   ⇒ The Save As dialog box is displayed.

4. Enter a destination filename and click [Save].
   ⇒ Conversion of project data begins.
Compressing and Saving a Project to Hard Disk

You can compress and save a project to a hard disk.
All projects must be closed before compression and saving.

CAUTION

A project can only be compressed and saved to:
- hard disk;
- or floppy disks

To compress and save a project to other memory media, you can first compress and save the project to a hard disk and then copy the saved file to the required memory media.

◆ Procedure ◆

(1) If a project is open, close it.
(2) Select [File]–[Save Project in Other Formats]–[Compress and Save] from the menu.
⇒ The Compress and Save Project dialog box is displayed.

(3) Specify the project to be compressed and saved, as well as the destination folder for saving the project.

TIP
You can click [Browse] and select the project to be compressed and saved from the displayed Select Project dialog box.

(4) Click [Save].
⇒ Project compression begins. The Compression Status dialog box is displayed during project compression. After compression is completed, saving begins. The Transfer Status dialog box is displayed during project saving. A confirmation dialog box is displayed when saving is completed.
Clicking [Stop] on the Compression Status dialog box or the Transfer Status dialog box aborts compression and saving.

(5) Click [OK].
⇒ The confirmation dialog box closes.

The compressed project file is saved with the name “Project Name.yp01”.
Compressing and Saving a Project to Floppy Disks

When saving a project to floppy disks, the project may be split into smaller pieces after compression and saved to multiple floppy disks if the project size exceeds the capacity of a floppy disk.

All projects must be closed before compression and saving.

**CAUTION**

A project can only be compressed and saved to:
- hard disk;
- or floppy disks

To compress and save a project to other memory media, you can first compress and save the project to a hard disk and then copy the saved file to the required memory media.

◆ Procedure ◆

1. If a project is open, close it.
2. Select [File]–[Save File in Other Formats]–[Compress and Save] from the menu.

⇒ The Compress and Save Project dialog box is displayed.

3. Specify the project to be compressed and saved, as well as a destination folder.

   **TIP**

   You can click [Browse] and select the project to be compressed and saved from the displayed Select Project dialog box. Specify the floppy disk as destination only if a floppy disk is inserted in the drive.

   **CAUTION**

   Always use floppy disks of identical capacities when saving a project on multiple floppy disks as the software calculates the number of disks required based on the capacity of the first floppy disk inserted in the drive.
(4) Click [Save].
⇒ The Compression Status dialog box is displayed during compression. A dialog box is then displayed to confirm the number of disks required based on the size of the compressed project.

(5) Click [OK].
⇒ A dialog box is displayed to confirm whether the first disk has been inserted.

(6) Check that a new floppy disk has been inserted in the drive and click [OK].
⇒ Saving begins. The Transfer Status dialog box is displayed, followed by the [Copying] dialog box during the save operation. If a project needs to be saved on multiple floppy disks, a dialog box is displayed to confirm that the next floppy disk has been inserted.

(7) Step 6 is repeated for each floppy disk.
⇒ When saving is completed, a confirmation dialog is displayed.
(8) Click [OK].
⇒ The confirmation dialog box closes.

**TIP**

Project files that are split and compressed are saved with names "Project File Name.yp01", "Project File Name.yp02", etc. If [Stop] is clicked during project saving, the compressed files may still remain in the work folder.
D2.2.9  Saving the Current Screen States

You can save the current screen states including the screen positions, screen sizes, scroll positions, and cursor positions of the program edit screen, program monitor screen, device monitor screen, and so on.

SEE ALSO

- For details on opening a project with saved screen states, see Section D2.2.10, "Opening a Project (and Restoring Screen States)."
- For details on redisplaying the current project with saved screen states, see Section D2.2.11, "Redisplaying the Current Project with Saved Screen States."

To save the current screen states, use the following procedure.

◆ Procedure ◆

(1) Select [File] - [Save Current Display Status] from the menu bar.
⇒ The current screen states are saved.

D2.2.10  Opening a Project (and Restoring the Screen States)

You can open a project with saved screen states.

SEE ALSO

- For details on saving the current screen states, see Section D2.2.9, "Saving the Current Screen States."
- For details on redisplaying the current project with saved screen states, see Section D2.2.11, "Redisplaying the Current Project with Saved Screen States."

To open a project with saved screen states, use the following procedure.

◆ Procedure ◆

(1) Select [File] - [Open Project (Restore Display Status)] from the menu bar.
⇒ The project is opened and the screen states are restored.

TIP

- The display conditions and display mode are also restored for the ladder program edit screen, program monitor screen, tag name definition screen, tag name definition monitor screen, and device monitor screen.
- The selection state is also restored for the CPU Properties window.

CAUTION

The screen positions, scroll positions, cursor positions, and selection states are restored only when the project is opened offline.
CAUTION

- Even if the [Display on top left of display area] checkbox is turned on in [Window Display Position] on the [Circuit Display/Input] tab on the Set up Environment dialog box, each screen is displayed with the restored screen state.
- If you restore previous screen states that were saved offline, the current communication port number in the environment setup is changed to the port number included in the previous screen states. If you restore screen states that were saved online, the current port number is not changed.

D2.2.11 Redisplaying the Current Project with Saved Screen States

You can redisplay the screens of the current project using the screen states saved before.

SEE ALSO

- For details on saving the current screen states, see Section D2.2.9, "Saving the Current Screen States."
- For details on opening a project with saved screen states, see Section D2.2.10, "Opening a Project (and Restoring Screen States)."

To redisplay the screens of the current project with saved screen states, use the following procedure.

◆ Procedure ◆

(1) Select [File] - [Restore Display Status] from the menu bar.
⇒ Each screen is redisplayed with the screen state restored.

TIP

- The display conditions and display mode are also restored for the ladder program edit screen, program monitor screen, tag name definition screen, tag name definition monitor screen, and device monitor screen.
- The selection state is also restored for the CPU Properties window.

CAUTION

- Even if the [Display on top left of display area] checkbox is turned on in [Window Display Position] on the [Circuit Display/Input] tab on the Set up Environment dialog box, each screen is displayed with the restored screen state.
- If you restore previous screen states that were saved offline, the current communication port number in the environment setup is changed to the port number included in the previous screen states. If you restore screen states that were saved online, the current port number is not changed.
- If a currently opened screen was not open when screen states were saved, restoring screen states does not affect the current state of the screen.
- When WideField3 is online, even if you restore the screen states that were saved offline, WideField3 remains online.
This chapter describes how to build and manage a project. Building or managing a project requires that you first open a project. All software operations in this chapter should be performed with an open project.

**TIP**

The functions of the Configuration window in WideField2 are now included in the Project Settings/Configuration window in WideField3. The following table shows the relationship between the previous Configuration window and the Project Settings/Configuration window.

### Table D3.1  Relationship between the Configuration Window and Project Settings/Configuration Window

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Project Settings/Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device Capacities</td>
<td>[Configuration] - [Device Setup] - [Device Area Setup]</td>
</tr>
<tr>
<td></td>
<td>[Configuration] - [Inter-CPU Shared Memory Setup]</td>
</tr>
<tr>
<td></td>
<td>[Configuration] - [FA Link Setup]</td>
</tr>
<tr>
<td>Power Failure/Local</td>
<td>[Configuration] - [Device Setup] - [Latch Range Setup at Power Failure]</td>
</tr>
<tr>
<td>Operation Control</td>
<td>[Configuration] - [Error Handling Setup]</td>
</tr>
<tr>
<td></td>
<td>[Configuration] - [Run Operation Setup]</td>
</tr>
<tr>
<td>Initial Data</td>
<td>[Configuration] - [Initial Data Setup]</td>
</tr>
<tr>
<td>D/O Setup</td>
<td>[Configuration] - [Input/Output Setup]</td>
</tr>
<tr>
<td>FA Link</td>
<td>[Configuration] - [FA Link Setup]</td>
</tr>
<tr>
<td>Sampling Trace</td>
<td>[Configuration] - [Sampling Trace Setup]</td>
</tr>
<tr>
<td>Communications Setup</td>
<td>[Configuration] - [Built-in Functions Setup]</td>
</tr>
<tr>
<td>ROM Setup</td>
<td>[Configuration] - [Built-in Functions Setup]</td>
</tr>
<tr>
<td>Interrupt Setup</td>
<td>[Configuration] - [Interrupt Setup]</td>
</tr>
<tr>
<td>Shared Refreshing</td>
<td>[Configuration] - [Inter-CPU Shared Memory Setup]</td>
</tr>
<tr>
<td>FL-net Refreshing</td>
<td>[Configuration] - [FA Link Setup]</td>
</tr>
<tr>
<td>Function Removal</td>
<td>[Configuration] - [Built-in Functions Setup]</td>
</tr>
</tbody>
</table>
### Table D3.2 Function Limitations for Each CPU Type

<table>
<thead>
<tr>
<th>CPU Modules</th>
<th>Limitations</th>
<th>SEE ALSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>F3SP22-0S,</td>
<td>In operation setup, the valid range for constant scan is from 1.0 ms to 190.0 ms.</td>
<td>D3.1.6</td>
</tr>
<tr>
<td>F3SP28-3S,</td>
<td>In interrupt setup, the valid range for the execution interval of the sensor CB is from 0.2 ms to 25.0 ms.</td>
<td>D3.1.9</td>
</tr>
<tr>
<td>F3SP38-6S,</td>
<td>In interrupt setup, the range and increment unit of the execution interval are limited.</td>
<td>D3.1.10</td>
</tr>
<tr>
<td>F3SP53-4S,</td>
<td>In built-in functions setup, [Deletion Setup for Built-in Functions] is unavailable.</td>
<td>D3.1.10</td>
</tr>
<tr>
<td>F3SP56-6S,</td>
<td>The script setup is unavailable.</td>
<td>G4.1</td>
</tr>
<tr>
<td>F3SP99-7S</td>
<td>In operation setup, the valid range for constant scan is from 1.0 ms to 190.0 ms.</td>
<td>D3.1.6</td>
</tr>
<tr>
<td>F3SP66-4S,</td>
<td>In interrupt setup, the valid range for the execution interval of the sensor CB is from 0.2 ms to 25.0 ms.</td>
<td>D3.1.9</td>
</tr>
<tr>
<td>F3SP67-6S</td>
<td>In interrupt setup, the range and increment unit of the execution interval are limited.</td>
<td>D3.1.10</td>
</tr>
<tr>
<td></td>
<td>In built-in functions setup, [Settings for Keeping Register Data Resident in ROM Pack] is unavailable.</td>
<td>D3.1.10</td>
</tr>
<tr>
<td></td>
<td>The script setup is unavailable.</td>
<td>G4.1</td>
</tr>
<tr>
<td>F3SP71-4N,</td>
<td>In operation setup, the valid range for constant scan is from 0.1 ms to 190.0 ms.</td>
<td>D3.1.6</td>
</tr>
<tr>
<td>F3SP76-7N</td>
<td>In interrupt setup, the valid range for the execution interval of the sensor CB is from 0.10 ms to 25.0 ms.</td>
<td>D3.1.9</td>
</tr>
<tr>
<td></td>
<td>In operation setup, built-in functions setup cannot be performed.</td>
<td>D3.1.10</td>
</tr>
<tr>
<td></td>
<td>FA Link setup is unavailable.</td>
<td>D3.1.14</td>
</tr>
<tr>
<td></td>
<td>Sampling trace setup cannot be performed from the Project Settings/Configuration window.</td>
<td>D3.1.15</td>
</tr>
<tr>
<td>F3SP71-4S,</td>
<td>In operation setup, the valid range for constant scan is from 0.1 ms to 190.0 ms.</td>
<td>D3.1.6</td>
</tr>
<tr>
<td>F3SP76-7S</td>
<td>In interrupt setup, the valid range for the execution interval of the sensor CB is from 0.10 ms to 25.0 ms.</td>
<td>D3.1.9</td>
</tr>
<tr>
<td></td>
<td>In operation setup, built-in functions setup cannot be performed.</td>
<td>D3.1.10</td>
</tr>
<tr>
<td></td>
<td>Sampling trace setup cannot be performed from the Project Settings/Configuration window.</td>
<td>D3.1.15</td>
</tr>
</tbody>
</table>

**TIP**

This section does not describe hardware-dependant limitations for each CPU type, such as the number of available devices.

**SEE ALSO**

- For details on limitations for each CPU type, refer to the user's manual for each type.
- For details on script setup, see Chapter G4, "Setting a Script in a Project" (Script).
D3.1 Building a Project

This section describes how to set up various settings to build a project.

**Project Settings/Configuration Window**

You can use the Project Settings/Configuration window to set up the settings for building a project. The Project Settings/Configuration window shows the setup items in tree view format.

Selecting a setup item in the tree switches the current screen to the screen for the selected item.

![Project Settings/Configuration Window](image)

**Figure D3.1 Project Settings/Configuration Window**

- **A** The setup item currently being edited.
- **B** The pane for setting up the settings.
- **C** The tree pane for the project setup items.
- **D** The tree pane for the configuration setup items.
- **E** Closes the window and enables the settings.
- **F** Closes the window without updating the settings.
- **G** Restores the currently displayed settings to defaults.
- **H** Displays help information.

**TIP**

"Configuration" is a function that allows you to preset features such as device capacities and operation method for a sequence CPU module in accordance with its intended use. The configuration setup items vary depending on the CPU type.
CAUTION

There are script setup items in the Project Settings/Configuration window in addition to the above setup items described in this manual. For details on script setup, see the volume entitled “Script.”

TIP

Each color used for folder icons in the tree panes has the following meaning.
- Blue: Configuration items or CPU properties
- Red: Setup items related to CPU operations
- Yellow: Setup items related to the project

To open the Project Settings/Configuration window and perform setup, use the following procedure.

◆ Procedure ◆

(1) Select [Project] - [Project Settings] from the menu bar.
⇒ The Project Settings/Configuration window is displayed.
(2) Select a setup item in a tree pane on the left side of the window.
⇒ The setup screen for the selected setup item is displayed.
(3) Modify the settings.
(4) After modifying the settings, click [OK].
⇒ The Project Settings/Configuration window is closed and the modified settings are saved.

TIP
To restore the settings on the screen to their default values, click [Default] and then click [Yes] in the displayed dialog box.
D3.1.1 Project Settings

Perform project settings to specify the project name, executable program name, and save destination folder of the project currently opened in WideField3. You can also check the detailed information of the execution block.

### Project Settings Screen

![Project Settings Screen](image)

- **A** Displays the project file name.
- **B** Displays the project title. You can edit the project title.
- **C** The executable program name is the same as the project file name.
- **D** Specify whether to use the configuration setup. If you use it, turn on the [Use] option button. If you do not use it, turn on the [Do Not Use] option button.
- **E** Updates the Execution Blocks Information.
- **F** Opens the Executable Program Details/Setup dialog box. On this dialog box, you can check or set the number of local devices to use.
CAUTION

If you select not to use the configuration, the program execution mode is always [Sequential Execution of All Blocks] even if you select [Specified Blocks].

SEE ALSO
For details on the program execution mode, see Section D3.1.3, "Executable Program Setup."

TIP
- From the Project Settings/Configuration window, you can open a dialog box for setting the number of local devices for each execution block. You can set the number of local devices to be used in the Executable Program Details/Setup dialog box. To open the Executable Program Details/Setup dialog box, click the [Common Tag Name Definitions/Local Device Detailed Setup] button in the Execution Blocks Information group box.
- If you turn on the [Do Not Use] option button for [Configuration: Use/Do not use], all settings of the configuration setup items remain their defaults. Therefore, the local device settings are disabled. In addition, the setup items in the [Configuration] tree are grayed out and cannot be changed.

SEE ALSO
For details on the settings in the Executable Program Details/Setup dialog box, see Section D3.1.16, "Component Block Setup."
D3.1.2 CPU Type Settings

Perform CPU type settings to specify the CPU type to be used in the project.
You can perform CPU type settings only when the Project Settings/Configuration
window is open. Before performing your operations, open the Project
Settings/Configuration window.

CPU Type Settings Screen

- A Opens a dialog box that shows the number of devices and device ranges for the
currently selected CPU type.
- B Displays a brief summary of the specifications of the currently selected CPU
type.
- C Displays a front image of the currently selected CPU type.
- D List of CPU types. Select the CPU type to be used in the project.

CAUTION

- After you selected a CPU type, unless you click the [OK] button to determine the
selected CPU type, you cannot display other screens for setting up the project.
- A program configuration contains elements for configuring an executable program.
The CPU type is one of these elements. Configuration setup is inherited only if the
CPU type is changed to another CPU type within the same model group. If the CPU
type is changed to a CPU type belonging to a different model group, the
configuration setup either changes to the previous configuration of the new CPU
type, or reverts to default so you need to create a new configuration and modify it.
The table below lists the CPU types in each model group.

Table D3.3 Model Groups of CPU Types

<table>
<thead>
<tr>
<th>Model Group Name</th>
<th>CPU Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group 1</td>
<td>F3SP05, F3SP08, F3SP21</td>
</tr>
<tr>
<td>Group 2</td>
<td>F3SP25, F3SP35, F3FP36</td>
</tr>
<tr>
<td>Group 3</td>
<td>F3SP28, F3SP38, F3SP53, F3SP58, F3SP59, F3SP66, F3SP67, F3SP71, F3SP76, F3SP22</td>
</tr>
</tbody>
</table>

- Even if configuration setup is inherited, the allowable value range for each setup
item may differ between CPU types. Hence errors may be detected during program
syntax checking.
TIP

Clicking [Device Range Details] displays a dialog box that shows the number of devices and device ranges for example as follows.

<table>
<thead>
<tr>
<th>Device</th>
<th>Range</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal Relay (I)</td>
<td>05575</td>
<td>05575</td>
</tr>
<tr>
<td>Shared Relay (E)</td>
<td>0944</td>
<td>2048</td>
</tr>
<tr>
<td>Extended Shared Relay (E)</td>
<td>4008</td>
<td>2048</td>
</tr>
<tr>
<td>Link Relay (L)</td>
<td>78192</td>
<td>16384</td>
</tr>
<tr>
<td>Special Relay (M)</td>
<td>984</td>
<td>984</td>
</tr>
<tr>
<td>Timer (T)</td>
<td>3972</td>
<td>3972</td>
</tr>
<tr>
<td>Counter (C)</td>
<td>3972</td>
<td>3972</td>
</tr>
<tr>
<td>Data Register (D)</td>
<td>66535</td>
<td>66535</td>
</tr>
<tr>
<td>Control Register (F)</td>
<td>62420</td>
<td>62420</td>
</tr>
<tr>
<td>File Register (R)</td>
<td>25244</td>
<td>25244</td>
</tr>
<tr>
<td>Link Register (V)</td>
<td>76192</td>
<td>16384</td>
</tr>
<tr>
<td>Special Register (Z)</td>
<td>1024</td>
<td>1024</td>
</tr>
<tr>
<td>Index Register (V)</td>
<td>255</td>
<td>255</td>
</tr>
<tr>
<td>Shared Register (R)</td>
<td>1024</td>
<td>1024</td>
</tr>
<tr>
<td>Extended Shared Register (E)</td>
<td>4008</td>
<td>3972</td>
</tr>
</tbody>
</table>

Figure D3.4  Number of Devices and Device Ranges
D3.1.3 Executable Program Setup

This subsection describes how to set up an executable program.

You can perform executable program setup only when the Project Settings/Configuration window is open. Before performing your operations, open the Project Settings/Configuration window.

In executable program setup, you can set up the following two setup items:
- Execution Block Components
- Protection Settings

What Is an Executable Program?

An executable program is a collection of data downloaded to a sequences CPU module. An executable program consists of the following elements.

- Configuration
  Configures the CPU environment. If this is not set up, the CPU runs with the default settings for the given CPU type.
- User log message
  Stores user log messages when a user log is used.
- Blocks
  These are ladder programs made up of function units. Block tag name definition can also be included as an element to be downloaded.
- Common tag name definition
  Maps tag names to addresses and is common to all blocks.
- Constant definition
  Maps constant names to constant values.

![Executable Program Components Diagram]

Figure D3.5 Executable Program Components

TIP

The components of an executable program vary depending on the CPU module type.

SEE ALSO

For details on CPU modules, see “Sequence CPU – Functions.”
**Execution Block Components Screen**

Set up execution block components to define the blocks to be executed in the sequence CPU module.

You can perform the following seven types of setting operations for the execution block components:

- Creating/deleting execution block components
- Adding/deleting a reference folder
- Registering/deleting blocks in the library
- Renaming a block registered with the library
- Registering/deleting blocks in the execution block components
- Changing the order of the executable programs
- Displaying/setting the details of the component blocks

You can set up the execution block components in the Execution Block Components screen.

![Diagram of Execution Block Components](D0301_07.VSD)

**Figure D3.6 Execution Block Components**

- **A** List of the block files for the current project.
  - [File Name] lists block file names.
  - [Definition] lists the numbers indicating the execution order of the blocks in the current execution block components. [SCB] is displayed for sensor control blocks.
  - [Ref] displays *** for blocks contained in the common library or projects other than the current project.
  - [Reference Folder] displays "Common Library" for blocks in the common library or displays a reference folder name for blocks in other projects.

- **B** The Executable Program tab allows for various settings in the execution block components, such as adding/deleting blocks, changing the execution order of the blocks, and selecting whether to execute specified blocks.

- **C** Use the Sensor Control Block tab to register sensor control blocks.
- D Select the execution mode of the executable program.
  If the [Sequential Execution of All Blocks] option button is turned on, all blocks
  registered in the execution block components are executed in the specified order.
  If the [Specified Blocks] option button is turned on, you can use the program monitor
during online to specify for each block to be executed or not. After downloading a
program, only the top block listed in the component definition is executed by default.

- E Moves the currently selected block up one position in the execution block list in
  the Executable Program tab.

- F Moves the currently selected block down one position in the execution block list
  in the Executable Program tab.

- G Deletes the currently selected block from the execution block components in the
  Executable Program tab.

- H Opens the Executable Program Details/Setup dialog box for the blocks
  registered in the execution block components.

- I Updates the number of steps.

- J Registers the currently selected block file in the execution block components.

- K Adds a reference folder. To register blocks contained in a project other than the
  current project, specify the folder of that project.

- L Registers the blocks selected in the list of the block files with the common
  library.

- M Deletes the selected block files from the common library.

- N Replaces the block selected in the execution block components with the block
  selected in the list of the block files.

- O Deletes one or more reference folders from the list of the block files.

- P Changes the block file name of a block in the block library.

SEE ALSO
For details on starting and stopping blocks, see Section K1.6, “Starting and Stopping Blocks” (Online).

CAUTION
If you select not to use the configuration, the program execution mode is always
[Sequential Execution of All Blocks] even if you select [Specified Blocks].
**Error Handling in the Execution Block Components**

In the execution block components, lines containing errors are displayed with a different background color.

![Image of Execution Block Components Errors](D0301_08.VSD)

**Figure D3.7 Example of Execution Block Components Errors**

**Table D3.4 List of Execution Block Components Errors**

<table>
<thead>
<tr>
<th>Error Item</th>
<th>Background Color</th>
<th>Cause</th>
<th>Troubleshooting</th>
</tr>
</thead>
<tbody>
<tr>
<td>The reference folder of the block does not exist.</td>
<td>Yellow</td>
<td>The reference folder is not registered with the block file list or the file name is duplicated in the list.</td>
<td>Verify the block file list and register the reference folder again.</td>
</tr>
<tr>
<td>The block does not exist in the library nor projects.</td>
<td>Red</td>
<td>The corresponding file does not exist in the current project, reference projects nor common library. The block may have been renamed or deleted.</td>
<td>Delete the relevant block from the execution block components.</td>
</tr>
</tbody>
</table>

**TIP**

In the list of the block files registered in the execution block components, duplicated block file names are displayed based on the priority order shown in the following table.

**Table D3.5 Priority Order of the Duplicated Block File Names**

<table>
<thead>
<tr>
<th>Priority</th>
<th>Reference Destination of the Block File</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td>Currently opened project</td>
</tr>
<tr>
<td>Low</td>
<td>Common library</td>
</tr>
<tr>
<td></td>
<td>Reference folders (in order of the folder names)</td>
</tr>
</tbody>
</table>
Adding/Deleting a Reference Folder

To add a reference folder, use the following procedure.

◆ Procedure ◆

   ⇒ The Execution Block Components screen is displayed.

2. Click [Add Reference].
   ⇒ A dialog box for selecting a project folder is displayed.

3. Select a project folder to add to the execution block components.

4. Click [OK].
   ⇒ The blocks contained in the reference folder are added to the execution block components.

Steps (2) to (4)
To delete one or more reference folders, use the following procedure.

◆ Procedure ◆

   ⇒ The Execution Block Components screen is displayed.
2. Click [Delete Reference].
   ⇒ A dialog box is displayed for selecting one or more folders to be deleted.
3. Select one or more reference folders to be deleted from the execution block components.
4. Click [OK].
   ⇒ The blocks in the specified reference folders are deleted from the execution block components.
CAUTION

If a block in an already-deleted reference folder is registered in the execution block components, the block generates an error in the execution block components. The line for the block that generates this error is displayed in yellow. Delete the blocks displayed as the error from the execution block components.

Figure D3.8  Example of Execution Block Components Errors That Occur When Reference Folders Are Deleted
Registering/Deleting Blocks in the Library

You can register blocks contained in the project folder with the common library. You can also delete registered blocks from the common library.

To register blocks with the library, use the following procedure.

◆ Procedure ◆

   ⇒ The Execution Block Components screen is displayed.

2. From the blocks listed in the execution block components, select blocks to be registered with the common library.

   **TIP**
   While you can select multiple blocks, only the blocks in the project folder can be registered. Even if you select blocks stored in another folder than the project folder, the blocks are not registered.

3. Click [Add].
   ⇒ A confirmation dialog box is displayed.

4. Click [Yes].
   ⇒ The selected blocks are registered with the common library.

   **TIP**
   If a block with the same name is already present in the common library, a warning message is displayed. To overwrite the block, click [OK]. To cancel the operation, click [Cancel].
To delete blocks from the library, use the following procedure.

◆ Procedure ◆

⇒ The Execution Block Components screen is displayed.

(2) From the blocks listed in the execution block components, select blocks to be deleted from the common library.

TIP
You can select multiple blocks.

(3) Click [Delete].
⇒ A confirmation dialog box is displayed.

(4) Click [Yes].
⇒ The selected blocks are deleted from the common library.
CAUTION

If you delete a block from the library while the block is already registered in the execution block components, the block generates an error in the execution block components.

The line for the block that generates this error is displayed in red.

Delete the blocks displayed as the error from the execution block components.

Figure D3.9   Example of the Block File Name Error That Occurs When a Block Is Deleted from the Library
Renaming a Block File Registered with the Library

You can change the block file name of a block registered with the common library. The relevant procedure is given below.

Procedure


⇒ The Execution Block Components screen is displayed.

(2) From the blocks whose reference destination is the common library, select a block to change its block file name.

(3) Click [Rename].

⇒ A dialog box for changing the block file name is displayed.

(4) Enter a new block file name.

(5) Click [OK].

⇒ The block file name is changed.
CAUTION

If you change the block file name of a block already registered in the execution block components, the block generates an error in the execution block components. The line for the block that generates this error is displayed in red. Delete the block displayed as the error and then register it again.

Figure D3.10  Example of an Execution Block Components Error That Occurs When the Block File Name of a Block in the Library Is Changed
Registering/Deleting Blocks in the Execution Block Components

You can register blocks with the program or sensor control block, which is downloaded and executed in the CPU module. You can also delete blocks from the executable program or sensor control block.

To register blocks with the executable program or sensor control block, use the following procedure.

◆ Procedure ◆

⇒ The Execution Block Components screen is displayed.

(2) To register blocks with the executable program, click the Executable Program tab. To register blocks with the sensor control block, click the Sensor Control Block tab.
⇒ The executable program registration list or sensor control block registration list is displayed.

(3) From the block list, select one or more blocks to be registered with the executable program or sensor control block in the execution block components.

TIP
You can select multiple blocks.

(4) Click [Add].
⇒ The selected blocks are registered with the executable program or sensor control block.
TIP
- You can drag a block from the block list and drop it into the executable program list or sensor control block list to register the block.
- You can also copy a block selected in the block list and paste it into the executable program list or sensor control block list to register the block.
- You can also drag a block file (.YBLK) directly from the explorer and drop it into a relevant list to register it.

TIP
A sensor control block allows independent high-speed constant scan, superior in performance to normal scanning. Only one ladder block can be assigned as a sensor control block. Registering only one sensor control block to a component definition with no normal block is not allowed.
Sensor control blocks are known as "Sensor CB" in WideField3.

SEE ALSO
For details on sensor control blocks, see "Sequence CPU – Functions."
To delete blocks from the executable program or sensor control block, use the following procedure.

◆ Procedure ◆

⇒ The Execution Block Components screen is displayed.

(2) To delete blocks from the executable program, click the Executable Program tab. To delete blocks from the sensor control block, click the Sensor Control Block tab.
⇒ The executable program registration list or sensor control block registration list is displayed.

(3) Select one or more blocks to be deleted from the executable program or sensor control block.

**TIP**
You can select multiple blocks in the executable program list.

(4) Click [Delete].
⇒ The selected blocks are deleted from the executable program or sensor control block.
Changing the Order of the Executable Programs

To change the order of the executable programs, use the following procedure.

◆ Procedure ◆

   ⇒ The Execution Block Components screen is displayed.

2. Click the Executable Program tab.
   ⇒ The executable program list is displayed.

3. In the executable program list, select a block of which you want to change the position in the list.

4. To move the selected block up one position in the list, click [Up]. To move the block down one position in the list, click [Down].
   ⇒ The order of the blocks is changed.
Displaying/Setting the Details of the Executable Program

To open the Executable Program Details/Setup dialog box, use the following procedure.

Procedure

   ⇒ The Execution Block Components screen is displayed.

2. Click [Details/Setup].
   ⇒ The Executable Program Details/Setup dialog box is displayed.

TIP

In the Executable Program Details/Setup dialog box, you can set the number of local devices to use.

SEE ALSO

For details on the settings in the Executable Program Details/Setup dialog box, see Section D3.1.16, "Component Block Setup."
# Protection Settings

You can set password protection for the project, blocks, and macros.
You can perform protection settings in the Protection Settings screen.

---

**Figure D3.11 Protection Settings Screen**

- **A** Click this button to set protection for the project.
- **B** Configure protection settings for blocks in the project.
- **C** Configure protection settings for macros in the project.

---

⚠️ **CAUTION**

When setting protection, make sure to safely manage the password.
We cannot accept any request from any user, including persons responsible for setting passwords, to clear passwords that have been set.
Setting Protection for the Project

To set protection for the current project, use the following procedure.

◆ Procedure ◆

(1) Select [Protection Settings] from the [Project Settings] tree in the Project Settings/Configuration window.
 ⇒ The Protection Settings screen is displayed.

(2) Click [Project].
 ⇒ The Set up Password dialog box is displayed.

(3) Enter a password and click [OK].
 ⇒ The protection password is updated or newly created.

TIP
The protection setting for the project is saved when you click the [OK] button on the Project Settings/Configuration window.

TIP
To remove the protection setting, click the [OK] button on the Set up Password dialog box without entering anything in both the [New Password] and [Confirm Password] text boxes.
Setting Protection for Blocks/Macros

To set protection for blocks/macros, use the following procedure.

Procedure

   ⇒ The Protection Settings screen is displayed.

2. Select the line of the block or macro for which you set protection, and double-click the line or press the [Enter] key.
   ⇒ The Set up Password dialog box is displayed.

   TIP
   When [Protection] is [Yes] for the block or macro, a password confirmation dialog box is displayed. Enter the correct password and click [OK].

3. Enter a password and click [OK].
   ⇒ The protection password is updated or newly created.

   TIP
   The protection setting for a block or macro is saved when you click the [OK] button on the Project Settings/Configuration window.

   TIP
   To remove the protection setting, click the [OK] button on the Set up Password dialog box without entering anything in both the [New Password] and [Confirm Password] text boxes.
D3.1.4 CPU Properties

Set up CPU properties to configure detailed settings of the sequence CPU module. You can set up CPU properties in the CPU Properties screen.

**Figure D3.12  CPU Properties Screen**

**SEE ALSO**

For details on CPU properties, see Chapter H5, "CPU Properties" (Online).
D3.1.5 User Log Message

You can add, edit, and delete user log messages.
You can configure user log messages in the User Log Message screen.

Figure D3.13 User Log Message Screen

- A Select whether to use user log messages.
- B Edit user log messages.

TIP

- If you double-click a message field, the field enters input mode, and you can copy and paste the message in the field.
- You can register up to 64 messages. Each message can be up to 32 characters. One or more messages can be left blank between messages.

SEE ALSO

To enable the user log messages you created, you need to download them to the CPU module. For details on including user log messages in the project to be downloaded, see Section D3.1.3, "Executable Program Setup."
D3.1.6 Run Operation Setup

Perform operation setup to configure the operations of the CPU module.

You can perform operation setup in the Run Operation Setup screen.

**Figure D3.14 Run Operation Setup Screen**

- **A** Select whether to use the constant scan. If you use it, enter a constant scan time that ranges from 0.1 ms to 190.0 ms.
- **B** Select whether to set a peripheral processing time. If you set it, enter a peripheral processing time that ranges from 0.1 ms to 190.0 ms.
- **C** Specify a scan monitoring time that ranges from 10 ms to 200 ms.
- **D** Select the momentary power failure detection mode.

**TIP**

The valid range of the constant scan time depends on the CPU type. For F3SP71 and F3SP76, it ranges from 0.1 ms to 190.0 ms. For other module types, it ranges from 1.0 ms to 190.0 ms.

**SEE ALSO**

For details on setting the peripheral processing time, see "Sequence CPU – Functions."
D3.1.7 Input/Output Setup

Use the Input/Output Setup screen to perform I/O setup and check the I/O usage status of the sensor CB in the current project. The I/O usage status in other projects can also be checked for reference.

![Diagram of Input/Output Setup Screen]

Figure D3.15 Input/Output Setup Screen

- **A** Allows you to check the I/O usage status of the program. The check results are displayed in the grids indicated as C. The color of check results corresponds to the background color of the guide message in the detailed information described in E below. When the I/O setup of other CPU modules is referenced, the information of the current project is displayed together with the information of other projects. Turn the checkbox on to allow checking the I/O usage status automatically when the Input/Output Setup screen is opened next time. Click the [Program Usage Status Check] button to update the usage status display.

- **B** Select a CPU slot number of the current project.

- **C** Displays the I/O usage status of the program. "***" is displayed for the slot whose setting is different from its default. "S" is displayed for the slot used for the sensor control block.

- **D** Specify the reference destination of the I/O setup of another project.

- **E** Displays a guide message about the I/O usage status of the program. The following table lists the guide messages to be displayed.

<table>
<thead>
<tr>
<th>Background Color</th>
<th>Guide Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red</td>
<td>Competing with I/O use setup of another CPU (project).</td>
</tr>
<tr>
<td>Red</td>
<td>Specified as not used but is accessed by program.</td>
</tr>
<tr>
<td>Light green</td>
<td>Normal. &quot;Slot used from this project.&quot;</td>
</tr>
<tr>
<td>Yellow</td>
<td>Normal. &quot;Can be used from this project but is not accessed from the program.&quot;</td>
</tr>
<tr>
<td>Orange</td>
<td>Normal. &quot;Setting is that used by other projects.&quot;</td>
</tr>
<tr>
<td>Lime</td>
<td>Normal. &quot;Slot used from the SCB of this project.&quot;</td>
</tr>
<tr>
<td>Gray</td>
<td>Not Used.</td>
</tr>
</tbody>
</table>


TIP

You can select multiple slots to set up them at once.

- F Use the Use Setup tab to select [Use], [Do Not Use], or [SCB] for the usage setting of each terminal set.
  [Data Code] allows you to select either [BIN] or [BCD] for data code.
  You can set up all terminals at once by using the [All] drop down list.

![Use Setup Tab](image1)

**Figure D3.16 Use Setup Tab**

- G Use the Input Setup tab to select an input sampling method for each terminal set.
  [Input Sampling] allows you to select [16ms], [1ms], [250μs], [62.5μs], or [Always] for the input sampling interval.
  You can set up all terminals at once by using the [All] drop down list.

![Input Setup Tab](image2)

**Figure D3.17 Input Setup Tab**

**CAUTION**

The input sampling interval options of [250μs], [62.5μs], and [Always] are available only for the following CPU types: F3SP22, 28, 38, 53, 58, 59, 66, 67, 71, and 76.
- Use the Output Setup tab to select an output setting for each terminal set for when the CPU module is stopped.

[Output When Stopped] allows you to select [Reset (OFF)] or [Hold] for the external output setting for when the CPU module is stopped.

You can set up all terminals at once by using the [All] drop down list.

![Output Setup Tab]

Figure D3.18 Output Setup Tab
D3.1.8 Device Setup

Device setup contains the following setup items:
- Device Area Setup
- Latch Range Setup at Power Failure

You can perform device setup only when the Project Settings/Configuration window is open. Before performing your operations, open the Project Settings/Configuration window.

Device Area Setup

You can perform device area setup to configure the use areas of global devices and local devices.

You can perform device area setup in the Device Area Setup screen.

Figure D3.19 Device Area Setup Screen

- A Global device area settings.
  [Start] displays the start address of each area.
  [End] displays the end address, as well as allows you to enter a value for the end address. The value depends on the CPU type.
- B Local device area settings.
  Turn on the [Set] checkbox if you use a local device. In [Start], enter the start address of the local device.
  [Used] displays the number of local devices currently used.
  [Available] displays the number of local devices currently available.
- C Device use area map.
You can use the timer use area map to specify the number of timers for each timer type.

- **TIP**
  Automatic setup of local devices is a function for automatically setting the number of local devices included in the block properties when local devices are used in the program.

  In addition, you can specify a reserve area that is automatically allocated for this number. For example, if you set the reserve area to 20% and the largest device number used in the program is /D00100, reservations are ensured for /D00001 ~ /D00120. However, since the number of reservations ensured is for the number of local devices that can be set, you should set the reserve area only as a guide.

  By specifying a reserve area, you can keep a sufficient area to continue your edit operation when you add a local device during online editing.

  If you specify to set local devices automatically, the [Auto Calc.] button is displayed instead of the [Re-retrieve] button.

**SEE ALSO**
For details on the procedure for setting local devices automatically, see Section F2.2, "Setting Local Devices."

---

**CAUTION**
If you click the [Default] button on the Device Area Setup screen, the settings in the setup for data latch at power failure are also restored to their defaults.
**Latch Range Setup at Power Failure**

Use the Latch Range Setup at Power Failure screen to set the data latch range at power failure.

![Image of Latch Range Setup at Power Failure screen]

- **A** Displays the device types.
- **B** Turn on/off the checkbox if you want to enable/disable the data latch at power failure for each device.
- **C** Enter the start address of the device for which the data latch at power failure is enabled. You can also set the start address in the data latch range map described in **E** below.
- **D** Enter the end address of the device for which the data latch at power failure is enabled.
- **E** Data latch range map. Move the start and end indicators to change the range. The data latch range at power failure is indicated in orange in the map.

![Image of Data Latch Range Map]

**CAUTION**

If you click the [Default] button on the Latch Range Setup at Power Failure screen, the settings in the device area setup are also restored to their defaults.
D3.1.9 Interrupt Setup

You can perform interrupt setup in the Interrupt Setup screen.

**TIP**

Interrupt setup is supported on the following CPU modules only: F3SP22, 28, 38, 53, 58, 66, 67, 71, and 76.

---

**Figure D3.24 Interrupt Setup Screen**

- **A** Select the priority of interrupts.
  - Select [Sensor CB interrupt has priority] to assign a higher priority to the sensor control block interrupt over the input interrupt.
  - Select [Input interrupt has priority] to assign a higher priority to the input interrupt over the sensor control block interrupt.

- **B** Specify the execution interval and timing of interrupts for the sensor control block.
  - In [Execution Interval], specify the interrupt execution interval for the sensor control block.
  - In [Timing of Interrupt], select [After Instruction] or [Immediate] for the timing of interrupt.

- **C** Select the timing of the input interrupt.

**TIP**

The valid range of the execution interval depends on the CPU type.

For F3SP71 and F3SP76, it ranges from 0.10 ms to 25.00 ms in increments of 0.01 ms.

For other module types, it ranges from 0.2 ms to 25.0 ms in increments of 0.1 ms.
D3.1.10 Built-in Functions Setup

Perform built-in functions setup to configure the built-in functions of the CPU module. You can perform built-in functions setup in the Built-in Functions Setup screen.

Figure D3.25 Built-in Functions Setup Screen

- A Setting for the PROGRAMMER/SIO ports.
  Select a communication mode from the [Communication Mode] drop down list. The available options depend on the CPU type.
  Turn on the [Use] checkbox of [Personal Computer Link Function] to use the Personal Computer Link function. This setup is required to use the CPU programming port for the Personal Computer Link function. If this item is checked, then turn on the [Checksum], [End Character], and [Protection] checkboxes as required.

- B Configure the settings for keeping the register data resident in the ROM pack.
  Turn on the [Keep register data resident in ROM pack] checkbox to keep register data resident in the ROM pack. Turn it off if you do not want to do so.
  If you select to keep register data resident, then turn on the [Data Register(D)] or [File Register(B)] option button and enter the start and end numbers. You can also set the start and end numbers by moving the indicators on the resident setup range map.

Figure D3.26 Resident Setup Range Map
Specify whether or not to use each built-in function by selecting [Use] or [Do Not Use] from the drop down list. If you select [Do Not Use], the relevant CPU function is disabled.

**Figure D3.27  [Operation Settings] Drop Down List in [Deletion Setup for Built-in Functions]**

**CAUTION**

Remote programming service is a tool command interface provided by CPU modules, which is also used by WideField3. Beware that disabling it would disallow subsequent connection by WideField3.

**TIP**

- The CPU communication port is provided on the front of the CPU module for connecting to a personal computer. It also provides the Personal Computer Link connection function.
- The ROM pack retains stored program data even after power is switched off. You can load and execute a program stored in the ROM at power-on.
- In built-in functions setup, [Settings for Keeping Register Data Resident in ROM Pack] is available for CPU modules other than F3SP66, 67, 71, and 76.
- Function removal can be specified only for CPU modules F3SP66 and F3SP67.
- To disable built-in functions on F3SP71 and F3SP76, use the operation protection settings on FA-M3 Defender.

**SEE ALSO**

- For details on the ROM pack and function removal, see “Sequence CPU - Functions.”
- For details on the operation protection settings on FA-M3 Defender, see Section H11.2.3, “Function Removal Setup” (Online).
D3.1.11  Error Handling Setup

Perform error handling setup to configure the settings for error handling. You can perform error handling setup in the Error Handling Setup screen.

Figure D3.28  Error Handling Setup Screen

[Error Detection Setup]
Select the [Perform Device Boundary Check] checkbox to enable to check whether device ranges are exceeded.

[Error-Time Action Setup]
To configure the error handling settings, select [Stop] or [Continue to run] from the [Operation Settings] drop down list for each error type, and click [OK].
D3.1.12 Initial Data Setup

Perform initial data setup to set the initial value range and initial values of data registers. You can perform initial data setup in the Initial Data Setup screen.

### Initial Data Setup Screen

The following describes the Initial Data Setup screen.

- **Figure D3.29 Initial Data Setup Screen**

  - A Set the range of initial values. Turn on the [Use] checkbox if you want to specify the initial value setup range or turn it off if you do not want to do so. To specify the start and end addresses of the initial value setup range, enter values directly or move the indicators in the initial value range map. The initial value setup range is indicated in orange in the map.

- **Figure D3.30 Initial Value Range Map**

  - **Table D3.7 Display Colors in the Initial Value Range Map**

<table>
<thead>
<tr>
<th>Color</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orange</td>
<td>Indicates the initial value setup range.</td>
</tr>
<tr>
<td>Green</td>
<td>Indicates the range assigned for devices.</td>
</tr>
</tbody>
</table>
- Set the initial values.
  From the [Display Format] drop down list, select decimal or hexadecimal format for
  the display format of the initial values.
  Click [Import from CSV File] to import the initial values from a CSV file.
  Click [Export to CSV File] to export the initial values as a CSV file.

**TIP**
- The settings in the initial data setup are saved in the configuration.
- Up to 1024 points can be contained in the initial value setup range.

**Registering/Editing the Initial Values**
To register/edit the initial value for an address of a device, select the cell corresponding
to the address and enter an initial value in the cell under [Initial Data Setup].
D3.1.13 Inter-CPU Shared Memory Setup

Perform inter-CPU shared memory setup to configure the ranges of the shared devices and the settings for shared refreshing.

You can perform inter-CPU shared memory setup in the Inter-CPU Shared Memory Setup screen.

![Inter-CPU Shared Memory Setup Screen](image)

- A Select the CPU module for the current project from the drop down list.
- B Opens the Check Sharing Settings dialog box. On the dialog box, you can compare the settings on the CPU module for the current project and the shared settings of other CPU modules.
- C Select the target CPU module for the current project, and set the setup ranges of the shared relay, shared register, extended shared relay, and extended shared register and also specify whether or not to refresh each of them.
  - [Setup Range] shows the range to be set.
  - [Start Address/Points] displays the start address and the number of points to be set.
- D To enable shared refreshing in peripheral processing, turn on the [Peripheral Process] option button. If you want to enable shared refreshing in control processing, turn on the [Control Process] option button.
- E Select whether to maintain simultaneity of shared device data.

**TIP**
- The shared refreshing function updates data of shared devices (such as shared relays and shared registers) for inter-CPU communications in a multi-CPU system.
- Shared memory refresh setup is supported on the following CPU modules only: F3SP22, 28, 38, 53, 58, 59, 66, 67, 71, and 76.

**SEE ALSO**
For details on simultaneity, see "Sequence CPU – Functions."
Comparing the Shared Settings with Other CPU Modules Settings

You can compare CPU shared memory settings between the current project and other projects to check the setup ranges of the shared devices.

The result of the comparison with other projects is displayed on the Check Sharing Settings dialog box. To open the Check Sharing Settings dialog box, click the [Check that sharing settings match with other CPUs] button on the Inter-CPU Shared Memory Setup screen.

**Figure D3.32 Check Sharing Settings Dialog Box**

- A Specify a project to be used on a relevant CPU module.
- B Updates the settings of each project.
- C Displays the shared device settings of each project in a map format. The map uses various colors according to the setting status.

**Table D3.8 Display Colors in the Shared Setting Map**

<table>
<thead>
<tr>
<th>Color</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orange</td>
<td>Indicates that the refresh setting is ON and the relevant CPU module writes data in this area.</td>
</tr>
<tr>
<td>Green</td>
<td>Indicates that the refresh setting is ON and each relevant CPU module reads data from this area.</td>
</tr>
<tr>
<td>Light yellow</td>
<td>Indicates that the refresh setting is ON but refresh range is not set in this area.</td>
</tr>
<tr>
<td>Gray</td>
<td>Indicates that the refresh setting is OFF in this area.</td>
</tr>
</tbody>
</table>

- D Displays a shared setting message listed in the table below.

**Table D3.9 Shared Setting Messages**

<table>
<thead>
<tr>
<th>Background Color</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red</td>
<td>Shared relay(E) settings differ between CPUs.</td>
</tr>
<tr>
<td>Red</td>
<td>Shared register(R) settings differ between CPUs.</td>
</tr>
<tr>
<td>Red</td>
<td>Extended shared relay(E) settings differ between CPUs.</td>
</tr>
<tr>
<td>Red</td>
<td>Extended shared register(R) settings differ between CPUs.</td>
</tr>
<tr>
<td>Cyan</td>
<td>All shared device settings are matching</td>
</tr>
</tbody>
</table>

- E Closes the Check Sharing Settings dialog box.
To compare the shared memory settings for the current project with the shared settings of other CPU modules, use the following procedure.

◆ Procedure ◆

(1) Select [Inter-CPU Shared Memory Setup] from the [Configuration] tree in the Project Settings/Configuration window.
⇒ The Inter-CPU Shared Memory Setup screen is displayed.

(2) Click [Check that sharing settings match with other CPUs].
⇒ The Check Sharing Settings dialog box is displayed.

(3) Click [Browse].
⇒ The Select Project dialog box is displayed.
(4) Select the project file of a project whose shared settings are to be compared with the settings of the current project, and click [Open].

⇒ The selected project file is displayed in the Check Sharing Settings dialog box.

(5) Repeat step (4) if you want to add another project whose shared settings are to be compared with the settings of the current project.

(6) Click [Check].
⇒ The shared setting map and shared setting message are updated.

(7) Click [Close].
⇒ The Check Sharing Settings dialog box is closed.
TIP

The following table shows the shared setting messages to be displayed and troubleshooting for each message.

**Table D3.10  Shared Setting Messages and Troubleshooting**

<table>
<thead>
<tr>
<th>Message</th>
<th>Troubleshooting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shared relay (E) settings differ between CPUs.</td>
<td>Adjust the setting range of the shared relay (E) for each project to match each other.</td>
</tr>
<tr>
<td>Shared register (R) settings differ between CPUs.</td>
<td>Adjust the setting range of the shared register (R) for each project to match each other.</td>
</tr>
<tr>
<td>Extended shared relay (E) settings differ between CPUs.</td>
<td>Adjust the setting range of the extended shared relay (E) for each project to match each other.</td>
</tr>
<tr>
<td>Extended shared register (R) settings differ between CPUs.</td>
<td>Adjust the setting range of the extended shared register (R) for each project to match each other.</td>
</tr>
</tbody>
</table>
D3.1.14 FA Link Setup

Perform FA link setup to configure the settings of the FA link and FL-net. You can perform the setup for the FA link and FL-net in the FA Link Setup screen.

**CAUTION**

- F3SP71-4N and F3SP76-7N do not support FA link setup.
- FA link setup is changed to [Do Not Use] if changing models from F3SP71-4S/76-7S to F3SP71-4N/76-7N and to [FL-net] if changing models from other CPUs to F3SP71-4N/76-7N.

**TIP**

When a WideField/WideField2 project is opened in WideField3, if the link type of the project is FA link and the number of link registers for each system is set to a value larger than 2048 or the number of link relays for each system is set to a value larger than 8192, the link type is changed to FL-net. In this case, the number of link registers and link relays are also changed to 2048 and 8192, respectively.

---

**Figure D3.33 FA Link Setup Screen**

- A  FA link or FL-net link system setup. Select manual or automatic setup.
- B  Select [Link Relay(L)] or [Link Register(W)] to switch the link device map.
- C  Select [Do Not Use], [FA Link], [FA Link High-Speed], or [FL-net] for each system.
- D  Specify a slot number if you turn on the [Manual setup] option button in A above.
- E  If you select [FA Link] or [FA Link High-Speed] in B, the FA Link Setup Tool is selected. If you select [FL-net] in B, select [FL-net Refresh Setup] or [FL-net Refresh Setup Tool Startup].
- F  Starts the tool selected in E or opens a setup dialog box.
- **G** Move the indicators in the map to assign a range to each link device. You can use the spin button to set the end address of the range.

- **H** Specify the end address of the range assigned to each link device.
FA Link Setup Tool

You can start the FA Link Setup Tool from the FA Link Setup screen. The relevant procedure is given below.

Procedure

1. Select [FA Link Setup] from the [Configuration] tree in the Project Settings/Configuration window.
   ⇒ The FA Link Setup screen is displayed.

2. Select [FA Link] or [FA Link High-Speed] from the link type drop down list for a system to be configured.

3. Click [Execute].
   ⇒ The FA Link Setup Tool is started.

TIP

To use online functions of the FA Link Setup Tool, WideField3 must be connected to the CPU module.

TIP

If WideField3 is not connected to the CPU module, a confirmation message box is displayed.

SEE ALSO

For details on the FA Link Setup Tool, see Section H7.1, "Station Assignment and Monitoring of FA Links" (Online).

TIP

F3SP71-4N and F3SP76-7N do not support FA link setup.
FL-net Refresh Setup

You can perform FL-net refresh setup by opening a setup dialog box from the FA Link Setup screen.

**TIP**

FL-net refresh setup is supported on the following CPU modules only: F3SP22, 28, 38, 53, 58, 59, 66, 67, 71, and 76.

---

**Figure D3.34   FL-net Refresh Dialog Box**

- **A** Turn on the [Peripheral Process] or [Control Process] option button to select the common data refresh mode.
- **B** Set the common data refresh ranges for FL-net1 and FL-net2.
  - Select [All Nodes] to include all nodes to the refresh range.
  - Select [Some Nodes] to specify whether or not to perform refreshing for each node by using the Set up Refresh Range dialog box.
  
  To open the Set up Refresh Range dialog box, click the [Set up] button after you select [Some Nodes] for the refresh range.
To open the dialog box for FL-net refresh setup from the FA Link Setup screen, use the following procedure.

◆ Procedure ◆

(1) Select [FA Link Setup] from the [Configuration] tree in the Project Settings/Configuration window.  
⇒ The FA Link Setup screen is displayed.

(2) Select [FL-net] from the link type drop down list for a system to be configured.

(3) Select [FL-net Refresh Setup] from the drop down list as shown in the right figure.

(4) Click [Execute].

⇒ The FL-net Refresh dialog box is displayed.
FL-net Setup Tool
You can start the FL-net Setup Tool from the FA Link Setup screen. The relevant procedure is given below.

◆ Procedure ◆

(1) Select [FA Link Setup] from the [Configuration] tree in the Project Settings/Configuration window.
⇒ The FA Link Setup screen is displayed.

(2) Select [FL-net] from the link type drop down list for a system to be configured.

(3) Select [FL-net Refresh Setup Tool Startup] from the drop down list as shown in the right figure.

(4) Click [Execute].
⇒ The FL-net Setup and Monitoring window is displayed.

TIP
To use online functions of the FL-net Setup Tool, you must connect WideField3 to the CPU module.

TIP
If WideField3 is not connected to the CPU module, a confirmation message box is displayed.

SEE ALSO
For details on the FL-net Setup Tool, see Section H7.2, "FL-net Setup and Monitoring" (Online).
D3.1.15 Sampling Trace Setup

This subsection describes how to perform sampling trace setup.

You can perform sampling trace setup in the Sampling Trace Setup screen.

![Sampling Trace Setup Screen](image)

**CAUTION**

For the CPU modules F3SP71 and F3SP76, the sampling trace settings can be read from the Project Settings/Configuration window, but cannot be changed. To perform sampling trace setup for these CPU modules, use the sampling trace tool.

**TIP**

Use the sampling trace tool to perform detailed settings of the sampling trace functions.

**SEE ALSO**

For details on the procedure for performing detailed settings of the sampling trace functions, see Section K4.3, “Sampling Trace Setup” (Online).

- **A** Turn on this checkbox to enable sampling trace settings when the program is downloaded to the CPU module.
- **B** Turn on the [Setup in this screen] option button to perform sampling trace based on the settings configured in this screen. You can also load the settings from a sampling trace setup file by turning on the [Load Sampling Trace Setup File] option button. To specify a sampling trace setup file, click [Browse] to open a dialog box and select a file to be loaded. Click [Apply] to apply the loaded settings to the screen.
- C Select a sampling method.
   Turn on the [TRC Instruction] option button to perform sampling when a TRC instruction is executed.
   Turn on the [Scan] option button to perform sampling when an END instruction is executed. An END instruction is always executed at the end of a scan cycle.
   Turn on the [Periodic] option button to perform sampling at specified fixed time intervals.
   A value of zero in [Delay Count] indicates that sampling is to be started at the timing when the trigger condition is true. If you specify a negative value in this field, sampling is started at an earlier point; and if you specify a positive value, sampling is started at a later point, depending on the specified delay count.

- D Specify a trigger condition.
   Enter the address of a device to be used for the trigger condition in [Device Address].
   Select the [Rising Edge of Specified Relay] option button to specify that the trigger condition becomes true when a rising edge occurs on the bit device used for the trigger condition.
   Select the [Falling Edge of Specified Relay] option button to specify that the trigger condition becomes true when a falling edge occurs on the bit device used for the trigger condition.
   Select [Data Coincidence] to specify that the trigger condition becomes true when the value of the word device for the trigger condition equals a specified value. In the spin box, enter a value to be used for the trigger condition.

- E Enter up to 16 relays and 4 registers to be sampled.
D3.1.16 Component Block Setup

This subsection describes how to set up component blocks.

You can perform component block setup in the Executable Program Details/Setup dialog box.

You can perform this setup only when the Project Settings/Configuration window is open. Before performing your operations, open the Project Settings/Configuration window.

Displaying the Executable Program Details/Setup Dialog Box

To open the Executable Program Details/Setup dialog box, use one of the following two procedures:

- Select [Project Settings] from the tree in the Project Settings/Configuration window, and click the [Common Tag Name Definitions/Local Device Detailed Setup] button.
- Select [Execution Block Components] from the tree in the Project Settings/Configuration window, and click the [Details/Setup] button.
Executable Program Details/Setup Dialog Box

In the Executable Program Details/Setup dialog box, you can view or set up the details of the project and executable program.

Figure D3.37  Executable Program Details/Setup Dialog Box

SEE ALSO

For details on the settings in the Executable Program Details/Setup dialog box, see Chapter B7, "Advanced Editing of Component Blocks" (Introduction and Troubleshooting).
Setting up the Entire Project

The following describes how to set up the entire project in the Executable Program Details/Setup dialog box.

- A Displays the project file name.
- B Displays the executable program name (i.e., the project name). This name is used as the name of the executable program (or the project) when the program is stored in the CPU module.
- C Displays step counts.
  - [All Blocks] is the step count of all blocks that make up the executable program.
  - [All] is the step count of all blocks including the step counts used for the tag name definitions and constant definitions stored in the CPU module.
- D Displays whether protection is enabled or not. [Yes] is displayed if protection is set; [No] is displayed if protection is not set. To set or change the protection password, double-click the protection password display area.
- E Displays the number of constant definitions currently stored and the maximum number of constant definitions that can be stored.
- F Displays the currently specified maximum number of common tag name definitions. You can change the maximum.
- G Displays whether or not to store common tag name definitions in the CPU module. You can change the setting of whether or not to store them.
  For WideField3R2.04 or later, [Common Tag Name Def.] of a newly created project is set to [Yes] by default.
- H Displays the total number of local devices of each type used in the entire project.
● Setting up the Executable Program

The following describes how to set up the executable program in the Executable Program Details/Setup dialog box.

- A Displays the execution order of execution blocks. [SCB] is displayed for sensor control blocks.
- B Displays the block file name of each execution block.
- C Displays the block name of each execution block. This block name is used when the block is stored in the CPU module. You can edit the block name. Enter a string of up to eight alphanumeric characters long that begins with two alphabetical characters. You can use ",", ",", and "_" in the name.
- D Displays step counts for each block.
  [Program] is the step count of the programs written in the relevant block.
  [All] is the step count of the programs including the step counts used for the tag name definitions and constant definitions stored in the CPU module.
- E Displays whether protection is enabled or not. [Yes] is displayed if protection is set; [No] is displayed if protection is not set. To set or change the protection password, double-click the protection password display area.
- F Displays the type of tag name definitions referenced by each block.
  [Common] indicates that the block refers to the common tag name definitions and block tag name definitions in the local devices.
  [Block] indicates that the block refers to only the block tag name definitions.
- G Displays whether or not to store tag name definitions or comments in the CPU module.
  [Tag Name Def.] displays [Yes] or [No] depending on whether to store tag name definitions in the CPU module. You can change the setting of whether or not to store them.
  [Comment] displays [Yes] or [No] depending on whether to store comments in the CPU module. You can change the setting of whether or not to store them.
For WideField3R2.04 or later, both [Tag Name Def.] and [Comment] of a newly created block are set to [Yes] by default.
- H Displays the total number of local devices of each type used in the block.
D3.2 Managing a Project

This section explains project management. The following project management functions are provided.

- Checking Programs
- Finding in Project
- Finding Instruction in Project
- Replacing Tag Name in Project
- Replacing Address in Project
- Displaying Device Usage Status
- Changing I/O Installation Position
- Converting Data
D3.2.1 Checking Programs

The Check Program function checks the following items and displays locations for detected errors.

- Connections between circuits
- Mapping between circuits and configurations

You can also check detailed information for a project during program checking.

To perform syntax check, use the following procedure.

◆ Procedure ◆

(1) Select [Project]–[Check Program] from the menu bar.

⇒ Program checking begins. The Program Check dialog box is displayed during syntax checking.

After syntax check is completed, the Results of Program Syntax Checking window is displayed.

TIP

Syntax checking is performed on saved programs and blocks. If you invoke the syntax checking function with a ladder program edit window (tag name definition window) open after having made changes in the window, a confirmation dialog box will be displayed.

If no errors are detected, a confirmation dialog box is displayed in place of the Results of Program Syntax Checking window.

If errors are detected, up to 256 errors and warnings (in that order) are displayed in the Results of Program Syntax Checking window.

(2) If an error is displayed, double-click the error or warning to display more details. Alternatively, use the arrow key to move the cursor to the error or warning and press [Enter].

⇒ The block containing the error or warning opens and the display jumps to the location of the error and the window focus moves to the circuit.

TIP

To display the location of the error or warning without changing the window focus, press any of the following keys.

- [SPACE] key: Displays the location of the error or warning at the current cursor position.
- [F3] key: Displays the location of the next error or warning after the current cursor position.
- [Shift]+[F3] keys: Displays the location of the previous error or warning before the current cursor position.

The Results of Program Syntax Checking window remains as the topmost window if you use the above operations to display the error or warning location without changing the focus.
Clicking [Details] on the Results of Program Syntax Checking window opens the Detailed Results for Program Check dialog box for you to check detailed information of the project.

![Figure D3.40 Detailed Results for Program Check Dialog Box](image)

If no errors and warnings are detected, clicking [Details] on the confirmation dialog box opens the Detailed Results for Program Check dialog box. If an error is detected during program syntax checking, correct detailed information is not displayed.

### Circuits Returning Errors at Program Check

Errors returned from program checking are classified into the following 2 types according to severity:

- **Error**
  - Critical errors. The download function is prohibited.
- **Warning**
  - Non-critical errors. Operation may not work as expected.

These error categories can be modified using the environment settings.

**SEE ALSO**

For details on changing the classifications, see Section D1.2.6, "Program Syntax Check Setup."

The table below summarizes the error types.

**Table D3.11 Programs Returning Errors at Syntax Check**

<table>
<thead>
<tr>
<th>Error Message</th>
<th>Description</th>
<th>Additional Information</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block not found.</td>
<td>Block is registered in the executable program component definition but its file is not found.</td>
<td>Block name</td>
<td>Error (not modifiable)</td>
</tr>
<tr>
<td>Macro not found.</td>
<td>An instruction macro called by a block does not exist.</td>
<td>Block name, instruction number, block name</td>
<td>Error (not modifiable)</td>
</tr>
<tr>
<td>Block conversion error.</td>
<td>Block was saved with error. Syntax checking is not done for blocks saved with errors.</td>
<td>Block name</td>
<td>Error (not modifiable)</td>
</tr>
<tr>
<td>Macro conversion error.</td>
<td>Macro was saved with error. Syntax checking is not done for macros saved with errors.</td>
<td>Macro name</td>
<td>Error (not modifiable)</td>
</tr>
<tr>
<td>Macro is not in CPU.</td>
<td>A macro used in a project is not found in the CPU.</td>
<td>Macro name</td>
<td>Error (not modifiable)</td>
</tr>
<tr>
<td>Mismatch in device capacity and latch range at power failure.</td>
<td>The device capacity and the power failure latch range do not match.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Invalid local device setting.</td>
<td>- The first address number of the local device is invalid. It exceeds the device range. - The first address of the local device is not defined.</td>
<td></td>
<td>Error (not modifiable)</td>
</tr>
<tr>
<td>Local device setup invalid for CPU model.</td>
<td>The local device setting is out of the usable range for the CPU selected for the project (during partial downloading).</td>
<td></td>
<td>Error (not modifiable)</td>
</tr>
<tr>
<td>Program step count exceeded limit.</td>
<td>The program step count exceeds the maximum allowed for the selected CPU type. (F3SP22-0S/SP28-3S/SP38-3S/SP53-4S/SP58-6S/SP59-7S/SP64-4S/SP67-6S/SP71-4N/SP76-7N/SP71-4S/SP76-7S CPU modules, the step count includes step counts for circuits and subcomments.)</td>
<td></td>
<td>Error (not modifiable)</td>
</tr>
<tr>
<td>Error Message</td>
<td>Description</td>
<td>Additional Information</td>
<td>Category</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>Project step count exceeded limit.</td>
<td>The program step count with tag name definition exceeds the maximum allowed for the selected CPU type.</td>
<td>Error (not modifiable)</td>
<td></td>
</tr>
<tr>
<td>Number of blocks exceeds the maximum limit for the model.</td>
<td>The number of blocks exceeds the maximum allowed for the selected CPU type.</td>
<td>Error (not modifiable)</td>
<td></td>
</tr>
<tr>
<td>Number of labels exceeds the maximum limit for the model.</td>
<td>The number of labels exceeds the maximum allowed for the selected CPU type.</td>
<td>Error (not modifiable)</td>
<td></td>
</tr>
<tr>
<td>Device is invalid or out of range</td>
<td>The address is out of range. The following are possible causes. - A global address exceeds the device capacities range in the configuration. - The number of local addresses used exceeds the number set in the configuration. - The number of device points used in the structure exceeds 2048. This error is generated when too many long-word members are used in a structure.</td>
<td>Block name, instruction number, invalid address</td>
<td>Error (not modifiable)</td>
</tr>
<tr>
<td>This instruction cannot be used on the selected CPU type.</td>
<td>The entered instruction is not supported by the selected CPU type.</td>
<td>Block name, instruction number, invalid instruction</td>
<td>Error (not modifiable)</td>
</tr>
<tr>
<td>This instruction parameter cannot be used on the selected CPU type.</td>
<td>To fix this problem, select a CPU that supports structures. The entered instruction is not supported by the selected CPU type.</td>
<td>Block name, instruction number, instruction parameter</td>
<td>Error (not modifiable)</td>
</tr>
<tr>
<td>Sensor CB cannot be used on the selected CPU type.</td>
<td>To fix this problem, select a CPU type that allows sensor control block or remove the block from the sensor control block list.</td>
<td>Error (not modifiable)</td>
<td></td>
</tr>
<tr>
<td>Indirect specification cannot be used on the selected CPU type.</td>
<td>Indirect specification representation is not supported for the selected CPU type.</td>
<td>Block name, instruction number, invalid instruction</td>
<td>Error (not modifiable)</td>
</tr>
<tr>
<td>No address assigned to tag name.</td>
<td>Address is not allocated in the tag name definition. The reference tag name definition may be incorrect. The structure object is yet undefined. A structure member does not exist.</td>
<td>Block name, instruction number, tag name</td>
<td>Error (not modifiable)</td>
</tr>
<tr>
<td>Preset value of timer/counter is out of range.</td>
<td>The timer or counter setting is out of range. Value is out of the timer precision range.</td>
<td>Block name, instruction number, invalid address</td>
<td>Error (not modifiable)</td>
</tr>
<tr>
<td>Invalid device for instruction.</td>
<td>An invalid device is used in an instruction.</td>
<td>Block name, instruction number, invalid address</td>
<td>Error (not modifiable)</td>
</tr>
<tr>
<td>Block for ACT/INACT does not exist.</td>
<td>A block for an ACT/INACT instruction is not registered in the executable program components definition.</td>
<td>Block name, instruction number, unregistered block name</td>
<td>Error (not modifiable)</td>
</tr>
<tr>
<td>Number of macros exceeds the maximum limit for the model.</td>
<td>The number of macros exceeds the maximum allowed for the selected CPU type.</td>
<td>Error (not modifiable)</td>
<td></td>
</tr>
<tr>
<td>MRET instruction is not at the end.</td>
<td>The last line of a macro block is not an MRET instruction.</td>
<td>Macro name</td>
<td>Error (not modifiable)</td>
</tr>
<tr>
<td>Duplicate use of coil</td>
<td>The same address is used in OUT, OUTN, DIFU and DIFD instructions. The second and subsequent instances are shown as error locations.</td>
<td>Block name, instruction number, duplicated address</td>
<td>Warning (Can be set)</td>
</tr>
<tr>
<td>Duplicate use of SET/RST</td>
<td>The same address is used in SET and RST instructions and the OUT, OUTN, DIFU and DIFD instructions. The second and subsequent instances are displayed as error locations.</td>
<td>Block name, instruction number, duplicated address</td>
<td>Warning (Can be set)</td>
</tr>
<tr>
<td>Duplicate use of timer/counter</td>
<td>The same timer or counter is used at more than one location. The second and subsequent instances are displayed as error locations.</td>
<td>Block name, instruction number, duplicated address</td>
<td>Warning (Can be set)</td>
</tr>
<tr>
<td>Duplicate use of label</td>
<td>The same label is used at more than one location. The second and subsequent instances are displayed as error locations.</td>
<td>Block name, instruction number, duplicated label</td>
<td>Warning (Can be set)</td>
</tr>
<tr>
<td>Duplicate use of interrupt I/O address</td>
<td>The same address is used more than once in INTP instructions. The second and subsequent instances are displayed as error locations.</td>
<td>Block name, instruction number, duplicated address</td>
<td>Warning (Can be set)</td>
</tr>
<tr>
<td>Error Message</td>
<td>Description</td>
<td>Additional Information</td>
<td>Category</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------------------------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>IL-ILC mismatch</td>
<td>IL-ILCs do not correspond one-to-one. IL-ILCs within a block must correspond one-to-one. Although nesting is allowed for F3SP22-0S/SP28-3S/SP38-3S/SP53-4S/SP58-6S/SP59-7S/SP66-4S/SP67-6S/ SP76-7N/SP71-4S/SP76-7S CPU models, the number of IL and ILCs must be identical. Furthermore, IL must precede ILC.</td>
<td>Block name, instruction number</td>
<td>Warning (Can be set)</td>
</tr>
<tr>
<td>SUB-RET mismatch</td>
<td>SUB-RETs do not correspond one-to-one. SUB-RETs within a block must correspond 1:1.</td>
<td>Block name, instruction number</td>
<td>Warning (Can be set)</td>
</tr>
<tr>
<td>INTP-IRET mismatch</td>
<td>INTP-IRETs do not correspond one-to-one. INTP-IRETs within a block must correspond 1:1.</td>
<td>Block name, instruction number</td>
<td>Warning (Can be set)</td>
</tr>
<tr>
<td>FOR-NEXT mismatch</td>
<td>FOR-NEXTs do not correspond one-to-one. FOR-NEXTs within a block must correspond 1:1.</td>
<td>Block name, instruction number</td>
<td>Warning (Can be set)</td>
</tr>
<tr>
<td>CALL-SUB mismatch</td>
<td>CALL destination label (subroutine) was not found.</td>
<td>Block name, instruction number, Label name</td>
<td>Warning (Can be set)</td>
</tr>
<tr>
<td>Jump destination not found.</td>
<td>A jump destination was not found.</td>
<td>Block name, instruction number, label name</td>
<td>Warning (Can be set)</td>
</tr>
<tr>
<td>HRD/HWR instructions exceed maximum limit of the model.</td>
<td>The number of high-speed READ/WRITE instructions exceeds the maximum allowed for the selected CPU type.</td>
<td></td>
<td>Warning (Can be set)</td>
</tr>
<tr>
<td>Block for ACT/INACT is registered more than once.</td>
<td>The same block name is found in 2 or more ACT/INACT instructions.</td>
<td>Block name, instruction number, error block name</td>
<td>Warning (Can be set)</td>
</tr>
<tr>
<td>Global device overlaps local device area.</td>
<td>A global device overlaps the local device range.</td>
<td>Block name, instruction number, device</td>
<td>Warning (Can be set)</td>
</tr>
<tr>
<td>Tag name converted to local/macro device.</td>
<td>Even if the reference tag name definition for a block is set to common tag name definition, if a tag name is assigned a local device in the block tag name definition, it will be converted to a local device. This message is displayed to indicate such a situation.</td>
<td>Block name, instruction number, tag Name</td>
<td>Warning (Can be set)</td>
</tr>
<tr>
<td>Inconsistent structure type.</td>
<td>The structure type definition names of the S and D parameters of an STMOV instruction are not the same.</td>
<td>Block name, instruction number</td>
<td>Error (not modifiable)</td>
</tr>
<tr>
<td>STRCT instruction error.</td>
<td>An invalid STRCT instruction has been found.</td>
<td>Block name, instruction number</td>
<td>Error (not modifiable)</td>
</tr>
<tr>
<td>Structure type definition file not found.</td>
<td>The structure type definition file specified in a structure object definition is not found in the common folder.</td>
<td>Block name, structure type definition name</td>
<td>Error (not modifiable)</td>
</tr>
<tr>
<td>Undefined constant</td>
<td>A constant name coded as an instruction parameter is not defined in the constant definition.</td>
<td>Block name, Instruction number, tag name</td>
<td>Error (not modifiable)</td>
</tr>
<tr>
<td>Constant name definition area is exceeded.</td>
<td>When evaluated, constant definition exceeds the maximum number of constant definition devices allowed.</td>
<td></td>
<td>Error (not modifiable)</td>
</tr>
<tr>
<td>Constant definition is not available for selected CPU.</td>
<td>Constant definition is used for a sequence CPU that does not support constant definition.</td>
<td></td>
<td>Error (not modifiable)</td>
</tr>
</tbody>
</table>

⚠️ **CAUTION**

- Program checking is performed on items registered in the executable program components definition and saved in a file.
- There are other error messages related to scripts in addition to those indicated in this manual. For details on script error messages, see the volume entitled "Script."
D3.2.2 Finding in Project
Use the Find in Project function to search and list all blocks where a specific device or comment is used.
You can open an offline edit window or monitor window directly from the project search results display window.

SEE ALSO
For details on the Find in Project function, see Section E4.1, "Find."

D3.2.3 Finding Instruction in Project
Use the Find Instruction in Project function to search and list all blocks where a specified instruction is used.

SEE ALSO
For details on the Find Instruction in Project function, see Section E4.1.2, "Finding Instruction."

D3.2.4 Replacing in Project
Use the Replace in Project function to replace tag names and addresses in blocks registered in the executable program for a project.

SEE ALSO
For details on the Replace in Project function, see Section E4.2.2, "Replacing over an Entire Project."
D3.2.5 Changing an Address Allocated to a Tag Name over an Entire Project

You can change an address allocated in a tag name definition over an entire project. To do so, use the project replace function.

Although you can also change the tag name definition, in general, it is effective only if each block refers to the tag name definition or if blocks are created using addresses.

■ Using Project Replace

Set the following conditions in the Replace in entire project tab on the Replace dialog box.
- Turn on the [Replace in Tag Name Definitions too] checkbox.
- Turn on the [Address->Address] option button for [Replacement Method].
- Turn on the [All Blocks] option button for [Replacement Range].

When you set these conditions, you can replace parameter addresses created in blocks and change address allocations in the block tag name definitions of all blocks and the common tag name definition. Through these operations, you can replace all occurrences of an address in a project.

⚠️ CAUTION

- During conversion of tag name definitions, conversion cannot be performed if the registered address exists. If a tag name definition cannot be converted, it will be skipped but conversion will still proceed with all other tag name definitions. Please check the results display.
- The wildcard character ("**") cannot be used.
D3.2.6 Changing a Tag Name Assigned to an Address over an Entire Project

You can change the tag name assigned to an address over an entire project.

Although you can also change tag name definition, if programs are developed using tag names, changing tag name definition alone does not change the tag names in blocks. Since you also need to change each tag name definition, the project replace function is convenient as it makes all required changes at the same time.

- **Using Project Replace**

  Set the following conditions in the Replace in entire project tab on the Replace dialog box.
  - Turn on the [Replace in Tag Name Definitions too] checkbox.
  - Turn on the [Tag Name->Tag Name] option button for [Replacement Method].
  - Turn on the [All Blocks] option button for [Replacement Range].

  When you set these conditions, you can replace the tag name in all blocks and change the tag names in the block tag name definition of each block and the common tag name definition.

  ![Figure D3.42 Changing a Tag Name Over an Entire Project](D0302_06.VSD)

- **CAUTION**

  - When performing replacement in tag name definitions, replacement cannot be done if the tag name is already registered. If a tag name definition cannot be converted, it will be skipped but conversion will still proceed with all other tag name definitions. Please check the results display.
  - The wildcard character ("*") cannot be used.
  - Replacement is not allowed for constant definition.
D3.2.7 Changing I/O Installation Position

Use the Change I/O Installation Position function when the slot where the module is mounted changes. This function changes the slot number parameters in all I/O devices, READ, WRITE, HRD and HWR instructions, etc. used within a project in a single operation. It also changes assignments in common tag name definitions and block tag name definitions at the same time.

**CAUTION**

The Change I/O Installation Position function cannot be used in blocks that refer to the common library or other projects.

To change the I/O installation position, use the following procedure.

◆ Procedure ◆

1. Close all blocks being edited.
2. Select [Project]–[Change I/O Installation Position] from the menu.
   ⇒ The Change I/O Installation Position dialog box opens.
3. Specify the old slot number, new slot number, and number of slots to replace.
   **TIP**
   Specify the old slot number and new slot number from 002 to 716.
   1 to 16 slots can be specified for replacement. In the example on the right, the installation positions of the I/O modules are to be changed from slot numbers 002 to slot numbers 003.
4. Specify the replacement range.
   **TIP**
   To perform the change for some blocks only, turn on the [Specified Block] option button and click [Browse]. Select the desired blocks from the displayed Block List dialog box and click [OK]. Control returns to the Change I/O Installation Position dialog box.
5. Click [OK].
   ⇒ Processing begins for changing the installation position of I/O slots. A Change I/O Installation Position – Progress dialog box is displayed with progress information.
TIP
To abort processing at any time, click [Cancel].

When processing is completed, a completion dialog box is displayed.

(6) Click [OK].
⇒ The completion dialog box closes.
  If processing ends successfully, the slot numbers of target I/O relays as well as READ, WRITE, HRD and HWR instructions are replaced. If an error occurs in a block during execution, changes are not performed for that block.

(7) Click [Close].
⇒ The Change I/O Installation Position dialog box closes.

SEE ALSO
For details on slot number, see "Sequence CPU – Functions."
### Change I/O Installation Position – Progress Dialog Box

The following table shows the messages that may be displayed in the Change I/O Installation Position – Progress dialog box.

**Table D3.12  Messages for Change I/O Installation Position Operation**

<table>
<thead>
<tr>
<th>Message</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>N items replaced.</td>
<td>N occurrences of the specified devices have been replaced successfully.</td>
</tr>
<tr>
<td>Skipped because error occurred when modifying tag name definition.</td>
<td>An error was encountered during replacement and processing was aborted. Replacement was not performed for the blocks.</td>
</tr>
<tr>
<td>No modification in tag name definition.</td>
<td>No tag name is defined with the specified slot and thus no replacement was done.</td>
</tr>
<tr>
<td>Only tag name definition is changed because there is no change in circuit.</td>
<td>Address allocations in tag name definitions has been changed.</td>
</tr>
<tr>
<td>Replacement cancelled.</td>
<td>Replacement was aborted because a user has clicked [Cancel].</td>
</tr>
<tr>
<td>File not found.</td>
<td>There is no block file defined in the components definition.</td>
</tr>
<tr>
<td>Cannot change block saved with error.</td>
<td>Project replacement cannot be performed for blocks saved with errors.</td>
</tr>
</tbody>
</table>

**CAUTION**

- When the [Number of Slots to Replace] specified is such that the range of the new slot numbers is beyond the existing slots, the following message will be displayed: “Invalid input value.”
  
  For instance, the following input generates an error:
  
  Old slot number     002  
  New slot number     016  
  Number of slots to replace     4  
  
  In this example, the 4 slots, 002 to 005, are to be changed to 016 to 019, but slots 017, 018, and 019 do not physically exist, leading to an error.

- If the slot range specified when changing I/O installation position is such that an address with a new slot is already allocated to a tag name in a tag name definition, an error is generated.
D3.2.8 Displaying a Device List
The device list function lists the usage status of each device.
By displaying a device list, you can check which devices are used or not in the project.

SEE ALSO
- For details on the device list, see Chapter E5, "Device List."
- For details on the cross reference, see Chapter E6, "Cross Reference."

D3.2.9 Obsolete Device List
You can display a list of devices used in all blocks.
The list does not include devices used in macros.

SEE ALSO
For details on Obsolete Device List, see Section E4.1.9, "Obsolete Device List."

D3.2.10 Finding a Circuit Comment-out in a Project
Use the Find Circuit Comment-out in Project function to search and list all blocks where circuit/instruction comment-out is used.

SEE ALSO
For details on the Find Circuit Comment-out in Project function, see Section E4.1.10, "Find Circuit Comment-out."
### D3.3 Managing Files

You can add, rename or delete project files. The following table summarizes each function, with its description and target file types.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Target Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insert File</td>
<td>Copies files of other projects and folders to the opened project.</td>
<td>- Block file (circuit and tag name definition files)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Common tag name definition file</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Constant definition file</td>
</tr>
<tr>
<td>Rename File</td>
<td>Changes the names of files in the opened project.</td>
<td>- Block file (circuit and tag name definition files)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Instruction macro file (circuit and tag name definition files)</td>
</tr>
<tr>
<td>Delete File</td>
<td>Deletes files in the opened project.</td>
<td>- Block file (circuit and tag name definition files)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Instruction macro file (circuit and tag name definition files)</td>
</tr>
<tr>
<td>Copy File</td>
<td>Makes a copy of a file in the current project with a different file name.</td>
<td>- Block file (circuit and tag name definition files)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Instruction macro file (circuit and tag name definition files)</td>
</tr>
</tbody>
</table>
D3.3.1 Adding Files

Use the Insert File function to add blocks, common tag name definition files, constant definition files from other projects to the open project.

The procedure for adding files is given below.

◆ Procedure ◆

(1) Select [Project]–[Insert File] from the menu bar.
⇒ The Select File dialog box opens.

TIP
This dialog box can also be opened from the [Block List] pop-up menu in the Project Window.

(2) Move the cursor to the source project folder and select the file you want to add.

TIP
You can switch the display between block tag name definition, common tag name definition and constant definition by selecting the appropriate file type from the file type drop-down list.

(3) Click [Select].
⇒ The Select File dialog box closes and the file is added.

TIP
If a file with the same name exists in the open project, a confirmation dialog box is displayed.

Click [Yes] to overwrite the existing file.
D3.3.2 Renaming Files

Use the Rename File function to change the names of block and macro files in a project. The procedure for changing the names of block and macro files is given below.

◆ Procedure ◆

1. Select [Project]–[Rename File] from the menu bar.

   TIP
   This dialog box can also be opened from the [Block List] pop-up menu in the Project Window. In addition, the [Rename File] dialog box is directly opened when selected from the Block/Macro Name pop-up menu.

   ⇒ The Select File dialog box opens.

2. Select the file to be renamed, and click [Select].

   ⇒ The Rename File dialog box opens.

3. Enter the new name and click [OK].

   ⇒ The name of the block or macro file is changed.

   TIP
   If a file with the same name exists in the open project, a confirmation dialog box is displayed.

   Figure D3.44 Confirmation Dialog Box

   Click [Yes] to overwrite the existing file.
D3.3.3 Deleting Files

Use the Delete File function to delete blocks and macro files in a project.
The procedure for deleting files is given below.

◆ Procedure ◆

(1) Select [Project]–[Delete File] from the menu bar.
⇒ The Select File dialog box opens.

TIP
This dialog box can also be opened from the [Block List] pop-up menu in the Project Window.
In addition, the delete confirmation dialog box is displayed when selected from the Block/Macro Name pop-up menu.

(2) Select the file to be deleted.
(3) Click [Select].
⇒ A dialog box opens to confirm the delete operation.

(4) Click [Yes].
⇒ The selected file is deleted.
D3.3.4 Copying Files

Use the Copy File function to make a copy of a block or macro file in a project with a different file name.

The relevant procedure for making a copy of a file with a different name is given below.

◆ Procedure ◆

(1) Select [Project]–[Copy File] from the menu bar.
⇒ The Select File dialog box opens.

**TIP**

This dialog box can also be opened from the [Block List] pop-up menu in the Project Window.
In addition, the [Copy File] dialog box is directly opened when selected from the Block/Macro Name pop-up menu.

(2) Select the file to be copied.
(3) Click [Select].
⇒ A dialog box is displayed for entering a new file name to be used for a copy of the file.

(4) Enter a new file name and click [OK].
⇒ A copy of the selected file is created with the specified name.
D3.4 Tag Names and Addresses

To promote program development using tag names, operations for editing programs and specifications relating to tag name definition have been changed in WideField3. The main changes are in the addition of two new functions.
- Entering only tag names in the program edit window
- Setting reference tag name definition for each block.

The user interface and internal file storage are also changed accordingly. This section describes tag name definition related settings outside of the tag name definition window, as well as data processing and cautions due to the difference in how internal data is stored within tag name definitions.

SEE ALSO
For details on the user interface for editing tag name definitions, see Chapter E2, "Tag Name Definition."

D3.4.1 About Reference Tag Name Definition

In WideField3, you can have two types of definitions within a single project: common tag name definition and block (macro) tag name definition.

By setting the reference tag name definition for each block, it gives you flexibility in having blocks that processes common data within a project or blocks that manages data on a block basis.

Setting Reference Tag Name Definition

By setting for each block the tag name definition to refer, you can achieve independent or shared data representation for blocks.

Changing Reference Tag Name Definition

To change the tag name definition that a block refers to, select [Edit]–[Local Device/Properties] from the menu bar. Change the setting for the [Reference Tag Name Definition] item on the Local Device/Properties dialog box.
If you want to change the reference tag name definition of a block registered in the execution block components, you can also use the Executable Program Details/Setup dialog box.

**SEE ALSO**
For details on the Executable Program Details/Setup dialog box, see Section D3.1.16, "Component Block Setup."

- **Common Tag Name Definition**
  Select this option to have a block refer to the common tag name definition. The block will refer to the common tag name definition to perform mapping between tag names, addresses and I/O comments within the block in all operations including program editing, downloading and uploading. However, when a tag name defined in the common tag name definition is also defined in the tag name definition as a local device, the block tag name definition has precedence.

- **Block Tag Name Definition**
  Select this option to have a block refer to the block tag name definition. The block will refer to the block tag name definition to perform mapping between tag names, addresses and I/O comments within the block in all operations including program editing, downloading and uploading.
D3.4.2 About Tag Name Design

In WideField3, you can write programs by entering only tag names in the ladder program edit window or you can allocate addresses and I/O comments for new tag names at the same time.

You can select to enter tag name with or without entering address and I/O comment at the same time by changing the setting for the [Set up Parameter Input] item in the Set up Environment dialog box.

Enter Tag Name with Address and I/O Comment

Turning on this setting opens the Address Assignment dialog box when you enter a tag name or address not registered in the tag name definition in a ladder program edit window.

Enter an address or I/O comment for the tag name. The data entered here will be stored to the reference tag name definition. Clicking [OK] without entering anything stores only the tag name to the tag name definition. Clicking [Cancel] stores nothing to the tag name definition.

TIP

Entering tag name with address and I/O comment is not available when entering structure members.
Enter Tag Name without Address and I/O Comment

Turning on this setting reflects entered data on the program immediately after you enter a tag name or address in a ladder program edit window.

An entered tag name is not stored to the tag name definition.

If you have entered a new tag name, you must allocate an address to the tag name using tag name definition before performing program syntax checking.

D3.4.3 Formats for Storing Parameters of Block Files

This section explains the different types of parameters stored in block files and how these data are changed by different operations.

What Is a Block File?

A file that stores ladder program data for a block is called a block file.

A block file contains ladder program data and the list of parameters for each instruction. The data for each instruction parameter may be stored as an address, a tag name, a constant name or a constant value.

Data Saved Through Editing a Block

Parameter data saved to a block is the same as the data entered during program editing.

When a tag name is entered in an Edit Block window, the tag name is saved in the file. If an address is entered instead, the address is saved in the file.

Data Saved Through Uploading or Online Edit Update

When you upload a program or reflect online edited changes to a file, WideField3 refers to the reference tag name definition and perform address-to-tag-name conversion regardless of the format of the entered data. An address is saved as a tag name if it is allocated to a tag name and saved as an address, if otherwise.
Effect on Edit Display

In program editing, data stored in files are displayed in instruction parameter setup dialog boxes. As shown in the following figure, even if the same parameter is entered, it may sometimes appear differently in an instruction parameter setup dialog box.

![Instruction Parameter Setup Dialog Box](image)

Effect on Find and Replace Operations

You can perform searches using either tag name or address regardless of how data is stored in a file. You can search using an address allocated to a tag name even if the block file is saved with tag names. Similarly, you can search using a tag name assigned to an address even if the block file is saved with addresses.

Online Operations

Although a compiled program is downloaded with all data in address format, instruction parameter changes made during online editing are displayed in tag names. Search operations in a program monitor ignores differences in data formats (address or tag name formats). When performing update processing for online edited changes, WideField3 performs address-to-tag-name conversion and stores the offline block file in tag name format.
Converting to Address Format

The Convert Data to Address Format function converts a block file to address format. Converting a block file to address format facilitates replacements by address. This function can also be used to migrate to address-based program development.

To convert data to address format, select [Project]–[Convert Data]–[Address Format] from the menu bar and perform conversion on the Convert to Addresses dialog box displayed.

You can also select specific blocks (macros) to be converted.

![Figure D3.49 Convert to Addresses Dialog Box](D0304_05.VSD)

**TIP**

If you invoke the convert data to address format function with a ladder program edit window (tag name definition window) open after having made changes in the window, a confirmation dialog box will be displayed.

**CAUTION**

- You cannot abort a conversion before its completion.
- If an error is encountered during conversion of a block, the conversion process skips over the block and proceeds to convert subsequent blocks.
Converting to Tag Names

The convert data to tag name format function converts a block file to tag name format. Converting a block file to tag name format facilitates replacements by tag name. To convert data to tag name format, select [Project]–[Convert Data]–[Tag Name Format] from the menu bar and perform conversion on the Convert to Tag Names dialog box displayed.

You can also select specific blocks (macros) to be converted.

![Convert to Tag Names Dialog Box](D0304_06.VSD)

**Figure D3.50  Convert to Tag Names Dialog Box**

**TIP**

If you invoke the convert data to tag name format function with a ladder program edit window (tag name definition window) open after having made changes in the window, a confirmation dialog box will be displayed.
D4. Creating and Managing Blocks and Macros

A program to be executed by the FA-M3 consists of block and macro files. Block and macro files store programs in the form of ladder diagrams. Each block or macro file has two types of names as follows:

- **Block or macro file name**
  A block or macro file name is used when the block or macro is handled on a PC. This is the name of the block file (.YBLK) or macro file (.YMCR).

- **Block or macro name**
  A block or macro name is used when the block or macro is stored in a CPU module. This name is a string of up to 8 characters.

A block or macro file name and a block or macro name are the same as each other, and usually you do not need to distinguish one from another.

**TIP**

The block and macro names mentioned above correspond to the block and macro names in WideField3.

A block or macro file name is used when a block or macro is handled on a PC. A block or macro name is used when a block or macro is stored in a CPU module.

This chapter describes how to open, close, and save block and macro files. Before you can create a ladder diagram program, you must open a project. The following description assumes that a project is already opened.

**Function Limitations for Each CPU Type**

<table>
<thead>
<tr>
<th>CPU Modules</th>
<th>Limitations</th>
<th>SEE ALSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>F3SP22-0S, F3SP28-3S, F3SP38-6S, F3SP53-4S, F3SP58-6S, F3SP59-7S</td>
<td>N/A</td>
<td>-</td>
</tr>
<tr>
<td>F3SP66-4S, F3SP67-6S</td>
<td>N/A</td>
<td>-</td>
</tr>
<tr>
<td>F3SP71-4N, F3SP76-7N</td>
<td>N/A</td>
<td>-</td>
</tr>
<tr>
<td>F3SP71-4S, F3SP76-7S</td>
<td>N/A</td>
<td>-</td>
</tr>
</tbody>
</table>

**TIP**

This section does not describe hardware-dependant limitations for each CPU type, such as the number of available devices.

**SEE ALSO**

For details on limitations for each CPU type, refer to the user’s manual for each type.
D4.1 Creating New Block and Macro Files

To create a new block or macro file, use the following procedure.

◆ Procedure ◆

(1) Confirm that a project is open.
(2) Select [File]–[New] from the menu bar.
   ⇒ The New dialog box opens with the Block/Macro tab displayed.
(3) Click the [Block] or [Macro] button.
(4) Enter a name for a file to be created in the [Block/Macro Name] text box.
   TIP
   When you enter a block or macro name, the entry is automatically reflected to the file name field.
(5) Select a reference tag name definition.
(6) Click [OK].
   ⇒ The edit ladder program window and the Local Device/Properties dialog box open.
(7) Sets the various items (properties) in the Local Device/Properties dialog box.
   TIP
   You can also change block/macro file properties later. To do so, open a file, and select [Edit]–[Local Device/Properties] from the menu bar to open the Local Device/Properties dialog box.

SEE ALSO
For details on the Local Device/Properties dialog box, see Section D4.4, “Editing Local Devices and Properties of Blocks and Macros.”

(8) Click [OK].
   ⇒ The Local Device/Properties dialog box closes.
**TIP**

- New block and macro files that are created must be saved before they are displayed in the project window. Files that have been saved are displayed under [Block List] or [Macro List] in the project window.

![Project Window (Example of Saving the BLK01 Block File)](D0401_03.VSD)

**Figure D4.1 Project Window (Example of Saving the BLK01 Block File)**

- After creating a block or macro file, create a ladder program in the edit ladder program window.

**SEE ALSO**

For details on how to save a block or macro file, see Section D4.3, “Saving Block and Macro Files.”
D4.2 Opening Block and Macro Files

This section describes how to open a block or macro file. To open a block or macro file, you can use one of the following two ways:
- Opening a block or macro in the project currently opened
- Opening a block or macro in a specified project

General Procedure

To open a block or macro in the project currently opened, use the following procedure.

◆ Procedure ◆

(1) Select [File]–[Open]–[Block/Macro] from the menu bar.
⇒ The Open File dialog box opens.
(2) Select a file.
(3) Click [Open].
⇒ The file opens to show the ladder program edit window.

Step (2)

Step (3)
TIP
You can also double-click a block or macro name in the project window to open it.

CAUTION
If you try to open a password-protected file, the Enter Password dialog box opens. Enter the correct password and click [OK] to open the file.

SEE ALSO
For details on the Local Device/Properties dialog box, see Section D4.4, "Editing Local Devices and Properties of Blocks and Macros."
Opening a Block or Macro in a Specified Project

When you open a block or macro in a specified project, select one of the following file open modes:
- Read mode
- Write mode

A block or macro opened in write mode can be saved by overwriting the existing file, but a block or macro opened in read mode cannot.

The procedure for opening a block or macro is nearly the same in either mode. To open a block or macro in read mode, use the following procedure.

 Procedure

(1) Select [File]–[Open]–[Reference Block (Read Mode)] from the menu bar.
⇒ The Open File dialog box opens.
(2) Select a file.
(3) Click [Open].
⇒ The selected file is opened and the ladder program edit window is displayed. The window title shows "(Read Mode)".

TIP
In read mode, you can open a block currently opened in another WideField3 instance.
D4.3 Saving Block and Macro Files

You can save an edited block or macro file with an existing file name (Save) or a new file name (Save As). In either case, the edited block or macro file are saved together with the tag name definition file.

To save a block or macro file, use the following procedure.

◆ Procedure ◆

(1) Ensure that the edit window for the block or macro file to be saved is active.

(2) Select [File]–[Save] from the menu bar.
   ⇒ The ladder program is converted and saved.

TIP

If a conversion error occurs, the error dialog box opens. Clicking [OK] displays a dialog box to confirm the save operation. Clicking [Yes] saves the file with error. Clicking [No] cancels the save operation.

CAUTION

If there are no conversion errors, the ladder diagram is converted to instruction words and saved. If a file is saved with errors, invalid parts of the ladder diagram is saved “as is”. The following operations cannot be performed for a block that has been saved with errors.

- Syntax checking of executable programs
- Downloading
- Data conversion
- Address conversion for project
- Changing I/O installation position in project
To save an edited file with a different file name, use the following procedure.

◆ Procedure ◆

1. Check that the edit window for the block or macro file to be saved is active.
2. Select [File]–[Save As] from the menu bar.
   ⇒ The Save As dialog box opens.

3. Enter a new file name in the [File name] text box.
   **TIP**
   You may not enter an existing file name.

4. Click [Save].
   ⇒ A dialog box is displayed for confirming whether to continue your edit operation with the file to be saved.

5. To edit the file to be saved with the specified file name, click [Yes]. To continue to edit the currently opened file, click [No].
   ⇒ The selected file is saved with the specified name.
TIP
- If a conversion error occurs, an error dialog box opens. Clicking [OK] displays a dialog box to confirm the save operation. Clicking [Yes] saves the file with errors. Clicking [No] cancels the save operation.

Figure D4.5 Confirmation Dialog Box
- To edit the original file of the block or macro file that you have saved with a different name, select [File]–[Open]–[Block/Macro] and reopen the original file.

CAUTION
If there are no conversion errors, the ladder diagram is converted to instruction words and saved. If a file is saved with errors, invalid parts of the ladder diagram is saved "as is". The following operations cannot be performed for a block that has been saved with errors.
- Syntax checking of executable programs
- Downloading
- Data conversion
- Address conversion for project
- Changing I/O installation position in project
D4.4 Editing Local Devices and Properties of Blocks and Macros

Use the Local Device/Properties dialog box to view or edit properties of block and macro files. To open the Local Device/Properties dialog box, use the following procedure.

◆ Procedure ◆

(1) Ensure that the editing window for the block or macro file to be viewed or edited is active.
(2) Select [Edit]–[Local Device/Properties] from the menu bar.
⇒ The Local Device/Properties dialog box opens.

Local Device/Properties Dialog Box Items

The layout of the Local Device/Properties dialog box is given below.

Figure D4.6 Local Device/Properties Dialog Box
- **A** Name  
The name of a block or macro file.

- **B** Title  
To change the title, enter a new title in the [Title] text box.  
The title may consists up to 24 characters.

- **C** Block/Macro Name  
The name of a block or macro used when the block or macro is stored in the CPU module. Up to 8 characters.

- **D** Step Count (including Tag Name Definitions)  
When the [Refresh Step Count] button is clicked, the block program step count is displayed. If tag name definitions are stored, the tag name definition step count is added to the block program step count and displayed.

- **E** Protection  
Indicates whether the block or macro file is password-protected.

- **F** Local Devices  
Specify the number of local devices (i.e., internal relays, data registers, file registers, timers, counters, and cache registers) to be used in the block being edited.

You can use up to the specified number of local devices during block or macro editing. For example, setting [Internal Relays] to 32 allows you to use devices /I1 to /I32 as local devices; setting a value 0 does not allow the use of local devices within the block.

**TIP**

The number of local devices must be set in the following units:

- Local internal relays  
  In units of 32 relays

- Local data registers  
  In units of 2 registers

- Local file registers  
  In units of 2 registers

- Local cache registers  
  In units of 2 registers

- Local timers  
  In units of 1 timer

- Local counters  
  In units of 1 counter

Only 10 ms timers can be used as local timers.

Before using local devices in a block, you must set up the block and device area accordingly.

When you set the number of local devices, you can automatically specify a reserve area.

**SEE ALSO**

- For details on local devices, see Chapter F2, "Using Local Devices."

- For details on setting a device area, see Section D3.1.8, "Device Setup."

- For details on the automatic setting of the number of local devices, see Section D1.2.4, "Circuit Display/Input Setup."
- **G** Reference Tag Name Definition
  Selects the reference tag name definition for tag names used in a block or macro file. By default, the [Common Tag Name Definition] option button is on.

- **H** Store to CPU (enabled for SP22-0S/28-3S/38-6S/53-4S/58-6S/59-7S/66-4S/67-6S/71-4N/76-7N/71-4S/76-7S)
  Specifies whether to store comments and tag name definitions along with programs at downloading. If this item is not selected for a block or macro file, then comments and tag name definitions are not stored to the CPU even if you specify to do so when downloading the block or macro.

  For WideField3R2.04 or later, when a new block or macro file is created, both [Store Circuit Comment/Subcomment] and [Store Block Tag Name Definition] are on by default.

- **I** [Set Protection] button and [Remove Protection] button
  When you try to open a password-protected file, the Enter Password dialog box opens to prompt for a password. You must also enter a valid password to remove password protection.

---

**WARNING**

When you set password protection, make sure that you are responsible for managing your passwords.

We cannot accept any request from any user, including persons responsible for setting passwords, to clear passwords that have been set.
D4.5 Opening CADM3 Format Blocks and Macros

You can open blocks or macros that have been created with the ladder program tool M3 (SF510-J3N and SF510-J3P). The relevant procedure is given below.

◆ Procedure◆

(1) Select [File]–[Open]-[CADM3 File] from the menu bar.
⇒ The Open CADM3 File dialog box opens.
(2) Select a CADM3-formatted file.
(3) Click [Open].
⇒ The CADM3 format file is converted to WideField3 format as a new block or macro file.
If the conversion ends successfully, a confirmation dialog box opens.

(4) Click [OK].
⇒ The converted data is displayed in the Open CADM3 File dialog box.
(5) Click [Close].
⇒ The name of the converted file is included in [Block List] or [Macro List] in the project window. The ladder program edit window opens.

CAUTION

- This operation is available only when a project is open.
- The original CADM3 format file remains intact.
- The block or macro tag name definition is used as the reference tag name definition.
- The results of conversion are saved in text format in the "Henkan.log" file in the project folder.

SEE ALSO

For details on how to open a CADM3 format executable program as a project, see Section D2.2.7, "Opening Projects in Other Formats."
D4.6 Using the Common Library

The common library function helps you to reuse your program resources by registering blocks that can be shared between different projects.

D4.6.1 Registering Blocks with the Common Library

You can register your blocks with the common library.

The blocks registered with the common library can be referred to from any projects. Reusing program resources in this way helps improve the development efficiency.

The relevant procedure is given below.

◆ Procedure ◆

(1) Select [Project]–[Add to Library...] from the menu bar.
⇒ The Select File dialog box opens.

(2) Select the block to register with the library, and click [Select].
⇒ A confirmation dialog box is displayed.

(3) Click [Yes].
⇒ The selected block is registered with the common library.
D4.6.2 Renaming Block Files in the Block Library

You can rename a block file registered with the common library.
The relevant procedure is given below.

◆ Procedure ◆

(1) Select [Project]–[Rename Library Block...] from the menu bar.
⇒ The Select File dialog box opens.

(2) Select the block file to rename, and click [Select].
⇒ The Rename File dialog box is displayed.

(3) Enter a new block file name in the text box.

(4) Click [OK].
⇒ The block file registered with the library is renamed.

CAUTION
If you rename a block file in the common library and the block file is already registered with the component definition, an error occurs in the component definition.
We recommend that before you rename a block file in the common library, you check that the block is not registered with the component definition of another project.

SEE ALSO
For details on the component definition, see Section D3.1.3, "Executable Program Setup."
D4.6.3 Deleting Registered Blocks from the Common Library

You can delete a registered block from the common library. The relevant procedure is given below.

◆ Procedure ◆

(1) Select [Project]–[Delete Library Block...] from the menu bar.
⇒ The Select File dialog box opens.
(2) Select the block to delete, and click [Select].
⇒ A confirmation dialog box is displayed.

(3) Click [Yes].
⇒ The registered block is deleted from the library.

⚠️ CAUTION

If you delete a block in the common library and the block is already registered with the component definition, an error occurs in the component definition.

We recommend that before you delete a block from the common library, you check that the block is not registered with the component definition of another project.
This manual describes operations such as creating and printing ladder programs using WideField3.
E1. Editing Programs

To create an executable program, you first edit a ladder program. Next, you define the program components. Always have a project open when you create an executable program or edit a ladder program. Here, we describe how to edit a ladder program.

Function Limitations for Each CPU Type

Table E1.1  Function Limitations for Each CPU Type

<table>
<thead>
<tr>
<th>CPU Modules</th>
<th>Limitations</th>
<th>SEE ALSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>F3SP22-0S, F3SP28-3S, F3SP38-6S, F3SP53-4S, F3SP58-6S, F3SP59-7S</td>
<td>Long-word index modification cannot be used.</td>
<td>E1.2.20</td>
</tr>
<tr>
<td></td>
<td>Double-long-word instructions cannot be used.</td>
<td>E1.2.10</td>
</tr>
<tr>
<td></td>
<td>Double-precision floating-point numbers cannot be used.</td>
<td>E1.2.17</td>
</tr>
<tr>
<td>F3SP66-4S, F3SP67-6S</td>
<td>Long-word index modification cannot be used.</td>
<td>E1.2.20</td>
</tr>
<tr>
<td></td>
<td>Double-long-word instructions cannot be used.</td>
<td>E1.2.10</td>
</tr>
<tr>
<td></td>
<td>Double-precision floating-point numbers cannot be used.</td>
<td>E1.2.17</td>
</tr>
<tr>
<td>F3SP71-4N, F3SP76-7N</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>F3SP71-4S, F3SP76-7S</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>

TIP

This section does not describe hardware-dependant limitations for each CPU type, such as the number of available devices.

SEE ALSO

For details on limitations for each CPU type, refer to the user's manual for each type.
E1.1 Cautions for Editing Ladder Program

This section describes the limitations in editing ladder programs and necessary precautions to avoid creating circuits that will cause conversion errors and circuits that cannot be amended.

E1.1.1 Limitations on Editing Ladder Program

Editing a ladder program inserts circuits to create a block. The following limitations apply when editing ladder programs.

- Maximum number of lines that can be created per block
  Each block may contain up to 20,000 lines of circuits, including circuit comments. Circuits may not be entered nor edited beyond this maximum limit.

- Maximum number of steps that can be created per block
  The F3SP22-0S/28-3S/38-6S/53-4S/58-6S/59-7S/66-4S/67-6S/71-4N/76-7N/71-4S/76-7S CPUs support up to 57,344 steps (56K steps) of circuits per block; the F3SP05/21/25/35/08/28-3N/38-6N/53-4H/58-6H, F3FP36 CPUs support up to 10,240 steps (10K steps) of circuits per block. If the number of steps entered exceeds the upper limit, an error occurs during conversion. To create circuits larger than this maximum, divide into smaller blocks.

- Maximum number of lines per circuit
  One circuit can have up to 25 lines. If the number of lines entered exceeds the upper limit, an error occurs during conversion.

- Maximum number of instructions per circuit
  One circuit can have up to 128 instructions. If the number of instructions entered exceeds the upper limit, an error occurs during conversion.

- Maximum number of continuation lines per circuit comment
  One circuit comment can have up to 25 lines. If the number of continuation lines entered exceeds the upper limit, an error occurs during conversion.

- Maximum number of continued circuits
  One line of circuit can have up to 2 continuation circuits. If the number of continuation circuits entered exceeds the upper limit, an error occurs during conversion.

E1.1.2 Circuits with Conversion Errors

The following circuits return errors (conversion errors) at the time of conversion.

Circuits with No Output Instruction

![Diagram of Circuits with No Output Instruction]

Figure E1.1 Circuits with No Output Instruction

TIP
Output instructions are instructions, such as OUT instructions that output data to devices. They are usually coded on the right hand side of a ladder program.
Multi-layered Circuits with No Output in the First Layer

Figure E1.2  Multi-layered Circuits with No Output in the First Layer

Disjointed Circuits

Figure E1.3  Disjointed Circuits

Circuits with Superfluous Branch Lines

Figure E1.4  Circuits with Superfluous Branch Lines

Circuits with Reverse Flow from Right to Left

Figure E1.5  Circuits with Reverse Flow from Right to Left
Circuits with Short-Circuit Line

Figure E1.6  Circuits with Short-Circuit Line

Circuits with 4 or More Continuation Lines

Figure E1.7  Circuits with 4 or More Continuation Lines

TIP

Continuation lines are circuits connected with a continuation symbol.
Circuits with Invalid Continuation

TIP
If a circuit is too wide to fit in the ladder program, you can continue the circuit on the next line using a continuation circuit.

CAUTION
Coding of continuation lines avoids errors similar to that shown below.

× Example of incorrect coding

✓ Example of correct coding
Circuits with Converging Lines

Figure E1.9  Circuits with Converging Lines

Circuits with AND-type instruction (UP, DWN, UPX, DWNX, INV instructions and continuous type application instructions) at invalid position

Figure E1.10  Circuits with And-type Instructions (e.g. INV instructions) at Invalid Position

Circuits with multiple branches in the same line to which the same logic is specified

Figure E1.10_2  Circuits with multiple branches in the same line to which the same logic is specified

⚠️ CAUTION

Sometimes, a circuit which is coded within 25 lines during editing may exceed 25 lines due to re-layout during conversion and fail to convert and return with a conversion error.
E1.1.3 Circuits that Cannot be Amended Using Ladder Programming

Sections (circuits) that cannot be recovered with a ladder program are displayed as mnemonics. These sections displayed in mnemonics can be deleted but cannot be corrected and added using the ladder program edit function. Delete these sections after entering an equivalent process using a ladder program.

**TIP**

Sections displayed in mnemonics can be changed with the mnemonic editing function if the program can be created as a circuit.

For details about the mnemonic editing function, see Section E1.2.43, “Mnemonic Editing Function”.
E1.2 Editing Ladder Programs

This function allows you to enter instructions and create ladder program circuits.

E1.2.1 Ladder Program Edit Screen Layout

Ladder Program Edit Screen Elements

The ladder program edit screen layout contains the following elements.

- **A** Instruction Palette
  Stores drawing elements which you can use to edit circuits using the mouse.

- **B** Ladder Program Edit Window
  Window for editing a ladder program. The window title displays “Edit Block” when editing a block file and “Edit Macro” when editing a macro file.

- **C** Status Bar
  Displays information about the edit window.

- **D** Output Window
  Displays the cross reference window.

- **E** Balloon List
  Displays a list of balloon comments in which you can edit and manage balloon comments.

- **F** Find Toolbar
  Used to search for instruction parameters.
# Instruction Palette

The instruction palette stores as icons drawing elements which can be used to edit a ladder programs in the ladder program edit screen.

## Table E1.2 Instruction Palette

<table>
<thead>
<tr>
<th>Icon</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Selector Icon" /></td>
<td>Selector</td>
<td>Returns the mouse indicator to the usual selector cursor. Clicking on a circuit element with the selector cursor selects a circuit element. Double-clicking a circuit element with the selector cursor displays the functions of the circuit element for user selection.</td>
</tr>
<tr>
<td><img src="image" alt="Contact A Icon" /></td>
<td>Contact A (LD AND)</td>
<td>Changes the mouse indicator to the Contact A cursor. Clicking anywhere inserts a contact A.</td>
</tr>
<tr>
<td><img src="image" alt="Contact B Icon" /></td>
<td>Contact B (LDN, ANDN)</td>
<td>Changes the mouse indicator to the Contact B cursor. Clicking anywhere inserts a contact B.</td>
</tr>
<tr>
<td><img src="image" alt="Contact A OR Icon" /></td>
<td>Contact A OR (OR)</td>
<td>Changes the mouse indicator to the Contact A OR cursor. Clicking anywhere inserts a contact A OR.</td>
</tr>
<tr>
<td><img src="image" alt="Contact B OR Icon" /></td>
<td>Contact B OR (OR)</td>
<td>Changes the mouse indicator to the Contact B OR cursor. Clicking anywhere inserts a contact B OR.</td>
</tr>
<tr>
<td><img src="image" alt="Contact A OR (both ends) Icon" /></td>
<td>Contact A OR (both ends)</td>
<td>Changes the mouse indicator to the Contact A OR (two ends) cursor. Clicking anywhere inserts a contact A OR (both ends).</td>
</tr>
<tr>
<td><img src="image" alt="Contact B OR (both ends) Icon" /></td>
<td>Contact B OR (both ends)</td>
<td>Changes the mouse indicator to the Contact B OR (two ends) cursor. Clicking anywhere inserts a contact B OR (both ends).</td>
</tr>
<tr>
<td><img src="image" alt="OUT instruction Icon" /></td>
<td>OUT instruction (Output coil)</td>
<td>Changes the mouse indicator to the OUT instruction cursor (output coil cursor). Clicking anywhere inserts an OUT instruction in the last column.</td>
</tr>
<tr>
<td><img src="image" alt="OUTN instruction Icon" /></td>
<td>OUTN instruction (Out Not)</td>
<td>Changes the mouse indicator to the OUTN instruction cursor (Out Not cursor). Clicking anywhere on the screen inserts an OUTN instruction in the last column.</td>
</tr>
<tr>
<td><img src="image" alt="Application instruction Icon" /></td>
<td>Application instruction</td>
<td>Changes the mouse indicator to the application instruction cursor. The Enter Instruction dialog box is displayed.</td>
</tr>
<tr>
<td><img src="image" alt="Instruction List Icon" /></td>
<td>Instruction List</td>
<td>Changes the mouse indicator to the instruction list cursor. The Instruction List dialog box is displayed.</td>
</tr>
<tr>
<td><img src="image" alt="TIM instruction Icon" /></td>
<td>TIM instruction (Timer)</td>
<td>Changes the mouse indicator to the TIM instruction cursor. Clicking anywhere inserts a TIM instruction in the last column.</td>
</tr>
<tr>
<td><img src="image" alt="CNT instruction Icon" /></td>
<td>CNT instruction (Counter)</td>
<td>Changes the mouse indicator to the CNT instruction cursor. Clicking anywhere inserts a CNT instruction in the last column.</td>
</tr>
<tr>
<td><img src="image" alt="SET instruction Icon" /></td>
<td>SET instruction</td>
<td>Changes the mouse indicator to the SET instruction cursor. Clicking anywhere inserts a SET instruction in the last column.</td>
</tr>
<tr>
<td><img src="image" alt="RST instruction Icon" /></td>
<td>RST instruction (Reset)</td>
<td>Changes the mouse indicator to the RST instruction cursor. Clicking anywhere inserts a RST instruction in the last column.</td>
</tr>
<tr>
<td><img src="image" alt="Insert Connection Line Icon" /></td>
<td>Insert Connection Line</td>
<td>Changes the mouse indicator to the connection line cursor. Dragging the mouse anywhere draws a connection line.</td>
</tr>
<tr>
<td><img src="image" alt="Delete Connection Line Icon" /></td>
<td>Delete Connection Line</td>
<td>Changes the mouse indicator to the delete connection line cursor. Dragging the mouse removes a connection line in an arbitrary location.</td>
</tr>
<tr>
<td><img src="image" alt="Continuation Line Icon" /></td>
<td>Continuation Line</td>
<td>Changes the mouse indicator to the continuation line cursor. Clicking anywhere inserts a continuation line.</td>
</tr>
<tr>
<td><img src="image" alt="Circuit Comment Icon" /></td>
<td>Circuit Comment</td>
<td>Changes the mouse indicator to circuit comment cursor. Clicking on any line inserts a subcomment.</td>
</tr>
<tr>
<td><img src="image" alt="I/O Comment Icon" /></td>
<td>I/O Comment</td>
<td>Switches to I/O Comment Input Mode. Double-clicking any element displays the I/O comment input field.</td>
</tr>
<tr>
<td><img src="image" alt="Subcomment Icon" /></td>
<td>Subcomment</td>
<td>Changes the mouse indicator to the subcomment cursor. Clicking anywhere inserts a subcomment.</td>
</tr>
<tr>
<td><img src="image" alt="Label Icon" /></td>
<td>Label</td>
<td>Changes the mouse indicator to the label cursor. Clicking anywhere inserts a label.</td>
</tr>
<tr>
<td><img src="image" alt="LDU instruction Icon" /></td>
<td>Load Differential Up instruction</td>
<td>Changes the mouse indicator to the Load Differential Up instruction cursor. Clicking anywhere inserts a Load Differential Up instruction.</td>
</tr>
<tr>
<td><img src="image" alt="LDD instruction Icon" /></td>
<td>Load Differential Down instruction</td>
<td>Changes the mouse indicator to the Differential Down instruction cursor. Clicking anywhere inserts a Load Differential Down instruction.</td>
</tr>
<tr>
<td><img src="image" alt="UP instruction Icon" /></td>
<td>Logical Differential Up instruction</td>
<td>Changes the mouse indicator to the Logical Differential Up instruction cursor. Clicking anywhere inserts a Logical Differential Up instruction.</td>
</tr>
<tr>
<td><img src="image" alt="DOWN instruction Icon" /></td>
<td>Logical Differential Down instruction</td>
<td>Changes the mouse indicator to the Logical Differential Down instruction cursor. Clicking anywhere inserts a Logical Differential Down instruction.</td>
</tr>
<tr>
<td><img src="image" alt="INU instruction Icon" /></td>
<td>Inverter instruction</td>
<td>Changes the mouse indicator to the Inverter instruction cursor. Clicking anywhere inserts an Inverter instruction.</td>
</tr>
<tr>
<td><img src="image" alt="Operation Guide Display Icon" /></td>
<td>Operation Guide Display</td>
<td>Displays the operation key guide on the program edit screen.</td>
</tr>
</tbody>
</table>
**Ladder Program Edit Window**

Ladder program edit windows are used to edit circuits. Here, ladder programs of up to 20,000 lines and 11 columns can be edited. Like all usual windows in Microsoft Windows, the edit window can be resized, split into panes, moved to a new display position, maximized to full screen, or minimized to an icon. The screen can be scrolled up and down using the vertical scroll bar or scrolled right and left using the horizontal scroll bar.

![Ladder Program Edit Window](E1012_02.VSD)

**Figure E1.12  Ladder Program Edit Window (Edit Block window example)**

- **A** Line Number/Instruction Number Display Area
  Displays the line number, instruction number and labels currently being edited.
- **B** TipHelp
  Displays tag name, device and I/O comment at the mouse cursor. You can choose to show or hide TipHelp in the environment setup.
- **C** Cell
  Area for writing the instruction mnemonic and parameters. The cell displays the circuit element, tag name, address, and I/O comments.
- **D** Vertical Split Bar
  Slide the split bar to split the window vertically into two panes. To cancel the split window, slide the split bar to the top or bottom position in the window, or double-click the split bar.
- **E** Position Cursor
  For specifying where to write instructions, and for selecting instructions. The cursor can be moved by using the mouse or the arrow keys.
- **F** Subcomment Display Area
  Displays sub-comments.
- **G** Vertical Scroll Bar
  For scrolling the ladder program edit window up and down. The window can also be scrolled a page at a time by pressing the [Page Up] and [Page Down] keys, and a line at a time by pressing the up/down arrow keys.
- **H** Horizontal Split Bar
  Slide the split bar to split the window horizontally into two panes. To cancel the split window, slide the split bar to the leftmost or rightmost position in the window, or double-click the split bar.
- **I** Horizontal Scroll Bar
  For scrolling the ladder program edit window sideways. The window can also be moved sideways a column at a time by pressing the left/right arrow keys.
TIP

- Other settings that can be done from the ladder program edit window include displaying or hiding display fields and enlarging or reducing the window size.
- By using the vertical or horizontal split bar, you can split a window either horizontally or vertically into up to two panes.

SEE ALSO

For details on showing or hiding display fields, enlarging or reducing the window size, see Section E1.5, “Switching Views”.

For details on environment setup, see Section D1.2, “Environment Setup.”

### Status Bar

The status bar displays information about the block being edited.

The status bar displays the number of lines, number of steps, and drawing mode (overwrite or insert mode) during ladder editing, as shown in the figure below.

![Status Bar](E0102_03.VSD)

**Figure E1.13 Status Bar**

- A Displays the number of lines, number of steps, number of comment steps for the block being edited. The number of steps is updated at conversion. It is not updated during editing. The number of comment steps is computed from the preset value of whether comments and tag name definitions are to be stored to the CPU.
- B Displays the drawing mode as 'Ins.' (Insert) or 'Ovr.' (Overwrite). You can toggle between the drawing modes by pressing the [Insert] key.
E1.2.2 Set up Circuit Display and Input

Selecting [Tools]–[Set up Environment] displays the Set up Environment dialog box. You can customize circuit editing and parameter input as well as set window display colors using the Circuit Display/Input tab on the displayed dialog box.

![Figure E1.14  Circuit Display/Input tab](Image)

SEE ALSO
For more details on the Circuit Display/Input tab on the Set up Environment dialog box, see Section D1.2.4, “Circuit Display/Input Setup.”

E1.2.3 Distinguishing Devices

You can distinguish tag names of global devices, local devices, unassigned tag names and undefined constant names using colors.

![Figure E1.15  Device Status Display in Circuit Edit](Image)

SEE ALSO
For more details on the Circuit Display/Input tab on the Set up Environment dialog box, see Section D1.2.4, “Circuit Display/Input Setup.”
E1.2.4 Basic Operations

Instructions used when editing a ladder program include basic instructions, as well as application instructions such as comparison instructions and arithmetic instructions.

There are basically 3 ways to enter an instruction in a ladder program edit window: using mouse, function keys or keyboard.

Mouse operation can be used to enter drawing elements by simply clicking the buttons on the instruction palette.

Function key operation can be used to enter instructions and parameters simultaneously by pressing the function keys.

In keyboard operation, you enter the mnemonics directly using the alphanumeric keys.

TIP

Basic instructions refer to a group of instructions that function like electrical circuit elements such as relays and coils. Except for some instructions such as the TIM instruction, basic instructions are single bit operations.

SEE ALSO

- For details on basic instructions, see “Sequence CPU Instruction Manual – Instructions” (IM34M06P12-03E).
- For details on input using the mouse and the alphanumeric keys, see the descriptions in Section E1.2.9, “Entering Basic Instructions” and Section E1.2.10, “Entering Application Instructions.”
- For details on input using alphanumeric keys, see Section E1.2.5, “Input Using Alphanumeric Keys.”
Special Operation Keys of Program Edit Window

The table below lists special operation keys that can be used in the program edit window.

### Table E1.3 Special Operation Keys of Program Edit Window

<table>
<thead>
<tr>
<th>Key Outline</th>
<th>Outline of Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>←→</td>
<td>Moves the position cursor to the left or right. These keys do not move the cursor to the line number or subcomment area.</td>
</tr>
<tr>
<td>↑↓</td>
<td>Moves the position cursor upwards or downwards.</td>
</tr>
<tr>
<td>Shift + ←</td>
<td>Moves the position cursor to the line number/instruction number display area.</td>
</tr>
<tr>
<td>Shift + →</td>
<td>Moves the position cursor to the subcomment display area.</td>
</tr>
<tr>
<td>Ctrl + arrow keys</td>
<td>Selects a range of cells.</td>
</tr>
<tr>
<td>Shift + ↑↓</td>
<td>Selects a range of lines.</td>
</tr>
<tr>
<td>Home</td>
<td>Jumps to the beginning of the line.</td>
</tr>
<tr>
<td>End</td>
<td>Jumps to the end of the line.</td>
</tr>
<tr>
<td>Ctrl+ Home</td>
<td>Jumps to the first line.</td>
</tr>
<tr>
<td>Ctrl+ End</td>
<td>Jumps to the last line.</td>
</tr>
<tr>
<td>Page Up (for displaying pages continuously)</td>
<td>Scrolls the edit screen upwards.</td>
</tr>
<tr>
<td>Page Down (for displaying pages continuously)</td>
<td>Scrolls the edit screen downwards.</td>
</tr>
<tr>
<td>Ctrl+ Page Up</td>
<td>Moves the position cursor to the previous circuit comment.</td>
</tr>
<tr>
<td>Ctrl+ Page Down</td>
<td>Moves the position cursor to the next circuit comment.</td>
</tr>
</tbody>
</table>
E1.2.5 Input Using Alphanumeric Keys

In the program edit window, you can enter or edit instruction words in mnemonic format or instruction parameters directly from the keyboard. The relevant procedure is given below.

⚠️ CAUTION ⚠️

Before inputting instructions using alphanumeric keys, you need to turn on the [Use parameter direct change input] checkbox in the [Parameter Direct Change Input Settings] group box on the Circuit Display/Input tab in the Set up Environment dialog box.

SEE ALSO

For details on the Circuit Display/Input tab in the Set up Environment dialog box, see Section D1.2.4, “Circuit Display/Input Setup.”

(1) Entering instruction words

◆ Procedure ◆

(1) Move the position cursor to the desired input position.

(2) Enter the first character of the mnemonic for the desired instruction using the keyboard.

⇒ The Enter Instruction dialog box is displayed. The character entered earlier is displayed in the text box.

(3) Proceed to enter the remaining letters of the instruction mnemonic.

TIP

For example, enter the following instructions.

LD X201
MOV D1 D2

(4) Click [OK].

⇒ The instruction is entered at the location of the position cursor.
SEE ALSO

You can use alphanumeric keys to write programs by inserting Contact A instructions, Contact B
instructions, and connection lines at the same time. The available input keys and function are listed in
the table below.

Table E1.4 Input Keys and Functions

<table>
<thead>
<tr>
<th>Input Keys</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>LD</td>
<td>Inserts a contact A not connected to the line above.</td>
</tr>
<tr>
<td>LDN</td>
<td>Inserts a contact B not connected to the line above.</td>
</tr>
</tbody>
</table>
| LD2        | Inserts at the position cursor a contact A connected
            on the left to the line above. |
| LDN2       | Inserts at the position cursor a contact B connected
            on the left to the line above. |
| OR         | Inserts a contact A not connected to the line above. |
| ORN        | Inserts a contact B not connected to the line above. |
| OR2        | Inserts at the position cursor a contact A connected
            at both ends to the line above. |
| ORN2       | Inserts at the position cursor a contact B connected
            at both ends to the line above. |

(2) Entering or modifying instruction parameters directly

Before entering or modifying instruction parameters directly, you need to turn on the
[Use parameter direct change input] checkbox and the [Parameter change has priority] option button in the [Parameter Direct Change Input Settings] group box on the Circuit
Display/Input tab in the Set up Environment dialog box.

◆ Procedure ◆

(1) Move the position cursor to the
position of the instruction parameter
to be entered or modified.

(2) Enter a parameter using the keyboard.
⇒ The Instruction Parameter Setup dialog
box is displayed in a format that depends
on the parameter you entered.

(3) Complete entering the instruction
parameter.

(4) Click [OK].
⇒ The instruction parameter at the position
cursor is updated.
E1.2.6 Entering Instructions Using Input Completion Function

When entering an instruction, the input completion function speeds up the process by displaying a list of instruction candidates beginning with the character that has been entered for selection.

The relevant procedure is given below.

◆ Procedure ◆

(1) Enter the first character of the instruction to be entered from the keyboard.
⇒ The Enter Instruction dialog box is displayed. The character entered earlier is displayed in the text box.

(2) Press the [▼] key.
⇒ A list of instructions beginning with the entered character is displayed.

TIP
- You can narrow the displayed candidate list by entering an instruction string instead of a character.
- Instruction strings are not case-sensitive.
- To hide the displayed instruction list, press the [ESC] key.

(3) Use the [▼] and [▲] keys to move in the displayed candidate list and press [Enter] to select the desired instruction.
⇒ The selected instruction is entered in the text box.

TIP
- Instead of pressing the [ENTER] key, you can also press the [TAB] key or double-click the mouse button to select an instruction.

TIP
If a checkmark has been added to [Automatically Display Candidate] for Instruction/Instruction Parameter Completion in the environment setup, a list of input candidates will automatically be displayed when two or more characters have been entered in the Enter Instruction dialog box.
E1.2.7 Entering Instruction Parameters Using Input Completion Function

When you are entering a device (tag name or address) as an instruction parameter, the input completion function displays a list of instruction parameter candidates for your selection based on a tag name definition or a history of recent past entries after you have entered the first character.

In the case of tag name definition, input candidates are extracted from the reference tag name definition (common tag name definition or block tag name definition) defined in the block properties. In the case of input history, input candidates are taken from the 256 most recently entered devices. As a new entry is added to the input history, the oldest entry is automatically deleted.

The required procedure is given below.

◆ Procedure ◆

(1) Enter an instruction, followed by a space character.
⇒ The Enter Instruction dialog box is displayed with the instruction and space entered earlier displayed in the text box.

(2) Enter the first character of the desired instruction parameter and press the [↓] key.
⇒ A list of tag names and addresses beginning with the character entered earlier is displayed. I/O comments are also displayed in the list.

TIP
- Pressing the [↓] key before entering any character for the instruction parameter displays a list of instruction parameter candidates extracted from the input history.
- Address candidates are not extracted from tag name definitions.
- Up to 5000 input candidates can be displayed.
- Extraction of input candidates is not case-sensitive.
- To hide a displayed input candidate list, press the [ESC] key.

(3) Use the [↑] and [↓] keys to move in the displayed candidate list and press [Enter] to select the desired instruction parameter.
⇒ The selected instruction parameter is entered in the text box.

TIP
Instead of pressing the [ENTER] key, you can also press the [TAB] key or double-click the mouse button to select an instruction parameter.

TIP
If a checkmark has been added to [Automatically Display Candidate] for Instruction/Instruction Parameter Completion in the environment setup, a list of input candidates will automatically be displayed when two or more characters have been entered in the Enter Instruction dialog box.
E1.2.8 Entering Structures Using Input Completion Function

You can make use of the input completion function when entering structure names or structure members as instruction parameters. The procedure is slightly different from the procedure for entering tag names and addresses using input completion. The relevant procedure is given below.

◆ Procedure ◆

1. Enter an instruction, followed by a space character.
   The Enter Instruction dialog box is displayed and the instruction and space entered earlier are displayed in the text box.

2. Enter the first character of the structure name to be entered and press the [+] key.
   ⇒ Structure names of instantiated structures beginning with the entered character is displayed along with tag names and addresses. Type names instead of I/O comments are displayed in structure names.

   TIP
   Structure name candidates are not extracted from the input history, but only from structure names registered in the tag name definition.

3. Use the [↑] and [↓] keys to move in the displayed candidate list and press [Enter] to select the desired structure name.
   ⇒ The selected structure name is entered in the text box.

   TIP
   Instead of pressing the [ENTER] key, you can also press the [TAB] key or double-click the mouse button to select a structure name.

4. Next, enter the period (‘.’) character.
(5) Press the [↑] key again.
⇒ All the members belonging to the structure are displayed in a list. I/O comments or device types are also displayed in the list.

**TIP**
- Structure members are not extracted from the input history, but only from structure members defined in the type definition.
- If an input string contains the period character ("."), the candidate list displays structure members; otherwise, it displays tag names or addresses.
- Extraction of candidates is not case-sensitive.
- To hide a displayed candidate list, press [ESC].

(6) Select a member in the same way you would select a tag name or address, and press [Enter].
⇒ The structure member is selected.

**TIP**
If a checkmark has been added to [Automatically Display Candidate] for Instruction/Instruction Parameter Completion in the environment setup, a list of input candidates will automatically be displayed when two or more characters have been entered in the Enter Instruction dialog box.

**TIP**
Other than the Enter Instruction dialog box, the input completion function for structures is also available in Instruction Parameter Setup dialog boxes similar to the one shown below.
E1.2.9 Entering Basic Instructions

Basic Instructions include instructions like “Contact A”, “Contact B”, “OUT Instruction”, etc. There are 3 ways to enter basic instructions.

SEE ALSO
For details on input using alphanumeric keys, see Section E1.2.5, “Input Using Alphanumeric Keys.”

Input Using Mouse

To enter input using mouse, use the following procedure.

◆ Procedure ◆

(1) Click on the Basic Instruction button on the instruction palette.
   ⇒ The mouse cursor changes to the indicator for the selected instruction.
(2) Click at the location where the instruction is to be entered.
   ⇒ The selected instruction is written at the clicked position and the parameter input field appears.

⚠️ CAUTION

When entering an output instruction, the instruction is assigned to the last column and not at the clicked location. Furthermore, a horizontal connect line leads from the clicked position.

(3) Enter the tag name or address into the parameter input field.
(4) Press [Enter].
   ⇒ The parameter is entered.

⚠️ CAUTION

The selected instruction remains as the current instruction after input (the mouse cursor indicator remains unchanged). To continue writing the same instruction, click on the desired input location. To reset the instruction (and restore the mouse indicator to its usual form), perform any of the following operations.
- Press [ESC];
- Right click; or
- Click on the Selector button on the instruction palette.
CAUTION

Up to 8 characters can be displayed in a single line in the input field.
If a tag name does not fit on one line, it will be displayed as shown below depending on whether or not [Display All Parameters in 2 Lines] is selected.

Up to 16 characters can be displayed for tag names or structure tag names. For tag names or structure tag names longer than 16 characters, the seventeenth and subsequent characters are not displayed.

Figure E1.18   Examples of Display of Tag Names

Input Using Function Keys

To enter input using function keys, use the following procedure.

◆ Procedure ◆

(1) Move the position cursor to the desired input position.
(2) Press the function key for the instruction to be entered.
   ⇒ The instruction is coded at the specified location and the parameter input field appears.
(3) Enter the tag name or device address and press [Enter].
   ⇒ The device is assigned to the instruction. The figure on the right shows an example for entering a Contact A instruction.

TIP

Pressing [ESC] cancels instruction input.
When you confirm the input, the position cursor moves to the next cell.
E1.2.10 Entering Application Instructions

You can enter application instructions using the Instruction List dialog box or the Enter Instruction dialog box.

The Instruction List dialog box lists all the instructions in different categories and allows you to search for the desired instruction word.

The Enter Instruction dialog box allows you to directly enter the instruction and is convenient if you know which instruction to enter.

**Input from the Instruction List Dialog Box**

Application instructions are classified into comparison instructions, rotate instructions, data transfer instructions, etc.

The Instruction List dialog box displays the instruction categories on the left and the instruction words on the right. The [Pulse Instruction] checkbox is enabled only if the Pulse option is available for the selected instruction word. The same goes for the [Long Word Instruction] and [Double Long Word Instruction] checkboxes.

![Instruction List Dialog Box](E0102_28.VSD)

Figure E1.19 Instruction List Dialog Box

You can enter application instructions from the Instruction List dialog box using the mouse or alphanumeric keys.

1. **Input Using Mouse**

   To input using mouse, use the following procedure.

   ◆ **Procedure** ◆

   1. **Click the Instruction List button** on the instruction palette.
      ⇒ The Instruction List dialog box is displayed.

   2. **Select the instruction category and instruction word from the respective list boxes.**

   3. **Click [OK].**
      ⇒ The Instruction List dialog box closes and the mouse cursor changes to the indicator for the selected instruction.
(4) Click the mouse at the location where the instruction is to be entered.
⇒ The instruction is entered at the clicked position without parameters.

⚠️ CAUTION

The selected instruction remains as the current instruction after input (the mouse cursor indicator remains unchanged). To continue coding the same instruction, click on another input location. To reset the instruction (and restore the mouse indicator to its usual form), perform any of the following operations.
- Press [ESC];
- Right click; or
- Click on the Selector button on the instruction palette.

When entering an output instruction, the instruction is assigned to the last column, not at the clicked location, and a horizontal connection line is drawn from the clicked position.

---

TIP

An Application instruction is displayed as “instruction mnemonic, parameter 1, parameter 2, …”. If the tag name is longer than 8 characters, it is displayed, beginning from the top line; if it has 8 or fewer characters, it is displayed on the bottom line.

- Ordinarily

_ MOV_ D00001 D00002

- Rising edge (pulse) instruction

_ MOV_ D00001 D00002

An arrow is displayed above the instruction.

- Long word instruction

_ MOV_ L00001 L00002

“L” is displayed above the instruction.

- Double-long-word instruction

D D00001 D00002 D00001 D00002

“D” is displayed above the instruction.

- Floating point instruction

F F00001 F00002 F00003

“F” is displayed above the instruction.

- Double-precision floating-point instruction

E E00001 E00002 E00003

“E” is displayed above the instruction.

- Continuous type application instruction

C C00001 C00002 C00003

“C” is displayed above the instruction.

Figure E1.20  Display Format of Application Instructions
(2) Input Using Function Keys

To input using function keys, use the following procedure.

◆ Procedure ◆

(1) Move the position cursor to where the instruction is to be entered.

(2) Press the [Shift]+[F4] (Instruction List) keys.

⇒ The Instruction List dialog box is displayed.

(3) Select the instruction category and instruction word and press [Enter].

⇒ The instruction is entered and the Parameter Setup dialog box opens.

(4) Enter the parameter and press [Enter].

TIP

You can use the [Tab] key to move between the fields in the Parameter Setup dialog box.

⇒ The instruction and parameters are entered at the cursor position.
Input from the Enter Instruction dialog box

Similar to entering basic instructions, there are also three ways to enter application instructions using the Enter Instruction dialog box.

(1) Input Using Mouse

To enter input using mouse, use the following procedure.

◆ Procedure ◆

(1) Click on the Application Instruction button [I] on the instruction palette.

⇒ The Enter Instruction dialog box is displayed.

(2) Enter the instruction into the text box (instruction input field).

TIP
Enter the instruction parameter(s) after the instruction word. Separate the instruction word and parameters with a space. Clicking [Cancel] cancels the instruction input operation and restores the mouse cursor indicator to its usual form.

(3) Click [OK].

⇒ The mouse cursor changes to the indicator for the selected instruction.

(4) Click at the location where the instruction is to be entered.

⇒ The instruction will be entered at the clicked position.

⚠️ CAUTION

The selected instruction remains as the current instruction after input (the mouse cursor indicator remains unchanged). To continue coding the same instruction, click on the desired input location. To reset the instruction (and restore the mouse indicator to its usual form), perform any of the following operations.

- Press [ESC];
- Right click; or
- Click on the Selector button [I] on the instruction palette.

When entering an output instruction, the instruction is assigned to the last column, not at the clicked location, and a horizontal connection line is drawn from the clicked position.
(2) Input Using Function Keys

To enter an application instruction using function keys, move the cursor to the desired position and press [F4] (Application Input). Enter the instruction into the text box (instruction input field) on the Enter Instruction dialog box and press [Enter].

(3) Input Using Alphanumeric Keys

To enter input using the alphanumeric keys, use the following procedure.

◆ Procedure ◆

(1) Move the position cursor to the desired instruction input position.
(2) Enter the first character of the mnemonic for the desired instruction.
   ⇒ The Enter Instruction dialog box is displayed. The character entered earlier is displayed in the text box (instruction input field).
(3) Continue to enter the mnemonic.
(4) Click [OK].
   ⇒ The instruction is entered at the location of the position cursor.

Modifying Instruction Parameters

You can modify the parameters of an application instruction entered previously. There are two ways to change a parameter: using mouse or keyboard.

(1) Modifying Instruction Parameters Using Mouse

◆ Procedure ◆

(1) Double-click the instruction of which the parameter is to be changed. Or, move the position cursor to the instruction and press [Enter].
   ⇒ The Instruction Parameter Setup dialog box is displayed.

   TIP
   Up to 8 characters can be displayed in the Instruction Parameter Setup dialog box.

(2) Change the parameter.
(3) Click [OK]. Or, press [Enter].
   ⇒ The parameter is changed.
TIP

- The leftmost input field of the Instruction Parameter Setup dialog box is for entering the instruction word. It is in the form of a drop-down list and allows you to change the instruction properties (long word instruction, double-long-word instruction, and pulse instruction) easily.

Further, for the CAL instruction, the operator can also be changed from the drop-down list.

Figure E1.21  CAL Instruction

- You may either enter an address or a tag name for an instruction parameter; the parameter for the entered instruction will be displayed as an address or tag name, according to what was entered.

SEE ALSO

For details on modifying parameters quickly, see Chapter B3, "How to Rapidly Enter and Modify Instructions and Parameters" (Introduction and Troubleshooting).

(2) Modifying Instruction Parameters Using Alphanumeric Keys

◆ Procedure ◆

1. Move the position cursor to the position of the instruction parameter to be entered or modified.

2. Enter a parameter using the keyboard.
   ⇒ The Instruction Parameter Setup dialog box is displayed in a format that depends on the parameter you entered.

3. Complete entering the instruction parameter.

4. Click [OK].
   ⇒ The instruction parameter at the position cursor is updated.

TIP

Before modifying instruction parameters using alphanumeric keys, you need to turn on the [Use parameter direct change input] checkbox and the [Parameter change has priority] option button in the [Parameter Direct Change Input Settings] group box on the Circuit Display/Input tab in the Set up Environment dialog box.
E1.2.11 Entering Timer Instruction

The section describes how to enter the frequently used TIM instruction (Timer instruction) and change its parameters.

The procedure is given below.

(1) Input Using Mouse

Click on the TIM Instruction button on the instruction palette. Check that the mouse cursor indicator changes into the TIM instruction cursor. Click on the desired location to enter the TIM instruction.

(2) Input Using Alphanumeric Keys

The procedure for input using the alphanumeric keys is given below.

◆ Procedure ◆

(1) Move the position cursor to where the instruction is to be entered.
(2) Press the [F11] (TIM instruction) key. ⇒ The TIM Instruction Parameter Setup dialog box is displayed.
(3) Enter the parameter.
(4) Click [OK]. ⇒ A TIM instruction is entered at the location of the position cursor.

Modifying Parameters

To modify the parameters of the TIM instruction, double-click the TIM instruction or press [Enter]. Modify the parameter in the TIM Instruction Parameter Setup dialog box and click [OK].

SEE ALSO

For details on modifying parameters quickly, see Chapter B3, "How to Rapidly Enter and Modify Instructions and Parameters" (Introduction and Troubleshooting).
E1.2.12 Entering Counter Instructions

The section describes how to enter the CNT instruction (Counter instruction) and change its parameters.

The procedure is given below.

(1) Input Using Mouse

Click on the CNT Instruction button on the instruction palette. Check that the mouse cursor indicator changes into the CNT instruction cursor. Click on the desired location to enter the CNT instruction.

(2) Input Using Alphanumeric Keys

The procedure for input using the alphanumeric keys is given below.

◆ Procedure ◆

(1) Move the position cursor to where the instruction is to be entered.
(2) Press the [Shift]+[F11] (CNT instruction) keys.

⇒ The CNT Instruction Parameter Setup dialog box is displayed.
(3) Enter the parameter.
(4) Click [OK].

⇒ A CNT instruction is entered at the location of the position cursor
(3) Modifying Parameters

To modify the parameters of the CNT instruction, double-click the CNT instruction or press [Enter]. Edit the parameter in the CNT Instruction Parameter Setup dialog box and click [OK].

**TIP**

The counter reset condition is entered on the next line.

![Reset condition and CNT instruction joined by connection line](E0102_50.VSD)

**Figure E1.22  Reset Condition for CNT Instruction**

**SEE ALSO**

For details on modifying parameters quickly, see Chapter B3, "How to Rapidly Enter and Modify Instructions and Parameters" (Introduction and Troubleshooting).
E1.2.13 Entering Scripts

SEE ALSO
For details on entering scripts, see Chapter G2, "Creating and Editing Scripts" (Script).
E1.2.14 Deleting Ladder Program Elements
To delete an element, say an instruction, of a ladder program, move the position cursor to the element to be deleted and select [Edit]–[Delete] on the menu or press the [Delete] key.

E1.2.15 Entering Constants and Constant Names
This subsection describes constants (decimal, hexadecimal, and timer preset values) and constant names among the instruction parameters.

- Entering Constants (decimal, hexadecimal and timer preset values)

  Entering Decimal Constants
  There are positive and negative decimal constants.
  The ranges that can be handled are given below.
  - Word instruction
    -32768 to 32767 (BIN)
  - Long word instruction
    -2147483648 to 2147483647 (BIN)
  - Double-long-word instruction

  Entering Hexadecimal Constants
  Hexadecimal constants are entered with the ‘$’ prefix.
  Up to 4 digits can be entered for word instructions; up to 8 digits can be entered for long word instructions; up to 16 digits can be entered for double-long-word instructions.

  Entering Timer Preset Values
  When using constants for timer preset values, suffix a value in seconds by “s” and a value in milliseconds by “ms”. Values smaller than a millisecond are entered up to one decimal place.

  TIP
  For ONDLY, OFDLY, and PULSE instructions, values smaller than a millisecond are entered up to third decimal place.

  The following figure shows an example with value 1s100.0ms (1 s and 100.0 ms)
  Seconds are displayed in the top row and milliseconds up to one decimal place are displayed in the bottom row.

  Figure E1.23 Example of Display of Timer Constant

  TIP
  - Long word instructions are computed in 32-bit units.
  - Double-long-word instructions are computed in 64-bit units.
## Constant Names

Constant name definition allows meaningful names to be assigned to constants used in programs. You can map constant names to constant values using constant name definition, and once defined, constant names are common within the project. Using constant names as instruction parameters improves program readability and simplifies modification of the constant value of the same parameter throughout the project.

**SEE ALSO**

For details on constant definition, see Chapter E3, "Constant Definition."

Constant names can be used in programs and entered just like address numbers and tag names.

To use a constant name in a program, a constant value must be assigned to the constant name in the constant name definition.

The constant value for a constant name need not be defined when the constant name is coded in a program but should be defined before program syntax check and downloading.

**CAUTION**

- A program containing a constant name with no value assigned generates an error and cannot be transferred to the CPU. Always define constant values for constant names.
- Constant names can generally be used in place of constants as instruction parameters but cannot be used in some instructions.

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Parameters that Do Not Allow the Use of Constant Names in Place of Constants</th>
</tr>
</thead>
<tbody>
<tr>
<td>HRD High-speed Read Instruction</td>
<td>Parameters 1, 2 and 4</td>
</tr>
<tr>
<td>HWR High-speed Write Instruction</td>
<td>Parameters 1, 2, 3 and 4</td>
</tr>
</tbody>
</table>
E1.2.16 Entering Character Strings

This subsection describes character strings among the instruction parameters.

- Entering Character Strings

WideField3 handles characters and text strings of alphanumeric and symbol characters. However, individual instructions may have their own restrictions. Text strings are enclosed within double quotation marks (" ").

Up to 2 ASCII characters can be entered for a word instruction; up to 4 ASCII characters can be entered for a long word instruction; up to 8 ASCII characters can be entered for a double-long-word instruction.

TIP

- To use a character string constant, which is too long to be coded as an instruction parameter, assign the string constant to a constant name in the constant definition, and code the constant name in the program.
- The semicolon (";") character cannot be entered directly as a character constant as it is used as a symbol for index modification. To use the semicolon as a character constant, you can either define a constant name for it or enter its hexadecimal character code ($3B).

SEE ALSO

For details about constant definition, see Chapter E3, "Constant Definition."
E1.2.17 Entering Floating Points

This subsection describes floating points among the instruction parameters.

- Entering Floating Points

  The use of floating point numbers is limited to a subset of instructions. Floating point numbers are entered with the ‘%’ prefix. Negative values are entered with a ‘-‘ (minus) sign after the ‘%’ prefix. Floating points can be used with long word instructions, double-long-word instructions and floating point instructions.

  Example: %1.1E10; %1.2; %-3.4

  Display is right-justified, with 8 characters for the fraction and 4 characters for the exponent.

  The display format is “%X.XXXXXXEYYY”.

  X.XXXXXX is the fraction, YYY is the exponent. A minus sign, if any, is displayed immediately before “X.XXXXXX” and “YYY”.

  The range of data that can be entered for long word instructions is approximately as follows:

  - Negative numbers -3.4e37 to -1.5e-38
  - Positive numbers 1.5e-38 to 3.4e37

  The range of data that can be entered for double-long-word instructions is approximately as follows:

  - Negative numbers -1.79e308 to -2.23e-308
  - Positive numbers 2.23e-308 to 1.79e308

  **TIP**

  The display format of floating points can be changed in environment setup.

  **SEE ALSO**

  For details on changing display format, see Section D1.2.4, “Circuit Display/Input Setup.”
E1.2.18 Entering Escape Sequences, Block Names, and Label Names

This subsection describes escape sequences, block names, and label names among the instruction parameters.

\section*{Entering Escape Sequences}

Special characters that cannot be entered directly can be entered as hexadecimal codes, by prefixing each byte of hexadecimal code with "\text{x}" and enclosing the escape sequences within double-quote (" ") characters.

Escape sequences can be combined with character strings.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{E102_56.VSD}
\caption{Example of Entering Escape Sequences}
\end{figure}

\textbf{TIP}

- To use an escape sequence character string, which is too long to be coded as an instruction parameter, assign the character string to a constant name in the constant definition, and code the constant name in the program.

- The double-quote (" ") character cannot be entered directly as a character constant as it is used as a symbol for escape sequence. To use the double-quote as a character constant, define a constant name for it and use the constant name instead.

- Parameters coded as character strings can be displayed in an escape sequence code by changing the display format. In addition, if parameters entered in escape sequence codes are codes corresponding to character strings, they can be displayed as character strings by changing the display format.

\textbf{SEE ALSO}

For details on constant definition, see Chapter E3, "Constant Definition."

\section*{Entering Block Names}

Block names are entered as parameters of ACT, INACT, MCALL, NCALL, and SCALL instructions. Entered names are always displayed in uppercase.

\section*{Entering Label Names in Jump or Subroutine Instructions}

Labels name are entered as parameters of JMP or CALL instructions.
E1.2.19 Input and Display of Indexed Devices

To enter an indexed device, insert a semi-colon after the device name or tag name, followed by an indexed register or a constant. Parameters of both basic instructions and application instructions are entered the same way.

The following are some input examples:
- X201; V001 for entering an address
- SW1; V001 for entering a tag name
- I001; 1000 for entering a constant index

The devices as modified by the above indexes are displayed in the 2 figures below. Tag names can also be used for index registers; an address is displayed for an indexed device and a constant for a constant indexed device.

Example of Display of Contact and Coil

The devices as modified by the above indexes are displayed in the 2 figures below. Tag names can also be used for index registers; an address is displayed for an indexed device and a constant for a constant indexed device.

Example of Display of Indexed Device

[Diagram showing device example]

Example of Display of Indexed Device

CAUTION

- Constant index modification is only supported on F3SP22-0S/28-3S/38-6S/53-4S/58-6S/59-7S/66-4S/67-6S/71-4N/76-7N/71-4S/76-7S modules.
- Constant index modification accepts any positive decimal number between 0 and 2,047 (inclusive). Any other input causes an input error.
- A constant indexed register cannot exceed the device area defined in the configuration.

Confirmation Dialog Box

- If a constant indexed device overlaps a device on the output side, a “duplicate usage of coil” or “duplicate SET/RST” error occurs.
- If a constant indexed device exceeds the setup range for local devices or overlaps the range for global devices, an error is generated.

SEE ALSO

For details on index modification, see “Sequence CPU Instruction Manual – Instructions” (IM34M06P12-03E).
E1.2.20 Input and Display of Devices with a Long-Word Index Modification

To enter a device with a long-word index modification, insert a semi-colon after the device name or tag name, followed by an index register with a suffix "L". Parameters of both basic instructions and application instructions are entered the same way.

The following are some input examples:
- D00001;V001L for entering an address
- SIGNAME;V001L for entering a tag name

The long-word index modification uses two index registers. For example, if V00001L is entered, values in V00001 and V00002 are used as long-word index values.

Only an odd address can be specified.

Example:
Correct: D00001;V001L for specifying an odd address
Incorrect: D00001;V002L for specifying an even address

The following figure shows an example of display of devices with a long-word index modification.

![Figure E1.27 Example of Display of Long-Word Index Modification](E0102_59.VSD)

**CAUTION**

- Long-word index modification is supported only on F3SP71-4N/76-7N/71-4S/76-7S.
- Long-word index modification cannot be used for entering a constant index.
- The devices that support a long-word index modification are: Data (D) registers, file (B) registers, and cache (F) registers.
- If devices with a long-word index modification are used as standard registers in a program, regardless of the suffix "L", the devices are processed according to the data size for the instructions used in the program.

![Figure E1.28 Example of Long-Word Index Modification Operation](E0102_60.VSD)

Data at B(00001+V001L) = B(00001+40000) = B40001 is transferred to F(00001+V003L) = F(00001+100000) = F100001.

"L" is appended for long-word index.
E1.2.21 Input and Display of Indirect Specification Devices

An indirect specification device stores the address number of a target device and through this, allows execution of various instructions via the device.

You can indicate an indirect specification device by prefixing the device name or tag name with an "@" character. The D, B, F, R, W, /D, /B and /F devices support indirect specification.

The FA-M3 provides 3 instructions for using indirect specification devices: SET@, ADD@ and MOV@.

Table E1.6 Instructions for Using Indirect Specification Devices

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Syntax</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>SET@</td>
<td>SET@</td>
<td>D1</td>
</tr>
<tr>
<td></td>
<td>D1: indirect specification not allowed</td>
<td></td>
</tr>
<tr>
<td></td>
<td>D2: indirect specification</td>
<td></td>
</tr>
<tr>
<td>ADD@</td>
<td>ADD@</td>
<td>L</td>
</tr>
<tr>
<td></td>
<td>D1: indirect specification</td>
<td></td>
</tr>
<tr>
<td></td>
<td>n: direct or indirect specification</td>
<td></td>
</tr>
<tr>
<td>MOV@</td>
<td>MOV@</td>
<td>D1</td>
</tr>
<tr>
<td></td>
<td>D1: indirect specification</td>
<td></td>
</tr>
<tr>
<td></td>
<td>D2: indirect specification</td>
<td></td>
</tr>
</tbody>
</table>

INPUT EXAMPLES:

```
[SET@  P  D0010  @B002]
```

On line 1, the address number of D0010 is stored in B002. Next, the content of the indirect address (D0010) in B002 is incremented.

```
[INC   @B002]
```

```
[INC  B002]
```

If line 2 is not coded as an indirect address, the content of B002 will be incremented instead.

```
[SET@  P  D0020  @B003]
[ADD@       @B003        2   ]
[MOV@  P @B003 @B004]
```

Devices used as operands in instructions used for indirect specification function as indirect specification devices.

In the above sample code, the SET@ instruction stores in B003 the address of D0020, to be used in indirect specification.

In the ADD@ instruction on line 2, the value 2 is added to the value of D0020, the address used for indirect specification stored in B003 by the preceding SET instruction in line 1. Hence, D0022 is now stored in B003.

In the MOV@ instruction on line 3, the address stored in B003 instruction is transferred to B004. As a result, D0022 is stored in B004.
CAUTION

- Indirect specification devices are only supported on F3SP22-0S/28-3S/38-6S/53-4S/58-6S/59-7S/66-4S/67-6S/71-4N/76-7N/71-4S/76-7S modules.
- Indirect specification devices (prefixed by "@") cannot be assigned to tag names.

Table E1.7  Tag Name and Address

<table>
<thead>
<tr>
<th>Tag Name</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓ Data</td>
<td>D0001</td>
</tr>
<tr>
<td>✗ Data</td>
<td>@D0001</td>
</tr>
</tbody>
</table>
E1.2.22 Input and Display of Tag Names, Structure Names and Structure Member Names

Devices used in programs can be assigned with tag names or handled as structures.

- **Tag Names**

  Tag names are names assigned to devices used in programs. Multiple tag names can be assigned within a project using the tag name definition function. Coding programs with tag names improves program reusability and readability. Multiple tag names can be grouped into a group tag name for convenience.

  **SEE ALSO**

  For details on tag name definition and group tag names, see Chapter E2, “Tag Name Definition.”

Tag names can be used in a program similarly to address numbers. Both global devices and local devices can be assigned to tag names.

You do not need to know the actual address assigned to a tag name when entering or editing a tag name in a program but you will need to assign an address to each tag name eventually using the tag name definition function.

You may either refer to a tag name in a program and then perform tag name definition later, or you may perform tag name definition before referring to the tag name in a program.

A tag name with an address already assigned using tag name definition is displayed in black, as shown in the left figure below. A tag name with no address assigned yet is displayed in a different color, as shown in the right figure below.

![Figure E1.29 Tag Names with and without an Address Assigned](E0102_61.VSD)

Tag names used in a block may either refer to block tag name definitions defined for each block, or common tag name definitions defined for the entire project.

**SEE ALSO**

- For details on entering tag names, see Section E1.3, “Entering Tag Names and Addresses”.
- For details on tag names, see Section E1.4, “Circuit and Tag Name Definition Reference”
- For details on block tag name definition and common tag name definition, see Chapter E2, “Tag Name Definition.”

**CAUTION**

A program containing a tag name with no address assigned generates an error and cannot be transferred to the CPU. Always assign an address to a tag name.
## Structure

Structures allow you to group devices used in programs by purpose and function and assign names to each group. It facilitates device management and improves program reusability.

To use a structure, you must perform structure type definition to create a basic template for the structure in a project as well as perform structure entity definition to instantiate the structure. Once defined, you can use the structure in place of a device in the program.

**SEE ALSO**

- For details on structures, see Chapter F4, “Structures.”
- For details on instantiation of structures, see Chapter E2, “Tag Name Definition.”

### CAUTION

Structures are only supported on F3SP22-0S/28-3S/38-6S/53-4S/58-6S/59-7S/66-4S/67-6S/71-4N/76-7N/71-4S/76-7S modules.

You can use either a structure name or the individual member names of an instantiated structure in a program.

### Using Structure Names

You can use a structure name to refer to the members of the structure as a group.

The following table lists structure instructions that can be used in a program to handle structures which group multiple devices of heterogeneous types together.

**Table E1.8 Structure Instructions**

<table>
<thead>
<tr>
<th>Instruction</th>
<th>P</th>
<th>L</th>
<th>Syntax</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>STMOV</td>
<td>✓</td>
<td>✗</td>
<td>STMOV</td>
<td>D1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>D1: Source Structure Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>D2: Destination Structure Name</td>
<td></td>
</tr>
<tr>
<td>STRCT</td>
<td></td>
<td>✓</td>
<td>STRCT</td>
<td>D1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>D1: Structure Pointer (Q)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>D2: Block Name</td>
<td></td>
</tr>
<tr>
<td>SCALL</td>
<td>✓</td>
<td>✗</td>
<td>SCALL</td>
<td>n</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>n: Macro Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>D1: Structure parameter 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>D2: Structure parameter 2</td>
<td></td>
</tr>
</tbody>
</table>

P: Pulse, L: Long
CAUTION

- Structures used in an STMOV instruction must be of the same type. Otherwise, an error occurs.
- Structure instructions cannot be used in program monitoring or online editing.
- When using structure instructions, the step count depends on the members of the structure used in the instruction.

In STMOV and SCALL instructions, specify parameters using structure names as shown in the following figure (the table shows examples for structures AAA and CCC).

Table E1.9 Example of Structure Instructions and Parameters

<table>
<thead>
<tr>
<th>Structure Instruction</th>
<th>Parameter Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAA</td>
<td>AAA[1].addr</td>
</tr>
<tr>
<td>AAA[1]</td>
<td>AAA[1].addr</td>
</tr>
<tr>
<td>CCC</td>
<td>CCC[1].addr</td>
</tr>
<tr>
<td>CCC[1]</td>
<td>CCC[1].addr</td>
</tr>
</tbody>
</table>

SEE ALSO

For details on structure instructions, see “Sequence CPU Instruction Manual – Instructions” (IM34M06P12-03E).

Using Structure Member Names

You can refer to a member of a structure similarly to a regular tag name. As shown in the figure below, bit device structure members and word device structure members are coded in the program with bit device structure tag names and word device structure tag names respectively.

![Reference to Structure Members](E0102_62.VSD)

You do not need to know the actual address assigned to an instantiated structure tag name when entering or editing a structure tag name in a program but you will need to assign an address to each structure tag name eventually using the tag name definition function.

You may either refer to a structure tag name in a program and define the tag name later, or you may pre-define a structure tag name before referring to it in a program.
A structure tag name of an instantiated structure with an address already assigned using tag name definition is displayed in black, as shown in the left figure below. A structure tag name with no address assigned yet is displayed in green, as shown in the right figure below.

![Figure E1.31 Structure Member Names with and without an Address Assigned](E0102_63.VSD)

Tag names for structures used in a block may either refer to block tag name definitions defined for each block, or common tag name definitions defined for the entire project.

**SEE ALSO**

For details on structure instantiation, see Chapter E2, “Tag Name Definition” and Section F4, “Structures.”

---

**CAUTION**

A program containing an uninstantiated structure with no address assigned causes an error and cannot be transferred to the CPU. Always assign an address to a structure.
E1.2.23 Inserting Connection Lines

There are three ways to edit (draw and delete) connection lines: using mouse, using function keys and using keyboard.

Input Using Mouse

Connection lines can be drawn or deleted in any direction, not limited to vertical and horizontal lines.

(1) Inserting Connection Lines Using the Mouse

To draw a connection line using the mouse, use the following procedure.

◆ Procedure ◆

(1) Click on the Insert Connection Line button on the instruction palette. The mouse cursor changes to the Insert Connection Line cursor.

(2) Drag the cursor from the start point to the end point.

(3) Release the mouse at the end point. A connection line is inserted. A bold line in inverted circuit color joining the start point and the current end point is displayed when the mouse is dragged.

⚠️ CAUTION

The Insert Connection Line cursor indicator remains unchanged after a line insertion. You can continue to draw more lines. To cancel, press the [Esc] key or right-click the mouse button.

(2) Deleting Connection Lines Using the Mouse

To delete a connection line using the mouse, click on the Delete Connection Line button on the instruction palette. Check that the mouse cursor has changed to the Delete Connection Line cursor and drag the cursor from the start point to the end point. Release the mouse at the end point to erase the connection line. A bold line in inverted circuit color joining the start and end points is displayed when the mouse is dragged.

⚠️ CAUTION

The mouse Delete Connection Line cursor indicator remains unchanged after the line deletion. You can continue to delete more lines. To cancel, press the [Esc] key or click the right mouse button.
Input Using Function Keys

The procedures for drawing and deleting a connection line using function keys differ for vertical connection lines and horizontal connection lines.

(1) Inserting Vertical Connection Lines Using the Function Keys

To insert a vertical connection line using function keys, use the following procedure.

◆ Procedure ◆

(1) Move the position cursor to where a line is to be inserted.
(2) Press the [F8] (Insert Vertical Connection Line) key.
⇒ A vertical connection line is added at the position where the position cursor is.

(2) Deleting Vertical Connection Lines Using Function Keys

To delete a vertical connection line using function keys, use the following procedure.

◆ Procedure ◆

(1) Move the position cursor to where a line is to be deleted.
(2) Press the [Shift]+[F8] (Delete Vertical Connection Line) keys.
⇒ A vertical connection line is deleted from the position where the position cursor is.

(3) Inserting Horizontal Connection Lines Using Function Keys

To insert a horizontal connection line using function keys, use the following procedure.

◆ Procedure ◆

(1) Move the position cursor to where a line is to be inserted.
(2) Press [F9] (Insert Horizontal Connection Line) key.
⇒ A horizontal line is inserted at the position cursor. The position cursor moves to the cell on the right.
## Input Using Keyboard

The procedures for drawing and deleting a connection line using the keyboard differ for vertical connection lines and horizontal connection lines.

### (1) Inserting or Deleting Vertical Connection Lines Using the Keyboard

To insert or delete a vertical connection line using the keyboard, use the following procedure.

**Procedure**

1. Move the cursor to the desired input position (or the vertical connection line to be deleted) using the arrow keys.
2. Press the [\] key.
   - A vertical connection line is inserted (or deleted).

**TIP**

The [\] key is used either to insert or delete a vertical connection line. If the position cursor is on a vertical connection line, pressing [\] deletes the line. Otherwise, pressing [\] inserts a vertical connection line.

### (2) Inserting Horizontal Connection Lines Using the Keyboard

To insert a horizontal connection line using the keyboard, use the following procedure.

**Procedure**

1. Move the position cursor to where a line is to be inserted.
2. Press the [-] key.
   - A horizontal line is inserted.

**TIP**

To delete a horizontal connection line, press the [DELETE] key.
E1.2.24 Entering Continuation Circuit

There are 3 ways to enter a continuation circuit: using mouse, using function keys or using keyboard. The respective procedures are given below.

◆ Procedure ◆

(1) When using the mouse, click on the Continuation Line button on the instruction palette and then click on the position where the continuation line is to be inserted.

(2) When using function keys, move the cursor to the position where the continuation line is to be inserted and press the [Shift]+[F9] (Continuation Line) keys.

(3) When using the keyboard, move the position cursor to the position where the continuation line is to be inserted and press the [>] key or the [<] key. ⇒ A continuation line is displayed.

TIP

- Input Using Mouse
  Clicking on the beginning of a line generates a continuation line to the previous line.
  Clicking mid-way in the line generates a continuation line to the next line.

- Input Using Keyboard or Function Keys
  When the position cursor is in the first column, a continuation line is generated from the previous line.
  When the position cursor is in any column other than the first column, a continuation line is generated to the next line.
- Each line of a continuation circuit can be continued up to 2 times.

(1) 1:1 Continuation Line

- Continued up to 2 times

(2) N:N Continuation Lines

- Continued up to 2 times for each line of a continuation circuit.

- In a continued circuit, the number of symbols continuing to the next line must tally with the number of continuation lines leading from the previous line. Otherwise, an error results.

TIP

For continuation line circuits that have been correctly converted, return numbers are shown. Return numbers are shown in the format "(xx-yy)", but part of the number may be hidden.

- For 1:1 continuation lines, "-yy" is hidden.
- For N:N continuation lines, "xx-" is hidden if a single return occurs.
E1.2.25 Entering Labels

Up to 6 ASCII characters can be entered for a label. You can enter a label using any of the following ways:

- clicking the Label button on the instruction palette;
- double-clicking the label area of a circuit;
- using function keys ([Ctrl]+[F6]);
- moving the cursor to the label area and pressing the [Enter] key;
- entering '(' from the keyboard.

An entered label may be edited and deleted similarly. To enter a label, use the following procedure.

◆ Procedure ◆

1. Click on the Label button on the instruction palette and then click on the line where the label is to be inserted. Or, move the position cursor to the line where the label is to be inserted and press the [Ctrl]+[F6] (Label) keys. Or, move the mouse cursor to the label area and double-click the mouse button. ⇒ The Edit Label dialog box is displayed.
2. Enter the label in the text box.
3. Click [OK]. Or, press the [Enter] key. ⇒ The Edit Label dialog box closes and the label is entered.

● Editing Labels

To edit a label, open the Edit Label dialog box in the same way as entering a label. Edit the label displayed in the text box and click [OK] or press the [Enter] key.

● Deleting Labels

To delete a label, move the position cursor to the label area and press the [Delete] key. Alternatively, open the Edit Label dialog box in the same way as entering a label. Delete all the characters of the label displayed in the text box and click [OK] or press the [Enter] key. The label is deleted.
E1.2.26 Inserting Empty Lines

To insert an empty line, move the position cursor to where the line is to be inserted and select [Edit]–[Insert Line] from the menu or press the [Shift]+[Insert] keys.

Figure E1.34 Inserting Empty Lines

**TIP**

To remove an empty line(s) immediately after insertion, select [Edit]–[Undo] from the menu bar or press the [Ctrl]+[z] keys.
E1.2.27 Selecting Circuits

- Selecting Lines

You must select lines before performing line operations (copy, move, delete). Lines can be selected using the mouse or the keyboard.

To select lines using the mouse, you can select a line number area by dragging a mouse or by clicking start and end line numbers of the line number area. In both cases, up to 500 lines can be selected at a time. It is not possible to select more than 500 lines at a time.

Dragging a Line Number Area

![Diagram of line selection](E0102_72.VSD)

Clicking Start and End Line Numbers of a Line Number Area

Click a start line number. Press and hold the [Shift] key and click an end line number. To cancel the selection, click anywhere other than the line number area.

First, click a start number

Then, press and hold the [Shift] key and click an end line

![Diagram of line selection](E0102_72_2.VSD)
To select lines using the keyboard, press the [Shift] key together with the up/down arrow keys.

Figure E1.36  Selecting Lines Using Keyboard (Example for Selecting Lines 2 and 3)

Selecting Cells

You must select a cell range before performing cell operations (copy, move, delete) in a circuit. Cell ranges can be selected using the mouse or the keyboard.

To select a cell range using the mouse, drag the mouse indicator over the area for the cell range. Up to 500 lines can be selected at a time. It is not possible to select more than 500 lines at a time.

Figure E1.37  Selecting Cells Using Mouse (Example for Selecting Column 1 on Line 2 through Column 4 on Line 4)
To select cells using the keyboard, press the [Ctrl] key together with the up/down/left/right arrow keys.

**TIP**

To select a single cell, move the position cursor to the desired cell.

Figure E1.38  Selecting Cells Using Keyboard (Example for Selecting Column 1 on Line 2 through Column 4 on Line 4)
E1.2.28 Deleting Circuits

● Deleting Lines

Lines in specified areas can be deleted in line units. Circuit comments and subcomments are also deleted.

To delete some lines, use the following procedure.

◆ Procedure ◆

1. Select the line(s) to be deleted.
2. Select [Edit]–[Delete Lines] from the menu, or press the [Delete] key.
   ⇒ A confirmation dialog box is displayed.
3. Click [Yes].
   ⇒ The selected line is deleted.

TIP
To recover lines immediately after deletion, select [Edit]–[Undo] from the menu bar or press the [Ctrl]+[z] keys.
Deleting Cells

Cells in specified areas can be deleted in cell units. Circuit comments and subcomments are also deleted. To delete selected cells, use the following procedure.

◆ Procedure ◆

(1) Select the range of cells to be deleted.
(2) Select [Edit]–[Delete] from the menu, or press the [Delete] key.
⇒ A confirmation dialog box is displayed.

(3) Click [Yes].
⇒ The selected range of cells is deleted.

TIP
- Deleted cells become empty.
- To recover cells immediately after deletion, select [Edit]–[Undo] from the menu bar or press the [Ctrl]+[z] keys.

CAUTION

Deletion is not allowed with the following selected cell ranges. Performing a deletion operation on an invalid selection range causes an error.
- The selection range contains edit-prohibit line(s) (circuit comment lines, temporarily deleted lines, mnemonic lines, circuit lines containing instructions that disallow online editing.)
- The selection range contains part of an instruction.
E1.2.29 Temporarily Deleting Lines and Restoring Temporarily Deleted Lines in Circuits

- **Temporarily Deleting Lines**

  Lines within a selected range can be deleted in line units. Temporarily deleted lines are displayed with a different background color.

  This function is handy for temporarily deleting lines of circuits while editing programs, when you are unsure whether to delete the lines permanently.

  Circuit comments and subcomments for the lines are also temporarily deleted.

  To temporarily delete some lines, use the following procedure.

  ◆ **Procedure◆**

  (1) Select the line(s) to be deleted temporarily.

  (2) Select [Edit]–[Delete Lines] from the menu.

  ⇒ The selected lines are temporarily deleted. The lines are displayed with a different background color.

  **TIP**

  - When a block is converted, any temporarily deleted lines are permanently deleted from the program.
  - You can use the circuit comment out function to temporarily activate, inactivate, or disable specific instructions or circuits without permanently deleting lines by converting a block.

  **SEE ALSO**

  For details on the circuit comment out function, see Section K2.3, “Commenting Out Circuits and Instructions” (Online).

- **Undo Temporary Delete**

  Temporarily deleted lines in a specified area can be recovered so that they can be edited as usual. To recover temporarily deleted lines, select the lines to be recovered and then select [Edit]–[Undo Temporary Delete] from the menu.

  **TIP**

  Circuits cannot be temporarily deleted or restored in cell units.
E1.2.30 Copying and Moving Circuits

- **Copying and Moving Lines**

Lines in specified areas can be copied and moved to specified positions in line units. Lines can be copied and pasted to the same or a different window. Circuit comments and sub-comments for the lines are also copied and pasted.

The required procedure is given below.

◆ Procedure ◆

1. **Procedure**
   
   (1) Select the line(s) to be copied or moved.
   
   (2) Select [Edit]–[Copy] or [Edit]–[Cut] from the menu.

   **TIP**
   - Selecting [Edit]–[Cut] deletes the selected lines from the window, which then becomes the object to be moved.
   - To recover lines immediately after a ‘cut’ operation, select [Edit]–[Undo] from the menu bar.

   (3) Move the position cursor to the destination for the copy or move operation.

   **TIP**
   - To move or copy to another window, move the position cursor to the destination position in the window.

   (4) Select [Edit]–[Paste] from the menu.

   ➞ The lines are inserted at the position cursor.

   **TIP**
   - To cancel a paste operation immediately after execution, select [Edit]–[Undo] from the menu bar or press the [Ctrl]+[z] keys.

   **TIP**
   - You can move a line using a drag and drop operation. While holding down the [Alt] key at the cursor position, drag a line and drop it to another line.
   - You cannot drop a circuit to the inactive pane in a circuit edit screen split into panes for displaying the same block or macro.
Copying and Moving Cells

Cells in specified areas can be copied and moved to specified positions in cell units. Cells can be copied and pasted to the same or a different window.

The required procedure is given below.

◆ Procedure ◆

(1) Select the range of cells to be copied or moved.

(2) Select [Edit]–[Copy] or [Edit]–[Cut] from the menu.

**TIP**
- Selecting [Edit]–[Cut] deletes the selected cells from the window, which then becomes the object to be moved.
- To recover the cells immediately after deletion, select [Edit]–[Undo] from the menu bar.

(3) Move the position cursor to the destination for the copy or move operation.

**TIP**
To move or copy to another window, move the position cursor to the destination position in the window.

(4) Select [Edit]–[Paste] from the menu.

⇒ The cells are entered at the position cursor.

**TIP**
- Pasting cells overwrites the contents at the destination.
- To cancel a paste operation immediately after execution, select [Edit]–[Undo] from the menu bar or press the [Ctrl]+[z] keys.

**TIP**
- You can move a cell using a drag and drop operation. While holding down the [Alt] key at the cursor position, drag a cell and drop it to another cell.
- You cannot drop a circuit to the inactive pane in a circuit edit screen split into panes for displaying the same block or macro.
CAUTION

Copying or Moving cells is not allowed in the following situations. Performing an invalid operation causes an error.

- **When performing a Cut or Copy operation:**
  - When the selection range contains edit-prohibit line(s) (circuit comment lines, temporarily deleted lines, mnemonic lines, circuit lines containing instructions that disallow online editing.)
  - The selection range contains part of an instruction.

- **When performing a Paste operation:**
  - The paste area is in the middle of an instruction.
  - The range does not allow the use of input-type picture elements.
  - The range does not allow the use of output-type picture elements.
  - Circuit comment lines
  - The data, if pasted, exceeds the vertical columns (horizontal rows) display area.
  - Prohibit-edit line (hidden lines, temporarily deleted lines, mnemonic display lines, circuit lines instructions that disallow online editing.
  - When the number of circuit lines after the paste operation exceeds the maximum number of lines allowed for the block.
  - When a receiving side drawing element is pasted not in the first column in a continuation circuit.
  - When a vertical continuation line is pasted on the last column.

### Copying and Moving Device Data

The device data of a circuit copied or pasted in the program edit screen can be pasted in the following screens or dialog boxes:

- Tag name definition screen
- Registered device monitor screen
- Screens in other applications that support device data

**TIP**

You can paste device data using a drag and drop operation. While holding down the [Alt] key at the cursor position, drag a device data and drop it to another cell.
E1.2.31 Copying and Moving Circuits Including Tag Name Definitions

Tag name definitions that are referred to in a selected circuit can be copied and moved together with the circuit. Circuits with tag name definitions can be copied and moved within the same ladder program edit screen and also to the ladder program edit screen in another project.

The paste destination of the tag name definitions pasted with a circuit varies depending on the type of the tag name definition referred to by the block or macro in the paste destination and the type of the device to be pasted.

<table>
<thead>
<tr>
<th>Type of Tag Name Definition Referred to in the Paste Destination</th>
<th>Type of the Device to Be Pasted</th>
<th>Paste Destination of Tag Name Definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common tag name definition</td>
<td>Local device/macro device</td>
<td>Block/macro tag name definition</td>
</tr>
<tr>
<td></td>
<td>Structure</td>
<td>Common tag name definition</td>
</tr>
<tr>
<td>Block/macro tag name definition</td>
<td>Local device/macro device</td>
<td>Block/macro tag name definition</td>
</tr>
<tr>
<td></td>
<td>Structure</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Global device</td>
<td></td>
</tr>
</tbody>
</table>

Use the following procedure to copy or move a circuit including tag name definitions.

◆ Procedure ◆

(1) Select the range to be copied or moved.

TIP

Circuits and tag name definitions can be copied and moved in both line units and cell units.

(2) Select [Edit]–[Copy] or [Edit]–[Cut] from the menu bar.

TIP

- Selecting [Edit]–[Cut] deletes the selected cells from the window, which then becomes the object to be moved.
- To recover the cells immediately after deletion, select [Edit]–[Undo] from the menu bar.

(3) Move the position cursor to the destination for the copy or move operation.

TIP

To move or copy to another window, move the position cursor to the destination position in the window.

(4) Select [Edit]–[Paste Special] from the menu.

⇒ The Paste Special dialog box opens.
(5) Turn on the [Circuits and Tag Name Definitions] option button, and click [OK].

⇒ After the circuit and tag name definitions are pasted, a completion dialog box is displayed.

**TIP**

If the tag name definition screen for the paste destination is close, the tag name definition screen opens when the circuit and tag name definitions are pasted.

**CAUTION**

- When a circuit and tag name definitions are pasted to a block or macro that refers to common tag name definitions, the process takes a long time depending on the maximum number of common tag name definitions.
- You cannot paste tag name definitions into the [Address list] tag name screen.
- You can perform the [Undo] operation only on the program edit screen. You cannot undo the past operation of tag name definitions on the tag name definition screen, but you can delete the pasted tag name definitions.
- When pasting tag name definitions causes to exceed the maximum number of common tag name definitions in the paste destination, if the tag name definitions can be pasted by increasing the maximum number of common tag name definitions, the maximum number is automatically increased.
- You cannot paste a structure object definition that contains a local device as a member into a block that refers to common tag name definitions. The relevant structure object definition is ignored.
- If a circuit copied/cut in online edit contains tag name definitions of a local device, the global address assigned to the local device is pasted as a tag name definition. The tag name definitions of the local device are not pasted.
TIP

- When a circuit including tag name definitions is moved, only the circuit is cut away. The tag name definitions are not cut away.

- If the copied or cut tag name definitions contain a tag name or address already defined in the paste destination block or macro, a dialog box is displayed for selecting whether or not to overwrite the tag name definition in the paste destination. If you select [Overwrite], the tag name definition in the paste destination is overwritten with the copied/moved tag name definition. If you select [Do not overwrite], the tag name definition in the paste destination remains unchanged.

---

Figure E1.39  Dialog Boxes Displayed When a Duplicated Tag Name Definition Is Found

- If any of the tag name definitions are overwritten and deleted, the deleted tag name definitions are saved in the "SigDefNotPasted.csv" file in the project folder of the paste destination.

- When copying or moving a circuit and tag name definitions causes to exceed the maximum number of tag name definitions that can be registered in the paste destination, the tag name definitions not pasted are outputted into the "SigDefNotPasted.csv" file stored in the project folder of the paste destination.

- I/O comments that can be pasted to a paste destination are I/O comments that can be displayed based on the display setting in the paste destination and extended I/O comments.
E1.2.32 Copying Circuit Images

Using the same procedure as for copying lines, select [Edit]–[Image Copy] to copy a circuit image to the clipboard in bitmap format. The copied circuit image can be pasted to software such as Microsoft Paint and MS-Word.

TIP

You cannot copy circuit images in cell units.

E1.2.33 Canceling an Operation

You can undo up to 10 operations that you have just performed. Some possible scenarios are as follows:

1. To recover an instruction or comment you have just deleted using the [Delete] key.
2. To delete an entered instruction immediately after input, and revert to a prior state.
3. To cancel parameter replacement, and revert to a prior state.
4. To cancel a delete, cut or paste operation on lines or columns immediately after execution, and revert to a prior state.
5. To remove empty blanks immediately after insertion, and revert to a prior state.

The following restrictions apply to the canceling of operations:

- If the total number of lines for the most recent 10 operations exceeds 500 lines, only the latest operations can be canceled.
- If an editing operation is performed after an operation has been canceled, the cancel information is reset.
- Once Hide/Show Circuits or Temporary Delete/Undo Temporary Delete operations are performed, previous operations can no longer be canceled.

In addition, the following operations, once performed, cannot be undone.

- Assignment of tag names to addresses (registration remains in the tag name definition)
- Modification of I/O comments

To undo an operation, select [Edit]–[Undo] from the menu or press the [Ctrl]+[z] keys.
Recovering Deleted Instruction or Comments

Pressing the [Delete] key deletes an instruction or comment at the position cursor. If you change your mind, you can restore a deleted item.

The following procedure illustrates how to recover a deleted TIM instruction.

◆ Procedure ◆

(1) Move the position cursor to the TIM instruction.

(2) Press the [Delete] key.
⇒ The TIM instruction is deleted.

(3) Select [Edit]–[Undo] from the menu or press the [Ctrl]+[z] keys.
⇒ The TIM instruction is restored.
Reverting to State Prior to Input

If you have entered an incorrect instruction, you can delete the instruction and revert to the previous state. The following procedure illustrates how to restore a TIM instruction overwritten by an invalid instruction.

◆ Procedure ◆

1. Move the position cursor to the TIM instruction.

2. Press the [F7] (OUT instruction) key in overwrite mode.
   ⇒ The OUT instruction is saved and the parameter input field is displayed.

   SEE ALSO
   For details on overwrite mode, see Section E1.2.34, "Overwrite Mode and Insert Mode."

3. Select [Edit]–[Undo] from the menu or press the [Ctrl]+[z] keys.
   ⇒ The OUT instruction is deleted and the TIM instruction is restored.

TIP
If you entered a character in the parameter input field, select [Edit]–[Undo] twice.
E1.2.34 Overwrite Mode and Insert Mode

There are two modes for entering instructions: overwrite mode or insert mode. Pressing the [Insert] key toggles between the two modes.

Overwrite mode is usually used for editing whilst Insert mode is used for inserting instructions.

- Overwrite Mode
  If there is an instruction at the position cursor when a new instruction is entered in overwrite mode, the instruction at the cursor is removed and replaced by the new instruction.

- Insert Mode
  If there is an instruction at the position cursor when a new instruction is entered in insert mode, the instruction at the cursor is shifted to the right and the new instruction is entered in its original position.

The following example illustrates the difference between Overwrite mode and Insert mode when an instruction is entered at the position cursor.

- Original condition

- In Overwrite Mode

- In Insert Mode

Figure E1.40  Difference between Overwrite Mode and Insert Mode (when an instruction is entered)
The following example illustrates the difference between Overwrite mode and Insert mode when the [Delete] key is used to delete an instruction at the position cursor.

In overwrite mode, the instruction at the position cursor is deleted. In insert mode, the instruction at the position cursor is deleted and the instruction on the right is shifted left to take its place.

- Original condition

- In Overwrite Mode
  The instruction located at the position cursor is deleted.

- In Insert Mode
  The instruction located at the position cursor is deleted and shifted left to take its place.

**Figure E1.41  Differences between Overwrite Mode and Insert Mode (when an instruction is deleted)**

**TIP**
Insert mode is valid only for input-type instructions. If an output-type instruction is entered in insert mode, it overwrites the existing output-type instruction.

**TIP**
Input-type instructions are instructions that input device states such as load instructions (LD instruction) and comparison instructions (CMP instruction). They are usually coded on the left side of a ladder diagram.
E1.2.35 Entering and Deleting Circuit Comments

Circuit comments can be coded in the same way as circuits. They are always displayed in the block edit window. Circuit comments are entered in line units. Up to 25 consecutive lines can be entered for each comment.

![Figure E1.42 Examples of Different Comment Types](image)

**Entering Circuit Comments**

You can enter a circuit comment using any of the following ways:
- clicking the Circuit Comment button on the instruction palette;
- using function keys ([Ctrl]+[F7]);
- entering "!" from the keyboard.

After any of the above operations, the same circuit comment edit dialog box is displayed.

1. **Input Using Mouse**

   To enter a circuit comment using the mouse, use the following procedure.

   ◆ Procedure ◆

   1. **Click the Circuit Comment button** on the instruction palette.
      ⇒ The mouse cursor indicator changes into the circuit comment cursor.

   2. **Click the mouse at the location where the circuit comment is to be entered.**
      ⇒ The circuit comment edit dialog box is displayed at the clicked position.

   3. **Enter the circuit comment.** To enter another line consecutively, press the [Alt]+[Enter] keys to break the line.

   4. **After entering all circuit comments, click the [OK] button at the right side of the dialog box to determine the entry.**

   **TIP**
   You can also press the [Enter] key to determine the entry of circuit comments.
The entered circuit comment is applied to the circuit.

**CAUTION**

The Circuit Comment cursor remains unchanged after the circuit comment input. You may continue to enter more circuit comments. To cancel, do one of the following:
- Press the [ESC] key;
- Right click or
- Click the Selector button on the instruction palette.

(2) Input Using Function Keys

To enter a circuit comment using function keys, move the position cursor below the position where the circuit comment is to be entered and press the [Ctrl]+[F7] (circuit comment) keys. Enter the circuit comment in the circuit comment edit dialog box and press the [Enter] key.

(3) Input Using [!] Key

Move the position cursor to the position where the circuit comment is to be entered and enter '!'. Enter the circuit comment in the circuit comment edit dialog box and press the [Enter] key.

**Editing Circuit Comments**

To edit an existing circuit comment, display it in the circuit comment edit dialog box, amend it and press the [Enter] key.

You can display the circuit comment edit dialog box using any of the following ways:
- Move the position cursor to the position of the circuit comment and press the [Enter] key or
- Double-click the circuit comment.

**Deleting Circuit Comments**

To delete a circuit comment, move the position cursor to the circuit comment to be deleted and select [Edit]–[Delete] from the menu (or press the [Delete] key).

E1.2.36 Creating Hierarchized Circuit Comments (To Be Supported)

WideField3 R2 or earlier does not support functions for creating hierarchized circuit comments.
E1.2.37 Entering and Deleting Subcomments

Subcomments can be written to output-type instructions. They are displayed in the subcomment display area beyond the right power rail. Subcomments are entered on instruction basis.

**TIP**
The right power rail is the vertical line drawn at the right end of a ladder diagram. It represents one of the two electrodes.

### Entering Subcomments

You can enter a subcomment using any of the following ways:
- clicking the Subcomment button on the instruction palette;
- double-clicking the subcomment area of a circuit;
- using function keys ([Ctrl]+[F9]);
- moving the cursor to the subcomment area and pressing the [Enter] key;
- entering ')’ from the keyboard.

#### (1) Input Using Mouse

To enter a subcomment using the mouse, use the following procedure.

**Procedure**

1. Click the Subcomment button on the instruction palette.
2. Click the line where the subcomment is to be entered.
   ⇒ The Enter Subcomment dialog box is displayed.
3. Enter the subcomment and click [OK].
   ⇒ The subcomment is entered.

#### (2) Input Using Function Keys

To enter a subcomment using the function keys, move the position cursor to where the subcomment is to be entered and press the [Ctrl]+[F9] (subcomment) keys. Enter the subcomment in the displayed Enter Subcomment dialog box and press the [Enter] key.

#### (3) Input in Subcomment Area

Move the cursor to the subcomment area and press the [Enter] key. Enter the subcomment in the displayed Enter Subcomment dialog box and press the [Enter] key.
Summarized Content:

**Editing Subcomments**

To edit an existing subcomment, display the Enter Subcomment dialog box, amend the subcomment and click [OK] or press the [Enter] key. You can display the Enter Subcomment dialog box using any of the following ways:

- Move the position cursor to the line for the subcomment and press the [Ctrl]+[F9] keys;
- Click the Subcomment button on the instruction palette and click the desired subcomment;
- Move the position cursor over the subcomment to be edited, and press the [Enter] key.

**Deleting Subcomments**

Move the position cursor over the subcomment to be deleted, and press the [Delete] key.

Alternatively, display the Enter Subcomment dialog box, delete all the text for the subcomment and click [OK].
E1.2.38 Entering and Deleting I/O Comments

I/O comments are written to devices and are displayed in the I/O comment display area. You can enter an I/O comment using the mouse or function keys.

Clicking the I/O Comment button on the instruction palette switches between the I/O Comment and Parameter input modes.

**Figure E1.46 I/O Comment Input Mode**

**TIP**
- To show the I/O comment display area, select [View]–[Specify Display Mode]–[Display I/O Comment] from the menu.
- You can also move the mouse cursor over a device and read its I/O comment using TipHelp.
- If a space character is included in an I/O comment and there should be a new line in the middle of the next character string, the new line occurs at the space character and the character string is displayed. The space character is not deleted.

**CAUTION**
- I/O comments can be set only for structure members defined in the structure type definition. You can set I/O comments for an instantiated structure from the individual circuit edit screens.
- I/O comment is not displayed when a parameter is a structure name. The I/O comment specified for a member in the structure type definition is displayed only when a parameter is specified as a member name. However, I/O comment is not displayed when the parameter is a member of a structure pointer (Q).

**Table E1.11 Displaying and Editing Parameter and I/O Comments**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Display of I/O Comment</th>
<th>Editing of I/O Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structure name e.g. “DATA[V01]”</td>
<td>Not displayed</td>
<td>Not allowed</td>
</tr>
<tr>
<td>Structure member name e.g. “DATA[10].Adr”</td>
<td>Displays I/O comment defined in the type definition</td>
<td>Not allowed</td>
</tr>
<tr>
<td>Structure pointer (Q) e.g. “Q1”</td>
<td>Not displayed</td>
<td>Not allowed</td>
</tr>
<tr>
<td>Member name of structure pointer (Q) e.g. “Q1.Adr”</td>
<td>Not displayed</td>
<td>Not allowed</td>
</tr>
</tbody>
</table>
Entering I/O Comments

(1) Input Using Mouse

To enter an I/O comment using the mouse, use the following procedure.

◆ Procedure ◆

(1) Click the I/O Comment button on the instruction palette.
   ⇒ The I/O Comment Input mode is activated.

(2) Double-click the position where the I/O comment is to be entered. Or, click the position and press the [Enter] key.
   ⇒ The I/O comment input field is displayed.

(3) Enter the I/O comment and press the [Enter] key.
   ⇒ The I/O comment is entered.

TIP

The entered I/O comment is updated in the reference tag name definition.
Pressing the [Esc] key exits the I/O comment input field.

(2) Input Using Function Keys

To input an I/O comment using the function keys, move the position cursor to the element where the I/O comment is to be entered and press the [Ctrl]+[F8] (I/O Comment) keys. Enter the I/O comment into the input field and press the [Enter] key.

Editing I/O Comments

To edit an existing I/O comment, display the I/O comment input field using the same procedure for entering an I/O comment. Amend the I/O comment displayed in the I/O comment input field and press the [Enter] key.

Figure E1.47   Editing an I/O Comment
■ Deleting I/O Comments

To delete an I/O comment, display the I/O comment input field using the same procedure for entering an I/O comment. Delete all the characters of the I/O comment and press the [Enter] key.

■ Enter Tag Name with Address and I/O Comment

The Enter Tag Name with Address and I/O Comment mode allows you to enter an I/O comment when you enter an address. The I/O comment is saved with the reference tag name definition.

SEE ALSO
For details on the Enter Tag Name with Address and I/O Comment mode, see Section E1.3, "Entering Tag Names and Addresses."

■ Saving I/O Comments

When the common tag name definition is referenced, I/O comments are saved as and when they are registered or edited during ladder program editing.

When the block tag name definition is referenced, I/O comments are not saved until the edited circuit is saved. Even if the displayed tag name definition is saved, I/O comments are not saved unless the edited circuit is saved. To save I/O comments, you must always save the edited circuit.
E1.2.39 Creating and Deleting Balloon Comments

Balloon comments can be used to describe a comment in any position in the circuit edit and monitor screens.

The created balloon comments are listed in the balloon list window.

The balloon list window can be shown or hidden by selecting [View]-[Balloon Comment List] from the menu bar.

Figure E1.48 Balloon List Window
- A Balloon comment status icon
- B Project title
- C Block name
- D Balloon comment name
- E Macro name

⚠️ CAUTION
- If a circuit is hidden at the position where a balloon comment is displayed, the balloon comment is also hidden. If the circuit is shown, the balloon comment is displayed again.
- Changes on a balloon comment cannot be undone by selecting [Edit]–[Undo] from the menu bar.
- Unlike the case of blocks, a balloon is saved when it is created and discarded when deleted.
- If you move a balloon during editing of a block but finish the editing by canceling it, the balloon is moved back to the line where it was placed when the block was last saved. On the other hand, if you create a new balloon during editing and finish the editing by canceling it, the newly created balloon is saved at the position specified during the editing. If the cancellation causes the balloon position during editing to be beyond the last line of the block, the balloon is moved to the last line of the block next time the block is opened.
- Balloons are saved separately from circuits. This allows you to create new balloons in reference blocks and to delete and edit existing balloons in reference blocks.

TIP
When you open the [online] tab of the balloon list, the project title shows the reference of the online balloon comment and monitor on the program monitor. "(CPU)" is displayed when the CPU is referred to, and "(When Downloaded)" is displayed when the balloon data at the time of the program download is referred to. If balloons are stored in the CPU, the CPU is always referred to.

SEE ALSO
You can use a balloon comment as a monitor. For details, see J2.6, "Balloon Monitor" (Online).
## Creating Balloon Comments

To create a balloon comment, use the following procedure.

### Procedure

1. Move the position cursor to where the balloon comment is to be placed in the program edit screen.
2. Select [Edit]–[Create Balloon Comment/Monitor] from the menu.
   - The Edit Balloon Comment dialog box is displayed.
3. Enter a balloon comment name and comment text, and click [OK].
   - The balloon comment is created.

### TIP
- The balloon comment is created in the first column of the line where the position cursor is positioned. If the position cursor is on a hidden line, the balloon comment is created in the hidden line and not displayed. To display the balloon comment, show the line.
- Up to 64 characters can be entered for a balloon comment name.
- Balloon comment names can contain special characters '-' (hyphen) and '_' (underscore).
- You can enter up 32 lines in a comment. Each line can be up to 64 characters.
- When a '[' character appears in a line in the comment, the subsequent ']' and '%' characters in the same line are regarded as the end position of a monitor format and the start position of a data type description, respectively.
- To use '[' as a literal character instead of a monitor format character, add a '\' character as a prefix before '['.

### SEE ALSO

For details on the format of the balloon monitor, see J2.6, "Balloon Monitor" (Online).
### Editing Balloon Comments

You can edit existing balloon comments from the Balloon Comment List window as follows.

Right-click the balloon to be edited in the Balloon Comment List window to open a popup menu. From the popup menu, select [Edit] to edit the balloon comment or select [Rename] to edit the balloon comment name. The Edit Balloon Comment dialog box is displayed.

Modify the balloon comment or its name in the Edit Balloon Comment dialog box, and click [OK].

### Balloon Comment Display Setup

You can perform balloon comment display setup to set the following items for a balloon comment:

- Background filling
- Line of the balloon comment
- Font

Balloon comment display setup can be performed in the Balloon Comment Display Setup dialog box.

![Balloon Comment Display Setup Dialog Box](E0102_130.VSD)

Figure E1.49  Balloon Comment Display Setup Dialog Box

- **A** Select the background color of the balloon comment.
- **B** Specify the transparency of the balloon comment. If you specify a higher level of transparency, you can display both the balloon comment and the part of the circuit covered by the balloon comment.
- **C** Select the border color of the balloon comment.
- **D** The border style (fixed to solid line) of the balloon comment. This control cannot be operated.
- **E** Select the line width of the balloon comment.
- **F** Specify the font of the balloon comment.
- **G** Saves the display setup as default. The setup saved as default applies to subsequent creation of new balloons.
◆ Procedure ◆

(1) Place the mouse cursor on the balloon comment name in the balloon list window or on the balloon comment in the ladder program edit window.

(2) Right-click the mouse to open a popup menu, and click [Display Settings] on the popup menu.  
⇒ The Balloon Comment Display Setup dialog box is displayed.

(3) Specify the background color and transparency.
(4) Specify the line.
(5) Specify the display font.
(6) Click [OK].

⇒ The settings are reflected in the balloon comment.
Showing and Hiding Balloon Comments

You can show and hide individual or all balloon comments. The relevant procedure is given below.

**TIP**
The display status of each balloon comment is indicated by an icon in the balloon comment tree. There are three icons as follows:

- ![icon] The balloon is in a "shown" state.
- ![icon] The balloon is in a "hidden" state.
- ![icon] The display position is incorrect and the balloon cannot be displayed.

### Showing and Hiding Individual Balloon Comments

To show and hide an individual balloon comment, use the following procedure.

**Procedure**

1. Select a balloon comment in the balloon list window.
2. Right-click the mouse to open a popup menu, and click [Hide/Display] on the popup menu.

   ⇒ The display status of the balloon comment is toggled between shown and hidden.

**TIP**
- If you double-click a balloon in an incorrect state, the balloon is redisplayed in the first line of the block or macro if the balloon can be redisplayed.
- If you drag a balloon in an incorrect state from the balloon list window and drop it into the block edit screen, the balloon is redisplayed at the position where the balloon is dropped.
- To delete all incorrect balloons from a block or macro, right-click the relevant block or macro in the balloon list window, and then click [Batch Delete of Invalid Status].
- To delete all incorrect balloons over the entire project, right-click [Project Balloon Comment List] in the balloon list window, and then click [Batch Delete of Invalid Status].

### Showing and Hiding All Balloon Comments

You can also show and hide all balloon comments for a block or macro or for an entire project.

To show and hide balloon comments for a block or macro, use the following procedure.

**Procedure**

1. Select a block or macro name in the balloon list window.
Right-click the mouse to open a popup menu, and click [Show All Balloon Comments] or [Hide All Balloon Comments] on the popup menu.

⇒ The display status of all the balloon comments for the block or macro is toggled between shown and hidden.

To show and hide balloon comments for an entire project, use the following procedure.

◆ Procedure ◆

1. Select a project name in the balloon list window.
2. Right-click the mouse to open a popup menu, and click [Show All Balloon Comments] or [Hide All Balloon Comments] on the popup menu.

⇒ The display status of all the balloon comments for the project is toggled between shown and hidden.
Resizing Balloon Comments

You can resize individual balloon comments. The relevant procedure is given below.

◆ Procedure ◆

(1) Move the mouse pointer to the border of the balloon comment.
⇒ The mouse pointer icon is changed to a double-headed arrow.
(2) Click and hold the left mouse button.
(3) While holding the left mouse button down, move the mouse pointer.
⇒ The size of the balloon comment is changed as you move the mouse pointer.
(4) Release the left mouse button.

Moving Balloon Comments

To move a balloon comment, use the following procedure.

◆ Procedure ◆

(1) Click any place in the balloon comment and hold the left mouse button down.
(2) While holding the left mouse button down in the balloon comment, move the mouse pointer.
(3) Release the left mouse button.

Deleting Balloon Comments

Right-click a balloon comment to open a popup menu, and then select [Delete] from the popup menu.

⚠️ CAUTION

- If you delete circuits in line units by operations such as deleting a circuit line and cutting selected line units in a circuit, any balloons in the deleted lines will also be deleted. In addition, only the circuits will be pasted to the paste destination.
- If you delete lines containing any balloons by, for example, selecting and cutting circuits in cell units, the contained balloons will automatically move to the preceding valid line.
- Changes on a balloon cannot be undone by selecting [Edit]–[Undo] from the menu bar.
Jumping to a Balloon Comment

To jump from the balloon list window to a balloon comment in the program edit screen, right-click the relevant balloon comment in the balloon list window to open a popup menu, and then select [To Display Position] from the popup menu. The program edit screen displays the place where the balloon comment is present and the position cursor is moved to the position of the balloon comment.

You can also jump to a balloon comment in the program edit screen by selecting the relevant balloon comment in the balloon list window and then pressing the [Enter] key.

TIP

- Jumping to a balloon in a hidden circuit opens the hidden circuit.
- In the program monitor screen, jumping to a balloon in the ladder program converted from scripts opens and displays the ladder program even when it has been hidden.

Copying Balloon Comments

Right-click a balloon comment to open a popup menu, and then select [Copy] from the popup menu.

Next, move the mouse to the position where you want to paste the balloon comment in the program edit screen, and then select [Paste] from the popup menu.

When you paste the balloon comment, the Edit Balloon Comment dialog box with the copied balloon comment text is displayed. Enter a name of the balloon comment to be pasted and click [OK] to paste the balloon comment.

TIP

- When a balloon comment is pasted, the comment text and display setup of the source balloon comment are copied.
- You can also paste a balloon comment by selecting [Paste] from the block or macro name popup menu of the [Balloon Comment List] window. If the Edit Block or Edit Macro window is open, the comment is pasted on the upper left corner of the window. When the Edit Block or Edit Macro window is not open, the comment is pasted at the start (leftmost position) of the window. The Edit Block or Edit Macro window does not open when you select [Paste] with a block or macro window closed.
- When copying to a different block where no balloon exists with the same name, the balloon is pasted with the source balloon name.
E1.2.40 Index View

The index view uses circuit comments as indexes, and hides circuits coded up to the next circuit comment. The index view helps you to have an overall flow of a ladder program before editing the details.

Figure E1.50 Conceptual View of Circuit Hiding

You can choose to hide or show individual circuits, as well as hide or show all circuits within a block.
Hide Circuits

You can use the following method to hide circuits:
- Hiding circuits in the program edit screen

(1) Hiding Circuits in the Program Edit Screen
To hide a circuit in the program edit screen, use the following procedure.

◆ Procedure ◆

(1) Move the position cursor to the circuit comment line.
(2) Select [Edit]–[Hide/Show Circuits]–[Hide Circuits] from the menu bar.
⇒ All circuits before the next circuit comment are hidden. The circuit comment of the hidden circuits is displayed with a different background color.

CAUTION
- When you hide the last circuit comment of a program, all circuits until the end of the program are hidden.
- Circuit comments of hidden circuits are displayed with a different background color to indicate that modification is not allowed.
- The index view mode remains even after the file is closed and reopened.
- Hidden circuits cannot be copied or moved.
Show Circuits
To redisplay a hidden circuit, move the position cursor to the comment of the hidden circuit and select [Edit]–[Hide/Show Circuits]–[Show Circuits] from the menu.

Hide All Circuits
To hide all circuits within a block, select [Edit]–[Hide/Show Circuits]–[Hide All Circuits] from the menu. This function, however, does not work if the circuit comment is not on the first line of the block.

Show All Circuits
To show all circuits within a block, select [Edit]–[Hide/Show Circuits]–[Show All Circuits] from the menu. When all circuits are shown, the line containing the position cursor is displayed at the top.

Customize Color of Hidden Circuits
You can customize the display color of circuit comments for hidden circuits on the Circuit Display/Input tab of the Set up Environment dialog box.
E1.2.41 Inserting and Deleting Page Breaks

You can enter a page break in the circuit edit screen.

- **Inserting Page Breaks**

  Move the position cursor to a line where the page break is to be inserted and select [Edit]–[Page Break] from the menu bar. An "@" symbol is displayed at the beginning of the line number. You can insert page breaks not only in circuits but also in circuit comments.

  ![Figure E1.51 Page Break](E0102_145.VSD)

  **TIP**
  - In WideField2, an "@" character is inserted as a page break at the beginning of a circuit comment. When WideField3 opens a block that contains a WideField2 page break, a page break is added at the beginning of the line number, and the "@" character at the beginning of the circuit comment does not work as a page break and is handled as a part of the circuit comment string.
  - You cannot insert a page break in a line currently being edited.

- **Deleting Page Breaks**

  To delete a page break from a circuit comment, edit the circuit comment to delete the "@" character. To delete a page break inserted in a line number, select [Edit]–[Delete Page Break] from the menu bar.

  **TIP**
  - When you print a ladder program that contains inserted page breaks, you can specify pages to be printed. For details on printing, see Chapter E8, "Printing."
  - You can insert an empty page by inserting a page break in an empty circuit comment. The empty page is printed as an empty page when a ladder program is printed.
  - You cannot delete a page break from a line currently being edited.

- **CAUTION**

  If you delete a line that contains a page break, the page break is also deleted.

- **Inserting Empty Pages**

  To insert an empty page, select [Edit]–[Insert Blank Page] from the menu bar.

  **TIP**
  You cannot insert an empty page in a line currently being edited.
E1.2.42 Changing Page Display

The circuit edit screen and program monitor screen can be split.

- Splitting a Window

  You can split the circuit edit screen horizontally or vertically.

  **TIP**

  You can split a screen only either horizontally or vertically.

  **Horizontal Splitting**

  To split the circuit edit screen horizontally, select [View]-[Page Display]-[Split Display Horizontally] from the menu bar. The [Split Display Horizontally] menu button is displayed with a checkmark.

  ![Figure E1.55 Horizontal Splitting](E0102_149.VSD)

  **Vertical Splitting**

  To split the circuit edit screen vertically, select [View]-[Page Display]-[Split Display Vertically] from the menu bar. The [Split Display Vertically] menu button is displayed with a checkmark.

  ![Figure E1.56 Vertical Splitting](E0102_150.VSD)
Cancelling a Split

To cancel a split state, select [View]-[Page Display]-[Split Display Vertically] or [Split Display Horizontally] from the menu bar to remove the checkmark displayed with either of these menu items.
E1.2.43 Mnemonic Editing Function

This function is used to edit programs in the mnemonic format in the program edit screen.

Programs that allow mnemonic editing can be used only when the program inside the block/macro is converted normally with the [Convert] operation.

In addition, mnemonic editing can be performed on a 1-circuit unit of programs at a time.

To perform mnemonic editing, use the following procedure.

◆ Procedure ◆

(1) Make sure the Edit Block or Edit Macro window is open.

(2) Move the cursor to the circuit where you will perform mnemonic editing, and press [Ctrl]+[Enter].

⇒ The Edit Mnemonics/Script pane opens.

(3) Code the mnemonic program.

TIP

If the program contains a circuit that has not been converted or a circuit that experienced a conversion error, the following message is displayed and mnemonic editing is not possible.

```
1 The circuit has been modified. Select [Convert].
```

Perform the [Convert] operation.

You can also create a new circuit in mnemonic editing. Move the position cursor to a line where no circuit is created. The line enters an editable state with a red diagonal hatch. Open the Edit Mnemonics/Script pane and code a mnemonic program.
(4) Click the [Save Changes] button to save the edited mnemonic program. ⇒ The converted program is applied to the circuit.

**TIP**

You can also use mnemonic editing to edit circuit comments and subcomments. The relevant procedures are given below.

- **Circuit Comment**
  
  An exclamation mark (!) is displayed as the first character in the mnemonic line.

- **Subcomment**
  
  An apostrophe (') is displayed at the end of the mnemonic line, and followed immediately by the character string for the subcomment.

**TIP**

If entering the NCALL instruction in mnemonic editing, you need to enter the instruction in a different way as shown below depending on the position where the instruction is entered.

- To enter the instruction at the beginning of a circuit: NCALLLD
- To enter the instruction in the middle of a circuit: NCALLAND
E1.2.44 Function for Commenting out Circuits

This function is used to temporarily activate, inactivate, or disable circuits or instructions. To set or remove a comment out, select an instruction or circuit to be commented out, select [Edit]-[Circuit Comment-out] to open a popup menu, and select [Forced On], [Forced Off], [Not Executed], [Remove] from the popup menu.

**CAUTION**

The function for commenting out circuits can be used only for circuits already converted.

**TIP**

When you copy a commented-out part from a circuit edit screen during offline, if the copy destination is offline, the comment out setting is also copied. If the copy destination is edited online, the comment out setting is not copied and only the circuit is copied.

**SEE ALSO**

- For details on the function for commenting out circuits, see Section K2.3, "Commenting out Circuits" (Online).
- For details on conversions, see Section E1.6, "Converting Circuits and Displaying Errors".

## Restrictions in the Function for Commenting out Circuits

- In circuit editing during offline, the following instructions can be commented out on a line basis:
  
  IL, ILC, SUB, RET, INTP, IRET, FOR, NEXT, MRET

  Even if these instructions are commented out, they can be downloaded and the comment-out settings remain unchanged. In that case, the commented-out instructions are not executed.

- The instructions in a commented-out line cannot be further commented out.
- In the Edit Mnemonics/Script screen, if you edit commented-out circuits, the comment out settings are removed.
- When you copy a commented-out part from a circuit edit screen during offline, if the copy destination is offline, the comment out setting is also copied. If the copy destination is edited online, the comment out setting is not copied and only the circuit is copied.
- You can use the comment out function in a block in read mode, but you cannot save or convert the block.
- You can comment out temporarily deleted lines. Note however that the commented-out temporarily deleted lines are deleted when converted.
E1.3 Entering Tag Names and Addresses

There are 2 modes for entering a device when editing a ladder program, namely, Enter Tag Name Without Address and Enter Tag Name With Address. You do not have to allocate an address to a tag name while editing a ladder program, but can do so after editing by modifying the tag name definition. The Enter Tag Name With Address mode, however, provides a convenient means to allocate the address and I/O comment to the tag name during program editing.

Use the [Set up Parameter Input] parameters on the Circuit Display/Input tab in the Set up Environment dialog box to select the desired mode of input.

SEE ALSO
For details on the Circuit Display/Input tab in the Set up Environment dialog box, see Section D1.2.4, “Circuit Display/Input Setup”.

E1.3.1 Enter Tag Name Without Address

In Enter Tag Name Without Address mode, a device is entered as a tag name during ladder program editing and an address is allocated to the tag name later. Select [Enter Tag Name Without Address and I/O Comment] in the [Set up Parameter Input] parameters on the Circuit Display/Input tab in the Set up environment dialog box.

You can also enter an address only without entering a tag name.

E1.3.2 Enter Tag Name With Address

The Enter Tag Name With Address mode provides a means to enter an address and I/O comment at the same time a device is registered with a tag name. Select [Enter Tag Name With Address and I/O Comment] in the [Set up Parameter Input] parameters on the Circuit Display/Input tab in the Set up environment dialog box.

You can also choose not to enter any address or I/O comment at this point, but to enter it later.

In Enter Tag Name With Address mode, the Address Assignment dialog box is displayed when a device is entered. Enter the address and/or the I/O comment and click [OK]. The reference tag name definition is saved.

Figure E1.57 Address Assignment Dialog Box in Enter Tag Name With Address Mode

TIP
When using structure tag names, the Address Assignment Dialog Box is not displayed because individual address and I/O comment assignment is not allowed.
E1.4 Circuit and Tag Name Definition Reference

There are 3 types of tag name definitions:
- Common tag name definition
- Block tag name definition
- Macro tag name definition

Common tag name definitions are common to a program. Using common tag name definitions, the same device can be associated with a common tag name over all blocks that refer to the common tag name definitions in a project.

Block tag name definitions are only valid within a block. Using block tag name definitions, the same device can be associated with different tag names in different blocks and used for different purposes in each block. It also avoids duplicate tag name errors when reusing program codes for a block.

TIP

You can set the reference tag name definition to be used in a circuit.

Local devices are registered in the block tag name definition even if the reference is to common tag name definition.

Macro tag name definitions define tag names to be used in an instruction macro.

Changing Reference from Common Tag Name Definition to Block Tag Name Definition

When you change the reference tag name definition, you can choose to read definitions for devices used in the block from the original reference tag name definition. When the reference is changed from common tag name definition to block tag name definition, a confirmation dialog box is displayed as shown in the following figure. Clicking [Yes] starts the merging process. Clicking [No] indicates not to perform merging. Clicking [Cancel] reverts to the original state.

Figure E1.58 Confirmation Dialog Box

Selecting Tag Name Definition Precedence

During merging, a Select Tag Name Definition Precedence dialog box is displayed. Specify the desired precedence and merge method, and start execution.
E1.5 **Switching Views**

On the ladder program edit window, you can switch between showing and hiding various information and use the Zoom function to display information in different sizes.

**SEE ALSO**

For details on configuring various display settings at once, see E1.5.4, "Setting Multiple Display Modes."

---

**E1.5.1 Switching Views for I/O Comments, Tag Names and Addresses, and Instruction Numbers**

You can switch views for the items listed in the table below.

<table>
<thead>
<tr>
<th>Table E1.13</th>
<th>View Function</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Item</strong></td>
<td><strong>Function</strong></td>
</tr>
<tr>
<td>Display I/O Comment</td>
<td>Displays I/O comments below ladder program components. The number of I/O comments displayed is set in the environment setup. I/O comments exceeding this limit are not displayed.</td>
</tr>
<tr>
<td>Display (Expanded) Instruction Parameter</td>
<td>Allows setting of the function for switching between 1-line display/2-line display for instruction parameters, the function for emphasized display of write parameters, escape sequence display, etc.</td>
</tr>
<tr>
<td>Display Instruction Number</td>
<td>Displays the instruction number of the first ladder program component below the line number. The instruction number is the five-digit number displayed below the line number; the instruction number of the first element of the circuit is prefixed with an ‘N’ character. If a label is defined for a line, the instruction number is not displayed.</td>
</tr>
<tr>
<td>Display Address</td>
<td>Toggles between display of tag names and addresses. In address display mode, the replace function is not available. Structure tags are always displayed as tag names regardless of the mode setting.</td>
</tr>
<tr>
<td>Display Instruction Range</td>
<td>Displays the instruction range of IL-ILC and other instructions used in pairs in the line number area.</td>
</tr>
</tbody>
</table>

To switch views, select [View]-[Specify Display Mode] and the respective item ([Display I/O Comment], [Display Instruction Number], [Display Address], [Display (Expanded) Instruction Parameter]), or [Display Instruction Range] from the menu. To cancel, select a menu item that is marked with a tick.

---

**Figure E1.60 View Menu (Example where Display I/O Comment and Display Instruction Number are selected)**

---
CAUTION

The Replace function cannot be used in address display mode.
Double-clicking to edit a device in address display mode displays the tag name as an initial value in the input field.

Figure E1.62  Initial Display of Tag Name

An instruction number is the number assigned to each mnemonic instruction when the ladder program is converted into mnemonics. Instruction numbers are not assigned to circuit comments, subcomments, and I/O comments. Before a circuit is converted, the displayed instruction numbers may be incorrect. Correct values will be displayed after conversion.

If an instruction parameter is a structure name, the I/O comment is not displayed. For structure member names and addresses assigned to a member, the comment defined for the member in the structure type definition is displayed.

Structure tag names are always displayed as tag names (the addresses are not displayed).

Comments for constant names are not displayed in the I/O comment area. To check the value or comment of a constant name, use TipHelp.

SEE ALSO

For details about the expanded display for the instruction parameter, see Section E1.5.2, “Expanded Display for Instruction Parameter”.
E1.5.2 Expanded Display for Instruction Parameter

These settings are for the display of the instruction parameter in the Edit Block and Edit Macro screens.

<table>
<thead>
<tr>
<th>Item</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compressed Display for Instruction Parameter</td>
<td>Allows an increase in the amount of program information displayed in a single screen by setting the instruction parameter display to a single line and compressing the program display vertically.</td>
</tr>
<tr>
<td>Emphasized Display for Write Parameter</td>
<td>Allows emphasized display for write parameter.</td>
</tr>
<tr>
<td>Escape Sequence Display</td>
<td>Allows setting the character string parameter enclosed in &quot;&quot; to either be displayed in escape sequence code or displayed as a character string.</td>
</tr>
</tbody>
</table>

### Compressed Display for Instruction Parameter

Allows an increase in the amount of program information displayed in a single screen by setting the instruction parameter display to a single line and compressing the program display vertically.

When instruction parameters are displayed in a single line, instruction parameters with long character strings are abbreviated. To check instruction parameters that have been abbreviated, you can switch between the mode that displays instruction parameters in two lines and the mode that displays instruction parameters in a single line.

![Example of Compressed Display for Instruction Parameter](E0105_04_VSD)

---

*Table E1.14 Expanded Display for Instruction Parameter Settings*

---

*Figure E1.63 Example of Compressed Display for Instruction Parameter*
### Emphasized Display for Write Parameter

Instruction parameters with the Write attribute are displayed with emphasis (in bold font).

Parameters where results are written by the instruction process are emphasized.

![Example of Emphasized Display for Write Parameter](E0105_05.VSD)

#### Escape Sequence Display

Sets the display of the character string parameter enclosed in " " to either be displayed in escape sequence code or displayed as a character string. Codes that cannot be displayed as character strings are displayed as escape sequences with no changes.

![Example of Escape Sequence Display](E0105_06.VSD)
E1.5.3 Display Instruction Range

These settings display the instruction ranges of IL-ILC and other instructions used in pairs or the ladder program converted from scripts in the line number display area of the Edit Block, Edit Macro, program monitor, or macro monitor windows.

<table>
<thead>
<tr>
<th>Item</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>IL-ILC Instruction Range</td>
<td>Displays the range from the IL instruction line to the ILC instruction line.</td>
</tr>
<tr>
<td>SUB-RET Instruction Range</td>
<td>Displays the range from the SUB instruction line to the RET instruction line.</td>
</tr>
<tr>
<td>INTP-IRET Instruction Range</td>
<td>Displays the range from the INTP instruction line to the IRET instruction line.</td>
</tr>
<tr>
<td>FOR-NEXT Instruction Range</td>
<td>Displays the range from the FOR instruction line to the NEXT instruction line.</td>
</tr>
<tr>
<td>SCRIPT Instruction Range</td>
<td>Displays the range of the ladder program where scripts are converted.</td>
</tr>
</tbody>
</table>

Display Instruction Range

Displays, from the right end of the line number display area to the left, the IL-ILC, SUB-RET, INTP-IRET, FOR-NEXT, and SCRIPT instruction ranges. For IL-ILC and FOR-NEXT, up to four stages are displayed with lines of different thickness.

Procedure

1. Make sure the Edit Block or Edit Macro window is open.
2. From [View]-[Specify Display Mode]-[Display Instruction Range] menu, click the instruction range to be displayed.

⇒ The range of the selected instructions is displayed.
TIP
- When editing a program, you have to convert it to correctly display the instruction range.
- You can change colors for instruction ranges by using the [Set up Circuit Components] settings on the Circuit Display/Input tab in Set up Environment.

SEE ALSO
For details on environment setup, see Section D1.2.4, "Circuit Display/Input Setup."

E1.5.4 Setting Multiple Display Modes
You can configure various display settings in the Display Mode Collective Setting dialog box. The relevant procedure is given below.

◆ Procedure ◆

(1) Select [View]-[Specify Display Mode]-[Display Mode Collective Setting] from the menu bar.
⇒ The Display Mode Collective Setting dialog box is displayed.

(2) Turn on the checkboxes of the items to be displayed.
(3) Click [OK] or [Apply].
⇒ The view of the ladder program edit screen is updated.

TIP
If you click [OK], the Display Mode Collective Setting dialog box is closed and the ladder program edit screen is updated.
If you click [Apply], the Display Mode Collective Setting dialog box remains open and the ladder program edit screen is updated.
SEE ALSO

For details on [Display Data Change Identification], [Display Details], and [Display Format] on the Display Mode Collective Setting dialog box, see J2.3.9, “Setting Multiple Display Modes” (Online).

**E1.5.5 Zooming Up and Down**

To enlarge the display, select [View]–[Zoom]–[Zoom Up] from the menu.

To reduce the display, select [View]–[Zoom]–[Zoom Down] from the menu.

The Zoom function provides 15 degrees of resizing: seven for enlargement, one for standard size, and seven for reduction.

The last zoom setting for a block is saved and used when the block is next displayed.

---

**Figure E1.67  Zoom Up and Zoom Down**
E1.6 Converting Circuits and Displaying Errors

Selecting [Edit]–[Convert] from the menu converts entered circuits into instruction words.

The following checks are performed at conversion.

- Is the circuit valid?
- Are instruction parameters entered correctly?
- Are labels entered correctly?

When the circuits are converted, temporarily deleted lines are deleted from the window and deleted lines cannot be recovered. At the same time, the circuits are rearranged and displayed.

If conversion ends successfully, a dialog box indicating normal completion appears.

If a conversion error is encountered, the error dialog box appears, and the position cursor moves to the error location.

CAUTION

The first error that is found during conversion is displayed in the error dialog box. Correct the error and resume conversion.

If a hidden circuit contains a conversion error, the circuit is expanded and the display jumps to the error location.

E1.6.1 Emphasized Display of Modified Line

During editing, programs that have not been converted can be identified.

Programs are remain unconverted for the following reasons:

- The [Convert] operation is not performed while editing a program.
- A conversion error occurred during the [Convert] operation.

This function allows you to verify the edited parts of the program that failed the conversion process by emphasizing these in red, as shown in the figure below.

Figure E1.68 Emphasized Display of Program being Modified

TIP

You can set the emphasis for modified lines to Show/Hide using the Circuit Display/Input in the environment setup.
E1.7 Importing/Exporting Circuit Comments and Subcomments

Circuit comment and subcomment data in a program can be imported from/exported to CSV data. Converting comment data to CSV data allows you to reuse comment data in other programs. This improves the efficiency of editing programs.

E1.7.1 Exporting Circuit Comments and Subcomments

Circuit comment and subcomment data in a program can be saved as CSV data. The relevant procedure is given below.

◆ Procedure ◆

(1) Make sure the Edit Block or Edit Macro window is open.

(2) Select [Edit]-[Output CSV Data for Circuit Comment/Subcomment] from the menu bar.

⇒ The Save As dialog box is displayed.

(3) Enter the file name to be saved, and click the [Save] button.

⇒ The circuit comment/subcomment data is saved in CSV format.
E1.7.2 Importing Circuits Comments and Subcomments

Circuit comment and subcomment data saved as CSV data can be loaded in the program edit screen. The relevant procedure is given below.

◆ Procedure ◆

(1) Make sure the Edit Block or Edit Macro window is open.

(2) Select [Edit]-[Load CSV Data for Circuit Comment/Subcomment] from the menu bar.
⇒ The Select File dialog box is displayed.

(3) Select the file to be imported, and click the [Select] button.
⇒ The Confirm Reading of Circuit Comment/Subcomment dialog box opens. This dialog box shows a list of comments set in the program and data for the comments that have been loaded.

(4) Adjust the positions of defined comments, and click the [Substitute] button.
⇒ The CSV data is imported into the program.

TIP
You can perform the following adjustments on the comments for loaded data: insert line, delete line, and clear set information.
E2. Tag Name Definition

The Tag Name Definition function assigns addresses and I/O comments to tag names. You can choose to enter only a tag name in the program of a block and then enter and allocate an address to the tag name later. Group tag names can also be used to group a set of functions.

There are 2 types of tag name definition: common tag name definitions which are shared among programs; and block tag name definitions which are valid only within a block. Macro tag name definitions which define tag names to be used in instruction macros is described together with block tag name definitions in this chapter. Therefore, for details on macro tag name definition, see the descriptions for block tag name definitions.

Tag name definition can only be done when a project is open.

- Function Limitations for Each CPU Type

<table>
<thead>
<tr>
<th>CPU Modules</th>
<th>Limitations</th>
<th>SEE ALSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>F3SP22-0S, F3SP28-3S, F3SP38-6S, F3SP53-4S, F3SP58-6S, F3SP59-7S</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>F3SP66-4S, F3SP67-6S</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>F3SP71-4N, F3SP76-7N</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>F3SP71-4S, F3SP76-7S</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>

TIP
This section does not describe hardware-dependant limitations for each CPU type, such as the number of available devices.

SEE ALSO
For details on limitations for each CPU type, refer to the user's manual for each type.
E2.1 Fundamentals of Tag Name Definition

This section describes the layout of the Tag Name Definition window, common tag name definition, block tag name definition and restrictions on tag name definitions.

E2.1.1 Tag Name Definition Window Layout

The layout of the tag name definition window is the same for block tag name definitions and common tag name definition.

Data can be copied and pasted between tag name definition windows and tables with assigned tag names created with MS-Excel or a similar program.

Furthermore, checks for duplicate tag names within a project are constantly performed, with the results displayed in various colors for distinction.

Structure object definition can also be performed on a tag name definition window. Structure tag names which have an object already defined are displayed in grey. Only structure object definition is carried out on a tag name definition Window; setup of various I/O comments cannot be done.

![Tag Name Definition Window](E0201_01.VSD)

Figure E2.1 Tag Name Definition Window (An Example of a Common Tag Name Definition Window)

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>SEE ALSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Definition area usage</td>
<td>SEE ALSO</td>
</tr>
<tr>
<td>B</td>
<td>Edit area</td>
<td>E2.2</td>
</tr>
<tr>
<td>C</td>
<td>Button for displaying the Project Settings/Configuration window</td>
<td>D3.1</td>
</tr>
<tr>
<td>D</td>
<td>Header</td>
<td>E2.8.1</td>
</tr>
<tr>
<td>E</td>
<td>Information area</td>
<td>E2.2.6</td>
</tr>
</tbody>
</table>

TIP

More than one I/O comment can be displayed.

SEE ALSO

- For details on displaying multiple I/O comments, see E2.8, "Multiple I/O Comments."
- For details on changing the maximum number of common tag name definitions, see E2.6, "Changing Maximum Number of Common Tag Name Definitions and Storing to CPU."
Common Tag Name Definition and Block Tag Name Definition

Common tag name definitions are usable throughout a project; block tag name definitions are valid only within a block. Within a project, multiple tag name definitions are allowed but a single reference tag name definition must be set for each block. Each block may refer to either the common tag name definition or a block tag name definition. Local devices always refer to a block tag name definition.

The following text describes how to set the reference tag name definition of a block to either the common tag name definition or a block tag name definition, and then discusses how data is handled for each of the situations.

Setting Reference for Tag Name Definitions

Reference for tag name definitions is set on block basis. The relevant procedure is given below.

Procedure

1. Check that the Edit Block Window is displayed.
2. Select [Edit]–[Local Device/Properties] from the menu.
   ⇒ The Local Device/Properties dialog box is displayed.
3. Turn on either the [Common Tag Name Definition] option button or the [Block Tag Name Definition] option button in the [Reference Tag Name Definition] group box and click [OK].
   ⇒ The Local Device/Properties dialog box closes and the reference tag name definition for the block is updated accordingly.
Common Tag Name Definition Reference

When the reference for tag name definition for a block is set to the common tag name definition, the block refers to the common tag name definition. Nevertheless, if a tag name coded in the program is allocated a local device address in the block tag name definition, or if a local device is coded within the block, the block tag name definition is referred.

Figure E2.2  Common Tag Name Definition Reference

The following table shows which tag name definition is referred to when a tag name is defined in both the common tag name definition and a block tag name definition.

Table E2.3  Tag Name Definition to Be Referred to

<table>
<thead>
<tr>
<th>Device Allocated to Tag Name in Block Tag Name Definition</th>
<th>Global Device</th>
<th>Local Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tag Name Definition to Be Referred to</td>
<td>Common tag name definition</td>
<td>Block tag name definition</td>
</tr>
</tbody>
</table>

Block Tag Name Definition

If the reference for tag name definition is set to block tag name definition, the block refers only to the block tag name definition. Address allocations in common tag name definitions are completely ignored.

Figure E2.3  Block Tag Name Definition Reference

TIP

When the reference tag name definition is changed from [Common Tag Name Definition] to [Block Tag Name Definition], common tag name definitions used within the block are extracted to the block tag name definition. Where addresses are duplicated, the block tag name definition has precedence. Structure object definitions are not merged.
E2.1.3 Restrictions on Tag Name Definitions

The following restrictions apply to tag name definitions.

- Number of Tag Name Definitions
  Up to 70,000 common tag names can be defined.
  Set the maximum number of common tag name definitions in Project Settings/Configuration.
  The maximum number of block tag name definitions for each block is fixed at 5,120.

- Duplicate allocation of address
  Addresses that are already registered cannot be registered again.

- Duplicate definition of tag name
  Tag names that are already registered cannot be registered again.

- Registration of Index-Modified Devices
  The following definitions are not allowed.

  Tag Name   Address
  SW 1       X00301;V01 ← Index Modification
  SW 2       @D0001 ← Indirect specification

- Identification of Structure Tag Names
  Tag names containing a period ("." ) represents structure tag names and hence, cannot be registered.

TIP
When a new project is created, templates of common tag name definitions can be read.

SEE ALSO
- You can set the maximum number of common tag name definitions in the Executable Program Details/Setup dialog box. For details on the Executable Program Details/Setup dialog box, see Section D3.1.16, "Component Block Setup."
- For details on reading templates of common tag name definitions, see E2.2.9, "Reading Templates of Common Tag Name Definitions."
E2.2 Editing Tag Name Definition

The basic operation of the tag name definition windows is the same for common tag name definitions, block tag name definitions, and macro tag name definitions.

SEE ALSO
For details on the find function in the tag name definition window, see Section E4.1.4, "Finding Tag Name Definitions."

E2.2.1 Basic Operations for Editing Tag Name Definition

Opening and Editing Tag Name Definition Files

First, you open a tag name definition file to display a Tag Name Definition window. Then, you can edit tag name definitions in the Tag Name Definition window. The relevant procedure is given below.

◆ Procedure ◆

1. In the case of common tag name definitions, double-click [Common Tag name Definition] in the Project window. In the case of block tag name definition, display the Edit Block window and select [Edit]–[Block Tag Name Definition] from the menu.
   ⇒ The tag name definition window is displayed. The figure on the right shows an example of a block tag name definition window.

2. Double-click the cell to be edited, or move the cursor to the cell to be edited and press the [F2] key.
   ⇒ System enters into Cell Edit mode.

3. Edit the cell content and press the [Enter] key.
   ⇒ Cell edit mode ends and the changes are confirmed.

TIP
- You can assign more than one I/O comment.
- If a block is opened in read mode, the block tag name definition screen for the block is opened in read mode. If the block is opened in write mode, the block tag name definition screen is opened in write mode.

SEE ALSO
For details on assigning multiple I/O comments, see E2.8.1, "Enabling to Display Multiple I/O Comments."
Saving Tag Name Definition File and Closing Tag Name Definition Window

The procedure for saving a tag name definition file that has been edited and closing a tag name definition window is given below.

◆ Procedure ◆

(1) Make the file to be saved the active window.
(2) Select [File]–[Save] from the menu. ⇒ The tag name definition file is overwritten and saved.
(3) Select [File]–[Close] from the menu. Or, click [Close] on the Tag Name Definition window. ⇒ The Tag Name Definition window closes.

CAUTION

- If an error such as duplicate definition of address or duplicate definition of tag name is detected, the background color of the line containing the error changes. When closing a tag name definition file containing lines with errors, the file cannot be saved without first discarding the errors. If a file contains lines with errors, a dialog box is displayed to confirm whether to discard lines with errors.

- If you attempt to close a file with unsaved tag name definition modifications, a dialog box is displayed to confirm whether to save the changes.
E2.2.2 **Save Function for Tag Name Definitions**

Common tag name definition and block tag name definitions are saved to actual projects (user files) at different times.

**Relationship Diagram for Saving of Tag Name Definitions**

The Save function on a block tag name definition window refers to saving the definitions to the block. Hence, the block tag name definitions are not saved to the project until and unless the block is saved to the project.

The Save function on a common tag name definition window saves the data directly to the project.

![Relationship Diagram for Saving of Tag Name Definition Files](E0202_05.VSD)

**Overview of Save Processing**

The following table compares the handling of common tag name definition and block tag name definition for various file save functions in WideField3.

**Table E2.4 Overview of Save Processing**

<table>
<thead>
<tr>
<th>File Save Function in WideField3</th>
<th>Saving to Common Tag Name Definition File</th>
<th>Saving to Block Tag Name Definition File</th>
</tr>
</thead>
<tbody>
<tr>
<td>[File]–[Close Project]</td>
<td>The latest information is saved.</td>
<td>The latest version is saved only upon confirmation to save the block tag name definition and also confirmation to save the block.</td>
</tr>
<tr>
<td>[File]–[Save Project]</td>
<td>The latest information is saved. If you select to ignore errors and perform saving, the previously saved file is saved.</td>
<td>All tag name definitions in edit mode are saved. If you select to ignore errors and perform saving, the previously saved file is saved.</td>
</tr>
<tr>
<td>[File]–[Save Project As]</td>
<td>The latest information is saved. Selecting not to discard error lines on the confirmation dialog aborts save processing.</td>
<td>All tag name definitions in edit mode are saved. Selecting not the discard error lines on the confirmation dialog aborts save processing.</td>
</tr>
<tr>
<td>[File]–[Save Project in Other Formats]–[Downgrade and Save]–[Convert to WideField format]</td>
<td>The latest information is saved in WideField format. Saving is terminated if error is detected during saving.</td>
<td>All tag name definition data in edit status is saved in WideField format. Saving is terminated if error is detected during saving.</td>
</tr>
<tr>
<td>[File]–[Save Project in Other Formats]–[Downgrade and Save]–[Convert to WideField2 format]</td>
<td>The latest information is saved in WideField2 format. Saving is terminated if error is detected during saving.</td>
<td>All tag name definition data in edit status is saved in WideField2 format. Saving is terminated if error is detected during saving.</td>
</tr>
<tr>
<td>[File]–[Close]</td>
<td>The latest information is saved.</td>
<td>Data is saved to the project as long as the block is saved after save processing for the tag name definitions. If the block is not saved, tag name definition reverts to the tag name definition saved previously</td>
</tr>
<tr>
<td>[File]–[Save As]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
E2.2.3 Error Handling

On tag name definition windows, lines and cells containing errors are displayed with a different background color. Lines containing input errors are displayed with yellow background whilst cells with errors are displayed with different background colors depending on the error type.

![Common Tag Name Definition](E0202_06.VSD)

**Figure E2.6  Error**

<table>
<thead>
<tr>
<th>Error Item</th>
<th>Background Color</th>
<th>Cause</th>
<th>Troubleshooting</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Duplicate address</td>
<td>Red</td>
<td>The same address is already defined within the tag name definition.</td>
<td>Change to a different address.</td>
</tr>
<tr>
<td>B Duplicate tag name</td>
<td>Red</td>
<td>The same tag name is already defined within the tag name definition.</td>
<td>Change to a different tag name.</td>
</tr>
<tr>
<td>C Invalid device</td>
<td>Blue</td>
<td>A device that cannot be used in an address position was registered.</td>
<td>Enter a valid address.</td>
</tr>
<tr>
<td>D Invalid tag name</td>
<td>Blue</td>
<td>A device name that cannot be used in a tag name position was registered.</td>
<td>Enter a valid tag name.</td>
</tr>
<tr>
<td>E Invalid comment</td>
<td>Blue</td>
<td>Length of I/O comment exceeded limit.</td>
<td>Shorten the I/O comment.</td>
</tr>
<tr>
<td>— Number of definitions exceeded</td>
<td>Pink</td>
<td>The number of tag name definitions in the address list exceeded the maximum number of tag name definitions allowed.</td>
<td>Reduce the number of tag name definitions.</td>
</tr>
</tbody>
</table>
E2.2.4 Copying, Cutting, Pasting and Deleting

You can copy, cut and paste a selected cell range on a tag name definition window. You can even copy and paste between tag name definition windows. The following procedure illustrates an example procedure for copying cells.

◆ Procedure ◆

(1) Select the cells to be copied on a tag name definition window.
(2) Select [Edit]–[Copy] from the menu.
(3) Make the tag name definition window where data is to be pasted the active window.
(4) Move the cursor to the position where data is to be pasted.
(5) Select [Edit]–[Paste] from the menu.
⇒ The copied data is pasted.

TIP
Similarly, you can select cells and perform cutting and deleting.

TIP
- You can specify a cell range in the tag name, address or I/O comment column for usual tag names. For tag names of structure object definitions, specify the selected range in line units.
- You can also paste tag name definitions from the circuit edit screen.
- You can move tag name definitions using a drag and drop operation. To do so, select a range, press the left mouse button down on the border of the selected range, and drag the selected range while holding down the [Alt] key. To drop the selected data into the circuit edit screen, release the [Alt] key and then drop the data. To drop the data as a text into MS-Excel or other applications, drop the data while holding down the [Alt] key.
SEE ALSO
For details on pasting tag name definitions in the circuit edit screen, see E1.2.31, "Copying and Moving Circuits Including Tag Name Definitions."

TIP
Tag name definitions copied or cut can be pasted into the following screens or dialog boxes:
- Registered device monitor
- Sampling Trace Setup dialog box
- Screens in other applications that support device data

CAUTION
- Input data validation is not performed when data is pasted. If an error occurs after data is pasted, lines containing errors are displayed as error lines. You cannot save data or exit without first removing the errors.
- Paste, cut and delete operations, once performed, cannot be undone using the “Undo” function.
E2.2.5 Sort

You can sort and rearrange tag name definitions. Sorting improves the efficiency of display, reference, and editing. Sorting can be performed in two modes: tag name order and address order.

SEE ALSO

For details on display modes, see Section E2.2.7, “Switching Display”.

Sorting is enabled only when the display format is set to display registered devices. The relevant procedure is given below.

◆ Procedure ◆

(1) Check that the tag name definition window to be sorted is displayed.
(2) Determine the display format.
(3) Select [Functions]–[Sort] from the menu.

⇒ Sorting is performed.

TIP
Definition errors are displayed at the end.

■ Display Order after Sorting

Sorting arranges and displays data according to the Display Format setting: ‘By Tag Name’ or ‘By Address’.

Undefined tag names or addresses and definition errors are sorted differently. The display order depends on the display format, as shown in the following table.

Table E2.6 Display Order after Sorting

<table>
<thead>
<tr>
<th>Display Format</th>
<th>By Address</th>
<th>By Tag Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully defined</td>
<td>Data is displayed in address order.</td>
<td>Data is displayed in tag name order.</td>
</tr>
<tr>
<td>Address yet unallocated</td>
<td>Such data is displayed at the beginning of the tag name definitions.</td>
<td>Such data is sorted and displayed together with tag name definitions already fully defined.</td>
</tr>
<tr>
<td>Tag name yet undefined</td>
<td>Such data is sorted and displayed together with tag name definitions already defined with tag names.</td>
<td>Such data is displayed at the end of the tag name definitions but before any definition errors.</td>
</tr>
<tr>
<td>Definition error</td>
<td>Such data is displayed at the end of the tag name definitions.</td>
<td>Such data is displayed at the end of the tag name definitions.</td>
</tr>
</tbody>
</table>
Example of Display: Registered Devices (by Addresses)

The following figure shows an example of a display sorted by address.

![Display Sorted by Address](image1)

- Address yet unallocated
- Fully defined
- Tag name yet undefined
- Definition error

Figure E2.7 Display Sorted by Address

Devices with undefined address are displayed at the beginning, sorted by tag name. All definition errors are displayed at the end.

Example of Display: Registered Devices (by Tag Names)

The following figure shows an example of a display sorted by tag name.

![Display Sorted by Tag Name](image2)

- Fully defined
- Address yet unallocated
- Tag name yet undefined
- Definition error

Figure E2.8 Display Sorted by Tag Name

Devices with undefined tag names are displayed at the end, sorted by address. Definition errors are displayed at the end.
E2.2.6 Read Circuits Function

The Read Circuits function reads tag names and addresses used in circuits. You can use this function when you wish to allocate addresses to tag names or to enter I/O comments for tag name definitions.

Prerequisites for Reading

Prerequisites for executing the Read Circuits function differ for block tag name definition and common tag name definition, as shown below.

- For block tag name definition, the block must have been converted.
- For common tag name definition, the block to be read must have been saved.

Results of Reading

When you execute the Read Circuits function, tag names and addresses used within the program are read and an asterisk (*) is displayed to the left of the cells of tag names or addresses used in the block for which reading is performed.

CAUTION

- If the number of tag names or addresses that are to be read from the program during a Read Circuits operation exceeds the number of tag name definitions that can be added, an error with the message “Number of registered devices exceeded limit. Some devices were not read.” is displayed and reading is aborted. If this error occurs with the common tag name definition, increase the maximum number of common tag name definitions in Project Settings/Configuration.
- Structure names, structure member names, structure pointer (Q), and member names of structure pointer (Q) are not read by the Read Circuits operation.

SEE ALSO

For details on the number of common tag name definitions, see Section E2.6, “Changing Maximum Number of Common Tag Name Definitions and Storing to CPU.”
Read Circuits Function for Block Tag Name Definition

The procedure for executing the Read Circuits operation for a block tag name definition is given below.

◆ Procedure ◆

(1) Check that the Block Tag Name Definition window is active.

(2) Select [Functions]–[Read Circuits] from the menu.

   ⇒ The Read Circuits dialog box is displayed.

   TIP
   If the block tag name definition has been edited but not yet saved, a confirmation dialog box is displayed before the Read Circuits dialog box appears. Click [Yes]. The Read Circuits dialog box is displayed.

(3) Turn on the displayed checkboxes as required and click [OK].

   ⇒ Tag names and addresses used in the block are read.

   TIP
   - [Reads global devices] checkbox
     When this checkbox is turned on, global devices used in a block are also read into the block tag name definition.
     When it is turned off, tag names and local devices are read, but global devices are not read.
   - [Deletes unused tag names] checkbox
     When this checkbox is turned on, all unused tag name definitions are deleted from the block tag name definition after the circuits are read.
Read Circuits Function for Common Tag Name Definition

The procedure for executing the Read Circuits function for common tag name definition is given below.

◆ Procedure ◆

1. Check that the Common Tag Name Definition window is active.
2. Select [Functions]–[Read Circuits] from the menu.
   ⇒ The Select Block/Macro dialog box is displayed.

3. Select the object to read.

   **TIP**
   To read from blocks, turn on the [Select target blocks] checkbox. To read from macros, turn on the [Select target macros] checkbox. To read from both blocks and macros, turn on both checkboxes.

4. Select the detailed read objects by turning on the desired option box under each checkbox.

   **TIP**
   - Turning on the [All Blocks] option button reads from all blocks registered in the component definition.
   - Turning on the [All Macros] option button reads from all macros registered in the component definition. Note that blocks referring to the common library or another project are not read.
   - Turning on the [Blocks Referring to Common Tag Name Definition] or [Macros Referring to Common Tag Name Definition] option button reads from all blocks registered in the block component definition, or respectively macro component definition, with its reference tag name definition set to common tag name definition. Note that blocks referring to the common library or another project are not read.
   - Turning on the [Select Blocks] or [Select Macros] option button allows you to select blocks, or respectively macros as targets for the Read Circuits operation.
(5) If you had turned on the [Select Blocks] or [Select Macros] option button earlier, click the [Browse] button.
⇒ The Block List or the Macro List dialog box is displayed.

**TIP**

The Block List dialog box displays block file names. The Macro List dialog box displays macro file names.

(6) Turn on checkboxes for blocks or macros to be read from the list box.

**Click [OK].**
⇒ Display returns to the [Select Block/Macro] dialog box.

(7) **Click [OK].**
⇒ The Select Block/Macro dialog box closes and the tag names or addresses are read.

---

**CAUTION**

- If an error message stating that "Number of registered devices exceeded limit. Some devices were not read." is displayed at the end of reading, increase the number of common tag number definitions.
- Structure names, structure member names, structure pointer (Q), and member names of structure pointer (Q) are not read by the Read Circuits operation.

---

**SEE ALSO**

For details on the number of common tag name definitions, see Section E2.6, "Changing Maximum Number of Common Tag Name Definitions and Storing to CPU."
E2.2.7 Switching Display

Tag name definition windows can be set to display registered devices only or to display a complete device list.
Registered devices refer to devices with completed tag name definition and can be displayed sorted by address or by tag name.
A device list displays the maximum number of devices for each device categories on a CPU type, regardless of whether the devices are registered.

Display Mode

There are three display modes.
- Registered Devices (by addresses)
  Displays registered tag name definitions only in ascending order of the addresses.
- Registered Devices (by tag names)
  Displays registered tag name definitions only in ascending alphabetic order of the tag names.
- Device List
  Displays all devices
Switching Display

To switch the display mode, select [View]–[Display Format], followed by either [Registered Devices (by Addresses)], [Registered Devices (by Tag Names)] or [Address List] from the menu.

<table>
<thead>
<tr>
<th>View</th>
<th>Project</th>
<th>Functions</th>
<th>Online</th>
<th>Debug/Maintenance</th>
<th>Tools</th>
<th>Window</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display Format</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Registered Devices (by Addresses)</td>
<td></td>
<td></td>
<td>All+2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Display Device</td>
<td>Registered Devices (by Tag Names)</td>
<td></td>
<td></td>
<td>All+2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Address List</td>
<td></td>
<td></td>
<td>All+2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure E2.10  Switching Display Modes

TIP

The inverted triangle in the item columns of the Tag Name Definition edit window can be used to distinguish between the two modes, as shown in the figures below.

![Figure E2.11  For Tag Name Display Style](E0202_21.VSD)

![Figure E2.12  For Address Display Style](E0202_22.VSD)

You can also switch the display by clicking the display item column for either [Tag Name] or [Address].

To switch the display of the device list, select [View]–[Display Device] followed by the device category to be displayed.

![Figure E2.13  Device List](E0202_23.VSD)
E2.2.8 Structure Object Definition

By editing tag name definitions, you can instantiate structures in their respective tag name definition locations based on structure type definitions defined in a project. Structures defined using structure type definition becomes usable in blocks after instantiation in tag name definition.

- Opening Tag Name Definition File and Performing Structure Object Definition

Open a tag name definition file and displays the tag name definition window. From the tag name definition window, display the Structure Object Definition dialog box to perform editing.

The relevant procedure is given below.

◆ Procedure ◆

1. In the case of common tag name definition, double-click [Common Tag Name Definition] in the project window. In the case of block tag name definition, display the [Edit Block] window and select [Edit]–[Block Tag Name Definition] from the menu.

   The tag name definition window is displayed.

2. Move the position cursor to the cell to be edited and select [Functions]–[Structure Object Definition] from the menu.

   TIP
   The figure shows an example of a Block Tag Name Definition window.

   ⇒ The Structure Object Definition dialog box is displayed.

3. Enter a name to be used for the structure after instantiation in the [Structure Name] text box.

   TIP
   Up to 7 characters can be specified for the structure name.

4. Select the structure type definition to be instantiated from the [Structure Type Definition] drop-down list.

Steps (3) to (7)
**TIP**

- You can read information (members, member properties, I/O comments) for the structure type selected in the [Structure Type Definition] drop-down list by clicking the [Details] button.
- Structure type definitions defined and saved in the WideField3 common folder (the default is the “Fam3com” folder in the folder where WideField3 is installed) are read.

(5) Specify the array size for the structure to be instantiated using the [Array Size] spin box.

**TIP**

Specifying a value of zero for the array size means that the structure to be instantiated is not an array.

(6) Specify actual addresses in the CPU to be allocated to the structure.

**TIP**

The internal addresses are automatically allocated according to the type of the structure members.

<table>
<thead>
<tr>
<th>Relay Devices</th>
<th>Register Devices</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relay</td>
<td>Integer(DEC)</td>
</tr>
<tr>
<td></td>
<td>Integer(HEX)</td>
</tr>
<tr>
<td></td>
<td>Integer(BCD)</td>
</tr>
<tr>
<td></td>
<td>Long Integer(DEC)</td>
</tr>
<tr>
<td></td>
<td>Long Integer(HEX)</td>
</tr>
<tr>
<td></td>
<td>Long Integer(BCD)</td>
</tr>
<tr>
<td></td>
<td>Float</td>
</tr>
<tr>
<td></td>
<td>Char</td>
</tr>
</tbody>
</table>

**Input Relay**

**Output Relay**

**CAUTION**

Mapping structure type object definition to local devices is not allowed in common tag name definition.

(7) Click [OK].

⇒ The Structure Body Definition dialog box closes and display returns to the tag name definition window.

**TIP**

Structures which have been instantiated are displayed in grey.
CAUTION

- Structure tag names are allocated respective addresses from relay devices, register devices, input relays (X) and output relays (Y) in the CPU. These address allocations should not duplicate the local device settings in the Project Settings/Configuration.

- An address that is automatically allocated according to the structure tag name setup is not displayed as an error during tag name definition even if it exceeds the usable range for a CPU type. If an error is indicated during program syntax checking, you should reallocate a valid address to the tag name.

- Devices are automatically allocated to a structure in tag name definition. You can also check whether devices automatically allocated to structures overlap devices assigned to tag names.

To do so, select [Functions]–[Check for Overlapping Structures] from the menu with the tag name definition window opened.

Figure E2.14  [Results of checking for overlapping structures:] dialog box

If overlapping allocations are found, they are displayed in a list, as shown in the above figure. You must correct the settings to remove these violations.
E2.2.9 Reading Templates of Common Tag Name Definitions

For common tag name definitions, you can use predefined I/O comments for two types of special devices: special relays (M) and special registers (Z).

Reading Templates

When creating a new project, you can select whether to read the predefined I/O comments of special relays (M) and special registers (Z) that are set in common tag name definitions.

This option is available only on the New Project dialog box.

◆ Procedure ◆

(1) Turn on the [Read default common tag name definition] checkbox at the bottom of the New Project dialog box.

⇒ Common tag name definitions including the default I/O comments of special relays (M) and special registers (Z) are created.
E2.3 Merging Tag Name Definitions

The Merge Tag Name Definition function extracts and merges selected tag name definition information from a merge source into the tag name definition of a destination block.

This merge source can either be the tag name definition of another block or the common tag name definition, but for ease of reference, will be referred to as tag name definition of the other block in this section.

You can select either to extract only tag name definition information corresponding to tag names or addresses found in the merge destination (that is, tag name definition of the destination block), or to extract all information from the other block’s tag name definitions.

Prerequisites for Merging

The following conditions must be satisfied for merging tag name definitions.

- Tag name definitions of the other block to be merged must be saved data. Only saved data in the other block’s tag name definition file can be merged. Tag name definition information of the other block currently being edited cannot be merged.
- The tag name definition display format must be set to Registered Devices. Merging is not allowed if the tag name definition display format of the destination block is set to Address List.

TIP

Block tag name definition files and common tag name definition files are saved at different times.
- A block tag name definition file is saved when you select to save the block.
- A common tag name definition file is saved when you select to save tag name definition.

CAUTION

Structure names are not merged.

Merge Operation for Tag Name Definitions

There are two options for merge method available for selection: the first option merges only tag name definitions of the other block that are relevant to the destination block; the second option reads all tag name definitions of the other block. In addition to the merge method, you also need to select the precedence for conflict resolution of duplicate tag name definitions. The basic merge operation is described below.

- When merging duplicate tag name definitions:

  If a tag name definition record of the destination block contains only a tag name, then the merge operation extracts and merges the address and I/O comment corresponding to this same tag name found in the other block’s tag name definition. If a tag name definition record of the destination block contains only an address (and an I/O comment), the merge operation extracts and merges the I/O comment corresponding to this same address found in the other block’s tag name definition. If this address is allocated to a tag name in the other block’s tag name, however, the tag name definition data is not merged.

- When the Read All Tag Name Definitions of the Other Block option is selected:

  The merge operation adds all tag name definitions of the other block to the tag name definition of the destination block. Any duplicate tag name definitions, however, are merged according to the precedence selection.
The precedence selection affects the merge operation as follows:

- If tag name definitions of the destination (this) block is given precedence:
  If the tag name or address of a tag name definition of the other block duplicates a definition in the destination block, its data is not merged.

- If tag name definitions of the other block is given precedence:
  If the tag name of a tag name definition of the other block duplicates a definition in the destination block, its address and I/O comment data are merged.
  If a tag name definition of the other block has an undefined tag name but its address (+ I/O comment) duplicates a definition in the destination block, its I/O comment data is merged.

---

**CAUTION**

If merging causes duplicate tag names or addresses, a duplicate use error is displayed at the place of duplication.

When the option to read all tag name definitions of the other block is selected, there is a possibility that the number of registered tag name definitions may exceed the maximum limit allowed. If this happens, merging proceeds up to the maximum number of registered tag name definitions allowed and skips subsequent processing.
E2.3.1 Merging Other Block’s Tag Name Definitions

The Merge Other Block’s Tag Name Definition function merges tag name definition information from another block tag name definition.

It can be executed from the menu of a Block Tag Name Definition window or a Common Tag Name Definition window.

![Figure E2.15 Merging Other Block’s Tag Name Definitions](E0203_01.VSD)

The following procedure lists the steps for merging information from the block tag name definition shown in the above figure.

◆ Procedure ◆

1. Check that the Block Tag Name Definition window for the merge destination (this block) is active.
2. Select [Functions]–[Merge Other Block’s Tag Name Definitions] from the menu.
   ⇒ The Open File dialog box is displayed.
3. Select the tag name definition file of another block to be used as the merge source and click [Open].
   ⇒ The Select Tag Name Definition Precedence dialog box is displayed.
4. Turn on the appropriate option button for the block to be given precedence.
   Click [OK].
   ⇒ Merging begins.
Examples for Merging of Tag Name Definitions

The following examples illustrate how tag name definition precedence selection and merge method selection affect the merge operation.

(1) Other block’s tag name definition has precedence and the merge method option to read all tag name definitions of the other block is not selected:

- **Before merging**
  
<table>
<thead>
<tr>
<th>Destination Block’s Tag Name Definitions</th>
<th>Other Block’s Tag Name Definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>sw1 100001  Comment I01</td>
<td>sw1 100004  Comment I04</td>
</tr>
<tr>
<td>sw2 100002  Comment I02</td>
<td>sw2 100005  Comment I05</td>
</tr>
<tr>
<td>sw3 100003  Comment I03</td>
<td>sw4 100006  Comment I06</td>
</tr>
</tbody>
</table>

- **After merging**
  
<table>
<thead>
<tr>
<th>Destination Block’s Tag Name Definitions</th>
<th>Other Block’s Tag Name Definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>sw1 100004  Comment I04</td>
<td>← Merged</td>
</tr>
<tr>
<td>sw2 100005  Comment I05</td>
<td>← Merged</td>
</tr>
<tr>
<td>sw3 100003  Comment I03</td>
<td>← Not merged</td>
</tr>
</tbody>
</table>

(2) Other block’s tag name definition has precedence and the merge method option to read all tag name definitions of the other block is selected:

- **Before merging**
  
<table>
<thead>
<tr>
<th>Destination Block’s Tag Name Definitions</th>
<th>Other Block’s Tag Name Definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>sw1 100001  Comment I01</td>
<td>sw1 100004  Comment I04</td>
</tr>
<tr>
<td>sw2 100002  Comment I02</td>
<td>sw2 100005  Comment I05</td>
</tr>
<tr>
<td>sw3 100003  Comment I03</td>
<td>sw4 100006  Comment I06</td>
</tr>
</tbody>
</table>

- **After merging**
  
<table>
<thead>
<tr>
<th>Destination Block’s Tag Name Definitions</th>
<th>Other Block’s Tag Name Definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>sw1 100004  Comment I04</td>
<td>← Merged</td>
</tr>
<tr>
<td>sw2 100005  Comment I05</td>
<td>← Merged</td>
</tr>
<tr>
<td>sw3 100003  Comment I03</td>
<td>← Not merged</td>
</tr>
<tr>
<td>sw4 100006  Comment I06</td>
<td>← Added</td>
</tr>
</tbody>
</table>

(3) Destination block’s tag name definition has precedence and the merge method option to read all tag name definitions of the other block is selected:

**TIP**

When the destination block’s tag name definition has precedence, the Read All Tag Name Definitions of the Other Merge Block merge option is always enabled.
Before merging

<table>
<thead>
<tr>
<th>Destination Block’s Tag Name Definition</th>
<th>Other Block’s Tag Name Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>sw1 I00001 Comment I01</td>
<td>sw1 I00004 Comment I04</td>
</tr>
<tr>
<td>sw2 I00002 Comment I02</td>
<td>sw2 I00005 Comment I05</td>
</tr>
<tr>
<td>sw3 I00003 Comment I03</td>
<td>sw4 I00006 Comment I06</td>
</tr>
</tbody>
</table>

After merging

<table>
<thead>
<tr>
<th>Destination Block’s Tag Name Definition</th>
<th>Other Block’s Tag Name Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>sw1 I00001 Comment I01← Not merged</td>
<td></td>
</tr>
<tr>
<td>sw2 I00002 Comment I02← Not merged</td>
<td></td>
</tr>
<tr>
<td>sw3 I00003 Comment I03← Not merged</td>
<td></td>
</tr>
<tr>
<td>sw4 I00006 Comment I06← Added</td>
<td></td>
</tr>
</tbody>
</table>

CAUTION

When merging tag name definitions of a block to common tag name definitions, tag name definitions assigned to local devices will not be merged.

E2.3.2 Merging Common Tag Name Definitions

The Merge Common Tag Name Definition function merges information from the common tag name definition into a block tag name definition.

The procedure is essentially the same as that for merging tag name definitions from another block.

SEE ALSO

For details on merging tag name definitions from another block, see Section E2.3.1, “Merging Other Block’s Tag Name Definitions.”

Differences between Merge Common Tag Name Definitions and Merge Other Block Tag Name Definitions Functions

The Merge Common Tag Name Definitions function differs from the Merge Other Block’s Tag Name Definitions function in the following ways:

- The data to be merged (merge source) is always the common tag name definition of the project.
- The following Select Tag Name Definition Preference dialog box is displayed.

![Select Tag Name Definition Preference Dialog Box](E0203_06.VSD)

Figure E2.16 Select Tag Name Definition Preference Dialog Box

- The Merge Common Tag Name Definitions function cannot be executed from the Common Tag Name Definition window.
  To merge common tag name definitions, display the Block Tag Name Definition window and select [Functions]–[Merge Common Tag Name Definitions] from the menu.
E2.4 Deleting Unused Tag Names

You can delete tag name definitions not used by the program from information registered in a tag name definition file. The procedure is slightly different for common tag name definition and block tag name definition.

**Block Tag Name Definition**

The following procedure describes how to delete unused tag name definitions from a block tag name definition.

◆ Procedure ◆

1. Check that the Block Tag Name Definition window is active.
2. Select [Functions]–[Delete Unused Tag Names] from the menu.
   ⇒ A confirmation dialog box is displayed.
3. Click [Yes].
   ⇒ Unused tag name definitions are deleted. The records are sorted and padded after deletion.

**Common Tag Name Definition**

The procedure for deleting unused tag name definitions from the common tag name definition is largely similarly to that for block tag name definition. However, you can specify multiple blocks and macros to be searched, instead of a single block or macro. When you select Delete Unused Tag Names from the menu, a Select Block/Macro dialog box is displayed for you to select the target blocks or macros to be searched.
The following items are displayed on the Select Block/Macro dialog box.

- [Select target blocks] checkbox.
  Sets blocks as operation targets.

- [All Blocks] option button
  Selects all blocks registered in the component definition of the executable program.

- [Blocks Referring to Common Tag Name Definition]
  Among the blocks registered in the executable program, selects blocks which have their reference tag name definition set to common tag name definition.

- [Select Blocks] option button
  Selects up to 1024 target blocks from the project folder. Clicking the [Browse] button displays a list for selection.

- [Select target macros] checkbox
  Sets macros as operation targets.

- [All Macros] option button
  Selects up to 256 macros in the project folder.

- [Macros Referring to Common Tag Name Definition] option button
  Selects up to 256 macros in the project folder which have their reference tag name definition set to common tag name definition.

- [Select Macros] option button
  Selects up to 256 target macros from the project folder. Clicking the [Browse] button displays a list for selection.

⚠️ CAUTION ⚠️

- Target macros must be macros stored in the project folder. A macro stored in the macro folder cannot be selected as a target macro. You must first select [Project]–[Extract Macro] from the menu to extract it to the project folder.

- Unused structure body definitions are not deleted.

- Parameters of some instructions in a program may be explicitly coded with a first device but actually occupy multiple devices. When deleting unused tag names, beware that other than the first device, the remaining implicit devices are not considered as being used by the instructions.
E2.5 Group Templates

The Group Template Definition function defines the members of a group tag name. It defines all information except the group name. Once a group template is created, you can simply attach different group tag names to it when performing tag name definition for groups having the same configuration. A group tag name is of the form [Group Name_Member Name]. Group templates define only the group members, without the group name.

E2.5.1 Creating Group Templates

The procedure for creating a group template is given below.

◆ Procedure ◆

(1) Select [File]–[New] from the menu.
⇒ The New dialog box is displayed.
(2) Click the [Common Data] tab.

(3) Click the [Group Template] button.

(4) Enter a file name in the [File Name] text box.
(5) Click [OK].

⇒ The Group Template Definition window is displayed.
TIP
Edit the group template definition on the Group Template Definition window.
Selecting [File]–[Close] after editing a definition displays a dialog box to confirm whether to save the changes. Group templates containing no error are saved in the common folder.

Figure E2.18  Group Template Definition window with Edited Data

E2.5.2 Using Group Templates
The procedure for using a group template created previously is given below.

◆ Procedure ◆

(1) On a tag name definition window, move the cursor to where a group template is to be entered.
(2) Select [Functions]–[Read Group] from the menu.
⇒ The Read Group Name dialog box is displayed.
(3) Enter a group name, select a group template and click [OK].

⇒ The group template is read, the entered group name is combined with the individual member names and written to the [Tag Name] cells.
E2.6 Changing Maximum Number of Common Tag Name Definitions and Storing to CPU

You can set or change the maximum number of common tag name definitions allowed. You can also specify whether to store common tag name definitions to the CPU in the Project Settings/Configuration (for F3SP22-0S/SP28-3S/SP38-6S/SP53-4S/SP58-6S/SP59-7S/SP66-4S/SP67-6S/SP71-4N/SP76-7N/SP71-4S/SP76-7S CPU modules). The relevant procedure is given below.

TIP

You can also check the maximum number of common tag name definitions versus the number of common tag name definitions currently defined by looking at the title bar of the Common Tag Name Definition edit window.

◆ Procedure ◆

(1) Select [Project]–[Project Settings] from the menu.
⇒ The Project Settings/Configuration window is displayed.

(2) Select [Execution Block Components] from the tree in the Project Settings/Configuration window, and click the [Details/Setup] button.
⇒ The Executable Program Details/Setup dialog box is displayed.

(3) Enter the number of tag name definitions for [Number of Common Tag Name Defs] in the [Entire Project] group box, and click [OK].

TIP

To store common tag name definitions in the CPU, select "Yes" for [CPU Storage] in the [Entire Project] group box, and click [OK].

⇒ The maximum number of common tag name definition allowed is updated.

⚠️ CAUTION

A large preset value for the Maximum Common Tag Name Definitions will lengthen the processing time for opening and closing projects, as well as other edit operations.
E2.7 Importing and Exporting Tag Name Definitions

You can import/export common tag name definitions and block tag name definitions to/from CSV data. Converting tag name definitions as CSV data allows you to reuse tag name definitions with other applications and project data. This improves the efficiency of editing tag name definitions.

E2.7.1 Exporting Tag Name Definitions

The procedure for saving defined tag name definitions as CSV data is given below.

◆ Procedure ◆

1. Check that the Tag Name Definition edit screen is active.

2. Select [Functions]–[Output CSV Data] from the menu bar.
   ⇒ The Save As dialog box is displayed.

3. Enter the file name to be saved, and click the [Save] button.
   ⇒ The tag name definition data is saved in CSV format.
E2.7.2 Importing Tag Name Definitions

The procedure for loading tag name definition data saved as CSV data as tag name definition data for WideField3 is given below.

TIP

If tag name definition files being imported have the same tag name or address as defined tag name definitions, matching definitions will be imported. If there is inconsistency in the definitions of tag names or addresses, priority (for import) will be given to tag name definitions that match.

◆ Procedure ◆

(1) Check that the Tag Name Definition edit screen is active.

(2) Select [Functions]–[Load CSV Data] from the menu bar.
⇒ The Select File dialog box is displayed.

(3) Select the file to be imported, and click the [Select] button.
⇒ The CSV data is imported to the tag name definition.

CAUTION

CSV data that can be loaded as tag name definition data includes information for identification in WideField3.

Therefore, the following message will appear for CSV data that does not include this information (or in which this information has been corrupted), and the data will not be loaded.

Figure E2.20 Tag Name Definitions Import Error
E2.8 Multiple I/O Comments

Up to three I/O comments can be set for each tag name or address in common tag name definitions and block tag name definitions. Multiple I/O comments allow you to describe different comments for various purposes including localization or maintenance.

E2.8.1 Enabling to Display Multiple I/O Comments

You can configure the I/O comment input area displayed in the common tag name definition window and block tag name definition window. You can edit the I/O comment header strings displayed in the tag name definition windows.

Enabling to Display Multiple I/O Comments

◆ Procedure ◆

(1) Select [Project] - [I/O Comment Display Settings] from the menu bar.
⇒ The I/O Comment Display Settings dialog box is displayed.

(2) Turn on the [Visible] checkboxes for the I/O comments to be displayed in the tag name definition windows, and click [OK].

⇒ The specified I/O comment input areas are displayed in the common tag name definition window or block tag name definition window.
Setting I/O Comment Header Strings

◆ Procedure ◆

(1) Select [Project] - [I/O Comment Display Settings] from the menu bar.
⇒ The I/O Comment Display Settings dialog box is displayed.

(2) Double-click the I/O comment header to be modified.

(3) Enter a header string and click [OK].
⇒ The specified string is displayed in the I/O comment header of the common tag name definition window or block tag name definition window.
E2.8.2 Assigning Multiple I/O Comments

You can assign multiple I/O comments in common tag name definitions and block tag name definitions.

The relevant procedure is given below.

Assigning Multiple I/O Comments

To assign multiple I/O comments to tag names or addresses, you need to turn on the [Visible] checkboxes on the I/O Comment Display Settings dialog box in advance.

Procedure

1. Open the common tag name definition screen or block tag name definition screen to be edited.

See Also

For details on edit operations in tag name definition screens, see Section E2.2.1, "Basic Operations for Editing Tag Name Definition."

2. Enter an I/O comment in the I/O comment input area.

⇒ The multiple I/O comments are assigned to the single tag name or address.
E2.8.3 Selecting I/O Comments to Be Displayed

You can select I/O comments to be displayed in the ladder program edit window or other windows. The relevant procedure is given below.

◆ Procedure◆

(1) Select [Project] - [I/O Comment Display Settings] from the menu bar.
⇒ The I/O Comment Display Settings dialog box is displayed.

(2) Select the [Active] checkboxes for the I/O comments to be displayed in the ladder program edit window or other windows, and click [OK].

⇒ The selected I/O comments are displayed in the ladder program edit window or other windows.

TIP
If you edit I/O comments in the ladder program edit window, only the I/O comments currently displayed are modified.
E2.8.4 Saving the I/O Comment Display Settings as Default

You can save the settings configured on the I/O Comment Display Settings dialog box as default to use the settings in other projects.

The relevant procedure is given below.

saving the I/O Comment Display Setting as Default

Procedure

(1) Select [Project] - [I/O Comment Display Settings] from the menu bar.

⇒ The I/O Comment Display Settings dialog box is displayed.

(2) Click [Save As Default].

⇒ The currently displayed settings are registered as default for WideField3 and used as the default settings when a new project is created.
E3. Constant Definition

The constant definition function maps constant names to constant values so that constant names can be used as instruction parameters in place of constant names within blocks. Constant names can be entered into program of blocks, and even have their constant values assigned or changed after program creation. Constant names can also be added to comments.

Each project allows the creation of only one constant definition file, which is common to all blocks within the project. Constant names that are used within macros referred by a project must also be defined in the constant definition file of the project so that the sequence CPU can execute the program using the defined constant values.

CAUTION

- Constant definition is only available on some sequence CPUs.
- Constant names can generally be used in place of constants as instruction parameters but cannot be used in some instructions.

Table E3.1 Instructions that Disallow the Use of Constant Names as Constants

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Parameters that Do Not Allow the Use of Constant Names in Place of Constants</th>
</tr>
</thead>
<tbody>
<tr>
<td>HWR</td>
<td>Parameters 2, 3 and 4</td>
</tr>
<tr>
<td>HRD</td>
<td>Parameters 1, 2, 3 and 4</td>
</tr>
</tbody>
</table>

SEE ALSO

For a list of sequence CPUs that support constant definition, see "Sequence CPU Instruction Manual – Instructions" (IM34M06P12-03E).
## Function Limitations for Each CPU Type

### Table 3.2  Function Limitations for Each CPU Type

<table>
<thead>
<tr>
<th>CPU Modules</th>
<th>Limitations</th>
<th>SEE ALSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>F3SP22-0S,</td>
<td>The constant definition function cannot be used.</td>
<td></td>
</tr>
<tr>
<td>F3SP28-3S, F3SP38-6S,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F3SP53-4S, F3SP58-6S,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F3SP59-7S</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F3SP66-4S, F3SP67-6S</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>F3SP71-4N, F3SP76-7N</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>F3SP71-4S, F3SP76-7S</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>

**TIP**

This section does not describe hardware-dependant limitations for each CPU type, such as the number of available devices.

**SEE ALSO**

For details on limitations for each CPU type, refer to the user’s manual for each type.
E3.1 Fundamentals of Constant Definition

This section describes the layout of the Constant Definition Window and restrictions on constant definition.

E3.1.1 Constant Definition Window Layout

Data can be copied and pasted between constant definition windows, as well as between constant definition windows and tables of constant names with assigned values created using MS-Excel or a similar program.

Checks on constant names and values defined in the constant definition are constantly performed, with the results indicated in various colors in the display.

![Constant Definition Window](image)

Figure E3.1 Constant Definition Window
E3.1.2 Restrictions on Constant Definition

The following restrictions apply to constant definitions.

- Number of constant definitions
  Up to 2,048 constants can be defined. The maximum number of constants that can be defined depends on the data type, size and string lengths of defined constant names. You can check the number of defined constants in the project properties and perform syntax check of constant definitions at downloading.

- Duplicate definition of constant name
  Constant names that are already defined cannot be defined again.

- Consistent data type and constant value
  You must define the constant value of a constant name within the range of its specified data type.

- Size and constant value
  You must define the constant value of a constant name within the range of its specified size.
E3.2 Editing Constant Definition

This section describes how to edit constant definition.

SEE ALSO
For details on the find function in the Constant Definition window, see Section E4.1.5, "Finding Constant Definitions."

E3.2.1 Basic Operations for Editing Constant Definition

■ Opening and Editing Constant Definition File

Firstly, you open constant definition data to display a Constant Definition window. You can then edit constant definitions in the Constant Definition window. The relevant procedure is given below.

◆ Procedure ◆

(1) Double-click [Constant Definition] in the Project window.
⇒ The Constant Definition Window is displayed.
(2) Double-click the cell to be edited, or move the cursor to the cell to be edited and press the [F2] key.
⇒ System enters into Cell Edit mode.

(3) Edit the cell content and press the [Enter] key.
⇒ Cell edit mode ends and the changes are confirmed.
Saving Constant Definition File and Closing Constant Definition Window

The procedure for saving edited constant definition data and closing a Constant Definition window is given below.

◆ Procedure ◆

(1) Make the constant definition file to be saved the active window.
(2) Select [File]–[Save] from the menu.
   ⇒ The constant definition file is overwritten and saved.

(3) Select [File]–[Close] from the menu.
   Or, click [Close] on the Constant Definition window.
   ⇒ The Constant Definition window closes.

TIP
- If an error such as duplicate constant name definition or invalid constant value is detected, the background color of the line containing the error changes.
- If you attempt to close constant definition with unsaved changes, a dialog box is displayed to confirm whether to save the changes.

CAUTION

Invalid constant definitions cannot be saved.
When a constant definition file is saved, invalid constant definitions will be discarded.
E3.2.2 Specification of Elements of Constant Definition

The table below shows the specification, valid data range and restrictions for each element of a constant definition.

Table E3.3 Specification and Restrictions on Elements of Constant Definitions

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
<th>Specification and Restrictions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant name</td>
<td>Specify a constant name</td>
<td>2 to 16 alphanumeric characters beginning with a '#' character. (length limit includes the '#' character)</td>
</tr>
<tr>
<td>Type</td>
<td>Specify the data type for the constant name.</td>
<td>One of the following: WORD, LONG, DLONG, STRING, BIN, FLOAT, DOUBLE</td>
</tr>
<tr>
<td>Size</td>
<td>Specify an appropriate data size when BIN or STRING data type is specified.</td>
<td>0 to 256. If 0 is specified, the system automatically allocates the required size for the specified constant value. If BIN or STRING data type is specified, the size defaults to 0 if nothing is specified.</td>
</tr>
<tr>
<td>Value</td>
<td>Specify a constant value.</td>
<td>The entered value must be consistent with the specified data type.</td>
</tr>
<tr>
<td>Comment</td>
<td>Enter a comment for the constant name.</td>
<td>Up to 32 characters.</td>
</tr>
</tbody>
</table>

Entering a value which is out of valid range for any element generates an error.

**Type Specifications**

The table below lists the data types that can be specified for a constant name, along with their restrictions.

Table E3.4 Specification and Restrictions of Constant Data Types

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Specification and Restrictions</th>
</tr>
</thead>
<tbody>
<tr>
<td>WORD</td>
<td>2-byte numeric value or character string</td>
<td>Numeric value from -32768 to 32767. Hexadecimal value prefixed by a '$' character. Character string enclosed within double-quotes ('').</td>
</tr>
<tr>
<td>LONG</td>
<td>4-byte numeric value or character string</td>
<td>Numeric value from -2147483648 to 2147483647. Hexadecimal value prefixed by a '$' character. Character string enclosed within double-quotes ('').</td>
</tr>
<tr>
<td>DLONG</td>
<td>8-byte numeric value or character string</td>
<td>Numeric value from -9223372036854775808 to 9223372036854775807. Hexadecimal value prefixed by a '$' character. Character string enclosed within double-quotes ('').</td>
</tr>
<tr>
<td>STRING</td>
<td>Character string of the specified size (An odd-length string can also be specified. When an odd-length string is stored in the CPU, a NULL is appended at the end of the string to adjust the length to an even length.)</td>
<td>Character string enclosed within double-quotes (''). There is no restriction on the characters used. A NULL is automatically appended at the end of the string. (Example: &quot;String&quot;)</td>
</tr>
<tr>
<td>BIN</td>
<td>Character string specified in hexadecimal format. The maximum allowable length of the character string is 2*specified size.</td>
<td>Hexadecimal digit pairs (00 to FF), prefixed by a '$' character. (Example: $0D0A)</td>
</tr>
<tr>
<td>FLOAT</td>
<td>Number in IEEE defined single precision floating-point format</td>
<td>Numeric value whose exponential part is from -38 to +38 (significant digits of a decimal number is approx. 6 - 7), prefixed by a &quot;%&quot; character. (Example: %1.234560E-012)</td>
</tr>
<tr>
<td>DOUBLE</td>
<td>Number in IEEE defined double precision floating-point format</td>
<td>Numeric value whose exponential part is from -308 to +308 (significant digits of a decimal number is approx. 15 - 16), prefixed by a &quot;%&quot; character. (Example: %1.234560E-064)</td>
</tr>
</tbody>
</table>

Escape sequence can be specified in a character string in the same way as entering an
escape sequence when editing a program.

SEE ALSO
For more details on how to enter an escape sequence, see "Entering Escape Sequences" in Section E1.2.18, "Entering Escape Sequences, Block Names, and Label Names."

E3.2.3  Error Handling
On constant definition windows, lines and cells containing errors are displayed with a different background color. Lines containing input errors are displayed with yellow background whilst cells with errors are displayed with different background colors depending on the error type.

<table>
<thead>
<tr>
<th>Error</th>
<th>Background Color</th>
<th>Cause</th>
<th>Troubleshooting</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Duplicate constant name</td>
<td>Red</td>
<td>The same constant name is already defined within the constant definition.</td>
<td>Delete lines for unwanted definitions.</td>
</tr>
<tr>
<td>B Invalid constant name</td>
<td>Blue</td>
<td>An invalid constant name is specified.</td>
<td>Enter a valid constant name.</td>
</tr>
<tr>
<td>C Invalid size</td>
<td>Blue</td>
<td>Specified size is inconsistent with the specified constant data type.</td>
<td>Enter a valid size.</td>
</tr>
<tr>
<td>D Invalid constant value</td>
<td>Blue</td>
<td>Specified constant value is invalid.</td>
<td>Enter a valid constant value consistent with the constant data type and size.</td>
</tr>
<tr>
<td>E Invalid comment</td>
<td>Blue</td>
<td>Length of I/O comment exceeded limit.</td>
<td>Shorten the I/O comment.</td>
</tr>
</tbody>
</table>

CAUTION
Constant definitions containing errors (error lines displayed with yellow cells or red cells) cannot be saved. You should amend or delete these error lines.
E3.2.4 Copying, Cutting, Pasting and Deleting
You can copy, cut and paste, and delete a selected cell range within a constant definition window.
You can also copy and paste between constant definition windows of different projects.

TIP
For instruction parameters that accept a constant definition name entry in the circuit edit screen, by dragging and dropping a constant name from the constant definition screen to the circuit edit screen, you can change an instruction parameter to the constant definition name.
To do so, select and drag a tag name from the constant definition screen and drop it to an instruction parameter in the circuit edit screen.
In this drag and drop operation, you do not need to press the [Alt] key.

E3.2.5 Sort
You can sort and rearrange constant definitions.

TIP
Constant definitions are saved according to the order in which they are defined, and are not automatically sorted.

■ Sort Order
The table below lists the available sort orders for constant definitions.

Table E3.6 Constant Definition Sort Orders

<table>
<thead>
<tr>
<th>Sort Order</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>By Constant Name</td>
<td>Rearranges constant definitions in ascending order of constant name.</td>
</tr>
<tr>
<td>By Type/Constant Name</td>
<td>Rearranges constant definitions in ascending order of type + constant name.</td>
</tr>
<tr>
<td>By Type/Value</td>
<td>Rearranges constant definitions in ascending order of type + value.</td>
</tr>
<tr>
<td>By Comment</td>
<td>Rearranges constant definitions in ascending order of comment.</td>
</tr>
</tbody>
</table>

◆ Procedure ◆

1) Check that the constant definition window to be sorted is displayed.
2) Select [Functions]–[Sort] from the menu, and then select one of the displayed sort orders.

⇒ Sorting is performed.
E3.2.6 **Read Circuits Function**

The Read Circuits function reads constant names used in circuits. You can use this function when you wish to define constant values for constant names.

- **Prerequisites for Reading**
  
  Prerequisites for executing the Read Circuits function differ for constant definition, and are given below:
  - The block to be read must have been saved.
  - The block to be read must be included in the component definition.

- **Results of Reading**
  
  When you execute the Read Circuits function, constant names used within the project are read and an asterisk (*) is displayed to the left of the cells of constant names used within the target block for which reading was performed.

![Figure E3.3 After Reading](E0302_04.VSD)

- **Read Circuits Function for Constant Definition**
  
  The procedure for executing the Read Circuits operation for constant definition data is given below.

  - **Procedure**

    1. Check that the Constant Definition window is active.
    2. Select [Functions]–[Read Circuits] from the menu.
       ⇒ The Select Block/Macro dialog box is displayed.
    3. Select the object type to be read.

      **TIP**
      
      To read from blocks, turn on the [Select target blocks] checkbox. To read from macros, turn on the [Select target macros] checkbox. To read from both blocks and macros, turn on both checkboxes.
(4) Select the detailed read objects by turning on the desired option button under each checkbox.

**TIP**

- Turning on the [All Blocks] option button reads from all blocks registered in the block component definition; turning on the [All Macros] option button reads from all macros registered in the macro component definition.

- The [Blocks Referring to Common Tag Name Definition] and [Macros Referring to Common Tag Name Definition] option buttons are disabled in the display.

- Turning on the [Select Blocks] or [Select Macros] option button allows you to select blocks, or respectively macros as targets for the Read Circuits operation.

(5) If you had turned on the [Select Blocks] or [Select Macros] option button earlier, click [Browse].

⇒ The Block List or the Macro List dialog box is displayed.

(6) Turn on checkboxes for blocks or macros to be read from the list box.

Click [OK].

⇒ Display returns to the [Select Block/Macro] dialog box.

(7) Click [OK].

⇒ The Select Block/Macro dialog box closes and the constant names are read.

---

**CAUTION**

- Target macros can only be macros stored in the project folder. A macro stored in the macro folder cannot be selected as a target macro. To select a macro stored in the macro folder, you must first select [Project]–[Extract Macro] from the menu to extract the macro to the project folder.

- Blocks referring to the common library or another project are not read.
E3.3 Deleting Unused Constant Names
You can delete constant name definitions not used by the program from constant definition data.

◆ Procedure ◆

(1) Check that the Constant Definition window is active.
(2) Select [Functions]–[Delete Unused Constant Names] from the menu.
  ⇒ The Select Block/Macro dialog box is displayed.
(3) Select the target object type to be checked in the same way as for the Read Circuits function.
(4) Select the target objects by turning on the desired option button under each checkbox.
(5) If you had turned on the [Select Blocks] or [Select Macros] option button earlier, click [Browse].
  ⇒ The Block List or the Macro List dialog box is displayed.
(6) Turn on checkboxes for blocks or macros to be checked from the list box.
  Click [OK].
  ⇒ Display returns to the [Select Block/Macro] dialog box.
(7) Click [OK].
  ⇒ The Select Block/Macro dialog box closes and lines for unused constant names are deleted.

⚠️ CAUTION ⚠️

- Target macros must be macros stored in the project folder. A macro stored in the macro folder cannot be selected as a target macro. To select a macro stored in the macro folder, you must first select [Project]–[Extract Macro] from the menu to extract the macro to the project folder.
- Constant name definitions used in blocks referring to the common library or another project are not deleted. Constant names used in blocks not in the current project are considered as unused and deleted.
E3.4 Displaying Number of Constant Devices Used

Constant definitions make use of constant devices of the CPU. You can check the usage of constant devices from the Project Settings/Configuration window. The relevant procedure is given below.

◆ Procedure ◆

(1) Double-click [Project Settings/Configuration] in the project window.

⇒ The Project Settings/Configuration window is displayed.

(2) Select [Execution Block Components] from the tree in the Project Settings/Configuration window.

(3) Click [Details/Setup].

⇒ [Constant Def. Storage Area] (number of constant definitions used/maximum number of constant definitions that can be defined) is displayed in the [Entire Project] group box on the Executable Program Details/Setup window.
E4. Find and Replace

This chapter describes find and replace functions of WideField3. In WideField3, you can find devices, comments, and instructions, and replace devices in the currently displayed circuit or the current project. Find and replace operations can be performed on circuits, on the project, on tag name definitions, and on constant definitions.
You can specify blocks registered in the component definition, blocks or macros in the project, or all blocks and macros in the project as the range of a find or replace operation.

⚠️ CAUTION

- Replace operations are disallowed in address display mode. Always exit from address display mode before initiating a Replace operation. To exit from address display mode, select [View]–[Specify Display Mode]–[Display Address] from the menu or press the [Ctrl] + [A] keys again (the menu item operates like a toggle).
- Replace operations are disallowed during circuit monitoring.
- Replace operations in the project are not performed on blocks referring to the common library or another project.
- When [Find in Project] is specified as a search condition, blocks and macros in the project folder are searched. Therefore, blocks referring to the common library or another project are not searched.
- Find and replace operations are not performed on the contents in balloon comments and balloon monitors.

○ Function Limitations for Each CPU Type

Table E4.1 Function Limitations for Each CPU Type

<table>
<thead>
<tr>
<th>CPU Modules</th>
<th>Limitations</th>
<th>SEE ALSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>F3SP22-0S, F3SP28-3S, F3SP38-6S, F3SP53-4S, F3SP58-6S, F3SP59-7S</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>F3SP66-4S, F3SP67-6S</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>F3SP71-4N, F3SP76-7N</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>F3SP71-4S, F3SP76-7S</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>

TIP
This section does not describe hardware-dependant limitations for each CPU type, such as the number of available devices.

SEE ALSO
For details on limitations for each CPU type, refer to the user’s manual for each type.
E4.1 Find
The find functions search blocks or the project for specified devices, comments, or instructions, and display the location where they are found. There are two types of find functions: Find Device/Comment and Find Instruction. The following describes each type of the find functions.

E4.1.1 Finding Devices/Comments
Using the Find Device/Comment function, you can search for any of the following seven objects:
- Tag name/address
- Structure
- Circuit comment
- Subcomment
- Block name/macro name
- Label
- I/O comment

You can specify one of the following two types of ranges as the search range for the Find Device/Comment function:
- Within the block currently opened
- Over the entire project

⚠️ CAUTION

You cannot search contents in balloon comments and balloon monitors.

TIP
Performing a search using a tag name as a search string also searches for any address allocated to the tag name. Similarly, searching for an address also searches for any tag name assigned to this address. For instance, if “SW1=X00301” is defined, then searching for “SW1” and searching for “X00301” will produce the same result.

Furthermore, if circuit data contains a base modified by a constant index, then the expanded address is also matched against the search string. For instance, the parameter “I001;3” matches both search string “I001” and “I004”. When searching using an index register, index modified addresses are also searched.

TIP
- To quickly refer to devices used in a block or project, use the cross reference function.
- To quickly refer to the list of devices used or unused in a block or project, use the device list function.

SEE ALSO
- For details on the cross reference, see Chapter E6, "Cross Reference."
- For details on the device list, see Chapter E5, "Device List."
# Searching in the Open Block

To find devices/comments within the block currently opened, use the Search in open block tab in the Find in dialog box.

![Find in Dialog Box](E0401_01.VSD)

**Figure E4.1 Find in Dialog Box**

- **A** Enter the search string. You may use the wildcard character ("*" ) in the search string. You can also select a string from the past search history.
- **B** Click this button to find next matches. If the search objects are found, the display switches to the window containing the search objects.
- **C** Specify the number of points to search. You may specify up to 2,048 points. Usually, 1 point is specified.
- **D** Select the search object.
  - Turn on the [Tag Name/Address] checkbox to include tag names and addresses in the search objects. [Displayed Devices] specifies the first device displayed in the circuit edit screen as the search object and performs a simple match search of the string. [Used Devices] includes in the search objects the used devices not displayed in the circuit edit screen ("hidden devices").
  - Turn on the [Circuit Comment] checkbox to include circuit comments in the search objects.
  - Turn on the [Subcomment] checkbox to include subcomments in the search objects.
  - Turn on the [Block/Macro Name] checkbox to include block names and macro names in the search objects.
  - Turn on the [Label] checkbox to include labels in the search objects.
  - Turn on the [I/O Comment] checkbox to include I/O comments in the search objects.

**SEE ALSO**

For details on hidden devices, see "# Searching for Hidden Devices of Tag Names and Addresses" in this section.
Click this button to show or hide the list of details of search results. The following table shows the contents displayed in the details of search results.

### Table E4.2 Details of Search Results

<table>
<thead>
<tr>
<th>Search Object</th>
<th>Contents Displayed in the List of Details of Search Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tag name/address</td>
<td>The line number of the position where the object is found is displayed.</td>
</tr>
<tr>
<td>Circuit comment</td>
<td>The circuit comment string is displayed.</td>
</tr>
<tr>
<td>Subcomment</td>
<td>The subcomment string is displayed.</td>
</tr>
<tr>
<td>Block name/macro name</td>
<td>The entire instruction that contains the search address is displayed.</td>
</tr>
<tr>
<td>Label</td>
<td>The label name is displayed.</td>
</tr>
<tr>
<td>I/O comment</td>
<td>The search I/O comment string and the entire instruction that contains the search address are displayed.</td>
</tr>
</tbody>
</table>

Select the search direction.

If [From Top] is selected, search proceeds from the first line to the last line of the block.

If [Upwards from Cursor] is selected, search proceeds from the position cursor location to the beginning of the block.

If [Downwards from Cursor] is selected, the search proceeds from the position cursor location to the end of the block.

Turning on this checkbox performs loop search. The search starts from the beginning when it reaches the end of the block.

Displays details of search results. Click [Details] to show or hide the details.

**TIP**

- If you double-click a line in the details of search results, the location where the selected device is used is displayed in another window and the cursor is positioned at the selected device in the window.

- If an address is specified for the search string, as many addresses as the specified number of search points are searched from the specified address. For example, if "X201" is specified for [Search String] and "32" is specified for [Search Points], the search range is from X201 to X232.
To search in the currently opened block, use the following procedure.

◆ Procedure ◆

(1) Select [Find]–[Find] from the menu.
⇒ The Find in dialog box is displayed.
(2) Click the Search in open block tab.

TIP
When you start a find operation in the program edit screen or tag name definition edit screen, the parameter at the cursor position is specified for the search string.

(3) Enter the string to search for.
(4) Specify the object to search for.
(5) To specify the number of search points, enter a value in [Search Points].

TIP
You can only specify the number of search points when the [Tag Name/Address] checkbox is selected for the search object.

(6) Select the direction for the search.
(7) Turn on the Loop Search checkbox to perform a loop search.
(8) Click [Find Next].
⇒ The search begins in the selected search direction. If the search object is found, the display switches to the window containing the search object.

TIP
If you changed the CPU type in the project, even if you select [Tag Name/Address] for the search object, devices not supported for the CPU type after changed cannot be found. To find these devices, select some other search object options than [Tag Name/Address].
TIP
- You can repeat the previous search in the downward direction, starting from the current cursor position even after closing the Find in dialog box by selecting [Find]–[Find Next]–[Find Next] from the menu or pressing the [F3] key. Pressing the [Shift]+[F3] keys repeats the previous search in the upward direction.
- If the search reaches the end of the block (or the beginning of the block in the case of a search in the upward direction) without finding the search object, a dialog box is displayed to inform that the search object is not found.
- Even if the specified search string is without index modification, the search may result in a device with index modification so long as it matches the search pattern.

CAUTION
- If the search object is found in a hidden circuit, the circuit is expanded.
- If you enter an invalid search string, an error message is displayed. A search string that begins with a numerical digit generates a search string error.
- The wildcard ("*") character can be used in the search string but matches only tag names and addresses assigned to tag names. Searching for an address not assigned to a tag name using the wildcard character ("*") is not allowed. The following restrictions apply to the use of the wildcard character.
  - No more than one wildcard character may be used (e.g. "SW*AB*").
  - The wildcard character cannot be in the middle of a search string (e.g. "SW*WF*").
  - The wildcard character cannot be used together with index modification (e.g. "**SW;V01").
- When an address name is specified as the search string, both direct addresses and indirect addresses are searched.
- If an indirect address (an address prefixed with "@") is entered as the search string, only indirect addresses are searched.
- When searching for a constant name, prefix the search string with a '#' character.
- When searching for structure tag names, an array type is not allowed.

TIP
The following are some examples of search strings and their matches.
- [SW*]     SW1, SW2, SW3, SW4 . . .
- [*SW]     WF_SW, YO_SW, KO_SW, GA_SW, WA_SW . . .
- [D0001]   D0001, @D0001 . . .
- [@D0001]  @D0001 . . .
- [MITA*]   MITAKA, MITAKA.STATION, MITAKA.STORE,
            MITAKA[10].STATION, MITATMP, MITATMP.STATION . . .
- [#DEF*]  #DEFINE . . .

SEE ALSO
For details on the functions for finding in structures, see Sections E4.3, "Finding and Replacing in Structures during Program Editing" and E4.4, "Finding and Replacing in Structures in the Project."
### Searching for Hidden Devices of Tag Names and Addresses

In the following instruction, while only "D1" is displayed in the program edit screen, devices D1 to D16 are actually used.

```
[BSET 100 D1 16]
```

Thus, devices D2 to D16 are used but not displayed in the program edit screen. A device that is not displayed in the edit screen but actually used is called "hidden device."

When the [Tag Name/Address] checkbox is selected for the search object and [Displayed Devices] is also specified, search operations ignore hidden devices. When [Used Devices] is specified, hidden devices are included in the search objects.

**TIP**

[Example]

```
[BSET 100 D1 16]
```

This instruction returns different search results depending on the specified search object setting, as shown below.

Displayed Devices: The search will only match the first device D1 displayed for the instruction parameter.

Used Devices: The number of points is 16 so it will match search string D1 to D16.

```
[BSET 100 D1 B1]
```

In this instruction, the number of points is specified by a device, which means that its range is determined during runtime. Hence, it will only match search string D1, regardless of whether [Displayed Devices] or [Used Devices] is specified.

### Specifying Tag Name/Address Range for Searching

If the search string is specified as a tag name or an address, the range of addresses searched for starts from the search string and extends for the specified number of Search Points. The number of Search Points can be specified with either [Displayed Devices] or [Used Devices].

**TIP**

[Example]

```
[BSET 100 D20 20]
```

For example, if the specified Search String is D030 and the specified value for Search Points is 3, this instruction returns different search results depending on the specified search object setting, as shown below.

Displayed Devices: The software searches for D30 to D32. These devices are used, but they are hidden devices not displayed for the instruction parameter. Therefore, they are not displayed in search results.

Used Devices: The software searches for D30 to D32. This example is displayed in the search result because the instruction uses devices D20 to D39.
CAUTION

If a search object is found in a hidden circuit, the circuit is expanded.

Searching over the Entire Project

To find devices/comments over the entire project, use the Search entire project tab in the Find dialog box.

- A Enter the search string.
  You may use the wildcard character ("*") in the search string. You can also select a string from the past search history.

- B Click this button to start searching.

- C Specify the number of points to search.
  You may specify up to 2,048 points. Usually, 1 point is specified.

- D Specify the search condition.
  Select [Blocks with Component Definitions] to search blocks currently registered in the component definition.
  Select [All Blocks in Project] to search all blocks in the project.
  Select [All Macros in Project] to search all macros in the project.
  Select [All Blocks/Macros in Project] to search all blocks and macros in the project.

- E Select the search object.
  Turn on the [Tag Name/Address] checkbox to include tag names and addresses in the search objects. [Displayed Devices] specifies the first device displayed in the circuit edit screen as the search object and performs a simple match search of the string. [Used Devices] includes in the search objects the used devices not displayed in the circuit edit screen ("hidden devices").
  Turn on the [Circuit Comment] checkbox to include circuit comments in the search objects.
Turn on the [Subcomment] checkbox to include subcomments in the search objects.

Turn on the [Block/Macro Name] checkbox to include block names and macro names in the search objects.

Turn on the [Label] checkbox to include labels in the search objects.

Turn on the [I/O Comment] checkbox to include I/O comments in the search objects.

SEE ALSO
For details on hidden devices, see "Searching for Hidden Devices of Tag Names and Addresses" in this section.

- F Turn on this button to enable jumping from a search result to the ladder program edit window.
- G Turn on this button to enable jumping from a search result to the Block Monitor window.
- H Displays details of search results.

TIP
- You can perform repeat a previous search in the downward direction, starting from cursor position, even after closing the Find Instruction Dialog Box by selecting [Find]–[Find Next]–[Find Next] from the menu or pressing the [F3] key. Pressing the [Shift]+[F3] keys repeats the previous search in the upward direction, starting from the current cursor position.
- The default jump destination is [Jump to Offline] (if the front window is the Edit Block or Edit Macro window) or [Jump to Online] (if it is the program monitor window).
- When the front window is other than the Edit Block or Edit Macro window or the program monitor window, the default jump destination is [Jump to Online] (if WideField3 is connected to the CPU) or [Jump to Offline] (if it is not connected).

CAUTION
When [Find in Project] is specified as a search condition, blocks and macros in the project folder are searched. Therefore, blocks referring to the common library or another project are not searched.

To search over the entire project, use the following procedure.

◆ Procedure ◆

(1) Select [Find]–[Find in Project] from the menu bar.
⇒ The Find in dialog box opens.

TIP
You can also open the Find in dialog box by selecting [Project] - [Find in Project] from the menu bar.
(2) Click the Search entire project tab.

**TIP**
If the Find in Project function is started in the program edit screen or the tag name definition edit screen, the parameter where the cursor is positioned is set as the search string.

(3) Enter a search string.
(4) Specify the search condition.
(5) Specify the search object.
(6) To specify the number of search points, enter a value in [Search Points].

**TIP**
You can only specify the number of search points when the [Tag Name/Address] checkbox is selected for the search object.

(7) Click [Find All].
⇒ Search results are listed at the bottom area in the Find in dialog box.

**TIP**
The search results display up to 1024 rows of block name, instruction number and instruction.

**TIP**
If you changed the CPU type in the project, even if you select [Tag Name/Address] for the search object, devices not supported for the CPU type after changed cannot be found. To find these devices, select some other search object options than [Tag Name/Address].

**TIP**
The details of search results are displayed in the Search entire project tab. By moving the cursor within the window and pressing the [Enter] key, you can continually display the search position.

Performing a search using a tag name as a search string also searches for any address allocated to the tag name. Similarly, searching for an address also searches for any tag name assigned to this address. For instance, if “SW1=X00301” is defined, then searching for “SW1” and searching for “X00301” will produce the same result.

Furthermore, if circuit data contains a base modified by a constant index, then the expanded address is also matched against the search string. For instance, the parameter “I001;3” matches both search
strings "1001" and "1004". When searching using an index register, index modified addresses are also searched.

You can use the wildcard character ("*") to search for a tag name or string constant. The following three wildcard search patterns are supported.

- "**data" : searches for tag name strings that end with "data".
- "data*" : searches for tag name strings that start with "data".
- "#ini*" : searches for constant name strings that start with "#ini".

**SEE ALSO**

For details on the functions for finding in structures in the project, see Section E4.4, "Finding and Replacing in Structures in the Project."

**CAUTION**

- You can also search for a tag name or address with index modification.
- When searching for a constant name, the search string must be prefixed by a "#" character.
- When searching for labels, the parameter area of the label instruction and the location of the label definition are displayed. For the location of the label definition, the first instruction number of the circuit of the label definition is displayed in the instruction number column and nothing is displayed in the instruction column.
- The Find in Project function searches the offline project file currently open. Online-edited changes or additions which are not yet reflected to the offline file are not searched.
E4.1.2 Finding Instruction

The Find Instruction function is used to search for instructions used in the block currently opened or in the project.

Searching in the Open Block

To search for instructions in the open block, use the Search in open block tab in the Find Instruction dialog box.

- A Select the instruction type to search for.
  Available options are: "- ()", "-/|", "-|-", "-|/|" and "Application Instruction."
- B Click this button to find next matches. If the search objects are found, the display switches to the window containing the search objects.
- C Select the search direction
  - If [From Top] is selected, the search proceeds from the first line to the last line of the block.
  - If [Upwards from Cursor] is selected, the search proceeds from the position cursor location to the beginning of the block.
  - If [Downwards from Cursor] is selected, the search proceeds from the position cursor location to the end of the block.
- D Enter the search string.
  Enter parameters for basic instructions. Input is optional.
  Enter instruction mnemonics for application instructions. You may also include parameters in the input.

TIP

To search for instructions with options, delimit the options with spaces as shown below.

| MOV L P D1 D2 | Searching with long word and pulse options. |
| MOV L D1 D2   | Searching with long word option. |

Instructions specified with options are searched even if the search string is specified without options.

- E Turning on this checkbox performs loop search. The search starts from the beginning when it reaches the end.
- F Click this button to show or hide the list of details of search results. The following table shows the contents displayed in the details of search results.

Table E4.3 Details of Search Results

<table>
<thead>
<tr>
<th>Line Number</th>
<th>Instruction Number</th>
<th>Search Result</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The line number of the position where the instruction is found is displayed.</td>
<td>The instruction number for the position where the instruction is found is displayed.</td>
</tr>
</tbody>
</table>

TIP

If you double-click a line in the details of search results, the location where the selected device is used is displayed in another window and the cursor is positioned at the selected device in the window.

- G Displays details of search results. Click [Details] to show or hide the details.
To find an instruction in the currently opened block, use the following procedure.

◆ Procedure ◆

(1) Select [Find]–[Find Instruction] from the menu.
⇒ The Find Instruction dialog box is displayed.

(2) Select the instruction type from the dropped down list.

(3) Enter the string to search for in the Instruction String text box.

TIP
To search for an instruction with options, include the options in the Instruction String text box.

(4) Select the search direction.

(5) Turn on the Loop Search checkbox to perform a loop search.

(6) Click [Find Next]
⇒ The search begins in the selected search direction. If the search object is found, the display switches to the window containing the search object.

TIP
- You can perform repeat a previous search in the downward direction, starting from cursor position, even after closing the Find Instruction Dialog Box by selecting [Find]–[Find Next]–[Find Next] from the menu or pressing the [F3] key. Pressing the [Shift]+[F3] keys repeats the previous search in the upward direction, starting from the current cursor position.
- If the search reaches the end of the block (or the beginning of the block in the case of a search in the upward direction) without finding the search object, a dialog box is displayed to inform that the search object is not found.
- Even if the specified search string is without index modification, the search may result in a device with index modification so long as it matches the search pattern.

CAUTION

If the search object is found in a hidden circuit, the circuit is expanded.
Searching over the Entire Project

To search for instructions over the entire project, use the Search entire project tab in the Find Instruction dialog box.

![Find Instruction Dialog Box](E0401_09.VSD)

- **A** Select the instruction type to search for.
  Available options are: "- ()", "-/()", "|- |", "|- |", and "Application Instruction."

- **B** Click this button to start searching.

- **C** Enter the search string.
  Enter parameters for basic instructions. Input is optional.
  Enter instruction mnemonics for application instructions. You may also include parameters in the input.

**TIP**

To search for instructions with options, delimit the options with spaces as shown below.

- **MOV L P D1 D2** Searching with long word and pulse options.
- **MOV L D1 D2** Searching with long word option.

Instructions specified with options are searched even if the search string is specified without options.

- **D** Specify the search condition.
  Select [Blocks with Component Definitions] to search blocks currently registered in the component definition.
  Select [All Blocks in Project] to search all blocks in the project.
  Select [All Macros in Project] to search all macros in the project.
  Select [All Blocks/Macros in Project] to search all blocks and macros in the project.

- **E** Turn on this button to enable jumping from a search result to the Block Monitor window.

- **F** Turn on this button to enable jumping from a search result to the ladder program edit window.
- **G** Displays details of search results. The following table shows the contents displayed in the details of search results.

<table>
<thead>
<tr>
<th>Block Name</th>
<th>Line Number</th>
<th>Instruction Number</th>
<th>Search Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>The block name of the block that contains the found instruction is displayed.</td>
<td>The line number of the position where the instruction is found is displayed.</td>
<td>The instruction number for the position where the instruction is found is displayed.</td>
<td>The entire instruction that contains the found instruction is displayed.</td>
</tr>
</tbody>
</table>

**TIP**

If you double-click a line in the details of search results, the location where the selected device is used is displayed in another window and the cursor is positioned at the selected device in the window.

**CAUTION**

When [Find in Project] is specified as a search condition, blocks and macros in the project folder are searched. Therefore, blocks referring to the common library or another project are not searched.

To find an instruction over the entire project, use the following procedure.

**◆ Procedure ◆**

(1) Select [Find]–[Find Instruction in Project] from the menu bar.

⇒ The Find Instruction dialog box opens.

**TIP**

You can also open the Find Instruction dialog box by selecting [Project] - [Find Instruction in Project] from the menu bar.

**TIP**

When you start an operation for searching over the entire project in the program edit screen, the instruction at the cursor position is specified for the search string.

(2) Click the Search entire project tab.

⇒ The Search entire project tab is displayed.

![Find Instruction dialog box](E0401_10.VSD)
(3) Select the instruction type to find. To search for an application instruction, select [Application Instruction].

(4) To search for an application instruction, enter an instruction string.

(5) Specify the search condition.

(6) Click [Find All].
Search results are listed at the bottom area in the Find Instruction dialog box.

TIP
By moving the cursor within the screen for searching for instructions in the project and pressing the [Enter] key, you can continually display the search position.

Performing a search using a tag name as a search string also searches for any address allocated to the tag name. Similarly, searching for an address also searches for any tag name assigned to this address. For instance, if “SW1=X00301” is defined, then searching for “SW1” and searching for “X00301” will produce the same result.

Furthermore, if circuit data contains a base modified by a constant index, then the expanded address is also matched against the search string. For instance, the parameter “I001;3” matches both search strings “I001” and “I004”.

You can directly jump to a relevant circuit from the results of find instruction in project window. Select one of the following option buttons as the jump destination.
- [Jump to Offline] option button
  Jumps to a ladder program edit window.
- [Jump to Online] option button
  Jumps to a Block Monitor window.

CAUTION
- You may also search for an instruction string with index modification.
- The Find Instruction in Project function searches the offline project file currently open. Online-edited changes or additions which are not yet reflected to the offline file are not searched.
- If the [Jump to Online] option button is turned on but the search object does not exist in the monitored block, the block cannot be correctly displayed.
E4.1.3 Finding the Next Candidate

The last search condition specified for the Find Device/Comment and Find Instruction function is saved even after the dialog box is closed. To search downwards for the next candidate starting from the cursor position, select [Find]–[Find Next]–[Find Next] from the menu or press the [F3] key. To repeat the search in the upward direction, press the [Shift]+[F3] keys instead.
E4.1.4 Finding Tag Name Definitions

You can search for tag names and comments in tag name definition windows. The relevant procedure is given below.

◆ Procedure ◆

(1) Check that the tag name definition window to be searched is the active window.
(2) Select [Find]–[Find] from the menu.
⇒ The Find dialog box is displayed.

(3) Enter the search string and select the search object and the search direction.
(4) Click [Find Next].
⇒ Search begins. The cursor moves to the search object, if found. If the search object is not found, an error is displayed.

TIP
You can modify a tag name definition, even with the Find dialog box displayed.

(5) Click [Cancel] or [Close] on the Find dialog box.
⇒ The Find dialog box closes.

TIP
Even after closing a Find dialog box, you can search downwards for the next candidate, starting from the position cursor location by selecting [Find]-[Find Next]-[Find Next] or by pressing the [F3] key. Pressing the [Shift]+[F3] keys searches upwards for the next candidate, starting from the position cursor location.
If you turn on the [Loop Search] checkbox, the search starts from the beginning when it reaches the end of the tag name definitions.
E4.1.5 Finding Constant Definitions

You can search for constant names, constant values and comments in constant definition windows.

The relevant procedure is given below.

◆ Procedure ◆

1. Check that the constant definition window to be searched is the active window.
2. Select [Find]–[Find] from the menu.
   ⇒ The Find dialog box is displayed.
3. Enter the search string and select the search object and the search direction.
4. Click [Find Next].
   ⇒ Search begins. The cursor moves to the search object, if it is found. If the search object is not found, an error is displayed.

   TIP
   You can modify a constant definition, even with the Find dialog box displayed.

5. Click [Cancel] or the button for closing the Find dialog box.
   ⇒ The Find dialog box closes.

   TIP
   Even after closing a Find dialog box, you can search downwards for the next candidate, starting from the position cursor location by selecting [Find]–[Find Next]–[Find Next] from the menu bar or by pressing the [F3] key.
   Pressing the [Shift]+[F3] keys searches upwards for the next candidate, starting from the position cursor location.
   If you turn on the [Loop Search] checkbox, the search starts from the beginning when it reaches the end of the constant definitions.
E4.1.6 Jumps

The jump function switches the display of the window to a specified line in a program. There are fifteen jump functions. These can be selected from the menu by selecting [Find]–[Jump] followed by the respective jump destination type.

**TIP**
The jump function works within the blocks and macros registered in the component definition.

- **Sequential Search (In<-->Out)**
  Searches for input and output positions of the selected relay devices in the program. When the position cursor is at an input contact, the cursor jumps to an output position where the same address is used. The display of the window is changed accordingly.

- **Set As Sequential Search Start Point**
  Sets the position of the selected device in the program as the start point of the sequential search.

- **Return to Sequential Search Start Point**
  The window displays the specified start point of the sequential search and the position cursor moves to the start position of the sequential search.

- **Back One**
  The window displays the previous position of the sequential search and the position cursor moves to the previous position.

- **Next Input/Next Output**
  Searches the selected devices in the program for the next input or output position where the address at the cursor position is used. The position cursor moves to the found position. The display of the window is changed accordingly.

- **Next Address**
  Searches the selected devices in the program for the next position where the address at the cursor position is used. The position cursor moves to the found position. The display of the window is changed accordingly.

- **Jumping to the First/Last Line**
  Select [Top]/[Bottom] to jump to the first/last line. The window displays the first/last line of the program, and the position cursor moves to the first ladder program component.

- **Jumping to a Specific Line**
  Select [Line] to jump to a specific line. The window displays the specified line at the top of the window, and the position cursor moves to the first ladder program component of the specified line.

- **Jumping to a Specific Instruction Number**
  Select [Instruction Number]. The window displays the circuit containing the specified instruction number at the top of the window, and the position cursor moves to the first ladder diagram element.

Selecting [Line] or [Instruction Number] opens the Jump dialog box. Enter the line number or the instruction number to jump to, and click [OK].
Figure E4.6  Jump Dialog Box (for Jumping to Specific Line)

- Jumping to the Beginning/End of a Line
  Select [Line Start]/[Line End]. The position cursor moves to the
  leftmost/rightmost position on the line.
- Jumping between Circuit Comments
  Select [Next Circuit Comment Line]/[Previous Circuit Comment Line]. The
  position cursor moves to the next/previous circuit comment.
E4.1.7 Jump to Tag Name Definition

The Jump to Tag Name Definition function displays defined tag name definitions for the device where the cursor is placed within the program.

The relevant procedure is given below.

◆ Procedure ◆

(1) Move the cursor to the device position to which to jump.

(2) Select [Find]-[Jump to Tag Name Definition].

⇒ The tag name definition edit screen opens, and the defined tag name is displayed as outlined.

TIP

If the target device has no defined tag name definitions, only the target name definition edit screen opens.

E4.1.8 Jump to Circuit Comment (To Be Supported)

WideField3 R2 or earlier does not support functions for jumping to circuit comments.
E4.1.9 Obsolete Device List

You can display device usage status in a list over the entire project or in blocks. This is a function that displays the status in the old style prior to the introduction of the device list function.

**TIP**

You can also check device usage status using the device list function.

**SEE ALSO**

For details on the device list, see Chapter E5, "Device List."

### Displaying Device Usage Status in Blocks

You can search for devices used in blocks. The device usage status displays the status based on converted circuits. Devices used in circuits containing errors are not displayed.

The relevant procedure is given below.

**Procedure**

1. Select [Find]–[Obsolete Device List] from the menu.
   - The Obsolete Device List dialog box is displayed.
2. Select the device to be displayed from the [First Device] drop down list.
3. Click [Display].
   - The device usage status is displayed, starting from the specified first device. 256 devices are displayed at a time.
4. If there are more than 256 devices, clicking [Next] displays the next 256 devices. Clicking [Previous] displays the previous 256 devices.
5. Click [Close].
   - The Obsolete Device List dialog box is closed.

**TIP**

Devices used in the program are prefixed by an asterisk.

The number of devices used (2 points, 4 points, 16 points, 32 points) is displayed to the right of the device address. Two points and four points are used for word device in a long word/double-long-word instruction, while 16 points and 32 points are used for bit device in word or long word instructions. When one point is used, it is not displayed. Devices used in special instructions that affect multiple devices, and devices modified by an index are displayed with a "Multi-point" remark on the right of the device address, instead of the number of used points. For a device displayed as "Multi-point," its following devices may be used.

Devices allocated to structure names and structure member names are all displayed as being used.

Devices modified by a constant index are displayed according to the usage status of the actual addresses after modification.
CAUTION

Only converted devices in blocks being edited are displayed.
To check the device usage status for the entire project, select [Project]–[Obsolete Device List].

Displaying Device Usage Status in the Project

You can display a list of devices used in all blocks registered in the component definition. The list does not include devices used in macros.
The relevant procedure is given below.

◆ Procedure ◆

(1) Select [Project] - [Obsolete Device List] from the menu bar.
⇒ The Obsolete Device List dialog box is displayed.

TIP
The project device usage status is displayed for saved blocks. If you invoke the function with a ladder program edit window (tag name definition window) open after having made changes in the window, a confirmation dialog window will be displayed.

(2) Select an address from the [First Device] drop-down list and click [Display].
⇒ The display of the devices changes.
(3) Click [Close].
⇒ The Obsolete Device List dialog box is closed.

TIP
If the blocks are registered in the component definition, even if the blocks refer to the common library or another project, the device usage status is displayed for the blocks. The device usage status is not displayed for the blocks not registered in the component definition.
CAUTION

- This function displays devices used in blocks registered in the executable program components definition and saved to files. It does not include devices in circuits containing conversion error for blocks saved with errors.

- Devices within the range set for local devices are displayed in gray. Devices displayed with a mark within the gray display area indicate global devices used within the local device range. An error or warning will be generated in this case depending on the program checking criteria set on the Program Syntax Check tab of the Set up Environment dialog box displayed by selecting [Tools]–[Set up Environment] from the menu.
E4.1.10 Find Circuit Comment-out

You can use the Find Circuit Comment-out function to search open blocks or an open project for circuit comment-out parts.

Searching in the Open Block

To find comment-out parts within the block currently opened, use the Search in open block tab in the Find Circuit Comment-out dialog box.

Figure E4.7 Find Circuit Comment-out Dialog Box

- A Select the comment-out type(s) to search for.
  - Turn on the [Forced On] checkbox to include activated comment-out parts in the search objects.
  - Turn on the [Forced Off] checkbox to include inactivated comment-out parts in the search objects.
  - Turn on the [Not Executed] checkbox to include non-execution comment-out parts in the search objects.
- B Click this button to find next matches. If the search objects are found, the display switches to the window containing the search objects.
- C Select the search direction.
  - If [From Top] is selected, search proceeds from the first line to the last line of the block.
  - If [Upwards from Cursor] is selected, search proceeds from the position cursor location to the beginning of the block.
  - If [Downwards from Cursor] is selected, search proceeds from the position cursor location to the last line of the block.
- D Click this button to remove all the specified comment-out types within the block. You can click this button when details of the search results are displayed.
- E Turning on this checkbox performs loop search. The search starts from the beginning when it reaches the end of the block.
- F Click this button to show or hide the list of details of search results. The following table shows the contents displayed in the details of search results.

<table>
<thead>
<tr>
<th>Table E4.5 Details of Search Results</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Comment-out Type</strong></td>
</tr>
<tr>
<td>The comment-out type of the found instruction is displayed.</td>
</tr>
</tbody>
</table>

**TIP**

If you double-click a line in the details of search results, the location where the selected device is used is displayed in another window and the cursor is positioned at the selected device in the window.

- G Displays details of search results. Click [Details] to show or hide the details.

Use the following procedure to search the currently opened block for comment-out parts.

**Procedure**

1. Select [Find]-[Find Circuit Comment-out] from the menu bar.
   ⇒ The Find Circuit Comment-out dialog box is displayed.
2. In [Comment-out type], select one or more objects to search for.
3. Select the direction for the search.
4. Turn on the [Loop Search] checkbox to perform a loop search.
5. Click [Find Next].
   ⇒ The search begins in the selected search direction. If the search object is found, the display switches to the window containing the search object.

**TIP**

- You can repeat the previous search in the downward direction starting from the current cursor position even after closing the Find Circuit Comment-out dialog box by selecting [Find]-[Find Next]-[Find Next] from the menu or pressing the [F3] key. Pressing the [Shift]+[F3] keys repeats the previous search in the upward direction.
- If the search reaches the end of the block (or the beginning of the block in the case of a search in the upward direction) without finding the search object, a dialog box is displayed indicating that the search object is not found.

**CAUTION**

The circuit is expanded if the search object is found in a hidden circuit.
Canceling All Circuit Comment-out in Open Block

Use the following procedure to cancel all circuit commenting-out parts in the currently open block.

⚠️ CAUTION

- You cannot perform a collective cancellation of circuit commenting-out in the program monitor window during online editing.
- You cannot restore the previous state after you cancel all circuit commenting-out.

◆ Procedure ◆

1. Open the Edit Block or Edit Macro window.
2. Select [Find]-[Find Circuit Comment-out] from the menu bar.
   ⇒ The Search in open block tab in the Find Circuit Comment-out dialog box opens.
3. From [Comment-out Type], select circuit comment-out type(s) you want to cancel collectively, and click [Details].
   ⇒ A list of search results containing the circuit comment-out in the open block is displayed.

Tips
The search result list shows comment-out types found in the open block, regardless of the search direction and loop search settings.
(4) Click [Remove All].

⇒ A dialog box is displayed to confirm whether to remove all the circuit comment-out type(s). Click [Yes] to remove the selected comment-out type(s). After the operation, an empty search results list is displayed.

**TIP**
The [Remove All] button is available when detailed search results are shown.

After detailed search results are displayed, changing any search comment-out type inactivates the [Remove All] button.
Searching throughout the Entire Project

Use the Search entire project tab in the Find Circuit Comment-out dialog box to find comment-out parts throughout the entire project.

- **A** Specify the search condition.
  - Select [Blocks with Component Definitions] to search blocks currently registered in the component definition.
  - Select [All Blocks in Project] to search all blocks in the project.
  - Select [All Macros in Project] to search all macros in the project.
  - Select [All Blocks/Macros in Project] to search all blocks and macros in the project.
- **B** Click this button to start searching.
- **C** Select the comment-out type(s) to search for.
  - Turn on the [Forced On] checkbox to include instructions with activated comment-out parts in the search objects.
  - Turn on the [Forced Off] checkbox to include instructions with inactivated comment-out parts in the search objects.
  - Turn on the [Not Executed] checkbox to include instructions and circuits with non-execution comment-out parts in the search objects.
- **D** Click this button to remove all the specified comment-out types within the range of the search conditions. You can click this button when the search results are displayed.
- **E** Turn on this button to enable jumping from a search result to the ladder program edit window.
- **F** Turn on this button to enable jumping from a search result to the Block Monitor window.
- **G** Displays details of search results. The following table shows the contents
displayed in the details of search results.

Table E4.6 Details of Search Results

<table>
<thead>
<tr>
<th>Block Name</th>
<th>Comment-out Type</th>
<th>Line Number</th>
<th>Instruction Number</th>
<th>Search Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>The block name of the block that contains the found instruction is displayed.</td>
<td>The comment-out type of the matched instruction is displayed.</td>
<td>The line number of the position where the object is found is displayed.</td>
<td>The instruction number for the position where the object is found is displayed.</td>
<td>The entire instruction that contains the matched instruction is displayed.</td>
</tr>
</tbody>
</table>

**TIP**

If you double-click a line in the details of search results, the location where the selected device is used is displayed in another window and the cursor is positioned at the selected device in the window.

---

**CAUTION**

When [Find in Project] is specified as a search condition, blocks and macros in the project folder are searched. Therefore, blocks referring to the common library or another project are not searched.

To find comment-out parts over the entire project, use the following procedure.

◆ **Procedure◆**

1. Select **[Find]- [Find Circuit Comment-out in Project]** from the menu bar.
   ⇒ The Find Circuit Comment-out dialog box is displayed.

   **TIP**
   You can also open the Find Circuit Comment-out dialog box by selecting **[Project] - [Find Circuit Comment-out in Project]** from the menu bar.

2. Click the Search entire project tab.
   ⇒ The Search entire project tab is displayed.

3. Specify the search condition.
4. In [Comment-out Type], select one or more objects to search for.
5. Click **[Find All]**.
   ⇒ Search results are listed at the bottom area in the Find Circuit Comment-out dialog box.
TIP
You can continually display the search position by moving the cursor within the screen for searching for circuit comment-out in the project and pressing the [Enter] key.

You can directly jump to a relevant circuit from the search results window of circuit comment-out in the project. Select one of the following option buttons as the jump destination.
- [Jump to Offline] option button
  Jumps to a ladder program edit window.
- [Jump to Online] option button
  Jumps to a Block Monitor window.

CAUTION
- The Find Circuit Comment-out in Project function searches the offline project file currently open. Online-edited changes or additions that are not yet reflected in the offline file are not searched.
- If the [Jump to Online] option button is turned on but the search object does not exist in the monitored block, the block cannot be correctly displayed.
Canceling All Circuit Comment-Out for Entire Project

Use the following procedure to cancel all circuit comment-out for a currently open project.

**CAUTION**

- You cannot perform a collective cancellation of circuit commenting-out in the program monitor window during online editing.
- You cannot restore the previous state after you cancel all circuit commenting-out.

◆ Procedure ◆

(1) Close all Edit Block and Edit Macro windows.

(2) Select [Find]- [Find Circuit Comment-out in Project] from the menu bar.
⇒ The Search entire project tab of the Find Circuit Comment-out dialog box opens.

(3) Select [Search Conditions] and circuit comment-out type(s) you want to cancel collectively from [Comment-out Type], and click [Find All].
⇒ A list of search results containing the circuit comment-out in the entire project is displayed.

(4) Click [Remove All].

**TIP**

The [Remove All] button is available when a list of search results is shown.

After a list of search results is displayed, changing any search comment-out type inactivates the [Remove All] button.
A dialog box is displayed to confirm whether to remove all the circuit comment-out type(s). Click [Yes] to remove the selected comment-out type(s). During a collective removal operation, the Remove Circuit Comment-out in Project - Progress dialog box is displayed. When the selected comment-out type(s) are removed, a completion dialog box is displayed.

**TIP**

To abort the operation, click [Cancel] in the Remove Circuit Comment-out in Project - Progress dialog box.

(5) Click [OK].
⇒ The completion dialog box closes.

(6) Click [Close] in the Remove Circuit Comment-out in Project - Progress dialog box.
⇒ The Remove Circuit Comment-out in Project - Progress dialog box closes.
E4.2 Replace

The replace function replaces a specified string with another string.
Using the replace function, you can replace any of the following three objects:
- Tag name/Address
- Constant name
- Structure

You can specify one of the following two types of ranges for the replace function:
- Within the block currently opened
- Over the entire project

**CAUTION**

- The Replace operation is not available in address display mode. Always exit from address display mode before performing a Replace operation. To exit from address display mode, select [View]–[Specify Display Mode]-[Display Address] from the menu again or press the [Ctrl]+[A] keys. (The menu item operates like a toggle)
- Replace operations are disallowed during circuit monitoring.
- You cannot replace contents in balloon comments and balloon monitors.

**TIP**

If you specify index registers (V001→V002) for a Replace operation, the index modification is also replaced.
### E4.2.1 Replacing within the Open Block

To replace devices within the block currently opened, use the Replace in open block (block name) tab in the Replace dialog box.

The replace function replaces a device with a specified device. Using the replace function, you can replace one relevant device, and also you can replace up to 2,048 points of successive addresses at once.

![Replace Dialog Box](E0402_01.VSD)

**Figure E4.9 Replace Dialog Box**

- **A** Input the tag name, constant name or device address to be replaced.
- **B** Clicking this item moves the position cursor to the location of the next tag name/address/constant name to be replaced.
- **C** Input the new replacement tag name, constant name or device address.
- **D** Clicking this button replaces the tag name/address/constant name at the position cursor.
- **E** Specify the number of points to replace. Specify the replacement range, starting from the specified tag name or address. This field is disabled when replacing constant names.
- **F** Clicking this button replaces all matches.
- **G** Replaces tag names or addresses displayed in the ladder program edit window with another tag name or address. After replacement, tag names with no registered addresses become unregistered tag names and changes color. Replacements are not reflected on tag name definitions. Replacements are only done in ladder program edit windows. There are 4 types of replacement:
  - Tag Name → Tag Name Replaces a tag name/constant name with another tag name/constant name.
  - Tag Name → Address Replaces a tag name with an address.
  - Address → Tag Name Replaces an address with a tag name.
  - Address → Address Replaces an address with another address.

  The wildcard (**) character can be used when replacing tag names or constant names.

  To replace constant names or structure tag names, specify [Tag name].
- **H** Select the search direction.
- **I** Turning on this checkbox performs loop search. The search continues from the beginning when it reaches the end of the block.
CAUTION

- The Replace operation is not available in address display mode. Always exit from address display mode before performing a Replace operation.

To exit from address display mode, select [View]–[Specify Display Mode]-[Display Address] from the menu again or press the [Ctrl]+[A] keys. (The menu item operates like a toggle)

- Clicking the [Find Next], [Replace] or [Replace All] button in the Replace dialog box replaces only objects entered as tag names if the preceding search was performed using a tag name, and replaces only objects entered as addresses if the preceding search was performed using an address.

The required procedure is given below.

◆ Procedure ◆

(1) Select [Find]–[Replace] from the menu.
⇒ The Replace dialog box is displayed.

(2) Click the Replace in open block (block name) tab.

TIP
When you start a Replace operation in the program edit screen or tag name definition edit screen, the parameter at the cursor position is specified for the search string.

(3) Enter the tag name or address to be replaced.

(4) Enter the new replacement tag name or address.

(5) Specify the number of replacement points.

(6) Select the search direction.

(7) To perform loop search, turn on the [Loop Search] checkbox.

(8) Select the replacement method.

(9) Click [Find Next].
⇒ The position cursor moves to the location of the next tag name or address to be replaced.

(10) Click [Replace].
⇒ The tag name/address at the position cursor is replaced.

(11) To continue with the next replacement, click [Find Next]. To replace all occurrences, click [Replace All].
CAUTION

Replacements in circuits are not reflected to tag name definitions. To reflect replacements to tag name definitions, use the Read Circuits function in tag name definition.

In ladder program edit, it is assumed that address assignments to tag names are not confirmed. Accordingly, when replacing tag names, only devices entered as tag names are replaced and when replacing addresses, only devices entered as addresses are replaced.

SEE ALSO

For details on the “Read Circuits” function for tag name definition, see Section E2.2.6, “Read Circuits Function”.

Restrictions on the Search and Replacement String

If an invalid search string is entered, an error message is displayed. A search string that begins with a numerical digit causes an invalid replacement string error.

The wildcard ("*") character can be used in the replacement strings but only for tag names and constant names. Replacing addresses containing the wildcard character ("*") is not allowed. The following restrictions apply to the use of the wildcard character.

- No more than one wildcard character may be used (e.g. "SW*AB*").
- The wildcard character cannot be in the middle of the string (e.g. "SW*WF").
- The wildcard character cannot be used together with index modification (e.g. "*SW;V01").
- Constant names must be prefixed by the '#' character.

When an address name is specified as the string to be replaced, both addresses used with direct specification and addresses used with indirect specification are replaced.

If an indirect address specification (an address prefixed with "@") is specified as the string to be replaced, only indirect address specifications are replaced.

The following restrictions apply to replacement of structure tag names.

- No more than 1 period (".") character may be used.
- Input is not allowed if the period (".") character is coded in an invalid position.
- A structure name must not exceed 7 characters (2 characters for the structure pointer (Q)) and a member name must not exceed 8 characters.
- Replacement of array types is not allowed.
- Structures cannot be entered with indirect specification.
- For Q devices, only Q1 and Q2 can be specified.

TIP

The following are some examples of search strings and their coverage.

```
- [SW*]     SW1, SW2, SW3, SW4 ...
- [*SW]     WF_SW, YO_SW, KO_SW, GA_SW, WA_SW ...
- [D0001]   D0001, @D0001 ...
- [@D0001]  @D0001 ...
- [MITAKA.STATION] MITAKA.STATION, MITAKA[10].STATION ...
- [MITA*]   MITAKA, MITAKA.STATION, MITAKA.STORE, MITAKA[10].STATION, MITATMP, MITATMP.STATION ...
- [*STATION] MITAKA.STATION, MITAKA[10].STATION, MITATMP.STATION ...
- [#DEF*]   #DEFINE ...
```
# Replacing using Wildcard (*)

The following table illustrates some examples of replacing "SW*" with "Switch*".

<table>
<thead>
<tr>
<th>Before Replacement</th>
<th>After Replacement</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>SW</td>
<td>Switch</td>
<td></td>
</tr>
<tr>
<td>SW1</td>
<td>Switch1</td>
<td></td>
</tr>
<tr>
<td>SW_AAA</td>
<td>Switch_AAA</td>
<td></td>
</tr>
<tr>
<td>Switch</td>
<td>×</td>
<td>Not replaced.</td>
</tr>
</tbody>
</table>

The following table illustrates some examples of replacing "SW;V0001" with "Switch;V0001" (with index modification).

<table>
<thead>
<tr>
<th>Before Replacement</th>
<th>After Replacement</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>SW;V0001</td>
<td>Switch;V0001</td>
<td></td>
</tr>
<tr>
<td>SW1;V0001</td>
<td>×</td>
<td>Not replaced.</td>
</tr>
<tr>
<td>SW_AAA;V0001</td>
<td>×</td>
<td>Not replaced.</td>
</tr>
<tr>
<td>Switch</td>
<td>×</td>
<td>Not replaced.</td>
</tr>
</tbody>
</table>

**CAUTION**

Deleting an index attached to a tag name during replacement is not allowed.

**SEE ALSO**

**E4.2.2 Replacing over an Entire Project**

To replace devices over the entire project, use the Replace in entire project tab in the Replace dialog box.

---

**CAUTION**

The function for replacing over the entire project cannot be used in blocks that refer to the common library or other projects.

---

**Figure E4.10 Replace Dialog Box**

- **A** Input the tag name, constant name or device address to be replaced.
- **B** Click this button to start a Replace operation.
- **C** Input the new replacement tag name, constant name or device address.
- **D** Specify the number of points to replace. Specify the replacement range, starting from the specified tag name or address. This field is disabled when replacing constant names.
- **E** Select the combination of replacement objects (tag name, address, and structure) for old and new devices.
  - To replace constant names, select [Tag Name→Tag Name].
- **F** Specify blocks to be replaced for the replacement range.
  - Turn on the [All Blocks] option button to perform replacement over all blocks registered in the executable program components definition.
  - Turn on the [Specified Block] option button to perform replacement over selected blocks. Click the [Browse] button to select blocks to be replaced.
- **G** Turn on this checkbox to also perform replacement in the tag name definition file. The replacement is performed on the common tag name definition and the tag name definition of the selected blocks in the replacement range. This checkbox is enabled only if any of the following option buttons of the [Replacement Method] is turned on: [Structure→Structure], [Address→Address] or [Tag Name→Tag Name].
To perform a Replace operation over the entire project, use the following procedure.

◆ Procedure ◆

(1) Close all blocks being edited.
(2) Select [Find]–[Replace in Project] from the menu bar.
⇒ The Replace dialog box opens.

TIP
You can also open the Replace dialog box by selecting [Project] - [Replace in Project] from the menu bar.

(3) Click the Replace in entire project tab.
(4) Set the conditions for the replacement.
(5) Click [OK].
⇒ Replacement begins and a progress replacement progress dialog box is displayed.

TIP
To abort replacement at any time, click [Cancel].

Steps (4) and (5)

After the replacement is completed, a completion dialog box is displayed.

(6) Click [OK].
⇒ The completion dialog box closes.
(7) Click [Close] on the project replacement progress dialog box.
⇒ The progress dialog box closes.

TIP
- The results of the replacement operation is stored in a log file called “Replace.log”. You can check the results by opening the text file using “Notepad” or any other text editor.
- You cannot specify replacement for local devices or devices with index modification.
- When performing replacement using an index register (V001 → V002) as a search string, an index modifier may also be replaced if it matches the search string.
CAUTION

- You cannot use the wildcard character ("*") in the string to be replaced.
- If an error is encountered during replacement, replacement is not performed for the block.

SEE ALSO
For details on the functions for replacing in structures, see Sections E4.3, "Finding and Replacing in Structures during Program Editing" and E4.4, "Finding and Replacing in Structures in the Project."

Replace in Project – Progress Dialog Box

Displays the progress of a replacement operation. The following messages may be displayed in the Replace in Project – Progress dialog box.

Table E4.9 Replacement Messages

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N items replaced.</td>
<td>N instances of the specified device has been replaced successfully.</td>
</tr>
<tr>
<td>Error encountered in circuit change.</td>
<td>An error was encountered during replacement and the replacement operation was aborted. Replacement was not performed on the specified blocks.</td>
</tr>
<tr>
<td>Error encountered in changing tag name definition.</td>
<td>An error was encountered during replacement and the replacement operation was aborted. Replacement was not performed on the specified blocks.</td>
</tr>
<tr>
<td>Only tag name is changed because there is no change in circuit.</td>
<td>The specified device is not used in circuits so no replacement was performed. However, replacement was performed on tag name definitions.</td>
</tr>
<tr>
<td>Tag name definition changed.</td>
<td>Tag name definition has been replaced successfully.</td>
</tr>
<tr>
<td>Tag name definition not changed.</td>
<td>The tag name was not defined for the specified device and hence no changes were made.</td>
</tr>
<tr>
<td>Replacement is terminated.</td>
<td>Replacement processing was aborted because a user selected the [Cancel] button.</td>
</tr>
<tr>
<td>File not found.</td>
<td>There is no block file defined in the components definition.</td>
</tr>
<tr>
<td>Cannot change block saved with error.</td>
<td>The Project Replace function cannot be used for blocks saved with errors.</td>
</tr>
</tbody>
</table>

Block List Dialog Box

Use the Block List dialog box to specify the blocks for replacement.

To display the Block List dialog box, turn on the [Specified Block] option button on the Project Replace dialog box and click [Browse].

Figure E4.11 Block List Dialog Box

TIP
During a replacement operation over the entire project, the replacement is not performed in macro blocks.
E4.3 Finding and Replacing in Structures during Program Editing

This section describes how to find and replace structure members and structure names. You can find or replace structure members and structure names using the Find Device/Comment or Replace Device/Comment function.

In finding and replacing structure members and structure names, structure members and structure names are handled as usual tag names. In addition to the descriptions in this section, all other descriptions for finding and replacing tag names are applied to finding and replacing in structures.

SEE ALSO
For details on how to find and replace tag names, see Sections E4.1, "Find" and E4.2, "Replace."

E4.3.1 Finding in Structures

This section describes how to find structure members and structure names during program editing.

To find structure members and structure names during program editing, you can use the Find Device/Comment function in the Search in open block tab.

■ Entering the Search String

Select [Find] - [Find] from the menu bar to open the Find dialog box. Turn on the [Tag Name/Address] radio button in [Search Object] and then enter a member name (in "structure.member" format) or structure name in the [Search String] text box.

■ Using the Wildcard

The use of wildcard in finding structure members is the same as for tag names.

Table E4.10  Examples of Uses of Wildcard

<table>
<thead>
<tr>
<th>Wildcard Entered for Find</th>
<th>Examples of Matches</th>
</tr>
</thead>
<tbody>
<tr>
<td>MITA*</td>
<td>MITAKA</td>
</tr>
<tr>
<td></td>
<td>MITAKA.STATION</td>
</tr>
<tr>
<td></td>
<td>MITAKA.STORE</td>
</tr>
<tr>
<td></td>
<td>MITAKA[10].STATION</td>
</tr>
<tr>
<td>MITAKA.*</td>
<td>MITAKA.STATION</td>
</tr>
<tr>
<td></td>
<td>MITAKA.STORE</td>
</tr>
<tr>
<td></td>
<td>MITAKA[10].STATION</td>
</tr>
<tr>
<td>*.STATION</td>
<td>MITAKA.STATION</td>
</tr>
<tr>
<td></td>
<td>MITAKA[10].STATION</td>
</tr>
<tr>
<td></td>
<td>MITATMP.STATION</td>
</tr>
</tbody>
</table>

■ Finding Arrays

- You can include an array index when entering the search string for a find operation.
- When searching for a structure member, arrays are considered in the search.

(For instance, if you enter the search string as “KOZO.rel1”, then an instruction parameter whose device range includes the address allocated to “KOZO.rel1” will return a match. Similarly, if you enter the search string as “KOZO[1].rel1”, then all instruction parameters whose device range include the address allocated to “KOZO[1].rel1” will return a match)
Replacing in Structures

This section describes how to replace structure members and structure names.
To replace structure members and structure names, you can use the Replace Device/Comment function in the Replace in open block (block name) tab.

■ Entering the Replace String

Select [Find] - [Replace] from the menu bar to display the Replace dialog box, and open the Replace in open block (block name) tab.
Turn on the [Tag Name → Tag Name] or [Tag Name → Address] radio button in [Replace] and then enter a member name (in "structure.member" format) or structure name in the [Old Device] text box.

■ Using the Wildcard

The use of wildcard in replacing structure members is the same as for tag names.

Table E4.11 Examples of Uses of Wildcard

<table>
<thead>
<tr>
<th>Wildcard Entered for Replace</th>
<th>Examples of Matches</th>
</tr>
</thead>
<tbody>
<tr>
<td>MITA*</td>
<td>MITAKA</td>
</tr>
<tr>
<td></td>
<td>MITAKA.STATION</td>
</tr>
<tr>
<td></td>
<td>MITAKA.STORE</td>
</tr>
<tr>
<td></td>
<td>MITAKA[10].STATION</td>
</tr>
<tr>
<td>MITAKA.*</td>
<td>MITAKA.STATION</td>
</tr>
<tr>
<td></td>
<td>MITAKA.STORE</td>
</tr>
<tr>
<td></td>
<td>MITAKA[10].STATION</td>
</tr>
<tr>
<td>*.STATION</td>
<td>MITAKA.STATION</td>
</tr>
<tr>
<td></td>
<td>MITAKA[10].STATION</td>
</tr>
<tr>
<td></td>
<td>MITATMP.STATION</td>
</tr>
</tbody>
</table>

■ Replacing Arrays

- You cannot include an array index when entering the search string for a replace operation. Otherwise an error will occur. For example, "KOZO[V01].*" is an invalid search string for the old device.
- If you enter a structure member or structure name as the search string for a replace operation, all structure members containing the name, regardless of whether they include arrays, are included in the search. For example, if you enter "KOZO.rel1" as the search string, then "KOZO.rel1," "KOZO[1].rel1," and "KOZO[*].rel1" are possible matches.
E4.4 Finding and Replacing in Structures in the Project

This section gives an overview of finding and replacing structure members and structure names over an entire project and describes the required procedures.

E4.4.1 Overview

■ Overview of the Find Function for Structure Members and Structure Names in the Project

Searching for structure members and structure names in the project is essentially the same as searching for tag names in the project.

Select [Project] - [Find in Project] from the menu bar to open the Find dialog box.

In the Search entire project tab of the Find box, enter a structure name, structure member name, or a string of the form "structure name.*" in the [Search String] text box. Structure member parameters and structure name parameters coded in the program are searched for the given string, in a similar way to searching for tag names.

SEE ALSO
For details of project search, see Section E4.1, "Find."

The table below lists the search results for some example search strings.

Table E4.12 Search Strings and Search Results

<table>
<thead>
<tr>
<th>Program Data</th>
<th>Search String</th>
<th>Search Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>dataS (tag name)</td>
<td>data</td>
<td>data</td>
</tr>
<tr>
<td>data (structure name)</td>
<td>data.reg1</td>
<td>data.reg1</td>
</tr>
<tr>
<td>data.reg1 (structure member)</td>
<td>data.*</td>
<td>data.reg1</td>
</tr>
<tr>
<td>data.rel1 (structure member)</td>
<td>data.*</td>
<td>data.reg1</td>
</tr>
<tr>
<td>data*</td>
<td>dataS</td>
<td>data</td>
</tr>
<tr>
<td></td>
<td>data</td>
<td>data.reg1</td>
</tr>
<tr>
<td></td>
<td>data.rel1</td>
<td>data.reg1</td>
</tr>
</tbody>
</table>

■ Overview of the Replace Function for Structure Members and Structure Names in the Project

The replace function for structures in the project uses a structure name as the search string.

Select [Project] - [Replace in Project] from the menu bar to open the Replace dialog box. Turning on the [Replace in Tag Name Definitions too] checkbox in the Replace in entire project tab on the Replace dialog box also replaces structure names in tag name definitions.

SEE ALSO
For details on replacement over a project, see Section E4.2, "Replace."

■ Find and Replace for Structure Arrays

- You cannot include array representation in search or replace strings. If a specified search or replace string contains an array, an error is generated. For example, "KOZO[V01].*" is not a valid search string.
- Search or replacement of structure members or structure names searches all elements of arrays, as well as non-array structures. For example, search string “KOZO.rel1” matches “KOZO.rel1”, “KOZO[1].rel1” and the pattern “KOZO[\*].rel1”.
E4.4.2 Procedure for Finding in Structures in the Project

This section describes how to search for a structure member or structure name over a project. The relevant procedure is given below.

◆ Procedure ◆

1. Select [Project]–[Find in Project] from the menu.
   ⇒ The Find in dialog box opens.
2. Click the Search entire project tab.
   ⇒ The Search entire project screen is displayed.
3. Enter a search string in the [Search String] text box, and click [Find All].
   ⇒ The Results of Project Search window is displayed.

TIP

Searching for a structure member name also searches for any address allocated to the structure member name; similarly, searching for an address also searches for any structure member name assigned to the address. For instance, if “St1.rew1=D00102” is defined in the tag name definition, then searching for “St1.rew1” or “D00102” with [Tag Name/Address] option selected produces the same result.
E4.4.3 Procedure for Replacing in Structures in the Project

This section describes how to replace a structure name over an entire project. The relevant procedure is given below.

◆ Procedure◆

(1) Open the desired project and close all blocks and tag name definitions being edited.
(2) Select [Project]–[Replace in Project] from the menu.
   ⇒ The Replace dialog box opens.
(3) Click the Replace in entire project tab.
   ⇒ The Replace in entire project screen is displayed.
(4) Enter the structure name to be replaced and the new structure name in the [Old Value] and [New Value] text boxes respectively, turn on the [Structure→Structure] option button in the [Replacement Method] group box and click [OK].

TIP
To replace structure members, turn on the [Tag Name→Tag Name] option button in the [Replacement Method] group box.

TIP
To also replace tag names in tag name definitions, turn on the [Replace in Tag Name Definitions too] checkbox.

⇒ Replacement begins and the Replace in Project–Progress dialog box is displayed with the progress of the project replacement.
(5) Click [Close].
⇒ The Replace in Project–Progress dialog closes.
E4.5 Simple Find

You can perform searches easily using the Find Bar from a project edit window, tag name definition window, monitor window or an online edit window. Two types of Find functions are available:

- Find Device
- Find in Project

The layout of the Find bar is shown below.

![Find Bar Layout](E0405_01.VSD)

- Enter the search string. You may include the wildcard character (**) in the search string.
- B Find Device
  Clicking this icon button searches for a device in the active window, and scrolls the screen to display the search string if it is found.
- C Find in Project
  Clicking this icon button performs a search within a project and displays the search result in a Results of Project Search window.
- D Select the search direction. This applies only when performing a Find Device operation.
  If [Top] is selected, the search proceeds downwards from the top.
  If [Down] is selected, the search proceeds downwards from the cursor position to the last line.
  If [Up] is selected, the search proceeds upwards from the cursor position to the first line.

TIP

To toggle between showing and hiding the find bar, use [View]–[Toolbar].
E4.5.1 Find Device

Clicking the Find Device icon on the Find Bar searches for a device in the active window, and scrolls the screen to display the search string if it is found. To find a device, use the following procedure.

◆ Procedure ◆

(1) Enter the search string in the text field on the Find Bar.

TIP
Pressing the Tab key in a ladder program edit window or a tag name definition window automatically copies the name of the device at the position cursor to the Find Bar.

(2) Select the search direction.

(3) Click the [Find] button or press the [Enter] key.

⇒ Search begins. The screen scrolls to display the search object if it is found.

TIP
You can perform steps 1 to 3 above using the [Ctrl]+[F3] keys. Pressing [Ctrl]+[F3] in a ladder program edit window or a tag name definition window displays the next candidate using the device at the position cursor as the search string. Search proceeds in the search direction displayed in the Find Bar.

(4) To continue, press the [Enter] key.

⇒ The position cursor moves to the next candidate found.

(5) To move the window focus to the ladder program edit window or tag name definition window at the end of a search, press the [Esc] key.

TIP
- After finding a search candidate using Find Device, you can continue to search downwards for the next candidate, starting from the position cursor location by selecting [Find]–[Find Next]-[Find Next] or by pressing the [F3] key. Pressing the [Shift]+[F3] keys searches upwards for the next candidate, starting from the position cursor location.
- Up to 16 search strings from previous searches are stored for each project. These are available for selection from the leftmost drop-down list box in the Find Bar for subsequent searches.

CAUTION
If the search object is found in a hidden circuit, the circuit will be displayed.

SEE ALSO
For details of finding devices, see Section E4.1.1, “Finding Devices/Comments”.

---

IM 34M06Q16-02E  3rd Edition: Apr. 26, 2013-00
E4.5.2 Find in Project

Clicking the Find in Project icon button on the Find Bar performs a search within a project and displays the search result in a Results of Project Search window.

◆ Procedure ◆

(1) Enter the search string in the text field on the Find Bar.

TIP
Pressing the Tab key in a ladder program edit window or a tag name definition window automatically copies the name of the device at the position cursor to the Find Bar.

(2) Click the [Find in Project] button or press the [Shift]+[Enter] keys.
⇒ Search begins. If the search object is found, the results are displayed in a Results of Project Search window.

TIP
Up to 16 search strings from previous searches are stored for each project. These are available for selection from the leftmost drop-down list box in the Find Bar for subsequent searches.

SEE ALSO
For details on searching within a project, see Section E4.1.1, “Finding Devices/Comments”.

Step (2)
E4.5.3 Changing Search Conditions

Searches initiated from the Find Bar proceed with the default condition below. This condition cannot be specified when initiating a search from the Find Bar.
- Loop Search='On'

You can, however, change this default condition from 'On' to 'Off'. To do so, modify file “FindCondition.ini” in the “Fam3sys” folder where WideField3 is installed using a text editor, save the file and restart WideField3.

To Disable Loop Search

Change the value of the Loop variable in file “FindCondition.ini” from “On” to “Off”.

Before change:
Device=Tag
Num=1
Start=Down
Loop=On

After change:
Device=Tag
Num=1
Start=Down
Loop=Off

CAUTION

Changing the value of a variable other than the Loop variable or changing the Loop variable to an invalid value will affect normal operation of WideField3.
E5. Device List

The device list function displays a list of devices used and not used in a project. You can create cross references from devices displayed in a device list. You can use the device list function only when a project is open. Before performing your operations, open a project.

**CAUTION**

The device list function cannot be used online.

**TIP**

This function also lists blocks registered in the component definition among the blocks that refer to the common library or another project.

**SEE ALSO**

For details on the cross reference, see Chapter E6, "Cross Reference."

### Function Limitations for Each CPU Type

<table>
<thead>
<tr>
<th>CPU Modules</th>
<th>Limitations</th>
<th>SEE ALSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td><strong>F3SP66-4S, F3SP67-6S</strong></td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td><strong>F3SP71-4N, F3SP76-7N</strong></td>
<td>None</td>
<td></td>
</tr>
<tr>
<td><strong>F3SP71-4S, F3SP76-7S</strong></td>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>

**TIP**

This section does not describe hardware-dependant limitations for each CPU type, such as the number of available devices.

**SEE ALSO**

For details on limitations for each CPU type, refer to the user's manual for each type.
E5.1 Fundamentals of Device List

This section describes the basic operations and restrictions of the device list function.

E5.1.1 Display and Layout of the Device List Dialog Box

Displaying the Device List Dialog Box

To open the Device List dialog box, select [Project] - [Device List] from the menu bar.

Screen Layout of the Device List Dialog Box

Contents displayed in the display pane of the Device List dialog box vary depending on the screen display mode.

There are three types of screen display modes for the display pane:
- Map mode
- Details mode
- By tag name definition

In addition, the dialog box has the icon description pane.

Figure E5.1  Device List Dialog Box

TIP

Either of the following two types of icons is used:
- 16 detailed icons
- 4 simple icons

To switch from "detailed icon mode" to "simple icon mode", click [Simple Display] and restart WideField3. To switch from "simple icon mode" to "detailed icon mode", click [Details Display] and restart WideField3.
● Screen Layout

The following figure shows the screen layout of the display pane.

Figure E5.2  Display Pane in the Device List Dialog Box

- A  Select the type of devices to be displayed. You can select one of the following four options to specify the type of devices:


Selecting [Local Device] displays all local devices in the project. If you select [Local Device], you cannot select [Project] nor [Component Blocks] for the search range.

Selecting [Macro Device] displays all macro devices in the project. If you select [Macro Device], you can select only [Project] for the search range.

Selecting [All Devices] displays all devices in the project. If you select [All Devices], you cannot select [BLOCK] nor [MACRO] for the search range.
TIP

Selecting [Global Device] displays the following types of devices:
- Input relay (X)
- Output relay (Y)
- Internal relay (I)
- Shared relay/extended shared relay (E)
- Special relay (M)
- Timer (T)
- Counter (C)
- Link relay (L)
- Date register (D)
- Cache register (F)
- File register (B)
- Special register (Z)
- Link register (W)
- Shared register/extended shared register (R)
- Index register (V)

Selecting [Local Device] displays the following types of devices:
- Internal relay (I)
- Timer (T)
- Counter (C)
- Date register (D)
- Cache register (F)
- File register (B)

Selecting [Macro Device] displays the following types of devices:
- Macro relay (H)
- Macro register (A)
- Macro index register (U)
- Pointer (P)

B Select the range of devices to be displayed. You can select one of the following four options to specify the range of devices:
Selecting [Project] searches for devices in all blocks and macros in the project and displays the found devices.
Selecting [Component Blocks] searches for devices in blocks currently registered in the component definition and displays the found devices.
Selecting [BLOCK] opens a popup menu that displays block names in the project. Select a block from the popup menu to display devices in the selected block.
Selecting [MACRO] opens a popup menu that displays macro names in the project. Select a macro from the popup menu to display devices in the selected macro.

C Click this button to switch the icon display mode between detailed icon mode and simple icon mode. Detailed icon mode and simple icon mode can represent 16 and 4 types of device usage states, respectively. The switched icon display mode becomes enabled when WideField3 is started next time.

D Click this button to switch the display mode to [Display on Output Window]/[Display Individually].

E Enter the address of a device to check the usage status of the device. After entering the address, press the [Enter] key. The position cursor jumps to the specified address. When [Auto Search] is turned on, the position cursor automatically jumps to the address indicated by the cursor position in the program edit window, without having to enter the address.
F Icon buttons.
Click the [Search] button to find a device for the specified address in the Device List dialog box.
Click the [Export] button to export the device list in CSV format.
Click the [Display Mode] button to switch the screen display mode.
Click the [Refresh] button to update the contents displayed in the dialog box.

G Sets the number of columns to be displayed. Specify the setting for each device type. The default values are 32 points for all relays and 20 points for all registers/timer/counter.

H [Show/Hide Legend] button. If you click [Show Legend], the icon description pane is shown and the button name is changed to [Hide Legend]. If you click [Hide Legend], the icon description pane is hidden and the button name is changed to [Show Legend].

SEE ALSO
For details on [Export], see Section E5.3, "Exporting a Device List."

(1) Map Mode

The following figure shows the screen layout in map mode.

This mode displays a map of devices. Currently unused devices are displayed as blank cells.

- A Column numbers. You can get the address for a cell by adding the row number and column number of the cell.
- B Row numbers. Each row number represents "the address of the first column of the row" - 1.
- C A cell. Each cell shows the usage status of the device for the address corresponding to the cell. Blank cells represent unused devices. Cells in gray represent devices not in the specified range.
- D Device type tabs. The sheet name represents the type of devices displayed in the sheet.
- E A tooltip. When the mouse cursor is positioned at a cell, a tooltip opens to display the usage status of the device corresponding to the cell.

**TIP**

If [All Devices] is selected for the type of the devices to be displayed, the map mode cannot be used.

(2) **Details Mode**

The following figure shows the screen layout in details mode. The screen in details mode lists devices currently used.

![Screen Layout in Details Mode](E0501_04.VSD)

**Figure E5.4  Screen Layout in Details Mode**

- A Displays the address of each device.
- B Shows the usage status of each device.
- C Indicates whether or not the device is used to be read from.
- D Indicates whether or not the device is used to be written to.
- E Indicates whether or not the device contains a numerical value.
- F Displays tag names.
- G Displays I/O comments.
- H Displays the name of the block or macro that uses the device.
- I Displays the cross reference information of the device.
(3) Displaying by Tag Name Definition

The following figure shows the screen layout for displaying by tag name definition.

Figure E5.5 Screen Layout for Displaying by Tag Name Definition

- **A** Displays the address of each device.
- **B** Shows the usage status of each device.
- **C** Indicates whether or not the device is used to be read from.
- **D** Indicates whether or not the device is used to be written to.
- **E** Indicates whether or not the device contains a numerical value.
- **F** Displays tag names.
- **G** Displays I/O comments.
- **H** Indicates the type of the tag name definition referred to by the device. If the device refers to the common tag name definition, "Common Tag Name Definition" is displayed. If the device refers to a block/macro tag name definition, the block name or macro name is displayed.
(4) **Icon Description Pane**

You can show or hide the icon description pane by clicking [Show/Hide Legend] on the Device List dialog box.

The icon description pane describes the following two types of icons:
- Detailed icons
- Simple icons

Detailed icons can represent 16 different states.
Simple icons can represent 4 different states.

You can switch between detailed icon mode and simple icon mode by clicking [Details Display] or [Simple Display] and restarting WideField3.

The following table describes the detailed icons.

<table>
<thead>
<tr>
<th>Number</th>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>Out of the specified range</td>
</tr>
<tr>
<td>1</td>
<td>(Blank)</td>
<td>Unused device</td>
</tr>
</tbody>
</table>
| 2      |      | Write  
- Duplicate use of SET/RST  
- With index modification  
- First device |
| 3      |      | Write  
- Duplicate use of coil  
- With index modification  
- First device |
| 4      |      | Write  
- Duplicate use of timer/counter  
- With index modification  
- First device |
| 5      |      | Write  
- With index modification  
- First device |
| 6      |      | Write  
- Duplicate use of SET/RST  
- First device |
| 7      |      | Write  
- Duplicate use of coil  
- First device |
| 8      |      | Write  
- Duplicate use of timer/counter  
- First device |
| 9      |      | Write  
- First device |
| 10     |      | Read  
- With index modification  
- First device |
| 11     |      | Read  
- First device |
| 12     |      | Numerical value  
- With index modification  
- First device |
| 13     |      | Numerical value  
- First device |
| 14     |      | Write  
- Hidden device |
| 15     |      | Read  
- Hidden device |
| 16     |      | Numerical value  
- Hidden device |
The following table describes the simple icons.

<table>
<thead>
<tr>
<th>Number</th>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>Out of the specified range</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>Unused device</td>
</tr>
<tr>
<td>2</td>
<td>![W]</td>
<td>Write</td>
</tr>
<tr>
<td>3</td>
<td>![R]</td>
<td>Read</td>
</tr>
<tr>
<td>4</td>
<td>![N]</td>
<td>Numerical value</td>
</tr>
</tbody>
</table>

There are priorities in displaying these usage status icons.
The priorities are as follows:
1) Used device > Hidden device
2) With index modification > Without index modification
3) Write > Read > Numerical value
4) Duplicate use of SET/RST > Duplicate use of coil > Duplicate use of timer/counter

Example:

```
-----------[DREF Y00210  Y00210]
-----------[STR Y00203  Y00205  Y00207]
```

In the above instruction example, the device Y00210 has the following usage states:
- Used device
- Read
- Write
- Numerical value
- Hidden device

Thus, in this example, according to the priorities, the usage status icon for Y00210 is ![W] (Used device + Write).

### Displaying the Device List on the Output Window

Click the [Display on Output Window] button in the Device List dialog box. A device list is displayed on the output window. In that state, click the [Display Individually] button to display the device list on an independent device list window.

### Setting the Number of Displayed Columns

To change the number of columns displayed in the Device List dialog box, select a value from the pull-down menu for the number of columns. Applicable values include [Auto Set], [8pt Dsply], [10pt Dsply], [16pt Dsply], [20pt Dsply], [30pt Dsply], [32pt Dsply], [40pt Dsply], [50pt Dsply], [64pt Dsply], [100pt Dsply]. Specify the setting for each device type. [Auto Set] automatically adjusts the number according to the resizing of the screen. The default values are 32 points for relays and 20 points for registers/timer/counter.
■ Refreshing the Device List Information

When the Device List dialog box is displayed, modifying the program causes inconsistency between the modified program and the information shown in the device list. To reflect the current program contents to the device list, click the [Refresh] button. Refreshing the device list may take some time depending on the program size.

■ Closing the Device List Dialog Box

To close the Device List dialog box, click the [Close] button.

TIP

If you close the project, the Device List dialog box is automatically closed.
E5.1.2 Finding a Device in a Device List

You can find a device in the Device List dialog box to check the usage status of the device.

To find a device, use the following procedure.

Procedure (A) enters an address manually. Procedure (B) specifies an address in the program edit window.

◆ Procedure (A)◆

(1) Select [Project]–[Device List] from the menu bar.
⇒ The Device List dialog box is displayed.
(2) Select the type of the device to find.

(3) Select the search range.
Enter the address of the device you want to find, and click the [Search] button or press the [Enter] key.

⇒ The position cursor moves to the usage status cells of the device for the specified address.

⚠️ CAUTION

You can specify only a device address for the search string. You cannot specify a tag name or structure name.

◆ Procedure (B)◆

1. Select [Project]–[Device List] from the menu bar.

⇒ The Device List dialog box is displayed.

2. Turn on [Auto Search] in the Device List dialog box.

3. Click the cell for the address to be specified in the program edit window.

⇒ The cursor jumps to the usage status cell of the device corresponding to the specified address.

TIP

- If you enter an invalid string as a search string, an error message is displayed.

- If you edit a ladder program while the Device List dialog box is open, the display contents in the Device List dialog box may be different from the usage status of devices actually used in the program.

- To reflect the current program contents to the Device List dialog box, click the [Refresh] button on the Device List dialog box.

SEE ALSO

For details on the icon that represents the device usage status on each cell, see Section E5.1.1, "Display and Layout of the Device List Dialog Box."
E5.1.3 Jumping to Used or Unused Devices

On the Device List dialog box, you can jump to the next or previous used or unused device of the currently selected device.

The relevant procedure is given below.

◆ Procedure ◆

1. **Procedure**
   - Click the [Display Mode] button on the Device List dialog box, and select [Map].
     - The screen display mode of the Device List dialog box is changed to map mode.
   - Select a cell in the window, right-click the cell to open a popup menu, and select the jump destination from the popup menu.
     - The window displays the next or previous used or unused device according to your operation above and the position cursor moves to the position of the relevant device.
E5.1.4 Restrictions on the Device List Function

This section describes the restrictions on the device list function.

The device list function has some restrictions on displaying usage status.

- **Devices Specified by an Index Modification**

  When a device list displays devices specified by an index modification, the device list does not reflect the index modification. Thus, while the devices are actually used, the Device List dialog box displays "Unused" for the usage status of the devices.

  ![Figure E5.6 Example of Index Modification: Type 1](image1)

  ![Figure E5.7 Example of Index Modification: Type 2](image2)
Devices Specified by a Constant Index Modification

When a device list displays devices specified by a constant index modification, the device list reflects the constant index modification.

![Diagram](E0501_12.VSD)

**Figure E5.8 Example of Constant Index Modification: Type 1**

Hidden Devices

When some devices such as long-word devices or double-long-word devices are used in a program, the devices may be partially hidden in the instructions. These devices are called hidden devices. Hidden devices are reflected in device lists.

The following figure shows a search result for when the data registers D2 and D11 are used as hidden devices.

![Diagram](E0501_14.VSD)

**Figure E5.10 Example of Hidden Devices**
### Indirect Specification and Index Modified Indirect Specification

Hidden devices used in an indirect specification instruction are reflected in device lists.

![Example of Indirect Specification](image1.png)

**Figure E5.11 Example of Indirect Specification**

Devices specified by an index modified indirect specification are not reflected in device lists. Thus, while the devices are actually used, the Device List dialog box displays "Unused" for the usage status of the devices.

![Example of Index Modified Indirect Specification](image2.png)

**Figure E5.12 Example of Index Modified Indirect Specification**

### Constant Definition

When a constant definition is used in an instruction, the Device List dialog box displays "Used" for each usage status of the device hidden due to the constant definition.

The following figure shows an example of the constant definition for when #CONST=10.

![Example of Constant Definition](image3.png)

**Figure E5.13 Example of Constant Definition**
Creating Cross References in the Device List Dialog Box

You can create cross references from a device selected in the Device List dialog box. To create cross references, use the following procedure.

◆ Procedure ◆

(1) In the Device List dialog box, select a device whose cross references are to be created.

TIP
You can create cross references in any of the three screen display modes: map mode, details mode, and mode for displaying by tag name definition.

(2) Double-click the cell of the selected device.
⇒ Cross references are created for the device corresponding to the specified cell.

SEE ALSO
For details on the cross reference, see Chapter E6, “Cross Reference.”
E5.3 Exporting a Device List

You can export the contents of the Device List dialog box to one or more CSV files. There are two types of export methods:

- Exporting from the currently displayed tab
- Exporting from all tabs

**TIP**

If a tab to be exported contains more than 65,535 rows, the contents are divided and saved in multiple CSV files. When the divided files are created, “[*]” is inserted before “.csv” in each file name of the second and subsequent files. Here, “*” denotes a sequential number starting with 2.

Example:

B.csv: The first to 65,535th rows are exported to this file.
B[2].csv: The 65,536th and subsequent rows are exported to this file.

**SEE ALSO**

For the number of devices available for each CPU type, see “Sequence CPU – Functions.”
E5.3.1 Exporting from the Currently Displayed Tab

To export from the currently displayed tab, use the following procedure.

◆ Procedure ◆

(1) While the tab whose contents to be exported is open on the Device List dialog box, click the [Export] button. ⇒ A popup menu is displayed.

(2) Click [Active Sheet].

⇒ The Save As dialog box is displayed.

(3) Enter a file name for the file to be saved, and click [Save].

TIP
A character representing the device type is entered in advance in the File Name text field of the Save As dialog box.

⇒ The contents of the currently displayed tab are saved in one or more CSV files.

TIP
- The default file name of the saved file is in the format of "character representing device type.csv".
- If you export a device list displayed in map mode, the contents saved in each exported file contains numbers representing the device's usage status as described in the current icon description pane. Thus, the meanings of the numbers vary depending on whether the icon display mode is simple icon mode or detailed icon mode.
E5.3.2  Exporting from All Tabs
To export from all tabs, use the following procedure.

◆ Procedure ◆

(1) Click the [Export] button on the Device List dialog box.
⇒ A popup menu is displayed.
(2) Click [All Sheets].
⇒ The Browse For Folder dialog box is displayed.
(3) Select the folder to which files are saved, and click [OK].
⇒ The contents of all tabs are saved in CSV files.

**TIP**
- The default file name of the saved file is in the format of “character representing device type.csv”.
- In the specified folder, WideField3 creates a new subfolder named “DV-Local Device-YYYYMMDD-HHmmSS-mmm”, where YYYY, MM, DD, HH, mm, SS, and mmm represent year, month, day, hour, minute, second, and millisecond, respectively. The subfolder name represents the date and time of the export operation. The contents of each tab in the Device List dialog box are exported to a CSV file that is saved in the subfolder.
- If you export a device list displayed in map mode, the contents saved in each exported file contains numbers representing the device’s usage status as described in the current icon description pane. Thus, the meanings of the numbers vary depending on whether the icon display mode is simple icon mode or detailed icon mode.
E6. Cross Reference

The cross reference function searches for addresses, tag names, structures, block names, macro names, constant names, or labels, and it displays the search result in a list format.

The search results are listed in the Cross-Reference tab on the output window and also in the search window.

The search window is a simplified version of the Cross-Reference tab of the output window and the operations in the search window are similar to those in the Cross-Reference tab.

This chapter mainly describes the operations in the Cross-Reference tab.

You can select a device from the cross reference search results and jump to the relevant device displayed in another window.

TIP

The search result also lists blocks registered in the component definition among the blocks that refer to the common library or another project.
Function Limitations for Each CPU Type

Table E6.1  Function Limitations for Each CPU Type

<table>
<thead>
<tr>
<th>CPU Modules</th>
<th>Limitations</th>
<th>SEE ALSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>F3SP22-0S, F3SP28-3S,</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>F3SP38-6S, F3SP53-4S,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F3SP58-6S, F3SP59-7S</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F3SP66-4S, F3SP67-6S</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>F3SP71-4N, F3SP76-7N</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>F3SP71-4S, F3SP76-7S</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>

TIP
This section does not describe hardware-dependant limitations for each CPU type, such as the number of available devices.

SEE ALSO
For details on limitations for each CPU type, refer to the user's manual for each type.
E6.1 Fundamentals of Cross References

This section describes the basic operations and restrictions of the cross reference function.

E6.1.1 Display and Layout of Windows for Cross References

- **Showing and Hiding the Output Window**

  Cross references are created on the Cross-Reference tab in the output window. You can show or hide the output window by selecting [View]-[Output Window] from the menu bar.

- **Screen Layout of the Cross-Reference Tab**

  To create cross references, use the Cross-Reference tab in the output window. The following figure shows the screen layout of the Cross-Reference tab.

  ![Cross-Reference Tab](E0601_01.VSD)

  **Figure E6.2 Cross-Reference Tab**

  - **A** Enter a device, tag name, constant name, block/macro name, or label to be searched for. You can use the wildcard character ("*") for a tag name, constant name, or index modification.
    Examples: tag*, D00001;*, tag*;*, #Card*
  - **B** Select the search range. You can select one of the following five ranges:
    Selecting [Project] searches all blocks and macros in the project.
    Selecting [Component Blocks] searches blocks currently registered in the component definition.
    Selecting [Active Block/Macro] searches blocks and macros that are currently active.
    Selecting [BLOCK] searches all blocks in the project or specified blocks.
    Selecting [MACRO] searches all macros in the project or specified macros.
  - **C** Enter instructions to be displayed in the search results. You can specify more than one instruction. To specify multiple instructions, insert "|" between instructions.
    Examples: LD | OUT, LD | OUT | MOV
- D Icon buttons.
  - Clicking the [Refresh] button updates the search results.
  - Clicking the [Sort] button sorts the search results.
  - Clicking the [Export] button exports the search results.
  - Clicking the [Search] button starts searching.

- E Turn on or off this checkbox to enable or disable the automatic search function.
  - When the automatic search function is enabled, clicking a device displayed on a screen automatically displays the search results for the device. This function can be used on the program edit window, program monitor window, tag name definition window, constant definition window, and device monitor window.

- F Turn on this checkbox to enable jumping to a monitor screen. When the [Online] checkbox is on, if you double-click an obtained cross reference of a device, you can jump to the monitor screen of the block or macro that uses the device.

- G Search result tabs. Clicking a tab opens a search result obtained by a search with the tab name. Up to 8 search result tabs can be created in the Cross-Reference tab. If a local device or search option is specified for a search, the tab name of the search result tab is displayed in blue.

- H Displays a search result.

**SEE ALSO**
- For details on [Sort], see Section E6.4, "Sorting Cross References."
- For details on [Export], see Section E6.5, "Exporting Cross References."

**CAUTION**

The search function for cross references only finds exact matches (no partial matches).
For example, when a device assigned with the tag name "tag001" is used in a program, even if you enter "tag" as a search object in the search text box, you cannot get "tag001" in the search results.
To search for the tag name "tag001" only, enter "tag001".
To find all tag names containing "tag", enter "tag*" using the wildcard character ("*").
To search for a constant name, enter ‘#’ at the beginning of your search string.
**Showing and Hiding the Search Window**

You can also use the search window to display cross references. By default, the search window is displayed on the left side of the screen as in the case of the project window, and you can use a vertically wide screen in WideField3. You can show or hide the search window by selecting [View]-[Search Window] from the menu bar.

**Search Window for Cross References**

The layout and operations of the search window are similar to those of the Cross-Reference tab in the output window.

![Search Window](E0601_02.VSD)

**Figure E6.3 Search Window**

**TIP**

The search window displays block/macro names, row numbers, and details only. If you want to list more detailed information, use the Cross-Reference tab in the output window.

**Search Objects for Cross References**

You can use the following search objects for cross references:
- Address
- Tag name
- Structure name (or member name in "structure name.member name" format)
- Constant name
- Label
- Block/macro name
E6.1.2 Creating Cross References

You can create cross references in the following screens:
- Program edit screen
- Program monitor screen
- Tag name definition edit screen (online/offline)
- Device monitor screen
- Registered device monitor screen
- Constant definition screen

There are two ways for creating cross references:
- Creating on the Cross-Reference tab
- Creating by specifying a device

You can use the following search objects for cross references:
- Address
- Tag name
- Structure name (or member name in "structure name.member name" format)
- Constant name
- Label
- Block name/macro name

SEE ALSO
For details on creating cross references on the search window, see Section E6.2, "Creating Cross References Automatically."

Creating Cross References on the Cross-Reference Tab

To create cross references on the Cross-Reference tab, use the following procedure.

◆ Procedure ◆

(1) Select [Find]-[Search Cross-Reference] from the menu bar.
⇒ The Cross-Reference tab is displayed in the output window.
(2) Enter a search string.
(3) Select the search range.

(4) If you want to specify an instruction, select the instruction from the instruction drop-down list.

**TIP**

To specify multiple instructions, insert "|" between instructions.

Example: MOV|MCALL

(5) Click the [Search] button or press the [Enter] key.

⇒ Search results are displayed in the Cross-Reference tab.

**CAUTION**

Up to 8 search result tabs can be created. If you perform another search operation while 8 tabs are already created, the left-most not-locked tab in the Cross-Reference tab is overwritten.

**TIP**

- You can change the column width on the search result tabs. The column width you changed is preserved and used in subsequent searches.
- When a search condition is specified, that is, an instruction filter is specified or the specified search range is other than the entire project, the tab name of the obtained cross references is displayed in blue.
- When cross references for a device are already created and present, if you create cross references for the device again, the tab of the existing cross references is overwritten. Thus, no new tab is created for the obtained cross references for the same device.
- When you try to create cross references, if no matched device is found, an empty cross reference result tab is created.
Creating Cross References by Specifying a Device

You can create cross references by specifying a device on a screen. There are six types of screens on which you can create cross references:

- Program edit screen
- Program monitor screen
- Tag name definition edit screen (online/offline)
- Device monitor screen
- Registered device monitor screen
- Constant definition screen

Use the following procedure to create cross references by specifying a device on a screen.

◆ Procedure ◆

(1) Select a device on the screen.
(2) Click the right mouse button.
⇒ A popup menu is displayed.

(3) Select [Search Cross-Reference].
⇒ Cross references are created in the Cross-Reference tab on the output window.

TIP
If you try to create cross references without opening the output window, the search window opens automatically.

SEE ALSO
For details on the search window, see E6.1.1, "Display and Layout of Windows for Cross References."
Locking and Unlocking Cross Reference Result Tabs

You can lock a cross reference result tab. By locking a cross reference result tab, you can avoid mistakenly closing the tab.

To lock or unlock a cross reference result tab, right-click the tab to open a popup menu, and select [Lock] from the popup menu. The [Lock] icon is displayed on the locked tab.

Figure E6.5   Locking and Unlocking a Cross Reference Result Tab

TIP
If you select [Close All] from the popup menu, the locked tabs are closed.

Closing Cross Reference Result Tabs

There are three ways for closing the cross reference result tabs:

- Closing the currently opened cross reference result tab
- Closing the cross reference result tabs other than the currently opened tab
- Closing all cross reference result tabs

Closing the Currently Opened Cross Reference Result Tab

To close the currently opened cross reference result tab, right-click the tab to open a popup menu, and select [Close] from the popup menu. You can also close the tab by clicking the mouse wheel.

Closing the Cross Reference Result Tabs Other Than the Currently Opened Tab

To close the cross reference result tabs other than the currently opened tab, right-click the currently opened tab to open a popup menu, and select [Close All Other Sheets] from the popup menu. Note however that this operation does not close the locked tabs.

Closing All Cross Reference Result Tabs

To close all cross reference result tabs, right-click the tab to open a popup menu, and select [Close All] from the popup menu. This operation also closes the locked tabs.
E6.1.3 Restrictions on the Cross Reference Function

This section describes the restrictions on the cross reference function.

The cross reference function has some restrictions on devices to be searched for and on search strings.

Restrictions on Devices

Some devices are not displayed in cross reference search results.

We describe below whether the following five types of devices are displayed:

- Devices specified by an index modification
- Devices specified by a constant index modification
- Hidden devices
- Indirect specification and index modified indirect specification
- Constant definition

Devices Specified by an Index Modification

Devices specified by an index modification are not reflected in cross references.

Figure E6.6  Example of Index Modification: Type 1

Figure E6.7  Example of Index Modification: Type 2
● Devices Specified by a Constant Index Modification

Devices specified by a constant index modification are reflected in cross references.

![Figure E6.8 Example of Constant Index Modification: Type 1](E0601_12.VSD)

![Figure E6.9 Example of Constant Index Modification: Type 2](E0601_13.VSD)

● Hidden Devices

When some devices such as long-word devices or double-long-word devices are used in a program, the devices may be partially hidden in the instructions. These devices are called hidden devices.

Hidden devices are reflected in cross references.

The following figure shows a search result for when the data registers D2 and D11 are used as hidden devices.

![Figure E6.10 Example of Hidden Devices](E0601_13.VSD)
Indirect Specification and Index Modified Indirect Specification

Hidden devices used in an indirect specification instruction are reflected in cross references.

Figure E6.11 Example of Indirect Specification

Devices specified by an index modified indirect specification are not reflected in cross references.

Figure E6.12 Example of Index Modified Indirect Specification

Constant Definition

When a constant definition is used in an instruction, all devices hidden due to the constant definition are displayed in the cross reference result.

The following figure shows an example of the constant definition for when #CONST=10.

Figure E6.13 Example of Constant Definition
Restrictions on Search Strings

The following table shows the restrictions on the search strings.

Table E6.2 Restrictions on Search Strings

<table>
<thead>
<tr>
<th>Search String</th>
<th>Maximum length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
<td>9</td>
<td>A valid device address.</td>
</tr>
<tr>
<td>Tag name definition</td>
<td>16</td>
<td>A string of up to 16 characters that begins with two alphabetic characters. Special characters '-', '!', and ':' can be used. The ':' character is used as a delimiter between a structure name and structure member name. You can use a string that can be used to identify a device.</td>
</tr>
<tr>
<td>Structure name</td>
<td>7</td>
<td>A structure name. Special characters '-' and '_' can be used. You can use a string that can be used to identify a device.</td>
</tr>
<tr>
<td>Structure member</td>
<td>8</td>
<td>A structure member.</td>
</tr>
<tr>
<td>Structure</td>
<td>24</td>
<td>A structure, index modified structure, structure member, or index modified structure member.</td>
</tr>
<tr>
<td>Constant name</td>
<td>16</td>
<td>A string that begins with '#'. Special characters '-' and '_' can be used.</td>
</tr>
<tr>
<td>Wildcard</td>
<td>25</td>
<td>A tag name definition, structure, or constant that contains the wildcard character ('*').</td>
</tr>
<tr>
<td>Input string</td>
<td>100</td>
<td>A string that can be entered.</td>
</tr>
<tr>
<td>Instruction</td>
<td>100</td>
<td>An instruction that can be entered.</td>
</tr>
<tr>
<td>Label</td>
<td>6</td>
<td>A string of up to six alphanumeric characters that begins with two alphabetic characters. Special characters '-' and '_' can be used.</td>
</tr>
<tr>
<td>Block/macro name</td>
<td>8</td>
<td>A string of up to eight alphanumeric characters that begins with two alphabetic characters. Special characters '-' and '_' can be used.</td>
</tr>
</tbody>
</table>

TIP

Leading and trailing spaces in a search string are ignored.

Restrictions on Search Objects

The cross reference function also searches blocks registered in the component definition among the blocks that refer to the common library or another project.
E6.2 Creating Cross References Automatically

Cross references can be created automatically.
You can create cross references automatically in the following screens:
- Program edit screen
- Program monitor screen
- Tag name definition edit screen (online/offline)
- Device monitor screen
- Registered device monitor screen
- Constant definition screen

To create cross references automatically, use the following procedure.

◆ Procedure ◆

(1) Make sure that the output window is open.
(2) Turn on the [Auto Search] checkbox on the Cross-Reference tab.

(3) In the screen, select a device whose cross references are to be created.
⇒ Cross references for the selected device are automatically created.

TIP
The behavior of the automatic search function for creating cross references depends on the type of devices as follows.
- Local device
  The cross reference result tab displays "Local device name [Block/macro name]" in blue.
  The search range is within the block in which the device is selected.
  The instruction filter follows the setting that is specified when the automatic search is executed.
- Global/macro device
  The cross reference result tab displays "Global/macro device name".
  The search range follows the setting that is specified when the automatic search is executed.
  The instruction filter follows the setting that is specified when the automatic search is executed.

CAUTION
For large projects, the automatic creation of cross reference may take time.
E6.3 Jumps

You can jump from the cross reference result created for a device to the position where the device is used in the program.

There are three types of jump destinations:
- Jumping to offline
- Jumping to online
- Jumping to a tag name definition

■ Jumping to Offline

You can jump from the cross reference result for a device to the position where the device is used in a program edit screen.

To jump to offline, perform one of the following three procedures:
- While the [Online] checkbox is off in the Cross-Reference tab, double-click a row in the cross reference result.
- Right-click a row in the cross reference result to open a popup menu, and click [Jump to Offline] on the popup menu.
- Use the arrow keys to position the cursor to a row in the cross reference result screen, and press the [Enter] key.

⚠️ CAUTION

- If you try to jump to a block or macro modified after cross references are created, you may jump to an incorrect position. In this case, click the [Refresh] button and perform the jump operation again.
- If the block or macro of a jump destination cannot be opened, a warning dialog box is displayed.
Jumping to Online

You can jump from the cross reference result for a device to the position where the device is used in a program monitor screen.

To jump to online, perform one of the following three procedures:
- While the [Online] checkbox is on in the Cross-Reference tab, double-click a row in the cross reference result.
- Right-click a row in the cross reference result to open a popup menu, and click [Jump to Online] on the popup menu.
- While the [Online] checkbox is on in the Cross-Reference tab, use the arrow keys to position the cursor to a row in the cross reference result screen, and press the [Enter] key.

CAUTION

- If the online project is different from the offline project, you may jump to an incorrect position. To jump to a correct position, download the project.
- If the block or macro of a jump destination cannot be found, a warning dialog box is displayed.
- If WideField3 is not connected to the CPU module, a warning dialog box is displayed.

SEE ALSO

- For details on downloading a project, see Chapter H2, "Downloading" (Online).
- For details on connecting to a CPU module, see Section H1.1, "Connecting and Disconnecting" (Online).

Jumping to a Tag Name Definition

You can jump from the cross reference result for a device to the position where the tag name for the device is defined in the tag name definition window.

To jump to a tag name definition, right-click a row in the cross reference result to open a popup menu, and click [Jump to Tag Name Definition] on the popup menu.

For a block referring to a block tag name definition, the jump destination is the block tag name definition window. For a block referring to a common tag name definition, the jump destination is the common tag name definition window.

CAUTION

If you try to jump to tag name definitions modified after cross references are created, you may jump to an incorrect position. In this case, click the [Refresh] button and perform the jump operation again.

TIP

If the device you specified for a jump operation is not assigned with any tag name definition, the position cursor in the tag name definition window of a jump destination is positioned to the top position of an undefined line. Edit the tag name definitions when needed.
E6.4 Sorting Cross References

You can sort obtained cross references. You can specify one of the following five types of sort orders:
- By component block
- By tag name
- By block/macro
- By instruction
- By parameter type (I/O type)

To sort cross references, use the following procedure.

◆ Procedure ◆

(1) Click the [Sort] button in the output window.  
⇒ A popup menu is displayed.

(2) Specify the sort order.  
⇒ The contents displayed in the output window are sorted.

TIP

When you sort by tag name, block/macro, instruction, or parameter type (I/O type), you can select either ascending or descending order. Click a sort item name to switch between ascending order and descending order. The upward icon (▲) and downward icon (▼) represent ascending order and descending order, respectively.

Figure E6.14 Ascending/Descending Order Icon
Sorting by Component Block

The following table shows the sort priorities to be used in sorting by component block.

Table E6.3 Priorities in Sorting by Component Block

<table>
<thead>
<tr>
<th>Priority</th>
<th>Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Blocks registered in the component definition</td>
</tr>
<tr>
<td>2</td>
<td>Blocks not registered in the component definition</td>
</tr>
<tr>
<td>3</td>
<td>Macros</td>
</tr>
<tr>
<td>4</td>
<td>Row numbers (in ascending order)</td>
</tr>
<tr>
<td>5</td>
<td>Column numbers (in ascending order)</td>
</tr>
</tbody>
</table>

Sorting by Tag Name

The following table shows the sort priorities to be used in sorting by tag name.

Table E6.4 Priorities in Sorting by Tag Name

<table>
<thead>
<tr>
<th>Priority</th>
<th>Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Tag names (in alphabetical order)</td>
</tr>
<tr>
<td>2</td>
<td>Block/macro names (in ascending order)</td>
</tr>
<tr>
<td>3</td>
<td>Row numbers (in ascending order)</td>
</tr>
<tr>
<td>4</td>
<td>Column numbers (in ascending order)</td>
</tr>
</tbody>
</table>

Sorting by Block/Macro

The following table shows the sort priorities to be used in sorting by block or macro.

Table E6.5 Priorities in Sorting by Block/Macro

<table>
<thead>
<tr>
<th>Priority</th>
<th>Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Block/macro names (in ascending order)</td>
</tr>
<tr>
<td>2</td>
<td>Row numbers (in ascending order)</td>
</tr>
<tr>
<td>3</td>
<td>Column numbers (in ascending order)</td>
</tr>
</tbody>
</table>

Sorting by Instruction

The following table shows the sort priorities to be used in sorting by instruction.

Table E6.6 Priorities in Sorting by Instruction

<table>
<thead>
<tr>
<th>Priority</th>
<th>Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Instructions (in alphabetical order)</td>
</tr>
<tr>
<td>2</td>
<td>Block/macro names (in ascending order)</td>
</tr>
<tr>
<td>3</td>
<td>Row numbers (in ascending order)</td>
</tr>
<tr>
<td>4</td>
<td>Column numbers (in ascending order)</td>
</tr>
</tbody>
</table>

Sorting by Parameter Type (I/O Type)

The following table shows the sort priorities to be used in sorting by parameter type (I/O type).

Table E6.7 Priorities in Sorting by Parameter Type

<table>
<thead>
<tr>
<th>Priority</th>
<th>Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Parameter Types (in alphabetical order)</td>
</tr>
<tr>
<td>2</td>
<td>Block/macro names (in ascending order)</td>
</tr>
<tr>
<td>3</td>
<td>Row numbers (in ascending order)</td>
</tr>
<tr>
<td>4</td>
<td>Column numbers (in ascending order)</td>
</tr>
</tbody>
</table>
E6.5 Exporting Cross References

You can export obtained cross references in CSV format. There are two types of export methods:
- Exporting from the currently displayed tab
- Exporting from all tabs

Exporting from the Currently Displayed Tab

You can export cross references from the currently displayed tab in CSV format. The relevant procedure is given below.

◆ Procedure ◆

1. Click the [Export] button in the output window.
   ⇒ A popup menu is displayed.
2. Select [Active Sheet].
   ⇒ The Save As dialog box is displayed.

TIP
When the Save As dialog box is opened, the current tab name is displayed as a file name of the file to be saved. Note however that special characters ‘:’ and ‘/’ are replaced with ‘_’, and the wildcard character ‘*’ is replaced with ‘%’ in the tab name.

3. Enter a file name and click [Save].
TIP

- If the export is completed successfully, a confirmation dialog box is displayed. If the export is failed, a warning dialog box is displayed.
- If obtained cross references contain more than 65,535 rows, the cross references are divided and exported to multiple CSV files. When the divided files are created, "[*]" is inserted before ".csv" in each file name of the second and subsequent files. Here, "***" denotes a sequential number starting with 2.
  
  Example:
  B00001.csv: The first to 65,535th rows are exported to this file.
  B00001[2].csv: The 65,536th and subsequent rows are exported to this file.
- Even if the project is modified, the displayed cross references remain unchanged until the [Refresh] button is clicked. Thus, to export the latest results, click the [Refresh] button before performing an export operation.

Exporting from All Tabs

You can export cross references from all tabs in CSV format. The relevant procedure is given below.

◆ Procedure ◆

(1) Click the [Export] button in the output window.

⇒ A popup menu is displayed.
(2) Select [All Sheets].

⇒ The Browse For Folder dialog box is displayed.
(3) Select the folder to which files are saved, and click [OK].
TIP

- In the specified folder, WideField3 creates a new subfolder named "YYYYMMDD-HHmmSS-mmm", where YYYY, MM, DD, HH, mm, SS, and mmm represent year, month, day, hour, minute, second, and millisecond, respectively. The subfolder name represents the date and time of the export operation. The contents of each cross reference tab are exported to a CSV file that is saved in the subfolder. For the file name for each tab, see "Exporting from the Currently Displayed Tab."

- If the export is completed successfully, a confirmation dialog box is displayed.

- If a CSV file cannot be saved, a warning dialog box is displayed.

- If the subfolder for storing CSV files cannot be created, a warning dialog box is displayed.
E7. Comparing Files

The File Comparison function compares the content of files associated with projects created in WideField3, and extracts and displays parts of difference.

The File Comparison function has the following two types of comparison methods:
- Compare Project
- Compare Window

With Compare Project, two sets of project data are specified and all files within both projects are compared.
With Compare Window, two sets of the same type of file data are specified and the files are compared. The user specifies a file in the currently open project as the comparison source file, and a file in another project as the comparison target file.

CAUTION

The File Comparison function does not compare the contents of balloon comments.
### Function Limitations for Each CPU Type

#### Table E7.1  Function Limitations for Each CPU Type

<table>
<thead>
<tr>
<th>CPU Modules</th>
<th>Limitations</th>
<th>SEE ALSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>F3SP22-0S, F3SP28-3S, F3SP38-6S, F3SP53-4S, F3SP58-6S, F3SP59-7S</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>F3SP66-4S, F3SP67-6S</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>F3SP71-4N, F3SP76-7N</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>F3SP71-4S, F3SP76-7S</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>

**TIP**

This section does not describe hardware-dependant limitations for each CPU type, such as the number of available devices.

**SEE ALSO**

For details on limitations for each CPU type, refer to the user's manual for each type.
E7.1 Basic File Comparison Operations

This section describes the basic File Comparison operations as well as settings for the File Comparison function.

E7.1.1 File Comparison Bar and Shortcut Keys

The operations that can be used only with the File Comparison function are found in a tool bar.

Operations are assigned for each button as shown in the table below.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Shortcut</th>
<th>Operation</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ctrl+5</td>
<td>Compare Project</td>
<td>Starts comparison between the project currently open, as the comparison source, and the other project.</td>
</tr>
<tr>
<td></td>
<td>Ctrl+6</td>
<td>Compare Window</td>
<td>Starts comparison between the window currently open, as the comparison source, and the other file of the same type.</td>
</tr>
<tr>
<td></td>
<td>Ctrl+8</td>
<td>Close Comparison Destination</td>
<td>Closes the comparison destination pane in the comparison results display, displaying only the comparison source pane.</td>
</tr>
<tr>
<td></td>
<td>Ctrl+0</td>
<td>Copy Comparison Results</td>
<td>Copies the project comparison results to the clipboard as text data.</td>
</tr>
<tr>
<td></td>
<td>Ctrl+Tab</td>
<td>Switch Pane</td>
<td>Switches the focus of the comparison source/comparison destination pane in the comparison results display.</td>
</tr>
<tr>
<td></td>
<td>Ctrl+9</td>
<td>Switch Split Display</td>
<td>Switches the pane structure in the comparison results display between up/down and left/right.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Enable/Disable Synchronized Scrolling</td>
<td>Enables or disables the synchronized scrolling of the comparison source and comparison destination panes.</td>
</tr>
<tr>
<td></td>
<td>Alt+↓</td>
<td>Next Difference</td>
<td>Moves the cursor from its current position (in a results display screen with differences) to the next difference at a lower position.</td>
</tr>
<tr>
<td></td>
<td>Alt+↑</td>
<td>Previous Difference</td>
<td>Moves the cursor from its current position (in a results display screen with differences) to the previous difference at a higher position.</td>
</tr>
<tr>
<td></td>
<td>Alt+Home</td>
<td>Initial Difference</td>
<td>Moves the cursor to the difference at the highest position in the window (in a results display screen with differences).</td>
</tr>
<tr>
<td></td>
<td>Alt+End</td>
<td>Final Difference</td>
<td>Moves the cursor to the difference at the lowest position in the window (in a results display screen with differences).</td>
</tr>
<tr>
<td></td>
<td>Alt+←</td>
<td>Copy to Comparison Source</td>
<td>Copies the content of the difference at the cursor position in the comparison destination pane to the comparison source.</td>
</tr>
<tr>
<td></td>
<td>Ctrl+7</td>
<td>Update with Latest Information</td>
<td>Recompares the same screen and file currently in the comparison results display, and refreshes the results screen.</td>
</tr>
</tbody>
</table>
E7.1.2 Optional File Comparison Settings

You can specify optional settings, such as comparison method and display method, for the File Comparison function.

This subsection describes the procedure regarding environmental setup for the File Comparison function. The relevant procedure is given below.

◆ Procedure ◆

(1) Select [Tools]-[Set up Environment] from the menu bar.
⇒ The Set up Environment dialog box opens.

(2) Click the [File Comparison] tab.

(3) Enter the optional settings for the File Comparison function, and click the [OK] button.
The optional File Comparison settings are shown below.

- **Comparison Method**
  You can specify whether to perform a comparison of devices using the conditions of the character string for the entered parameter with no change, or convert to the actual address indicated by the parameter before comparing, when comparing programs (blocks/macros).

- **Comparison Target**
  You can specify whether to set the comment type as a comparison target when comparing programs (blocks/macros).

- **Comparison Source Location**
  You can specify the display location for the panes in the comparison results window.

- **Color**
  You can specify the display color for comparison results in the comparison results window.
E7.2 Compare Project

With the Compare Project function, two projects are specified and the content of all files included in both projects are compared.

E7.2.1 Compare Project

The target files and comparison details for the Compare Project function are shown in the table below.

Table E7.3 Targets for the Compare Project Function

<table>
<thead>
<tr>
<th>File Type</th>
<th>Comparison Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component Definition</td>
<td>Compares block components defined in the component definition, and determines if they are the same or different.</td>
</tr>
<tr>
<td>Properties</td>
<td>Compares the content of project properties, such as CPU type and project titles, and determines if they are the same or different.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Compares configuration settings, and determines if they are the same or different.</td>
</tr>
<tr>
<td>User Log Message</td>
<td>Compares user log messages settings, and determines if they are the same or different. If there is only one user log messages file, No Comparison Destination/No Comparison Source is displayed.</td>
</tr>
<tr>
<td>Common Tag Name Definition</td>
<td>Compares the content of common tag name definitions, and determines if they are the same or different.</td>
</tr>
<tr>
<td>Constant Definition</td>
<td>Compares the content of constant definitions, and determines if they are the same or different. If there is only one constant definition file, No Comparison Destination/No Comparison Source is displayed.</td>
</tr>
<tr>
<td>Block</td>
<td>Compares programs of blocks with the same file name in projects, and determines if they are the same or different. If there is only one block with the same name, No Comparison Destination/No Comparison Source is displayed.</td>
</tr>
<tr>
<td>Block Properties</td>
<td>Compares properties of blocks with the same file name in projects, and determines if they are the same or different. If there is only one block with the same name, No Comparison Destination/No Comparison Source is displayed.</td>
</tr>
<tr>
<td>Block Tag Name Definition</td>
<td>Compares tag name definitions of blocks with the same file name in projects, and determines if they are the same or different. If there is only one block with the same name, No Comparison Destination/No Comparison Source is displayed.</td>
</tr>
<tr>
<td>Macro</td>
<td>Compares programs of macros with the same file name in projects, and determines if they are the same or different. If there is only one macro with the same name, No Comparison Destination/No Comparison Source is displayed.</td>
</tr>
<tr>
<td>Macro Properties</td>
<td>Compares properties of macros with the same file name in projects, and determines if they are the same or different. If there is only one macro with the same name, No Comparison Destination/No Comparison Source is displayed.</td>
</tr>
<tr>
<td>Macro Tag Name Definition</td>
<td>Compares tag name definitions of macros with the same file name in projects, and determines if they are the same or different. If there is only one macro with the same name, No Comparison Destination/No Comparison Source is displayed.</td>
</tr>
</tbody>
</table>

SEE ALSO

For details on the configuration, see Section D3.1, "Building a Project."

CAUTION

This function cannot compare blocks, block properties, block tag name definitions, macros, macro properties, and macro tag name definitions that refer to the common library or another project.
Compare Project Procedure

This subsection describes the Compare Project procedure. The relevant procedure is given below.

◆ Procedure ◆

(1) Make sure the project to be used as the comparison source is open.
(2) Select [File]-[Compare Project] from the menu bar.

TIP
You can also perform startup from File Comparison Bar and the Project Window.

⇒ The Start Project Comparison dialog box opens.

(3) Specify the project to be used as the comparison destination for Project in the Comparison Destination group box. Click the [Browse] button.

TIP
You can select the projects that have been compared in the past from the Project list.

⇒ The Select Project dialog box opens.

(4) Specify the project file to be used as the comparison destination, and click the [Open] button.
(5) Specify the target for project comparison in Comparison Target, and click the [OK] button.

TIP
When you turn on the [Block] checkbox in Comparison Target, if you turn on the [All Blocks with Component Definitions] option button, the [Compare by Component Definition Registration] checkbox becomes enabled. If you want to compare in component definition order, turn on the [Compare by Component Definition Registration] checkbox.

⇒ Project comparison is performed, and the Project Comparison Results List opens.
E7.2.2 Project Comparison Results List

After comparing each file in a project, the results of the Compare Project function are displayed in a list as shown below.

![Project Comparison Results List Screen](E0702_04.VSD)

- **A Title Bar**
  Displays the window title and the comparison source/comparison destination project file names.

- **B Project Folder Name**
  Displays the path for the project folder for the comparison source/comparison destination.

- **C Comparison Results List**
  Displays a list of comparison results for each file in the project.

- **D Number of Differences**
  Displays the number of files in the comparison results with differences (Do not Match, No Comparison Destination, No Comparison Source) as a number of differences.

- **E [Help] Button**
  Displays help information for the Comparison function.

- **F Comparison Time**
  Displays the time when the Compare Project function was executed.
Content Displayed in the Project Comparison Results List

The content displayed in the Project Comparison Results List is shown in the table below.

Table E7.4 Project Comparison Results List Details

<table>
<thead>
<tr>
<th>Item</th>
<th>Display Content and Tips</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>File Name</strong></td>
<td>Displays the file type or file name of the file compared.</td>
</tr>
<tr>
<td><strong>File Type</strong></td>
<td>Since the following files are normally the only files included in a project, they are shown as the names of file types.</td>
</tr>
<tr>
<td></td>
<td>• Component Definition</td>
</tr>
<tr>
<td></td>
<td>• Properties</td>
</tr>
<tr>
<td></td>
<td>• Configuration</td>
</tr>
<tr>
<td></td>
<td>• User Log Message</td>
</tr>
<tr>
<td></td>
<td>• Common Tag Name Definition</td>
</tr>
<tr>
<td></td>
<td>• Constant Definition</td>
</tr>
<tr>
<td><strong>File Name</strong></td>
<td>Since files related to blocks/macros might or might not be included in a project, they are shown as individual block and macro names.</td>
</tr>
<tr>
<td></td>
<td>• Block</td>
</tr>
<tr>
<td></td>
<td>• Block Properties</td>
</tr>
<tr>
<td></td>
<td>• Block Tag Name Definition</td>
</tr>
<tr>
<td></td>
<td>• Macro</td>
</tr>
<tr>
<td></td>
<td>• Macro Properties</td>
</tr>
<tr>
<td></td>
<td>• Macro Tag Name Definition</td>
</tr>
<tr>
<td><strong>Type</strong></td>
<td>Displays the file type.</td>
</tr>
<tr>
<td><strong>Comparison Results</strong></td>
<td>Displays the comparison results as Match/Do not Match/No Comparison Source/No Comparison Destination.</td>
</tr>
<tr>
<td><strong>Comparison Source Component</strong></td>
<td>Displays the number defined in a project’s component definition for block files in the comparison source project.</td>
</tr>
<tr>
<td></td>
<td>Differences in component definition numbers cannot be comparison targets.</td>
</tr>
<tr>
<td><strong>Comparison Destination Component</strong></td>
<td>Displays the number defined in a project’s component definition for block files in the comparison destination project.</td>
</tr>
<tr>
<td></td>
<td>Differences in component definition numbers cannot be comparison targets.</td>
</tr>
<tr>
<td><strong>Comparison Source Date</strong></td>
<td>Displays the date and time each file in a comparison source project was created (or last modified).</td>
</tr>
<tr>
<td></td>
<td>Differences in creation date/time cannot be comparison targets.</td>
</tr>
<tr>
<td><strong>Comparison Destination Date</strong></td>
<td>Displays the date and time each file in a comparison destination project was created (or last modified).</td>
</tr>
<tr>
<td></td>
<td>Differences in creation date/time cannot be comparison targets.</td>
</tr>
</tbody>
</table>

SEE ALSO

For details on the configuration, see Section D3.1, "Building a Project."
E7.2.3 Operations on the Project Comparison Results List

You can perform the following operations on the project comparison results list.
- Copy Comparison Results
- Display Comparison Results for Component Definitions
- Display Comparison Results for Project Properties
- Display Comparison Results for Configurations
- Display Comparison Results for User Log Messages
- Display Comparison Results for Common Tag Name Definitions
- Display Comparison Results for Constant Definitions
- Display Comparison Results for Blocks
- Display Comparison Results for Block Properties
- Display Comparison Results for Block Tag Name Definitions
- Display Comparison Results for Macros
- Display Comparison Results for Macro Properties
- Display Comparison Results for Macro Tag Name Definitions

SEE ALSO
- For details about program comparison results for blocks/macros and comparison results displays for block/macro tag name definitions, see Section E7.3, “Compare Window”.
- For details on the configuration, see Section D3.1, "Building a Project."

■ Copy Comparison Results

You can copy a project comparison results list to the clipboard as text data.
Since results lists are saved to the clipboard in CSV format, they can be used in text editors.
Display Comparison Results

The comparison source and comparison destination for a file type selected from the project comparison results list can be lined up in the same window to check the comparison results.

This subsection describes the display procedure for comparison results for configurations as an example.

Use the following procedure to display comparison results.

◆ Procedure ◆

1. Make sure that project comparison is performed, and that the Project Comparison Results List is displayed.

2. Double-click the item whose comparison results you want to check.

⇒ The content of the comparison results is displayed.
Details of the Comparison Results Window for Project Setup

The comparison results for configuration are displayed in a table format that compares comparison source and comparison destination, as shown in the figure below.

Example of Operation Control for Configuration

Displayed items that have parts of difference are identified by color.

Table E7.5 Display Colors in the Comparison Screen (Default)

<table>
<thead>
<tr>
<th>Display Color</th>
<th>Display Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>White</td>
<td>Indicates identical content in the comparison source and comparison destination.</td>
</tr>
<tr>
<td>Pink</td>
<td>Indicates a difference between two setting values.</td>
</tr>
<tr>
<td>Light gray</td>
<td>Indicates the absence of an applicable item on one side.</td>
</tr>
<tr>
<td>Dark gray</td>
<td>Indicates the absence of an applicable item in both the comparison source and comparison destination.</td>
</tr>
</tbody>
</table>

TIP

You can modify the display color settings in the environment setup.

You cannot modify the setting items from the comparison results screen. If the settings have been updated in a location other than the comparison screen, either perform the comparison again or select [Update with Latest Information].

SEE ALSO

For details about program comparison results for blocks/macros and comparison results displays for block/macro tag name definitions, see Section E7.3, “Compare Window”.

Figure E7.2 Comparison Results Screen (Example of Operation Control for Configuration)
The following table shows the relationship between the configuration in the Project Settings/Configuration window and the comparison results screen.

### Table E7.6  Relationship of the Items in Configuration Comparison Results and the Project Settings/Configuration Window

<table>
<thead>
<tr>
<th>Configuration Comparison Results</th>
<th>Project Settings/Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device Capacities</td>
<td>[Configuration] - [Device Setup] - [Device Area Setup]</td>
</tr>
<tr>
<td></td>
<td>[Configuration] - [Inter-CPU Shared Memory Setup]</td>
</tr>
<tr>
<td></td>
<td>[Configuration] - [FA Link Setup]</td>
</tr>
<tr>
<td>Script Setup</td>
<td>[Configuration] - [Script Setup]</td>
</tr>
<tr>
<td>Power Failure/Local</td>
<td>[Configuration] - [Device Setup] - [Latch Range Setup at Power Failure]</td>
</tr>
<tr>
<td>Operation Control</td>
<td>[Configuration] - [Error Handling Setup]</td>
</tr>
<tr>
<td></td>
<td>[Configuration] - [Run Operation Setup]</td>
</tr>
<tr>
<td>Initial Data</td>
<td>[Configuration] - [Initial Data Setup]</td>
</tr>
<tr>
<td>DIO Setup</td>
<td>[Configuration] - [Input/Output Setup]</td>
</tr>
<tr>
<td>FA Link</td>
<td>[Configuration] - [FA Link Setup]</td>
</tr>
<tr>
<td>Sampling Trace</td>
<td>[Configuration] - [Sampling Trace Setup]</td>
</tr>
<tr>
<td>Communications Setup</td>
<td>[Configuration] - [Built-in Functions Setup]</td>
</tr>
<tr>
<td>ROM Setup</td>
<td>[Configuration] - [Built-in Functions Setup]</td>
</tr>
<tr>
<td>Interrupt Setup</td>
<td>[Configuration] - [interrupt Setup]</td>
</tr>
<tr>
<td>Shared Refreshing</td>
<td>[Configuration] - [inter-CPU Shared Memory Setup]</td>
</tr>
<tr>
<td>FL-net Refreshing</td>
<td>[Configuration] - [FA Link Setup]</td>
</tr>
<tr>
<td>Function Removal</td>
<td>[Configuration] - [Built-in Functions Setup]</td>
</tr>
</tbody>
</table>

---

**CAUTION**

If using scripts, there is script setup comparison in addition to the configuration comparison described in this manual.
Details of the Comparison Results Window for Common Tag Name Definitions/Constant Definitions

The comparison results for common tag name definitions and constant definitions are displayed in a table format that compares comparison source and comparison destination, as shown in the figure below.

![Figure E7.3 Comparison Results Screen (Example for Common Tag Name Definition)](image)

Since you can specify items as you choose for common tag name definitions and constant definitions, identical items are displayed on the same line. If an item of common tag name definitions exists for only one side, the display is corrected so that line is left blank.

Displayed items that have parts of difference are identified by color.

### Table E7.7 Display Colors in the Comparison Screen (Default)

<table>
<thead>
<tr>
<th>Display Color</th>
<th>Display Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>White</td>
<td>Indicates identical content in the comparison source and comparison destination.</td>
</tr>
<tr>
<td>Pink</td>
<td>Indicates a difference between two setting values.</td>
</tr>
<tr>
<td>Light gray</td>
<td>Indicates the absence of an applicable item on one side.</td>
</tr>
</tbody>
</table>

**TIP**

You can modify the display color settings in the environment setup.

You can edit the setting values for only the comparison source pane in the common tag name definition/constant definition comparison results screen. In addition, you can select the settings for the comparison destination and copy them to the comparison source. To display the comparison results again after editing, perform [Update with Latest Information].

**SEE ALSO**

For details about program comparison results for blocks/macros and comparison results displays for block/macro tag name definitions, see Section E7.3, "Compare Window".

In addition, the operations used on the common tag name definition/constant definition comparison results screen are the same for both block/macro tag name definitions. For details, see Section E7.3.3, "Operations in the Comparison Results Screen".
E7.3 Compare Window

With the Compare Window function, two sets of the same type of file data are specified and the files are compared.

E7.3.1 Compare Window

The files used as comparison targets are shown in the table below.

<table>
<thead>
<tr>
<th>File Type</th>
<th>Comparison Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common Tag Name Definition</td>
<td>Compares the content of common tag name definitions, and determines if they are the same or different. The comparison results are shown in a list comparing comparison source/comparison destination.</td>
</tr>
<tr>
<td>Constant Definition</td>
<td>Compares the content of constant definitions, and determines if they are the same or different. The comparison results are shown in a list comparing comparison source/comparison destination.</td>
</tr>
<tr>
<td>Block</td>
<td>Compares the programs of two block files, and determines if they are the same or different. Displays the comparison source/comparison destination of the comparison results as ladder programs. When starting the comparison, you can select whether to simultaneously compare the block properties. The comparison results of the block properties are shown in a list comparing comparison source/comparison destination.</td>
</tr>
<tr>
<td>Block Tag Name Definition</td>
<td>Compares two block tag name definitions, and determines if they are the same or different. The comparison results are shown in a list comparing comparison source/comparison destination.</td>
</tr>
<tr>
<td>Macro</td>
<td>Compares the programs of two macro files, and determines if they are the same or different. Displays the comparison source/comparison destination of the comparison results as ladder programs. When starting the comparison, you can select whether to simultaneously compare the macro properties. The comparison results of the macro properties are shown in a list comparing comparison source/comparison destination.</td>
</tr>
<tr>
<td>Macro Tag Name Definition</td>
<td>Compares two macro tag name definitions, and determines if they are the same or different. The comparison results are shown in a list comparing comparison source/comparison destination.</td>
</tr>
</tbody>
</table>

Compare Window Procedure

This subsection describes the Compare Window procedure.

The Compare Window function performs a comparison of two files, with the window open on WideField3 as the comparison source and the other specified as the comparison destination.

This subsection describes the procedure for block files. The relevant procedure is given below.

◆ Procedure ◆

(1) Make sure the file to be used as the comparison source is open.
(2) Select [File]-[Compare Window] from the menu bar.

TIP

You can also perform startup from the File Comparison Bar, the Project Window, and the pop-up menu.

⇒ The Start Window Comparison dialog box opens.
(3) Specify the file to be used as the comparison destination for File in the Comparison Destination group box. Click the [Browse] button.

**TIP**
You can select the files that have been compared in the past from the File list.

⇒ The Select File dialog box is displayed.

(4) Specify the file to be used as the comparison destination, and click the [Select] button.

(5) Specify the target for window comparison in Comparison Target, and click the [OK] button.

⇒ Window comparison is performed, and the results window opens.
E7.3.2 Display Details of the Window Comparison Results

The types of window comparison results are shown below.

- Display Comparison Results for Blocks/Macros
- Display Comparison Results for Block/Macro Properties
- Display Comparison Results for Block/Macro Tag Name Definitions
- Display Comparison Results for Common Tag Name Definitions
- Display Comparison Results for Constant Definitions

Details of Comparison Results for Blocks/Macros

The block/macro comparison results are displayed in a format that compares comparison source and comparison destination in ladder programs, as shown in the figure below.

Lines with the same program content are displayed on the same line of the ladder programs in the comparison results screen. If a line with the same content exists for only one side, the display is corrected so that line is left blank.

Displayed items that have parts of difference are identified by color.

Table E7.9 Display Colors in the Comparison Screen (Default)

<table>
<thead>
<tr>
<th>Display Color</th>
<th>Display Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>White</td>
<td>Indicates identical content in the comparison source and comparison destination.</td>
</tr>
<tr>
<td>Pink</td>
<td>Indicates a difference between the content of two programs.</td>
</tr>
<tr>
<td>Light gray</td>
<td>Indicates the absence of an applicable line in one program.</td>
</tr>
</tbody>
</table>
TIP
- You can modify the display color settings in the environment setup.
- When the comparison results screen is opened, the scrolling of the comparison source and comparison destination panes are synchronized. To disable the synchronized scrolling, select [Compare Files]-[Display/Scroll Differences Simultaneously] from the menu bar to remove the corresponding checkmark.

You can perform editing in the same way as with the normal program edit function for only the comparison source pane in the ladder program comparison results screen. In addition, you can select the settings for the comparison destination and copy them to the comparison source.

To display the comparison results again after editing, perform [Update with Latest Information].

SEE ALSO
For details on editing programs in the comparison results screen, see Section E7.3.3, "Operations in the Comparison Results Screen."

Details of Comparison Results for Block/Macro Properties

The comparison results for block/macro properties are displayed in a table format that compares comparison source and comparison destination, as shown in the figure below.

![Comparison Results Screen (Example for Block Properties)](E0703_05.VSD)

Figure E7.5 Comparison Results Screen (Example for Block Properties)

Displayed items that have parts of difference are identified by color.

<table>
<thead>
<tr>
<th>Display Color</th>
<th>Display Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>White</td>
<td>Indicates identical content in the comparison source and comparison destination.</td>
</tr>
<tr>
<td>Pink</td>
<td>Indicates a difference between two setting values.</td>
</tr>
<tr>
<td>Light gray</td>
<td>Indicates the absence of an applicable item on one side.</td>
</tr>
<tr>
<td>Dark gray</td>
<td>Indicates the absence of an applicable item in both the comparison source and comparison destination.</td>
</tr>
</tbody>
</table>

TIP
You can modify the display color settings in the environment setup.

You cannot modify the setting items from the comparison results screen. If the settings have been updated in a location other than the comparison screen, either perform the comparison again or select [Update with Latest Information].
Details of Comparison Results for Block/Macro Tag Name Definitions or Common Tag Name Definitions

The comparison results for block/macro tag name definitions or common tag name definitions are displayed in a table format that compares comparison source and comparison destination, as shown in the figure below.

![Comparison Results Screen (Example for Block Tag Name Definitions)](E0703_06.VSD)

Figure E7.6  Comparison Results Screen (Example for Block Tag Name Definitions)

Since you can specify items as you choose for block/macro tag name definitions or common tag name definitions, identical items are displayed on the same line. If an item exists for only one side, the display is corrected so that line is left blank.

Displayed items that have parts of difference are identified by color.

<table>
<thead>
<tr>
<th>Display Color</th>
<th>Display Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>White</td>
<td>Indicates identical content in the comparison source and comparison destination.</td>
</tr>
<tr>
<td>Pink</td>
<td>Indicates a difference between two setting values.</td>
</tr>
<tr>
<td>Light gray</td>
<td>Indicates the absence of an applicable item on one side.</td>
</tr>
</tbody>
</table>

TIP

You can modify the display color settings in the environment setup.

You can edit the setting values for only the comparison source pane in the block/macro tag name definition or common tag name definition comparison results window. In addition, you can select the settings for the comparison destination and copy them to the comparison source.

To display the comparison results again after editing, perform [Update with Latest Information].
Details of Comparison Results for Constant Definitions

The comparison results for constant definitions are displayed in a table format that compares comparison source and comparison destination, as shown in the figure below.

![Comparison Results Screen (Example of Constant Definitions)](E0703_07.VSD)

Unlike the comparison for tag name definitions, the constant definitions are compared in the order of items currently displayed. Displayed items that have parts of difference are identified by color.

Table E7.12  Display Colors in the Comparison Screen (Default)

<table>
<thead>
<tr>
<th>Display Color</th>
<th>Display Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>White</td>
<td>Indicates identical content in the comparison source and comparison destination.</td>
</tr>
<tr>
<td>Pink</td>
<td>Indicates a difference between two setting values.</td>
</tr>
<tr>
<td>Light gray</td>
<td>Indicates the absence of an applicable item on one side.</td>
</tr>
</tbody>
</table>

**TIP**

You can modify the display color settings in the environment setup.

You can edit the setting values for only the comparison source pane in the constant definition comparison results window. In addition, you can select the settings for the comparison destination and copy them to the comparison source.

To display the comparison results again after editing, perform [Update with Latest Information].
E7.3.3 **Operations in the Comparison Results Screen**

The comparison results screen includes functions for viewing differences detected during a comparison.

This subsection uses the comparison results display screen of ladder programs to describe the operations in the comparison results screen.

You can perform the following operations in the comparison results screen:
- Switch pane
- Modify split display
- Move to difference
- Update with latest information
- Display differences and enable synchronized scrolling

**Switch Pane**

This operation allows you to switch the active pane in the 2-pane structure of the comparison results screen. The relevant procedure is given below.

◆ **Procedure◆**

(1) Make sure the comparison results screen is displayed.

(2) Select [Compare Files]-[Switch Pane] from the menu bar.

⇒ The active pane is switched.

**TIP**

You can also switch the active pane by pressing the [Ctrl] + [Tab] keys.
Modify Split Display

You can select either left/right or up/down as the pane layout in the comparison results display screen. The relevant procedure is given below.

◆ Procedure ◆

(1) Make sure the comparison results screen is displayed.

(2) Select [Compare Files]-[Split Display Horizontally] from the menu bar.
⇒ The pane structure switches to an up/down layout.

TIP
Remove the checkmark from [Split Display Horizontally] to switch to a left/right split display pane structure.

Move to Difference

When there are multiple differences, use this operation to move the cursor to the next difference. The relevant procedure is given below.

◆ Procedure ◆

(1) Make sure the comparison results screen is displayed.
(2) Select [Compare Files]-[Next Difference] from the menu bar.  
⇒ The cursor moves to the next difference.

**TIP**
You can perform similar operations with [Previous Difference], [Initial Difference], and [Final Difference].

⚠️ **CAUTION**

Mnemonic/script editing is not possible using the comparison results screen.

## Update with Latest Information

If there have been modifications to the content while displaying the comparison results, use this function to perform a comparison again and update the screen display. The relevant procedure is given below.

**◆ Procedure ◆**

1. **Procedure**
   
   (1) Make sure the comparison results screen is displayed.

   (2) Select [Compare Files]-[Update with Latest Information] from the menu bar.  
   ⇒ The comparison is performed again, and the screen is updated.

   **TIP**
   In the operation example, the comparison is performed again applying conditions to the comparison source program that are consistent with the comparison destination program.
Display Differences and Enable Synchronized Scrolling

By default, when the comparison results screen is opened, differences are displayed and synchronized scrolling is enabled. In this state, you can edit only blocks in the comparison source.

If you disable displaying differences and synchronized scrolling, you can edit blocks in both the comparison source and comparison destination.

To enable or disable the function of displaying differences and activating synchronized scrolling, select [Compare Files]-[Display/Scroll Differences Simultaneously] from the menu bar.

When this function is enabled, a checkmark is displayed beside the [Display/Scroll Differences Simultaneously] menu item.

This function has the following restrictions:

- **When [Display/Scroll Differences Simultaneously] Is On**
  - Only blocks in the comparison source can be edited.
  - The scrolling of the comparison source and comparison destination panes is synchronized.
  - Differences between the comparison source and comparison destination are displayed.

- **When [Display/Scroll Differences Simultaneously] Is Off**
  - Blocks in both the comparison source and comparison destination can be edited.
  - The scrolling of the comparison source and comparison destination panes is not synchronized.
  - Differences between the comparison source and comparison destination are not displayed.

**CAUTION**

When [Display/Scroll Differences Simultaneously] is off, if you edit the comparison source or comparison destination, you cannot turn on [Display/Scroll Differences Simultaneously].

**TIP**

When [Display/Scroll Differences Simultaneously] is off, programs edited in the comparison source and comparison destination are converted separately.
E8. Printing

This chapter describes how to print programs created with WideField3 and information displayed in windows.

Function Limitations for Each CPU Type

<table>
<thead>
<tr>
<th>CPU Modules</th>
<th>Limitations</th>
<th>SEE ALSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>F3SP22-0S, F3SP28-3S, F3SP38-6S, F3SP53-4S, F3SP58-6S, F3SP59-7S</td>
<td>Constant definitions cannot be used.</td>
<td>E8.4.1</td>
</tr>
<tr>
<td>F3SP66-4S, F3SP67-6S</td>
<td>N/A</td>
<td>-</td>
</tr>
<tr>
<td>F3SP71-4N, F3SP76-7N</td>
<td>N/A</td>
<td>-</td>
</tr>
<tr>
<td>F3SP71-4S, F3SP76-7S</td>
<td>N/A</td>
<td>-</td>
</tr>
</tbody>
</table>

TIP

This section does not describe hardware-dependant limitations for each CPU type, such as the number of available devices.

SEE ALSO

For details on limitations for each CPU type, refer to the user’s manual for each type.
Types of Printing

The Print function supports printing of active windows (Print Screen) and data components of a project (Print Project).

Printing a Screen

The Print Screen function prints the active window. The following windows can be printed.
- Edit Block and Edit Macro windows
- System log display, user log display, and operation log display windows
- Group template definition window
- Block monitor and macro monitor windows
- Alarm display window

TIP

When printing instruction parameters of circuits, addresses allocated to tag names are also printed, but addresses of structure members are not printed.

Printing a Project

The Print Project function batch-prints all types of data components (project information, project settings/configuration, blocks, macros, and device list) of a project.

TIP

When printing instruction parameters of circuits, addresses allocated to tag names are also printed, but addresses of structure members are not printed.
### Print Setup

Prior to printing from WideField3, you can perform setup for printer settings, print target settings, page format settings, print layout settings and detailed settings as required.

- **Printer Setup**
  Sets the printer model, paper size, and paper orientation.

- **Print Target Specification**
  Select either to print a print target selected from project data or to print the current active window.

- **Page Format Settings**
  Sets the output paper, margins, cover, header/footer, and blocks/macros to be outputted.

- **Print Layout Settings**
  Sets the project information, project settings/configuration, and output format.

- **Detailed Print Setup**
  Sets additional information (such as cross references and tag name definitions) for printing circuits, cover text, header and footer print items.

---

**CAUTION**

You can perform printing only when a project is open.

---

### Printing Operations

You can perform printing operations from the [File] menu of the menu bar. You can perform the following printing operations from the menu bar only when the conditions described below are met:

- **[Print Setup]**
  Available when a project is open or when a printable screen is selected.

- **[Print Preview]**
  Available when a project is open or when a printable screen is selected.

- **[Print]**
  Available when a project is open or when a printable screen is selected.

---

**TIP**

Even if you do not open the project, you can print log files when the screen of the log files to print is open.

---

**SEE ALSO**

For details on various log files, see Chapter K3, "Logs and Alarms" (Online).
**E8.1 Printer Setup**

This section describes how to perform printer settings (printer name, paper size and document orientation).

◆ **Procedure◆**

(1) Select [File]–[Print Setup] from the menu.
⇒ The Print Setup dialog box opens.
(2) Click [Printer Setup].

(3) Select a printer model from the list box.

**TIP**  
The printer list displays all printers connected to the computer.

(4) Select the paper size and paper source.

(5) Select the page document by turning on the [Portrait] or [Landscape] option button.

(6) Click [Properties] to perform printer detailed settings as required.

(7) Click [OK].
⇒ The printer settings are enabled.

⚠️ **CAUTION**

Printer setup of a printer cannot be done unless the printer is connected.
E8.2 Print and Print Preview

This section describes the Print and Print Preview functions. The Print function consists of the Print Project function and the Print Screen function. You can use the Print and Print Preview functions from the Print Setup dialog box.

E8.2.1 Print Setup Dialog Box

This subsection describes the Print Setup dialog box.

To open the Print Setup dialog box, while a project is open, select [File] - [Print] from the menu bar.

![Figure E8.1 Print Setup Dialog Box](E0802_01.VSD)

- **A Tree view**
  Allows you to specify print targets and provides various setting menu items.
- **B Read from Default**
  Reads the print setup saved as default. Not only the settings currently displayed in the screen but also all other settings in the tree view are changed to default settings.
- **C Save as Default**
  Saves the current settings as default. You can read the saved settings by clicking the [Read from Default] button. The settings are used when you print a newly created project for the first time.
- **D Print Preview**
  Displays a print preview.
- **E Save Settings**
  Saves the current settings and closes the Print Setup dialog box. The settings are specific for each project. After the settings are saved, when the Print Setup dialog box is opened in the same project, the settings are referenced. Not only the settings currently displayed in the screen but also all other settings in the tree view are saved.
- **F** Print
  Saves all current settings, performs printing, and closes the Print Setup dialog box.

- **G** Printer Setup
  Opens the Print Setup dialog box. In the Print Setup dialog box, you can select the printer to use, paper size, page orientation, and so on.

- **H** Retrieve Page No.
  Gets the output page number of the currently selected print target.

---

**CAUTION**

If the program size is huge, [Retrieve Page No.] takes time.

---

**SEE ALSO**

- For details on printer setup, see Section E8.1, "Printer Setup."
- For details on specifying print targets, see Section E8.2.3, "Printing a Project."
- For details on the page format settings, see Section E8.3, "Page Format Settings."
- For details on print layout settings, see Section E8.4, "Print Layout Settings."

---

**TIP**

You can use the print preview function also by selecting [File]-[Print Preview] from the menu bar.
E8.2.2 Printing a Screen

The following table lists the windows that can be printed using the Print Screen function.

<table>
<thead>
<tr>
<th>Window</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edit Block</td>
<td>Prints the circuits of the block being edited. Converted items are printed.</td>
</tr>
<tr>
<td>Edit Macro</td>
<td>Prints the circuits of the instruction macro being edited. Converted items are printed.</td>
</tr>
<tr>
<td>System log display</td>
<td>Prints the system log being displayed. The log can be read from the file or from the CPU.</td>
</tr>
<tr>
<td>User log display</td>
<td>Prints the user log being displayed. The log can be read from the file or from the CPU.</td>
</tr>
<tr>
<td>Operation log display</td>
<td>Prints the operation log being displayed. The log can be read from the file or from the CPU.</td>
</tr>
<tr>
<td>Group template definition</td>
<td>Prints the group template being edited.</td>
</tr>
<tr>
<td>Alarm display</td>
<td>Prints the alarm monitor currently being displayed.</td>
</tr>
<tr>
<td>Program monitor</td>
<td>Prints the circuits in the monitor window. Blocks and instruction macros can be printed. Printed content is either the same content displayed when the program monitor is started, or content changed in online editing.</td>
</tr>
</tbody>
</table>

To print a screen, use the following procedure.

◆ Procedure ◆

(1) Make the screen to be printed the active window.

(2) Select [File]–[Print] from the menu bar.

⇒ The Print Setup dialog box appears.

(3) In the Print Target Specification screen, turn on the [Print the Currently Active Window] option button.

(4) Click [Print].

⇒ Printing begins.

TIP

- You can check the print image prior to printing by clicking [Print Preview]. If you click the [Print] button or [Print Preview] with a ladder program edit window (tag name definition window) open after having made changes on the window, a confirmation dialog box will be displayed.

- When you print the Edit Block or Edit Macro window or a program monitor window, tag name definitions are not printed.
E8.2.3  Printing a Project

With the Print Project function, you can select data components of a project and perform batch printing. The following table lists items that can be printed.

<table>
<thead>
<tr>
<th>Target Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cover</td>
<td>Prints the cover with a border.</td>
</tr>
<tr>
<td>Project Information</td>
<td>Prints the component blocks of the project in the form of a tree structure.</td>
</tr>
<tr>
<td>Project Settings/Configuration</td>
<td>Prints the project settings/configuration.</td>
</tr>
<tr>
<td>Program (Ladder)</td>
<td>Prints blocks. You can print only programs saved in files.</td>
</tr>
<tr>
<td>Program (Macro)</td>
<td>Prints macros. You can print only programs saved in files.</td>
</tr>
<tr>
<td>Device List</td>
<td>Prints a device list.</td>
</tr>
</tbody>
</table>

### Selecting Print Targets

You can select print targets for printing a project and arrange the print order of the targets.

You can perform these operations in the Print Target Specification screen of the Print Setup dialog box.

- **A** Print Target
  Select either [Print the Currently Active Window] or [Select from Project Data and Print]. If you want to perform project printing, turn on the [Select from Project Data and Print] option button.

- **B** Print target list
  Displays the currently specified print targets. Clicking the [+] button shows blocks and macros that make up the program. Clicking the [-] button hides the blocks and macros. The settings in this pane become enabled when the [Select from Project Data and Print] option button is turned on.
[Output Page]
Displays sequential page numbers for when all print targets currently selected are printed.

- C Up/Down
Changes the print order of the items selected in the print target list.
[Up]
Moves the item currently selected in the print target list up one level.
[Down]
Moves the item currently selected in the print target list down one level.

- D Add/Delete
Add or delete print targets.
[Add]
Adds the item currently selected in the item list at the cursor position in the print target list.
[Delete]
Deletes the item currently selected in the print target list.

- E Non-printed target list
Displays a list of targets not to be printed. The items in this list are not printed.

TIP
By default, all items are specified to be printed.
Adding Print Targets

To add a print target, use the following procedure.

Procedure

1. Confirm that a project is opened.
2. Select [File]–[Print] from the menu bar.
   ⇒ The Print Setup dialog box is displayed.
3. Click [Print Target Specification] in the tree.
   ⇒ The Print Target Specification screen is displayed.
4. Turn on the [Select from Project Data and Print] option button.
5. From the item list, select the item to be added to the print target list.
6. Click [Add].

⇒ The selected item is included in the print targets and added to the print target list.
Deleting Print Targets
To delete a print target, use the following procedure.

Procedure
(1) Confirm that a project is opened.
(2) Select [File]–[Print] from the menu bar.
   ⇒ The Print Setup dialog box is displayed.
(3) Click [Print Target Specification] in the tree.
   ⇒ The Print Target Specification screen is displayed.
(4) Turn on the [Select from Project Data and Print] option button.
(5) From the print target list, select the item to be deleted from the print target list.
(6) Click [Delete].
   ⇒ The selected item is deleted from the print target list and added to the item list.

Steps (5) and (6)
Changing the Order of Print Targets
To change the order of print targets, use the following procedure.

◆ Procedure ◆

(1) Confirm that a project is opened.
(2) Select [File]–[Print] from the menu bar.
⇒ The Print Setup dialog box is displayed.
(3) Click [Print Target Specification] in the tree.
⇒ The Print Target Specification screen is displayed.
(4) Turn on the [Select from Project Data and Print] option button.
(5) From the print target list, select the item whose print order is to be changed.
(6) To move the selected item up one level in the print order, click [Up]. To move it down one level, click [Down].

⇒ The print order is changed.
**Procedure for Using Print Project Function**

The procedure for printing a project is given below.

◆ **Procedure◆

(1) Confirm that a project is opened.
(2) Select [File]–[Print] from the menu bar.
⇒ The Print Setup dialog box appears.
(3) Turn on the [Select from Project Data and Print] option button in the Print Target Specification screen.
(4) Select print targets and specify the print order.
(5) Click [Page Format Settings] to configure the page layout settings.

SEE ALSO
For details on the page layout settings, see Section E8.3, "Page Format Settings."

(6) Click [Print Layout Settings] to configure the print layout settings.

SEE ALSO
For details on print layout settings, see Section E8.4, "Print Layout Settings."

(7) Click [Print].
⇒ Printing begins.
TIP
You can check the print image prior to printing by clicking on [Print Preview].

CAUTION
If some of the blocks to be printed are protected, the Enter Password dialog box appears with a list of protected blocks. Enter all the passwords and click [OK] to begin printing.

Figure E8.3  Enter Password Dialog Box
E8.2.4 Procedure for Previewing Print Image

You can click [Print Preview] in the Print Setup dialog box to switch to the Print Preview window that displays an image of the printout.

Figure E8.4  Print Preview

The following table summarizes the functions of the Print Preview buttons.

Table E8.4  Functions of Buttons in the Print Preview Window

<table>
<thead>
<tr>
<th>Button</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Print</td>
<td>Cannot function.</td>
</tr>
<tr>
<td>Next Page</td>
<td>Displays the print image for the next page.</td>
</tr>
<tr>
<td>Prev Page</td>
<td>Displays the print image for the previous page.</td>
</tr>
<tr>
<td>Two Page</td>
<td>Displays the print image for two pages at a time.</td>
</tr>
<tr>
<td>Zoom In</td>
<td>Enlarges the image on the display.</td>
</tr>
<tr>
<td>Zoom Out</td>
<td>Reduces the image on the display.</td>
</tr>
<tr>
<td>Close</td>
<td>Closes the print preview window.</td>
</tr>
</tbody>
</table>

*1: Clicking the [Two Page] button toggles between one-page and two-page layout.

You may click on [Close] to close the Print Preview window and return to the Print Setup dialog box.

TIP

WideField3 R2 or earlier does not support the function for printing from the Print Preview window.
E8.3 Page Format Settings

This section describes how to set up a page format. The Print function prints contents with a layout defined in page format settings.

You can set the following items in page format settings:

- Configuring paper settings and margins
- Editing a cover
- Editing headers and footers
- Configuring program output settings

You can perform page format settings in the Print Setup dialog box.

To open the Print Setup dialog box, select [File]–[Print] from the menu bar.
E8.3.1 Configuring Paper Settings and Margins

You can configure paper settings, margins, and page numbers.

The following figure shows the page layout.

![Page Layout Diagram](image1)

Figure E8.5   Page Layout

To configure paper settings, margins, and page numbers, select [Page Format Settings]-[Paper/Margins Settings] from the tree on the left side of the Print Setup dialog box.

![Print Setup Dialog](image2)

Figure E8.6   Paper/Margins Settings

- **A** Paper
  Displays the paper type set in the Print Setup dialog box.

- **B** Orientation
  Displays the orientation set in the Print Setup dialog box.

- **C** Margins
  Specify the top, bottom, left, and right margins with the spin buttons. Each margin value can be from 0 to 30 mm.

- **D** Page Number
  [Format]
  Select [None], [Sequential Number], or [Item-Page Number]. [None] prints no page numbers. [Sequential Number] prints sequential numbers on each page. [Item-Page Number] prints page numbers with the “XX-YY” format where “XX” is an item number and “YY” is the page number within the item. [Item-Page Number] is enabled only for project printing. The default setting is [None].
Select [Left], [Center], or [Right] for the print position of the page number. The default setting is [Center].

Set the page number to be printed on the first page. For sequential numbering format, enter the initial value for the page number. For Item-page number format, enter the initial value for the item number. The default setting is 1. You can set a page number between 1 and 100.

SEE ALSO
For details on printer setup, see Section E8.1, "Printer Setup."
E8.3.2 Editing a Cover

A cover is printed based on the settings you edited. The following figure shows the cover layout.

![E8.7 Cover Layout](Image)

To edit a cover, select [Page Format Settings]-[Edit Cover] from the tree on the left side of the Print Setup dialog box.

![E8.8 Edit Cover](Image)

- **A** Project Title
  Turn on the checkbox to print the project title.
- **B** Document No.
  Turn on the checkbox to print the document number.
- **C** Document No. text box
  Enter the string to be displayed in the document number area. You can enter up to 64 characters for a document number.
- **D** Date
  Turn on the checkbox to print the date.
  Date to be printed
  Select either [Print Project Update Date] or [Print with Date Printed] for the date to be printed.
- E  Additional Information
  Turn on the checkbox to print additional information.
- F  Additional information text box
  Enter the string to be displayed in the additional information area. The additional information can be set for up to 64 characters horizontally and 16 lines vertically. To start a new line, press the [Enter] key. Line and character format attributes cannot be used.

Each item has the same print format settings. The following setting items are available:

- Select Font
  Select [(Env. setup display font)], [(Print settings default)], or [(Set)] for the fonts for printing. Selecting [(Set)] opens the Font dialog box.
- Output Font
  Displays the fonts for printing.
- Size
  Displays the character size for printing.
- Style
  Displays the character style for printing.
- Alignment
  Select the print position from the drop down list.

⚠️ CAUTION

If the characters you entered do not fit the area of the project title, document number, additional information, or date, modify the font settings or line break positions.
E8.3.3 Editing Headers and Footers

The headers and footers are printed based on the settings you edited. The following figure shows the page layout.

![Diagram showing page layout with headers, footers, margins, and print area]

Figure E8.9 Page Layout

To edit the header and footer, select [Page Format Settings]-[Header/Footer] from the tree on the left side of the Print Setup dialog box.

![Print Setup dialog box with header and footer settings]

Figure E8.10 Header/Footer

- **A** Print Header
  
  Turn on the checkbox to print the headers.

- **B** Header text box
  
  Enter the string to be displayed in the header area. The header can be set for up to 108 characters horizontally and 8 lines vertically. To start a new line, press the [Enter] key. Line and character format attributes cannot be used in headers.

- **C** Print Footer
  
  Turn on the checkbox to print the footers.

- **D** Footer text box
  
  Enter the string to be displayed in the footer area. The footer can be set for up to 108 characters horizontally and 8 lines vertically. To start a new line, press the [Enter] key. Line and character format attributes cannot be used in footers.
Each item has the same print format settings. The following setting items are available:

- **Select Font**
  Select [(Env. setup display font)], [(Print settings default)], or [(Set)] for the fonts for printing. Selecting [(Set)] opens the Font dialog box.

- **Output Font**
  Displays the fonts for printing.

- **Size**
  Displays the character size for printing.

- **Style**
  Displays the character style for printing.

- **Alignment**
  Select the print position from the drop down list.

⚠️ **CAUTION**

If the characters you entered do not fit the header/footer area, modify the font settings or line break positions.

---

**TIP**

The following text strings can be entered in the header or footer to insert various data items (the text strings are not case-sensitive).

- **&DATE** Prints the current date. Example: 07/21/1998.
- **&TIME** Prints the current time. Example: 15:30.
- **&CPU** Prints the name of the selected CPU. Example: F3SP21-0N.
  This item is valid for the Print Project function only. It is ignored by the Print Screen function.
E8.3.4 Configuring Program Output Settings

To configure program output settings, select [Page Format Settings]-[Program] from the tree on the left side of the Print Setup dialog box.

- **A Block tab and Macro tab**
  Switch between the program output setting screen for blocks and that for macros.

- **B Detailed setting area for program output**
  [Select]
  Turn on the checkboxes to print the blocks or macros.
  [Component]
  Displays the order of the component definitions. [SCB] is displayed for sensor control blocks.
  [Block Name]/[Macro Name]
  Displays block/macro names.
  [Comment]
  Turn on the checkboxes to print circuit comments, subcomments, and I/O comments.
  [Tag Name]
  Turn on the checkboxes to print the tag names for devices used in the program.
  [Cross-Reference]
  WideField3 R2 or earlier does not support a function for printing a circuit screen with cross references.
  [Print Range]
  Enter the print range for each block or macro. To print all contents of the relevant block or macro, leave the setting blank. To specify pages to be printed, enter "Pnn-mm" where "nn" and "mm" are the print start and end page numbers, respectively. To specify lines to be printed, enter "Lnn-mm" where "nn" and "mm" are the print start and end line numbers, respectively.
TIP
When you specify a page range for [Print Range], enter page numbers counted from the beginning of each block or macro.

You can specify a print range as follows.
- To print all contents
  Leave the print range text box empty.
- To print a specified page range
  Enter a page number or page range prefixed by the letter 'P' or 'p'. 'p' is automatically capitalized to 'P'.
  Example: To print page 99, enter as follows.
  P99 or p99
  Example: To print pages 1 to 10, enter as follows.
  P1-10 or p1-10
- To print a specified line range
  Enter a line number or line range prefixed by the letter 'L' or 'l'. 'l' is automatically capitalized to 'L'.
  Example: To print line 99, enter as follows.
  L99 or l99
  Example: To print lines 1 to 100, enter as follows.
  L1-L100 or l1-l100

Cross references are printed when a device list is printed. To print a device list, select the device list as a print target.

To specify whether or not to print balloon comments, select [Balloon Comment/Hidden Circuit] in block/macro settings.

TIP
If printing a screen from block or macro editing, the following program output settings are valid.
- Regardless of whether there is a check in the [Select] settings, the settings for the block or macro to print the screen for are valid.
- Regardless of whether there is a check in the [Comment] or [Tag Name] settings, printing is done as if there is a check.
- The set values for [Print Range] are valid.

SEE ALSO
- For actual print examples, see Section E8.5, "Print Layout."
- For details on selecting print targets, see Section E8.2.3, "Printing a Project."
- For details on configuring device list settings, see Section E8.4.4, "Device List Settings."
- For details on configuring block/macro settings, see Section E8.4.3, "Block/Macro Settings."
- For details on printing a screen, see Section E8.2.2, "Printing a Screen."
E8.4 Print Layout Settings

This section describes how to set up a print layout. The Print function prints contents with a layout defined in print layout settings.
In print layout settings, you can configure settings related to the following items:
- Project information
- Project settings/configuration
- Blocks/macros
- Device list

Before setting up a print layout, open the Print Setup dialog box. To open the Print Setup dialog box, select [File]–[Print] from the menu bar.
E8.4.1 Project Information Settings

To configure project information settings, select [Print Layout Settings]-[Project Information] from the tree on the left side of the Print Setup dialog box.

![Print Setup Dialog]

- **A** Print format settings
  - [Select Font]
    Select [(Env. setup display font)], [(Print settings default)], or [(Set)] for the fonts for printing project information. If you select [(Set)], the Font dialog box is displayed for selecting any fonts.
  - [Output Font]
    Displays the fonts for printing.
  - [Size]
    Displays the character size for printing.
  - [Style]
    Displays the character style for printing.

- **B** Output item settings
  - [Project Components]
    Turn on the checkbox to print the list of blocks and macros that make up the project.
  - [Constant Definition]
    Turn on the checkbox to print constant definitions.

⚠️ **CAUTION**

Selecting [Project Components] prints only blocks registered in the component definition and macros used in the program.

**SEE ALSO**

For an actual print example of project information, see Section E8.5.1, "Print Layout for Project Information."
E8.4.2 Project Settings/Configuration

To configure the settings for project settings/configuration, select [Print Layout Settings]-[Project Settings/Configuration] from the tree on the left side of the Print Setup dialog box.

Figure E8.13   Settings for Project Settings/Configuration

- **A** Print format settings
  - [Select Font]
    Select [(Env. setup display font)], [(Print settings default)], or [(Set)] for the fonts for printing. Selecting [(Set)] opens the Font dialog box.
  - [Output Font]
    Displays the fonts for printing.
  - [Size]
    Displays the character size for printing.
  - [Style]
    Displays the character style for printing.

- **B** Initial Data Setting Table Style
  - [Print Grid Lines]
    Turn on the checkbox to print the initial data setting table with a border.
  - [Automatic]
    Turn on the radio button to automatically set the number of rows and number of columns for the initial data setting table to be printed.
  - [Specify Lines/Columns]
    Turn on the radio button to specify the number of rows and number of columns between 0 and 99 using the spin buttons.

**TIP**

If lines to be printed do not fit the specified paper width or if 0 is specified, the table is printed with the automatically calculated number of rows and number of columns.
- C. Project settings/configuration

Set the style of project settings/configuration. The items whose checkboxes are turned off are not printed.

[Device Capacities]
Turn on the checkbox to print device capacities.

[Script Device Assignment]
Turn on the checkbox to print script device assignment.

[Inter-CPU Shared Memory Setup]
Turn on the checkbox to print the settings of inter-CPU shared memory setup.

[FA Link Setup]
Turn on the checkbox to print the settings of FA link setup.

[FL-net Refresh Setup]
Turn on the checkbox to print the settings of FL-net refreshing setup.

[Input/Output Setup]
Turn on the checkbox to print the settings of I/O setup.

[Error Handling Setup]
Turn on the checkbox to print the settings of error handling setup.

[Interrupt Setup]
Turn on the checkbox to print the settings of interrupt setup.

[Run Operation Setup]
Turn on the checkbox to print the settings of operation setup.

[Built-in Functions Setup]
Turn on the checkbox to print the settings of built-in function setup.

[Sampling Trace Setup]
Turn on the checkbox to print the settings of sampling trace setup.

[Initial Data Setup]
Turn on the checkbox to print the settings of initial data setup.

[User Log Message]
Turn on the checkbox to print user log message.

[Specify Unit] button
Opens the Specify Unit dialog box.

⚠️ CAUTION

If using scripts, you can specify [Script Setup] printing enable/disable in addition to the project settings/configuration indicated in this manual.
Specify Unit Dialog Box

When you print the settings of I/O setup, you can select the units to be printed. Use the Specify Unit dialog box to select the units to be printed.

To open the Specify Unit dialog box, click the [Specify Unit] button in the [Project Settings/Configuration] group box.

![Specify Unit Dialog Box](E0804_03.VSD)

Figure E8.14  Specify Unit Dialog Box
E8.4.3 Block/Macro Settings

To configure block/macro settings, select [Print Layout Settings]-[Block/Macro] from the tree on the left side of the Print Setup dialog box.

![Print Setup Dialog]

**Figure E8.15 Block/Macro Settings**

- **A** Print method setting
  - [Compact printing]
    Turn on the checkbox to perform compact printing.
  - [B/W Printing]
    Turn on the checkbox to perform monochrome printing.
- **B** Print setting item list
  Select a print setting item to set.
- **C** Detailed print settings
  Configure detailed print method settings for the item selected in B.

**CAUTION**

In [Print Layout Settings]-[Block/Macro], you can also set up script print settings, in addition to the settings explained in this manual. For details on script print setup, see the volume entitled "Script."
**Compact Printing**

You can select whether to use compact printing. Compact printing prints pages in a vertically compact format by removing unnecessary blank lines from the circuits to be printed. By using compact printing, you can print more circuit information on each page and reduce the number of pages to be printed.

![Figure E8.16 Output when Using Compact Printing](E0804_05.VSD)

![Figure E8.17 Output when Not Using Compact Printing](E0804_06.VSD)

**CAUTION**

- If you perform compact printing for a program saved with errors, the program may not be printed correctly.
- Among the print setting items, [Circuit/Symbol], [Parameter], and [Label] are always printed.

**SEE ALSO**

For an actual example of printing blocks or macros, see Section E8.5.3, "Print Layout for Circuits."

**B/W Printing**

You can print in black and white.
The following shows the setting screen for each item.

**Circuit/Symbol**

Figure E8.18  Circuit/Symbol

- A Print
  Circuits and symbols are always printed.
- B Color
  Select the color for printing. You can select [(Environment setup display color)], [(Default settings display color)], or [(Set)]. Selecting [(Set)] opens the Color dialog box.
- C Print format settings
  [Select Font]
  Select [(Env. setup display font)], [(Print settings default)], or [(Set)] for the fonts for printing. Selecting [(Set)] opens the Font dialog box.
  [Output Font]
  Displays the fonts for printing.
  [Size]
  Displays the font size for printing.
  [Style]
  Displays the font style for printing.
  [Alignment]
  Select the print position from the drop down list.

**TIP**
If you select [B/W Printing], the color setting is ignored when printing is performed.
- **A** Print
  Parameters are always printed.

- **B** Color
  Select the color for printing. You can select [(Environment setup display color)], [(Default settings display color)], or [(Set)]. Selecting [(Set)] opens the Color dialog box.

- **C** Print format settings
  - [Select Font]
    Select [(Env. setup display font)], [(Print settings default)], or [(Set)] for the fonts for printing. Selecting [(Set)] opens the Font dialog box.
  - [Output Font]
    Displays the fonts for printing.
  - [Size]
    Displays the character size for printing.
  - [Style]
    Displays the character style for printing.
  - [Alignment]
    Select the print position from the drop down list.

**TIP**

If you select [B/W Printing], the color setting is ignored when printing is performed.
I/O Comment

- A Print
  Turn on the checkbox to print I/O comments.

- B Color
  Select the color for printing. You can select [(Environment setup display color)], [(Default settings display color)], or [(Set)]. Selecting [(Set)] opens the Color dialog box.

- C Print format settings
  [Select Font]
  Select [(Env. setup display font)], [(Print settings default)], or [(Set)] for the fonts for printing. Selecting [(Set)] opens the Font dialog box.
  [Output Font]
  Displays the fonts for printing.
  [Size]
  Displays the character size for printing.
  [Style]
  Displays the character style for printing.
  [Alignment]
  Select the print position from the drop down list.

- D Number of Output Lines
  Specify the number of lines using the spin button for when I/O comments are printed. You can specify the number of lines between 1 and 4.
Line Number

Figure E8.21  Line Number

- **A** Print
  The checkbox for line numbers is always turned on.

- **B** Color
  Select the color for printing. You can select [(Environment setup display color)], [(Default settings display color)], or [(Set)]. Selecting [(Set)] opens the Color dialog box.

- **C** Print format settings
  [Select Font]
  Select [(Env. setup display font)], [(Print settings default)], or [(Set)] for the fonts for printing. Selecting [(Set)] opens the Font dialog box.
  [Output Font]
  Displays the fonts for printing.
  [Size]
  Displays the character size for printing.
  [Style]
  Displays the character style for printing.
  [Alignment]
  Select the print position from the drop down list.
- **A** Print
  Labels are always printed.

- **B** Color
  Select the color for printing. You can select [(Environment setup display color)], [(Default settings display color)], or [(Set)]. Selecting [(Set)] opens the Color dialog box.

- **C** Print format settings
  [Select Font]
  Select [(Env. setup display font)], [(Print settings default)], or [(Set)] for the fonts for printing. Selecting [(Set)] opens the Font dialog box.
  [Output Font]
  Displays the fonts for printing.
  [Size]
  Displays the character size for printing.
  [Style]
  Displays the character style for printing.
  [Alignment]
  Select the print position from the drop down list.
**Circuit Comment**

- **A** Print
  
  Turn on the checkbox to print circuit comments.

- **B** Color
  
  Select the color for printing. You can select [(Environment setup display color)], [(Default settings display color)], or [(Set)]. Selecting [(Set)] opens the Color dialog box.

- **C** Print format settings

  [Select Font]
  
  Select [(Env. setup display font)], [(Print settings default)], or [(Set)] for the fonts for printing. Selecting [(Set)] opens the Font dialog box.

  [Output Font]
  
  Displays the fonts for printing.

  [Size]
  
  Displays the character size for printing.

  [Style]
  
  Displays the character style for printing.

  [Alignment]
  
  Select the print position from the drop down list.
**Subcomment**

Figure E8.24  Subcomment

- **A** Print
  Turn on the checkbox to print subcomments.

- **B** Color
  Select the color for printing. You can select [(Environment setup display color)], [(Default settings display color)], or [(Set)]. Selecting [(Set)] opens the Color dialog box.

- **C** Print format settings
  - [Select Font]
    Select [(Env. setup display font)], [(Print settings default)], or [(Set)] for the fonts for printing. Selecting [(Set)] opens the Font dialog box.
  - [Output Font]
    Displays the fonts for printing.
  - [Size]
    Displays the character size for printing.
  - [Style]
    Displays the character style for printing.
  - [Alignment]
    Select the print position from the drop down list.

**Balloon Comment/Hidden Circuit**

Figure E8.33  Balloon Comment/Hidden Circuit
- **A** Print Balloon Comments
  Turn on the checkbox to print balloon comments.

- **B** Print Color
  Select [Set Color] or [B/W] for the color for printing balloon comments. If you select [Set Color], each balloon comment is printed in a color specified for the balloon comment.

- **C** Print Conditions for Hidden Circuits
  Select [Print Hidden Circuits] or [Do not Print Hidden Circuits] for the print layout of hidden circuits. By default, the [Do not Print Hidden Circuits] option button is turned on.

The following figure shows an example of printing with hidden circuits.

![Figure E8.34 Printing with Hidden Circuits](image)

(If necessary, this line indicates a hidden circuit.)
Figure E8.35  Grid

- **A Print**
  Turn on the checkbox to print grids.

- **B Color**
  Select the grid color. You can select [(Environment setup display color)], [(Default settings display color)], or [(Set)]. Selecting [(Set)] opens the Color dialog box.
E8.4.4 Device List Settings

To configure device list settings, select [Print Layout Settings]-[Device List] from the tree on the left side of the Print Setup dialog box.

- A Print Conditions
  - [Print in Address Order]
    - Turn on the option button to print in the order of addresses.
  - [Unused addresses registered in common tag name definitions are also printed]
    - Turn on the checkbox to print a device list with addresses that are registered in the common tag name definition but not used in the program.
  - [Print in Tag Name Order]
    - Turn on the option button to print in the order of tag names.
  - [Device Type/Range Setup]
    - Opens the output device selection dialog box.

- B Print Column Setup
  - [Address]
    - The address column is always printed.
  - [Tag Name]
    - The tag name column is always printed.
  - [I/O Comment]
    - Turn on the checkbox to print the I/O comment column.
[Cross-Reference]
Turn on the checkbox to print cross references in the device list. When cross references are output, block names and line numbers are always output. Turn on the checkboxes for column numbers and instruction names to print them.

**TIP**
Currently active I/O comments are printed.

**SEE ALSO**
For details on assigning multiple I/O comments, see Section E2.8, "Multiple I/O Comments."

- **C Table Style**
  Specify the number of rows and number of columns for the table to be printed between 0 and 99 using the spin buttons.
  [Print Display Frame]
  Turn on the checkbox to print the border of the table.

**TIP**
- The term "column" in the table style does not refer to the column of each item specified in [Print Column Setup] but refers to a set of columns of all specified items. Thus, for example, if [I/O Comment] and [Cross-Reference] are turned off for [Print Column Setup], the two columns [Tag Name] and [Address] are considered as one column when printed. Also, when cross references are output, the column number is always printed as 1.
- If the rows or columns do not fit the specified paper width or if 0 is specified, the table is printed with the automatically calculated number of rows and number of columns.

- **D Font**
  [Select Font]
  Select [(Env. setup display font)], [(Print settings default)], or [(Set)] for the fonts for printing. Selecting [(Set)] opens the Font dialog box.
  [Output Font]
  Displays the fonts for printing.
  [Size]
  Displays the character size for printing.
  [Style]
  Displays the character style for printing.

**SEE ALSO**
For an actual example of printing a device list, see Section E8.5.5, "Print Layout for Device List."

---

**Output Device Selection Dialog Box**
You can print a device list with selected devices or a specified device range.
You can perform these settings on the output device selection dialog box.
To open the output device selection dialog box, click the [Device Type/Range Setup] button.
The following figure shows the setting screen for all devices.
Figure E8.37  Setting Screen for All Devices

If you want to specify the range of addresses to be printed for each device, click the relevant device type tab. The basic operations are the same between device tabs. Here we describe the example of data registers (D).

The following figure shows an example in which only the data registers D1 to D100 and D10001 to D10100 are specified to be printed.

Figure E8.38  Setting Screen for Each Device
E8.5 Print Layout

This section describes the print layout for the major print items.

E8.5.1 Print Layout for Project Information

This subsection describes the print layout for project information.

- **Requirements for Printing**

  When [Select from Project Data and Print] is selected in the Print Target Specification screen of the Print Setup dialog box and [Project Information] is selected for the print target.

- **Print Contents**

  The contents are printed based on the settings configured in the Project Information screen, which can be opened by selecting [Print Layout Settings]-[Project Information] in the Print Setup dialog box.

![Figure E8.39 Print Layout for Project Components](E0805_01.VSD)
E8.5.2 Print Layout for Project Settings/Configuration

This subsection describes the print layout for project settings/configuration.

● Requirements for Printing

When [Select from Project Data and Print] is selected in the Print Target Specification screen of the Print Setup dialog box and [Project Settings/Configuration] is selected as a print target.

● Print Contents

The contents are printed based on the settings configured in the Project Settings/Configuration screen, which can be opened by selecting [Print Layout Settings]- [Project Settings/Configuration] in the Print Setup dialog box.

---

Figure E8.40 Print Layout for Project Settings/Configuration
E8.5.3 Print Layout for Circuits

This subsection describes the print layout for circuits.

- **Requirements for Printing**
  - When [Select from Project Data and Print] is selected in the Print Target Specification screen of the Print Setup dialog box and a program (ladder) or macro is selected as a print target.
  - When [Print the Currently Active Window] is selected in the Print Target Specification screen of the Print Setup dialog box opened while the Edit Block or Edit Macro window is open.

**TIP**

When a screen with circuits is printed, the circuits and tag name definitions currently being edited are printed.

- **Print Contents**

The contents are printed based on the settings configured in the Block/Macro screen, which can be opened by selecting [Print Layout Settings]-[Block/Macro] in the Print Setup dialog box.

The following figure shows a print layout obtained when [Select from Project Data and Print] is selected in the Print Target Specification screen.

![Figure E8.41 Print Layout for Circuits](E0805_03.VSD)

**CAUTION**

Constant values mapped to constant names are printed up to a maximum of 8 characters.
E8.5.4 **Print Layout for System Log**

This subsection describes the print layout for the system log.

- **Requirements for Printing**
  
  When [Print the Currently Active Window] is selected in the Print Target Specification screen of the Print Setup dialog box opened while the system log display window is open.

- **Print Contents**
  
  The following figure shows a print layout of the system log.

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Message</th>
<th>Code</th>
<th>Block Name</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>2011/05/24</td>
<td>15:52:25</td>
<td>Startup completed</td>
<td>01-00</td>
<td>SLOT1E001</td>
<td></td>
</tr>
<tr>
<td>2011/05/24</td>
<td>15:51:50</td>
<td>Power Off</td>
<td>02-00</td>
<td>SLOT1E001</td>
<td></td>
</tr>
<tr>
<td>2011/05/24</td>
<td>15:51:35</td>
<td>Startup completed</td>
<td>01-00</td>
<td>SLOT1E001</td>
<td></td>
</tr>
<tr>
<td>2011/04/28</td>
<td>17:41:30</td>
<td>Power Off</td>
<td>03-00</td>
<td>SLOT1E001</td>
<td></td>
</tr>
<tr>
<td>2011/04/28</td>
<td>09:20:26</td>
<td>Startup completed</td>
<td>01-00</td>
<td>SLOT1E001</td>
<td></td>
</tr>
<tr>
<td>2011/04/28</td>
<td>20:21:31</td>
<td>Power Off</td>
<td>03-00</td>
<td>SLOT1E001</td>
<td></td>
</tr>
<tr>
<td>2011/04/27</td>
<td>09:22:13</td>
<td>Startup completed</td>
<td>01-00</td>
<td>SLOT1E001</td>
<td></td>
</tr>
<tr>
<td>2011/04/26</td>
<td>19:51:16</td>
<td>Power Off</td>
<td>02-00</td>
<td>SLOT1E001</td>
<td></td>
</tr>
<tr>
<td>2011/04/26</td>
<td>15:52:54</td>
<td>Startup completed</td>
<td>01-00</td>
<td>SLOT1E001</td>
<td></td>
</tr>
<tr>
<td>2011/04/26</td>
<td>15:52:59</td>
<td>Power Off</td>
<td>02-00</td>
<td>SLOT1E001</td>
<td></td>
</tr>
<tr>
<td>2011/04/26</td>
<td>08:35:21</td>
<td>Startup completed</td>
<td>01-00</td>
<td>SLOT1E001</td>
<td></td>
</tr>
<tr>
<td>2011/04/25</td>
<td>19:40:59</td>
<td>Power Off</td>
<td>03-00</td>
<td>SLOT1E001</td>
<td></td>
</tr>
</tbody>
</table>

**Figure E8.42  Print Layout for System Log**
E8.5.5 Print Layout for Device List

This subsection describes the print layout for the device list.

● Requirements for Printing

When [Select from Project Data and Print] is selected in the Print Target Specification screen of the Print Setup dialog box and [Device List] is selected as a print target.

● Print Contents

The contents are printed based on the settings configured in the Device List screen, which can be opened by selecting [Print Layout Settings]-[Device List] in the Print Setup dialog box.

![Device List](image1)

Figure E8.43 Print Layout for Device List (When Not Outputting Cross References)

![Device List](image2)

Figure E8.44 Print Layout for Device List (When Outputting Cross References)
In a device list, addresses, tag names, I/O comments (when specified to be output), cross references (when specified to be output) are printed in this order.

The print format of cross references is as follows:

Table E8.4 Printing Cross References

<table>
<thead>
<tr>
<th>Device Type</th>
<th>Cross Reference Information Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Device</td>
<td>bbbb(ll-cc)INST</td>
<td>bbbb: Block name</td>
</tr>
<tr>
<td>Local Device</td>
<td></td>
<td>ll : Line number</td>
</tr>
<tr>
<td></td>
<td></td>
<td>cc : Column number (when specified to be output)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INST : Instruction name (when specified to be output)</td>
</tr>
</tbody>
</table>

**TIP**
- Currently active I/O comments are printed.
- If there is too much cross reference information, a new line occurs at the cross reference information delimiter "," or the cross reference element delimiter "(" or ")".

**SEE ALSO**
For details on assigning multiple I/O comments, see Section E2.8, "Multiple I/O Comments."
This manual describes object ladder functions of WideField3.
This chapter describes how to create and use macros.

### Function Limitations for Each CPU Type

<table>
<thead>
<tr>
<th>CPU Modules</th>
<th>Limitations</th>
<th>SEE ALSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>F3SP22-0S, F3SP28-3S, F3SP38-6S, F3SP53-4S, F3SP58-6S, F3SP59-7S</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>F3SP66-4S, F3SP67-6S</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>F3SP71-4N, F3SP76-7N</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>F3SP71-4S, F3SP76-7S</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>

**TIP**

This section does not describe hardware-dependant limitations for each CPU type, such as the number of available devices.

**SEE ALSO**

For details on limitations for each CPU type, refer to the user's manual for each type.
F1.1 About Macros

F1.1.1 What are Macros?

Macros are a series of processing instructions defined as one instruction. By assigning a name to a macro, you can then use the macro just like any other existing instructions.

F1.1.2 The Purpose of Macros

Using macros has the following advantages.

- Increases program readability
  Program readability generally decreases as the size of a program grows. As a result, debugging and maintenance become more difficult. Using macros, however, increases program readability, facilitating debugging and maintenance.

- Reduces the number of program steps
  Coding a program with repeated group of instructions increases the number of program steps. Bringing these steps into a single macro reduces them to a single program step.

- Promotes modular programming
  This facilitates reuse of past resources. By reusing macros, which have undergone testing, this will reduce the time required for debugging. Furthermore, macros can be created independently, allowing many developers to work in parallel.

- Accumulates know-how
  Processes that are difficult to achieve using existing instructions are accumulated as original know-how in the form of macros. By protecting these macros, secrecy is maintained.

- Easier to use than subroutines
  Macros offer the following benefits, not available with subroutines:

    - A macro can be considered and used as a black box.
      A subroutine may change the value of devices used internally. Accordingly, you must be careful not to use such devices outside the subroutine. When using a macro, however, you do not have to pay any attention to its internal devices.

    - Parameter passing to a macro is easier.
      With subroutines, you have to pre-define the interface specifying which devices are for input and which devices are for output. With macros, however, you can dynamically change the number of parameters and the devices used for the interface.

    - You do not have to include macros in a program.
      You have to include subroutines, but not macro definitions, in a program. Macros are automatically linked and transferred to the CPU.
F1.1.3 Precautions When Using Macros

The following restrictions apply to macros. Abide by these restrictions when using macros.

- **Compatible CPU types**
  Some CPU types do not support the use of macros.

- **Maximum number of macros**
  The maximum number of macros allowed in an executable program depends on the CPU type. The same macro, however, can be used as many times as required in an executable program.

- **Maximum number of macros allowed in a macro block**
  Only one macro definition can be created in a macro file. A macro file has the same name as the macro contained therein.

- **Maximum number of parameters that can be passed to a macro**
  Up to 16 parameters can be passed by a macro calling instruction to a macro. Input and output macro calling instructions allow up to three parameters to be registered. To register more than three parameters, use the PARA instruction.

- **Compatible devices**
  Normal devices, local devices and the following macro devices can be used in a macro body:
  - Pointer (P)
  - Macro relay (H)
  - Macro register (A)
  - Macro index register (U)
  - Structure pointer (Q)
  Considering the nature of macros, we recommend that you use only these P, H, A, U, Q devices in a macro definition.
  A pointer (P) is a register that stores the parameter to be transferred to a macro definition. Structure pointers are used in a structure macro. Macro relays (H), macro registers (A) and macro index registers (U) are equivalent to internal relays (I), data registers (D) and index registers (V) respectively, and may be used in a macro definition.

- **Coding of tag name**
  In a macro definition, tag names can be used for normal and local devices and macro devices P, H, A, and U.

- **Using local devices in macros**
  If local devices are used in a macro, do not call the same macro twice. Save the macro with a different name and call the new macro instead.

**TIP**

Macro devices are provided for exclusive use in macro instructions. Macro devices include pointers (P), macro relays (H), macro registers (A), macro index registers (U) and structure pointers (Q). Macro devices and devices that are used in a block occupy different areas. Hence, macros using macro devices can be highly independent.

Pointers are devices to which parameters passed to a macro instruction are copied in a macro. They are used when arguments are to be computed in a macro. Up to 16 pointers can be used.

**SEE ALSO**

- For details on macro devices P, H, A, and U, see "Sequence CPU – Functions."
- For details on macro devices Q and the limitation of structure macros, see Section F4.5, "Structure Macros."
F1.1.4  Output Macro Call and Input Macro Call Instructions

- **Output Macro Call instruction**
  A macro call instruction to be used on the output side in a block.

- **Input Macro Call instruction**
  A macro call instruction to be used on the input side in a block.

- **Output of Input Macro instruction**
  This is the instruction used by a macro called by the input macro calling instruction to return a macro execution result to the calling block. If a register device is specified as a parameter, the instruction execution result is off when the value of the register is zero, and on when the value of the register is non-zero.

- **Macro instructions**
  Macro instructions can be created independently of the input or output of the calling side. A macro called by the input macro calling instruction requires an NMOUT instruction to return the instruction execution result.

![Figure F1.1 Macro Instructions](F0101_01.VSD)

NMOUT instruction result is reflected here.

**Input macro instruction parameter**
**Input macro call instruction**
**Output macro instruction parameter**
**Output macro call instruction**

F1.1.5  Macro Devices and Macro Local Devices

- **Macro Devices**
  Macro devices A, H, and U are provided for exclusive use with macro instructions. They can be shared by multiple macros.

  Macro devices are not affected by the block that invokes the macro. However, when the same macro device is used by multiple macros, the macros complete for the device.

  Macro devices are basically used in instructions that complete execution in one scan.

- **Macro Local Devices**
  Macro local devices used in a macro are independent of other macros. Unlike normal macro devices (A, H, and U), which are shared among macros, macro local devices used in one macro are not affected by other macros.

  Using macro local devices allows you to develop macros that complete execution in multiple scans.

  Local devices of macros, like those of blocks, may use internal relays (/I), data registers (/D), file registers (/B), timers (/T), and counter (/C).

  As many macro local devices as specified in the Local Device/Properties dialog box are automatically allocated. The following figure illustrates how macro local devices are implemented as actual devices, using internal relays (/I) as an example. Macro local devices are allocated after block local devices.
Figure F1.2 Implementation of Macro Local Devices as Actual Devices

CAUTION

Observe the following points when using macro local devices:
- If a macro uses local devices, you may not call the same macro again. If you need the same macro, save it as a macro with a different name before calling it.
- Do not nest macros.
F1.1.6 About Structure Macros

You can develop macros that use structures as their arguments. These macros are called structure macros.

Structure macros and ordinary macros use the same file naming rules and file management methods but different instructions and devices.

SEE ALSO
For details on structure macros, see Chapter F4, "Structures."

Instructions

- **STRCT Instruction**
  Structure macros must have a STRCT instruction at their beginning.
  The STRCT instruction declares the types of the arguments (structure type definition tag names) for the structure macro.

  \[ \text{---------------- [ STRCT Q1 Structure type definition tag name ]} \]

- **SCALL Instruction**
  This is the instruction used by a block to call a structure macro.
  Specify one structure macro name and two structure names as parameters.
  Set the parameters for structure names to zero if they are not required.

Q Devices

Structure arguments passed from a block are given virtual Q structure type names in a macro.

The Q designation uses the "Q1.<member name>" format, for example, where Q1 is handled like a structure type.

The two Q devices that can be used are Q1 and Q2.
F1.2 Developing Macros

This section describes how to develop macros. Macros created in a project are registered in a macro folder. You cannot directly edit or debug macro definitions stored in a macro folder. You can think a macro folder as a library where debugged macroinstructions are registered. Normally, you would create and temporarily store macros in a project, retrieve and debug them, and register them in a macro folder when you finished debugging.

Figure F1.3 Creating Macros

- Initial setup of macros
  Specify where macro definitions are to be stored.
- Creating macros
  Create macro definitions in a project. Or, retrieve registered macros for debugging.
- Using macros
  Insert codes in a block to call a macro created earlier.
- Transferring macros
  Transfer macros to the CPU using the project download function.
- Debugging macros
  Debug macros from a monitor screen.
- Registering macros
  Register macros in the macro folder so that it can be used by other projects.

SEE ALSO
For details on transferring and debugging macros, see Chapter H14, "Macros" (Online).
F1.2.1 Initial Setup of Macros

Initial setup sets a folder for storing macros. The folder acts like a library where debugged macros are registered and stored. Use the following procedure.

◆ Procedure ◆

(1) Confirm that WideField3 is open.

(2) Select [Tools]–[Set up Environment] from the menu bar.

⇒ The Set up Environment dialog box is displayed with the Folder Setup tab selected.

(3) Enter a macro folder name in the Macro Folder text box, and click [OK].

⇒ The Set up Environment dialog box closes and control returns to WideField3.

Step (3)
F1.2.2 Creating Macros

You cannot put macro definitions in an ordinary block. To create a new macro, specify a macro name and enter the macro definition. There are two types of macro definitions: those in a project that are yet to be debugged and those in a macro folder that have already been debugged. Before you can debug and modify macro definitions, you must retrieve them to a project folder.

To create new macros, use the following procedure.

◆ Procedure ◆

(1) Confirm that the project in which the macro is to be created is open.

SEE ALSO
For details on how to open a project, see Section D2.2.2, “Opening a Project.”

(2) Select [File]–[New] from the menu bar.
⇒ The New dialog box is displayed with the Block/Macro tab selected.

(3) Click [Macro], enter a macro name in the [Block/Macro Name] text box.

(4) Select the tag name definition to be referred to, and click [OK].
⇒ The Local Device/Properties dialog box is displayed.
(5) Enter an appropriate value for input item on the Local Device/Properties dialog box and click [OK].

⇒ The Edit Macro window is displayed.

(6) Create a macro in the Edit Macro window.

SEE ALSO
For details on how to create macros, see Chapter D4, "Creating and Managing Blocks and Macros."

(7) Close the Edit Macro window.
⇒ A dialog box is displayed confirming whether to save the macro.

(8) Click [Yes].
⇒ Control returns to WideField3.

⚠️ CAUTION

Macro and block names must be unique.
Retrieving Macros from a Macro Folder

To retrieve macros that are registered in a macro folder, use the following procedure.

◆ Procedure ◆

(1) Confirm that the project is open.

SEE ALSO
For details on how to open a project, see Section D2.2.2, “Opening a Project.”

(2) Select [Project]–[Extract Macro] from the menu bar.
⇒ The Select File dialog box is displayed.

(3) Enter in the File Name text box the name of the macro to be retrieved, and click [Select].
⇒ The Edit Macro window opens with the selected macro displayed.

Displaying Macro Files Stored in a Project

To open macro files stored in a project, use the following procedure.

◆ Procedure ◆

(1) Confirm that the project is open.

SEE ALSO
For details on how to open a project, see Section D2.2.2, “Opening a Project.”

(2) Select [File]–[Open]–[Block/Macro] from the menu bar.
⇒ The Open File dialog box is displayed.
(3) Select a macro file by clicking, or enter in the File Name text box the name of the macro file to be opened and click [Open].
⇒ The Edit Macro window opens with the selected macro displayed.

## Editing Macros

The same functions for editing ladder programs are provided for editing macros. You can edit macros or define tag names using the same procedure as ladder program editing. In macro editing, you can also use macro relays (H), macro registers (A), macro index registers (U), pointers (P), and structure pointers (Q), which are available only for macros.

⚠️ **CAUTION**

You must always put a MRET (macro return) instruction at the end of a macro.
A macro that is called by the Call Input Macro instruction must end with one MRET (macro return) instruction or at least one NMOUT (macro execution result within a macro block) instruction.

### TIP

When the NMOUT (macro execution result within a macro block) instruction is executed, a logical operation result from the execution of an input-type macro instruction is transferred to the next instruction.

## Editing Local Device Setup and Properties for Macros

You can edit macro local device setup and properties, similar to editing with a block, by selecting [Edit]–[Local Device/Properties] from the menu bar.
F1.2.3 Using Macros

To use a macro, follow the same procedure for entering an application instruction in normal program block editing.

There are two ways to enter a macro instruction: typing and selecting. To input a macro instruction from the keyboard, use the following procedure.

Example

```
MCALL  EXTMOV  Parameter 1  Parameter 2  Parameter 3
```

In offline editing, the [Instruction Categories] list box in the Instruction List dialog box includes [Macros in project] and [Macros in macro folder] in its display.

In online editing, the [Instruction Categories] list box in the Instruction List dialog box includes [Macros in CPU] in its display.

Displayed instructions include the following macros:

<table>
<thead>
<tr>
<th>Instruction Categories</th>
<th>Macro Instructions Displayed in Instruction List</th>
</tr>
</thead>
<tbody>
<tr>
<td>Macros in project</td>
<td>Macro instructions created in a project</td>
</tr>
<tr>
<td>Macros in macro folder</td>
<td>Macro instructions in the macro folder</td>
</tr>
<tr>
<td>Macros in CPU</td>
<td>Macro instructions downloaded to the CPU</td>
</tr>
</tbody>
</table>

Parameters 1, 2, and 3 are to be set by the calling side. You can use any device. In macro definitions, parameters 1, 2, and 3 correspond to pointers P1, P2, and P3 respectively.

While you can pass up to 16 parameters to a macro definition, you must use the PARA instruction before the MCALL (NCALL) instruction if you intend to pass more than three parameters.

Example

```
PARA   4  D00001
```

This example uses D1 as the fourth parameter of a macro instruction.
F1.2.4 Registering Macros

You can register debugged macros in a macro folder so that they can be used in other projects.

To register a macro, use the following procedure.

◆ Procedure ◆

1. Confirm that a project is open.

   SEE ALSO
   For details on how to open a project, see Section D2.2.2, “Opening a Project.”

2. Select [Project]–[Register Macro] from the menu bar.
   ⇒ The Select File dialog box is displayed.

3. Select a file or enter a file name and click [Select].

   TIP
   If a macro with the same name is already registered in the folder, a dialog box appears to confirm whether you really want to overwrite the existing macro.

   ⇒ A dialog is displayed to confirm whether to delete the macro in the project.

4. Click [Yes] if you want to delete the macro in the project; click [No] if you want to retain the macro in the project.
# F2. Using Local Devices

This chapter describes how to set up and use local devices in programs.

## Function Limitations for Each CPU Type

Table F2.1  Function Limitations for Each CPU Type

<table>
<thead>
<tr>
<th>CPU Modules</th>
<th>Limitations</th>
<th>SEE ALSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>F3SP22-0S, F3SP28-3S, F3SP38-6S, F3SP53-4S, F3SP58-6S, F3SP59-7S</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>F3SP66-4S, F3SP67-6S</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>F3SP71-4N, F3SP76-7N</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>F3SP71-4S, F3SP76-7S</td>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>

**TIP**

This section does not describe hardware-dependant limitations for each CPU type, such as the number of available devices.

**SEE ALSO**

For details on limitations for each CPU type, refer to the user’s manual for each type.
F2.1 What are Local Devices?

Local devices are independent devices in a block. Unlike normal devices, local devices are used only within a block. Using local devices, you can program highly independent ladder programs without concerning yourself with the allocations of the local devices within the program (You only need to allocate some normal devices to be used as local devices).

The following devices can be used as local devices:

- Internal relay (I)
- Data register (D)
- Cache register (F)
- File register (B)
- Timer (T) (only timers with 10 ms precision)
- Counter (C)

Local devices use a different addressing notation from normal devices. Local devices are prefixed with a slash (/) before their address number.

Notation Examples

/I00001 Local internal relay
/D0001 Local data register
/F0001 Local cache register
/B0001 Local file register
/T0001 Local timer
/C0001 Local counter

Address numbers of local devices are taken as consecutive numbers, starting with 1 and continuing for the quantity set up for each block. For example, if 32 internal relays (I) are designated as local devices, then the addresses of the devices will be /I00001 to /I00032.

Local devices offer the following merits:

- Allocations of local devices are transparent to the user throughout the entire program.
  Without local devices, a global change in device allocation in a program requires you to change device allocations of each block. With local devices, global device allocations are transparent to each block.
- Facilities reuse of blocks in other projects.
  Program codes involving local devices need not be modified when used in other projects.
### F2.2 Setting Local Devices

Setting up local devices involves setting up 2 items: firstly, the number of local devices to be used in each block and secondly, the location for allocating all local devices in a project.

The figure below illustrates how local devices for each block is allocated. In this setup example, blocks 1, 2, and 3 use 32, 64, and 96 internal relays (I) as local devices respectively; allocated starting with device number I00321.

![Figure F2.1 Allocation of Local Devices](F0002_01.VSD)

**Table F2.2 Example for Allocation of Local Devices**

<table>
<thead>
<tr>
<th>Block</th>
<th>Local Device Address</th>
<th>Actual Device Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block 1</td>
<td>/I1 to /I32</td>
<td>I321 to I352</td>
</tr>
<tr>
<td>Block 2</td>
<td>/I1 to /I64</td>
<td>I353 to I416</td>
</tr>
<tr>
<td>Block 3</td>
<td>/I1 to /I96</td>
<td>I417 to I512</td>
</tr>
</tbody>
</table>
F2.2.1 Block Setup for Local Devices

The procedure for specifying the number of local devices used for each block is given below.

◆ Procedure◆

(1) Confirm that the Edit Block window is displayed.

SEE ALSO
For details on how to display the Edit Block window, see D4, "Creating and Managing Blocks and Macros."

(2) Select [Edit]–[Local Device/Properties] from the menu bar.
⇒ The Local Device/Properties dialog box opens.

(3) Specify the number of local devices for each device type, and click [OK].

TIP
In this setup example, the following local devices are to be used in blocks.
Internal relay /I1 to /I32
Data register /D1 to /D10
Cache register /F1 to /F16
File register /B1 to /B10
Timer /T1 to /T10
Counter /C1 to /C10
Set the quantity for the local devices in units of 32 for internal relays, units of 2 for data registers, cache registers, and file registers, units of 1 for timers and counters.

⇒ Control returns to the Edit Block window.
F2.2.2 Project Setup for Local Devices

When you use local devices in blocks or macros for a project, you need to specify the start address of the local device area.

- Specifying the Start Address of the Local Device Area and the Number of Local Devices to Be Used

Use the following procedure to specify the start address in a global device from which local devices are allocated and the number of local device to be used.

◆ Procedure ◆

1. Ensure that the project is open.
2. Double-click [Project Settings/Configuration] in the project window.
   ⇒ The Project Settings/Configuration window is displayed.
4. Below [Local Device] in [CPU Device Area Setup], turn on the [Set] checkbox of each local device to be used and enter the start address of the local device.

   **TIP**
   [Used] displays the total number of local devices currently used in each block in the project. Check the total number and set the start address so that the device range is not exceeded.

   **SEE ALSO**
   For details on opening and operating the [Device Area Setup] window, see Section D3.1.8, "Device Setup."

5. Click [OK].

   **CAUTION**
   Note that the total number of local devices does not include local devices used in macros. Therefore, when you set a start address, consider the number of local devices currently used in macros so that the device range will not be exceeded.
Automatic Reservation of the Number of Local Devices to Be Used

You can use the function for calculating the number of local devices used in a project and automatically reserving the number of local devices to be used. You can also specify the reserve area to enable adding local devices in online editing.

To automatically reserve local devices, use the following procedure.

◆ Procedure ◆

(1) Select [Tools] - [Set up Environment] from the menu bar.
⇒ The Set up Environment dialog box is displayed.

(2) Click the Circuit Display/Input tab.
⇒ The Circuit Display/Input screen is displayed.

(3) Turn on the [Set Automatically] checkbox in [Set Number of Local Devices].

(4) Use the [Reserve] spin box to enter the rate of the number of devices to be added to the number of local devices actually used in the component blocks. For example, when 32 local devices are used in blocks, if "100%" is entered for [Reserve], 64 local devices are reserved.

(5) Click [OK].

CAUTION

When you use local devices in macros, do not enable the automatic reservation.
F2.3 Programming with Local Devices

Local devices are used in exactly the same way as normal devices. All instructions that can use internal relays (I), data registers (D), cache registers (F), file registers (B), timers (T), and counters (C) can also use local devices. The only difference in programming between normal and local devices is that the address of a local device is prefixed with a slash (/). Devices allocated to local devices are displayed with a different color.

Figure F2.2  A Sample Program Using Local Devices
F2.4 Reusing Programs Containing Local Devices

In cases where you need to modify part of a standard program, as is often required in customizing design for a piece of equipment, using local devices in the initial program development will facilitate program reuse because you would not need to reallocate devices.

Next, we will explain how to add customized blocks.

Figure F2.3 Reusing Blocks Using Local Devices
F2.4.1 Reusing Blocks

When you use blocks in another project, you do not have to modify codes for local devices because they will not conflict with devices of other blocks. You may have to modify codes involving normal devices before reusing the blocks. You can make use of the Replace in Projection function or Change I/O Installation Position function to make batch changes.

To reuse blocks, use the following procedure.

◆ Procedure ◆

1. Open a new project.

SEE ALSO
For details on how to open a project, see Section D2.2.2, "Opening a Project."

2. Select [Project]–[Insert File] from the menu bar.
⇒ The Select File dialog box is displayed.
3. Enter in the File Name text box the file name of the block to be reused, and click [Select].
⇒ The specified block is added to the new project.
F2.4.2 Project Setup

To add blocks to be reused to the execution block, use the following procedure.

◆ Procedure ◆

(1) Confirm that a project is opened.

(2) Double-click [Project Settings/Configuration] in the project window.
   ⇒ The Project Settings/Configuration window is displayed.

   ⇒ The Execution Block Components screen is displayed.

(4) In the block list (left-side list in the window), select a block to be added to the executable program.

(5) In the execution block list (right-side list in the window), click the position at which the selected block is to be added.

(6) Click [Add].
   ⇒ The selected block is added to the specified position in the execution block list.

(7) Click [OK].
   ⇒ The execution block components are updated and the Project Settings/Configuration window is closed.
F3. Using Group Tag Names

Tag names can be grouped into a single group tag name to be used like a data structure. Group tag names allow data relationship, structure and clustering to be easily understood at a glance. This function is especially useful when there are groups of data with the same components. By standardizing the names of group members, group tag names also help in program standardization.

● Function Limitations for Each CPU Type

Table F3.1 Function Limitations for Each CPU Type

<table>
<thead>
<tr>
<th>CPU Modules</th>
<th>Limitations</th>
<th>SEE ALSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>F3SP22-0S, F3SP28-3S, F3SP38-6S, F3SP53-4S, F3SP58-6S, F3SP59-7S</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>F3SP66-4S, F3SP67-6S</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>F3SP71-4N, F3SP76-7N</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>F3SP71-4S, F3SP76-7S</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>

**TIP**

This section does not describe hardware-dependant limitations for each CPU type, such as the number of available devices.

**SEE ALSO**

For details on limitations for each CPU type, refer to the user's manual for each type.
F3.1 Defining Group Tag Names

This section explains how to assign group tag names, using a multi-axis positioning module as an example.

Group tag names are implemented by defining the component members of a group (group template definition) and assigning a group name to the component members (read group name).

---

Figure F3.1  Defining Group Tag Names

### Defining the Component Members of a Group

For example, the F3NC52-0N module has its I/O relays defined as shown in the table below. The module is capable of controlling two axes, with similar relays defined for each axis.

In the table below, X☐☐☐01 and X☐☐☐17 are input relays for Operation Instruction ACK for axes 1 and 2 respectively.

Table F3.2  Definition of Input Relays (X) of the F3NC52-0N Positioning Module

<table>
<thead>
<tr>
<th>Input relay number</th>
<th>Operation when ON</th>
</tr>
</thead>
<tbody>
<tr>
<td>AX1 (axis 1)</td>
<td>AX2 (axis 2)</td>
</tr>
<tr>
<td>X☐☐☐01</td>
<td>X☐☐☐17</td>
</tr>
<tr>
<td>X☐☐☐02</td>
<td>X☐☐☐18</td>
</tr>
<tr>
<td>X☐☐☐03</td>
<td>X☐☐☐19</td>
</tr>
<tr>
<td>X☐☐☐04</td>
<td>X☐☐☐20</td>
</tr>
<tr>
<td>X☐☐☐05</td>
<td>X☐☐☐21</td>
</tr>
<tr>
<td>X☐☐☐06</td>
<td>X☐☐☐22</td>
</tr>
<tr>
<td>X☐☐☐07</td>
<td>X☐☐☐23</td>
</tr>
<tr>
<td>X☐☐☐08</td>
<td>X☐☐☐24</td>
</tr>
<tr>
<td>X☐☐☐09</td>
<td>X☐☐☐25</td>
</tr>
<tr>
<td>X☐☐☐10</td>
<td>X☐☐☐26</td>
</tr>
<tr>
<td>X☐☐☐11</td>
<td>X☐☐☐27</td>
</tr>
<tr>
<td>X☐☐☐12</td>
<td>X☐☐☐28</td>
</tr>
<tr>
<td>X☐☐☐13</td>
<td>X☐☐☐29</td>
</tr>
<tr>
<td>X☐☐☐14</td>
<td>X☐☐☐30</td>
</tr>
<tr>
<td>X☐☐☐15</td>
<td>X☐☐☐31</td>
</tr>
<tr>
<td>X☐☐☐16</td>
<td>X☐☐☐32</td>
</tr>
</tbody>
</table>

---

Tag name Address
START Y00233
STOP Y00234
ERROR X00201
PAUSE Y00235
In this example, a group template for the input relays (X) is defined as follows. The name of the group template is "MOTION."

![Group Template Definition Window](image)

**Figure F3.2 Group Template Definition Window**

Using this group template definition, for example, you can represent all Start Operation Instruction ACK terminals (X□□□01 and X□□□17) with a single member name DctACK.

**SEE ALSO**

For details on how to create group templates, see Section E2.5.1, "Creating Group Templates."

---

### Assigning Group Names

You can open the Block Tag Name Definition window to define group tag names for axes 1 and 2. To do this, use the following procedure.

**◆ Procedure ◆**

1. **(1) Confirm that the Block Tag Name Definition window is open.**

   **SEE ALSO**

   For details on how to display the Block Tag Name Definition window, see Section E2.2.1, "Basic Operations for Editing Tag Name Definition."

   ![Block Tag Name Definition Window](image)

2. **(2) Select [View]–[Display Format]–[Address List] from the menu bar.**

3. **(3) Select [View]–[Display Device]–[Input Relay (X)] from the menu bar.**

   **TIP**

   In this example, we assume that the module is mounted in slot 2.
(4) Click the address row (i.e., X00201) to be defined, and select [Functions]–[Read Group] from the menu bar.
⇒ The Read Group Name dialog box is displayed.
(5) Select a group template (i.e., MOTION), enter a group name (i.e., AX1) in the Group Name text box, and click [OK].

**SEE ALSO**

For details on how to create group templates, see Section E2.5.1, "Creating Group Templates."

⇒ Control returns to the Block Tag Name Definition window. Devices X00201-X00216 are assigned tag names in the form of "AX1_<member name>".

(6) Click the next address row to be defined (i.e., X00217), and select [Functions]–[Read Group] from the menu bar.
⇒ The Read Group Name dialog box is displayed.
(7) Select the same group template (i.e., MOTION), enter another group name (i.e., AX2) in the Group Name text box, and click [OK].

⇒ Control returns to the Block Tag Name Definition window. Devices X00217-X00232 are assigned tag names in the form of "AX2_<member name>". This concludes the assignment of group tag names. As each device is assigned a tag name in the form of "group name + member name" (e.g. device X201 assigned with tag name AX1_DctACK and device X271 assigned with tag name AX2_DctACK), it is easy to see the corresponding axis and signal function.
F3.2 Programming Using Group Tag Names

The following sample program uses group tag names defined in Section F3.1.

Positive direction JOG stepping (X axis)

Turn ON request for jog stepping

Turn OFF request for jog stepping

Write parameters

Positive direction jog stepping relay ON

Executing command flag ON

Positive direction jog stepping relay OFF

Executing command flag OFF

Positive direction JOG stepping (Y axis)

Turn ON request for jog stepping

Write parameters

Positive direction jog stepping relay ON

Executing command flag ON

Positive direction jog stepping relay OFF

Executing command flag OFF

Figure F3.3 Sample Program Using Group Tag Names

In this example, the same circuit pattern is repeated for the X and Y axes. Reading the program, you will notice that you can understand the meaning of devices (and their signals) represented using tag name more easily than the devices represented using address numbers.

Group templates eliminate the need to define every signal individually. In addition, if you define group templates systematically, you can standardize naming conventions and improve program readability.

While the above example is for a 2-axis positioning module, tag name definition using group templates is especially useful for a module with many channels or a system using a number of identical modules.
F4. Structures

This chapter describes WideField3 structures.

**TIP**

For details on the online functions of structures, see Chapter H13, "Structures" (Online).

### Function Limitations for Each CPU Type

<table>
<thead>
<tr>
<th>CPU Modules</th>
<th>Limitations</th>
<th>SEE ALSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>F3SP22-0S, F3SP28-3S, F3SP38-6S, F3SP53-4S, F3SP58-6S, F3SP59-7S</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>F3SP66-4S, F3SP67-6S</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>F3SP71-4N, F3SP76-7N</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>F3SP71-4S, F3SP76-7S</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>

**TIP**

This section does not describe hardware-dependant limitations for each CPU type, such as the number of available devices.

**SEE ALSO**

For details on limitations for each CPU type, refer to the user's manual for each type.
What Is a Structure?

A structure is a group of data represented under a unified name. Grouping data into a structure improves device representation and readability. While you can define individual tag names to devices required to implement a particular function, you can also assign a single structure name to a group of devices and represent individual data within the mechanism of the structure as members of the structure. The FA-M3 provides instructions and macros, which support structure parameters.
Structure Implementation Example

The figure below shows a structure implementation example using a matrix.

In this example, a program is to be created for several sets of inspection data, each consisting of four data items.

![Data for Inspected Item 0](Kensa0.png)

![Data for Inspected Item 1](Kensa1.png)

![Data for Inspected Item 2](Kensa2.png)

**Figure F4.1 Structure Implementation Example**

- **Data Simplification and Processing Using Structures**

  In the above example, you can represent multiple inspection data items with a single structure named "KENSA" to simplify data representation. Each structure contains four data items as members under a single name KENSA. The program uses the format "KENSA[<array number>].TIME" to represent a particular TIME data item and can thus refer to all TIME data items by simply incrementing the array number. This approach simplifies the program and improves program readability.

- **Using Structures with WideField3**

  To use a structure, you must first define the structure type using WideField3. In the above example, set the 4 inspection data as structure members, each with its own data type and comment, to create the structure type.

  Next, define the structure objects in the project. You can use a structure in a program only after defining a structure object, which includes defining a structure name and an array, and assigning a starting address.
Structure Related Functions

Structures are implemented using all the WideField3 functions. However, you will mainly use the following functions to create and debug structures.

- **Editing Functions**
  - Block edit
  - Tag name definition (structure object definition)
  - Structure type definition
  - Macro-related functions

- **Debugging Functions**
  - Block monitor
  - Registered device monitor

CAUTION

F3SP□□□□S and F3SP71/76 CPUs support all structure-related functions.

SEE ALSO

For details on CPU type limitations, see Chapter H1, "Overview of Online Functions" (Online).
F4.1 Overview of Structures

F4.1.1 Using Structures in WideField3

This Section describes how to write programs using structures. The flowchart below illustrates the relevant procedure.

Figure F4.2 Programming Using Structures

- A Create a structure type definition file for an arbitrary structure name in the Structure Type Definition window. In the Structure Type Definition window, create a structure type by entering the structure members and their data types and I/O comments. The structure type definition file is saved in the common folder.
- B Open a project for the program, followed by a block tag name definition or common tag name definition file. Next, define structure objects in the Structure Object Definition dialog box. In the Structure Object Definition dialog box, first set a structure name. Then, set the structure name to refer to a structure type definition created earlier. You can now use the structure type in the ladder diagram edit window.

TIP
Before you can use tag names in program editing, you must establish the relationship between a tag name and its address using tag name definition. Likewise, before you can use a structure in program editing, you must first specify a structure name and its structure type definition.

- C Enter structure parameters in the Edit Block window.
F4.1.2 Structure-related Limitations

The following restrictions apply to use of structures.

### Table F4.2 Structure Specifications

<table>
<thead>
<tr>
<th>Item</th>
<th>Parameter</th>
<th>Limitation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compatible CPUs</td>
<td>F3SP□□□-DS and F3SP71/76 only</td>
<td></td>
</tr>
<tr>
<td>Structure character string</td>
<td>Structure name</td>
<td>Up to 7 alphanumeric characters, beginning with 2 letters. The hyphen (&quot;-&quot;), underscore (&quot;_&quot;), and special characters can be used for the third and subsequent characters.</td>
</tr>
<tr>
<td></td>
<td>Structure member</td>
<td>1 to 8 alphanumeric characters, including the hyphen (&quot;-&quot;), and underscore (&quot;_&quot;) special characters.</td>
</tr>
<tr>
<td></td>
<td>Separator</td>
<td>The period (&quot;.&quot; ) mark is used as the separator between a structure name and a structure member name.</td>
</tr>
<tr>
<td>Structure member</td>
<td>Number of structure members</td>
<td>Up to 2048 members. Up to 2048 devices may be used for each device type. (A long-word member uses two devices. If only register devices are used, up to 2048 are allowed. This limitation applies to each device type, regardless of members of other device types.)</td>
</tr>
<tr>
<td>Array parameter and array specification</td>
<td>Number of arrays</td>
<td>32,767 maximum</td>
</tr>
<tr>
<td></td>
<td>Array number</td>
<td>0 to 32766 (except for STMOV instruction) 0 to 2047 (for STMOV instruction)</td>
</tr>
<tr>
<td></td>
<td>Index modifier</td>
<td>Structure member index modification Not allowed (e.g. &quot;structure. member; V01&quot; is invalid)</td>
</tr>
<tr>
<td>Structure macro</td>
<td>Structure argument</td>
<td>Up to two arguments</td>
</tr>
<tr>
<td></td>
<td>Structure name representation</td>
<td>When structures are passed as macro arguments, assign structure names &quot;Q1&quot; and &quot;Q2&quot;. Structure names are represented as Q1 and Q2; members are represented as &quot;Q1.AAA&quot;.</td>
</tr>
<tr>
<td></td>
<td>Macro nesting</td>
<td>A macro cannot call a structure macro. A macro must not contain a SCALL instruction.</td>
</tr>
<tr>
<td>I/O comment</td>
<td></td>
<td>Up to 32 characters</td>
</tr>
<tr>
<td>Number of steps</td>
<td>If structure members are used</td>
<td>The number of steps is calculated the same way as for constant indexing.</td>
</tr>
</tbody>
</table>

SEE ALSO

For details on the number of steps with index modifications, see "Sequence CPU – Instructions User's Manual."
F4.2 Defining Structure Types

F4.2.1 What Is a Structure Type Definition?

A structure type definition is a definition file defining the members of a structure and their data types under a single structure name.

In a structure type definition, you define a structure type definition name, the members of the structure, as well as the data types and I/O comments of the members.

Structure types created using structure type definition are used as data types to declare structure objects in structure object definition. A structure type definition file is required to define structure objects.

A structure type definition consists of the following components.

● Structure Type Definition File Management

A structure type definition file has a file name extension of ".YGRS."

Structure type definition files are stored and managed in the common folder, together with group templates.

● Structure Type Definition Name

A structure type definition name is a name assigned to a group of structure members.

It is used as the name of the structure type definition file.

This same structure type definition name is also used in structure object definitions.

● Member

A member is a constituent unit (range of devices) of a structure.
Data type

Each member has a data type. Members are allocated different types of devices according to their data types. Each member must be assigned a data type.

Table F4.3  Structure Data Type

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Device Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relay</td>
<td>Relay device</td>
<td></td>
</tr>
<tr>
<td>Integer (DEC)</td>
<td>Register device</td>
<td>Displays as decimal in the registered device monitor.</td>
</tr>
<tr>
<td>Integer (HEX)</td>
<td>Register device</td>
<td>Displays as hexadecimal in the registered device monitor.</td>
</tr>
<tr>
<td>Integer (BCD)</td>
<td>Register device</td>
<td>Displays as BCD (binary coded decimal) in the registered device monitor.</td>
</tr>
<tr>
<td>Long integer (DEC)</td>
<td>Register device</td>
<td>Requires two registers. Displays as decimal in the registered device monitor.</td>
</tr>
<tr>
<td>Long integer (HEX)</td>
<td>Register device</td>
<td>Requires two registers. Displays as hexadecimal in the registered device monitor.</td>
</tr>
<tr>
<td>Long integer (BCD)</td>
<td>Register device</td>
<td>Requires two registers. Displays as BCD (binary coded decimal) in the registered device monitor.</td>
</tr>
<tr>
<td>Char</td>
<td>Register device</td>
<td>Displays as characters in the registered device monitor.</td>
</tr>
<tr>
<td>Float</td>
<td>Register device</td>
<td>Requires two registers. Displays as floating point in the registered device monitor.</td>
</tr>
<tr>
<td>Input relay</td>
<td>Input relay</td>
<td></td>
</tr>
<tr>
<td>Output relay</td>
<td>Output relay</td>
<td></td>
</tr>
</tbody>
</table>

I/O Comments

You can enter an I/O comment for each member. Usage of I/O comments in structures is exactly the same as tag name definitions.
F4.2.2 Creating a New Structure Type Definition
Before using a structure, you must first create a structure type definition. To do this, use the following procedure.

◆ Procedure ◆

(1) Select [File]–[New] from the menu bar.
⇒ The New dialog box will be displayed.

(2) Click the Common Data tab, and select [Structure Type Definition].

(3) Enter a structure type definition name in the File Name text box, and click [OK].

⇒ The Structure Type Definition window is displayed.
F4.2.3 Changing Structure Type Definitions

Structure type definitions can be modified. You can add members to or remove members from a structure type definition file. As changes to a structure type definition file is automatically reflected over a project, there is no need to also change structure object definitions. To modify a structure type definition, use the following procedure.

◆ Procedure ◆

1. Select [File]–[Open]–[Structure Type Definition/Group Template] from the menu bar.
   ⇒ The Open File dialog box is displayed.

2. Select a structure type definition file (with file extension “.YGRS”) and click [Open].
   ⇒ The Structure Type Definition window is displayed.

   **TIP**
   Selecting a group template file opens the Group Template Definition window.
F4.2.4 Exiting the Structure Type Definition Window and Saving Changes to File

To exit structure type definition and save changes made to a file, use the following procedure.

◆ Procedure ◆

(1) Click the [Close] button at the upper right corner of the Structure Type Definition window, or select [File]–[Close] from the menu bar.
⇒ A save confirmation dialog box is displayed.
(2) To save, click [Yes].
⇒ The structure type definition file is saved and the Structure Type Definition window closes.

⚠️ CAUTION

If no data type is defined or there is an error in the definition, an error message is displayed and the save operation is cancelled.

TIP

- To save changes made to a structure type definition file, select [File]–[Save As], or [File]–[Save] from the menu bar.
- When [Save As] is selected, a dialog box is displayed to confirm whether or not to continue your edit operation with the saved file. To continue your edit operation with the saved file, click [Yes]. To edit the file opened before saving the file, click [No].
F4.2.5  Editing Structure Type Definitions

You define structure members, as well as their data types and I/O comments using structure type definition. The relevant procedure is given below.

◆ Procedure ◆

(1) Check that the Structure Type Definition window is open.

(2) Double-click a [Member Name] cell, or move the cursor to a cell and press the [F2] key. Next, enter a member name.

(3) Select a data type from the [Data Type] drop-down list on the line where you have entered a member name.

(4) Double-click the [I/O Comment] cell on the same line, or move the cursor to the cell and press the [F2] key. Next, enter an I/O comment.

(5) Repeat steps (2) to (4) for other members, as required.
Other Editing Operations

- **Removing a Cell**
  To remove a cell, move the cursor to the cell, and press the [Delete] key.

- **Removing a Range of Cells**
  To remove a range of cells, select the cell range, and press the [Delete] key.

- **Copying a Cell or a Range of Cells**
  To copy data to the clipboard, select a cell or a range of cells and select [Edit]–[Copy] from the menu bar.

- **Pasting Cells**
  To paste the content of the clipboard to a range of cells (destination cells), move the cursor to the first destination cell and select [Edit]–[Paste] from the menu bar. You can paste member names, data types, and I/O comments either collectively or individually. If invalid data is pasted to a cell, an error message will appear when the file is saved.

- **Undoing**
  The undo function is not available.

- **Find**
  The find function is not available for member names.
F4.3 Defining Structure Objects

F4.3.1 Overview of Structure Object Definition

Structure object definition is an edit function for declaring structure names. It allows structures to be used in a project or a block.

In structure name declaration, you specify a structure name, the name of a structure type definition file to be used by the structure name, the array size, and the starting addresses allocated to each device used by the members.

Structure object definition is a sub-function of the common tag name definition or block/macro tag name definition functions.

■ Elements of a Structure Object Definition

Table F4.4 Elements of a Structure Object Definition

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
<th>Limitation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structure name</td>
<td>Any name given to a structure type</td>
<td>Up to 7 alphanumeric characters, beginning with 2 letters.</td>
</tr>
<tr>
<td>Structure type definition</td>
<td>A structure type definition file which the structure name refers to.</td>
<td>A structure type definition file in the common folder.</td>
</tr>
<tr>
<td>Array</td>
<td>Specifies a structure array size.</td>
<td>0 to 32767, 0 to 2047 for STMOV instruction.</td>
</tr>
<tr>
<td>Relay device</td>
<td>Relay-type members are allocated devices sequentially, starting with the specified device.</td>
<td>Internal relays (L/I) are allowed. (local devices are only allowed in block tag name definitions).</td>
</tr>
<tr>
<td>Register device</td>
<td>Register-type members are allocated devices sequentially, starting with the specified device.</td>
<td>Data registers (D, /D), cache registers (F, /F), and file registers (B, /B) are allowed. (local devices are only allowed in block tag name definitions).</td>
</tr>
<tr>
<td>Input relay</td>
<td>Input relay-type members are allocated devices sequentially, starting with the specified device.</td>
<td>The maximum number of points allowed for a module is not taken into consideration.</td>
</tr>
<tr>
<td>Output relay</td>
<td>Output relay-type members are allocated devices sequentially, starting with the specified device.</td>
<td>The maximum number of points allowed for a module is not taken into consideration.</td>
</tr>
</tbody>
</table>

CAUTION

Although input and output relays can be set regardless of the maximum allowed for a module, if an unavailable relay number is allocated in structure object definition, an error will be generated at the time of downloading or execution. Be extra careful when using structure arrays.

Figure F4.3 Invalid I/O Relay Settings in Structure Object Definition

- A Number of input relays to be used for each array
- B The last address of the input relays allocated in the structure X00264 and later addresses are invalid.
- C The last address of the output relays allocated in a structure
Structure Names in A Tag Name Definition Edit Window

This section explains the display of structure names in a Block Tag Name Definition window.

Figure F4.4  Structure Names in a Tag Name Definition Window

- **A**  Tag name and address
  Displays tag names, with corresponding addresses and I/O comments.
  Data is displayed as black text against a white background.
- **B**  Structure name
  Only structure names are shown.
  Data is displayed as black text against a gray background.

⚠️ **CAUTION**

All tag names in the tag name definition display in a block monitor are displayed as inactive so it is difficult to distinguish between tag names and structure names. Double-clicking a displayed structure tag name, or moving the cursor to a structure tag name and pressing the [F2] key opens the Structure Object Definition dialog box.
F4.3.2 Structure Object Definition Dialog Box

This Section describes the content of the Structure Object Definition dialog box.
You can define a structure object for a structure name in the Structure Object Definition dialog box.

![Structure Object Definition Dialog Box Diagram]

Figure F4.5 Structure Object Definition dialog box

- A Specify a structure name.
- B Select a structure object definition file name.
- C Specify the array size.
- D Select a relay device type.
- E Select a register device type.
- F Specify the starting address allocated for each device type.
- G Displays the last device number allocated to the structure members.
- H Displays the number of devices required for each structure array element.
- I Use this button to view the content of the structure type definition specified in the [Structure Type Definition] field.
F4.3.3 Editing Structure Object Definitions

To create a new structure object definition in the common block tag name definition or a block tag name definition, use the following procedure.

◆ Procedure ◆

(1) Confirm that the Common Tag Name Definition or Block Tag Name Definition window is open.

(2) Move the cursor to a line with a structure name or a blank line, and select [Functions]–[Structure Object Definition] from the menu bar.

⇒ The Structure Object Definition dialog box is displayed.

(3) Enter a structure name in the [Structure Name] text box.

(4) Select a structure type definition from the [Structure Type Definition] drop-down list.

(5) Specify the array size using the [Array Size] spin box. Specify "0" if no array is to be used.

(6) Select a relay device type and a register device type from the [Relay Device] and [Register Device] drop down lists respectively.

(7) Specify the first address allocated for each device type using the [Start Number] spin box.

(8) Check that all entries are correct, and click [OK].

⇒ The Structure Object Definition dialog box closes. The registered structure is displayed in a cell in black text against a gray background in the Common Tag Name Definition window or Block Tag Name Definition window.
- **Modifying**
  You can modify a structure object definition for an existing structure name.
  In the Common or Block Tag Name Definition window, double-click a cell containing a structure name displayed as black text against a gray background, or move the cursor to such a cell and select [Functions]–[Structure Object Definition] from the menu bar. The Structure Object Definition dialog box will be displayed.
  Modify the entries in the same way as for creating a new structure object.

  **TIP**
  You can change the structure name in a Structure Object Definition dialog box. Once the structure name is changed, the old structure name is no longer defined. To keep the old structure object definition, first copy the structure tag name, and then change the structure name of the copy.

- **Deleting**
  To delete a structure object definition, move the cursor to the cell containing its structure name, and press the [Delete] key.
F4.3.4 Devices Allocated to Structures

A structure defined in structure object definition acquires normal and local devices as structure member devices as required by the relevant structure type definition file.

■ Number of Devices Required

The number of devices required by a structure depends on the following parameters:

- Number of members as defined in the structure type definition
- Member data type as defined in the structure type definition
- Array size as defined in the structure object definition

For a member of relay data type, one relay corresponds to one relay device and is so acquired. For a member of integer or character data type, one word corresponds to one register device and is so acquired. For a member of long integer or floating data type, one long word corresponds to two register devices and is so acquired.

■ How Devices are Allocated

Devices of each device type acquired in a structure occupy contiguous areas, beginning with the starting number for allocating the device type specified in the structure object definition.

■ Displaying Allocated Devices

Figure F4.6  Structure Object Definition (displaying allocated devices)

For each device type, devices with addresses between the [Start Number] and [Last Number] displayed in the Structure Object Definition window are devices allocated to the structure name.
F4.3.5 Tag Name Definition Window Operations

The Common Tag Name Definition or Block Tag Name Definition window is used to allocate addresses to tag names and to define structure objects for structure names.

This section describes and compares the display and operations of tag name definition lines and structure name definition lines in the Common Tag Name Definition window and the Block Tag Name Definition window.

Table F4.5 Comparison between Behaviors of Structure Names and Tag Names in Tag Name Definition Windows

<table>
<thead>
<tr>
<th>Item or Function</th>
<th>Condition e.g. area</th>
<th>Description for Each Object Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content displayed in tag name definition</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tag name</td>
<td>Structure name</td>
<td>Tag name</td>
</tr>
<tr>
<td>Address</td>
<td>Blank</td>
<td>Address</td>
</tr>
<tr>
<td>I/O comment</td>
<td>Blank</td>
<td>I/O comment</td>
</tr>
<tr>
<td>Display sequence</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Registered devices (by address)</td>
<td>Structure names are displayed behind tag names</td>
<td></td>
</tr>
<tr>
<td>Registered Devices (by tag name)</td>
<td>Structure names are interleaved with tag names in the display (in alphabetic order)</td>
<td></td>
</tr>
<tr>
<td>Address list</td>
<td>Structure names are not displayed</td>
<td></td>
</tr>
<tr>
<td>Display of definition errors</td>
<td>Tag name</td>
<td>Display is the same as for tag names</td>
</tr>
<tr>
<td>Text input</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tag name</td>
<td>Structure object definition edit window is displayed.</td>
<td>Tag name is overwritten.</td>
</tr>
<tr>
<td>Address</td>
<td>Structure object definition edit window is displayed.</td>
<td>Address is overwritten.</td>
</tr>
<tr>
<td>I/O comment</td>
<td>Structure object definition edit window is displayed.</td>
<td>I/O comment is overwritten.</td>
</tr>
<tr>
<td>Double clicking</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tag name</td>
<td>Structure object definition edit window is displayed.</td>
<td>Edits tag name</td>
</tr>
<tr>
<td>Address</td>
<td>Structure object definition edit window is displayed.</td>
<td>Edits address</td>
</tr>
<tr>
<td>I/O comment</td>
<td>Structure object definition edit window is displayed.</td>
<td>Edits I/O comment</td>
</tr>
<tr>
<td>Copying cells</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tag name</td>
<td>The entire structure object definition is copied.</td>
<td>The tag name is copied</td>
</tr>
<tr>
<td>Address</td>
<td>Ignored</td>
<td>The address is copied</td>
</tr>
<tr>
<td>I/O comment</td>
<td>Ignored</td>
<td>The I/O comment is copied</td>
</tr>
<tr>
<td>Pasting a tag name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tag name</td>
<td>The structure is deleted and the structure name becomes a tag name.</td>
<td>Becomes a tag name.</td>
</tr>
<tr>
<td>Address</td>
<td>Ignored</td>
<td>Error</td>
</tr>
<tr>
<td>I/O comment</td>
<td>Ignored</td>
<td>Pasted</td>
</tr>
<tr>
<td>Pasting a structure name</td>
<td>Tag name</td>
<td>Structure is overwritten.</td>
</tr>
<tr>
<td>Address</td>
<td>Structure is overwritten.</td>
<td>Structure is overwritten.</td>
</tr>
<tr>
<td>I/O comment</td>
<td>Structure is overwritten.</td>
<td>Structure is overwritten.</td>
</tr>
<tr>
<td>Deleting a cell</td>
<td>Tag name</td>
<td>Structure name is deleted.</td>
</tr>
<tr>
<td>Address</td>
<td>Ignored</td>
<td>Tag name is deleted</td>
</tr>
<tr>
<td>I/O comment</td>
<td>Ignored</td>
<td>Address is deleted</td>
</tr>
<tr>
<td>Find</td>
<td>Tag name or address search</td>
<td>Tag names are searched.</td>
</tr>
<tr>
<td>Sort</td>
<td></td>
<td>Handled the same way as tag names.</td>
</tr>
<tr>
<td>Reading circuits</td>
<td></td>
<td>Not available</td>
</tr>
<tr>
<td>Merging common tag name definitions</td>
<td></td>
<td>Not available</td>
</tr>
<tr>
<td>Merging other block tag name definitions</td>
<td></td>
<td>Not available</td>
</tr>
<tr>
<td>Deleting unused tag names</td>
<td></td>
<td>Cannot be deleted.</td>
</tr>
</tbody>
</table>
F4.3.6 Checking Structure Devices for Overlaps

While tag names and addresses in tag name definitions are automatically checked for overlaps at the time of data entry, device addresses allocated to structure members are not automatically checked for overlaps.

To check for device overlaps in structure object definitions, use the Check for Overlapping Structures function as follows.

◆ Procedure ◆

(1) Confirm that the Common Tag Name Definition window or Block Tag Name Definition window is open, and select [Functions]–[Check for Overlapping Structures] from the menu bar.

⇒ The Results of Checking for Overlapping Structures window is displayed.

Display of Results

The Results of Checking for Overlapping Structures window lists overlapping structure names or tag names by displaying the starting and ending addresses of the corresponding devices.

An error is displayed if overlap check cannot be done because the structure type for a structure name is undefined or otherwise.

Up to 256 instances of overlaps may be displayed.

- **[Tag Name]**
  Overlapping structure names and tag names are displayed.

- **[Overlap]**
  For tag name overlaps, the name of the overlapping device is displayed.
  For structure name overlaps, the range of overlapping devices is displayed e.g. "D00001 – D00016".

**CAUTION**

Overlap checking is not available with other types of tag name definition.
F4.4 Programming Using Structures

This section describes how to use structure parameters in the Edit Block window.

F4.4.1 Display of Structure Parameters in Program Editing Windows

This section describes how structure parameters are displayed in the Edit Block window, as compared to tag name parameters.

### Difference between Structures and Tag Names

#### Table F4.6 Difference between Structures and Tag Names

<table>
<thead>
<tr>
<th>Item</th>
<th>Details</th>
<th>Structure Member</th>
<th>Structure Name</th>
<th>Tag Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display of instruction parameters</td>
<td>Normal display</td>
<td>Displays &quot;&lt;structure name&gt;.&lt;member&gt;&quot;</td>
<td>Displays structure name.</td>
<td>Displays tag name.</td>
</tr>
<tr>
<td></td>
<td>Address display mode</td>
<td>Displays &quot;&lt;structure name&gt;.&lt;member&gt;&quot;</td>
<td>Displays structure name.</td>
<td>Displays allocated address.</td>
</tr>
<tr>
<td>Display color of instruction parameter</td>
<td>Defined</td>
<td>Black (A)</td>
<td>Black</td>
<td>Brown</td>
</tr>
<tr>
<td></td>
<td>Defined as local device</td>
<td>Brown (B)</td>
<td>—</td>
<td>Brown</td>
</tr>
<tr>
<td></td>
<td>Undefined</td>
<td>Blue (C)</td>
<td>Blue</td>
<td>Blue</td>
</tr>
<tr>
<td>I/O comment</td>
<td>Display</td>
<td>Displays I/O comments as assigned in structure type definition (G).</td>
<td>No display (H)</td>
<td>Displays assigned I/O comment.</td>
</tr>
<tr>
<td>TipHelp</td>
<td>Display</td>
<td>Displays structure type definition content for a specific member (F).</td>
<td>No display</td>
<td>Displays tag name definition content.</td>
</tr>
</tbody>
</table>

Note: Letters A to H in the table refer to the parts of Figure F4.7 indicated by the corresponding letters.

#### Limitations on Display of Structures

The following limitations apply to the display of structure member names.

- If the full name of a structure member, including array representation, cannot be displayed, the name is displayed as the parameter string followed by "...". In this case, you can use TipHelp to confirm its full name.
F4.4.2 Entering Structures in a Program Editing Windows

This section describes how to enter structure parameters in the Edit Block window. To enter structure parameters, use the following procedure.

◆ Procedure ◆

1. Confirm that the Edit Block window is open.
2. Double-click a cell to be edited, or move the cursor to a cell and press the [Enter] key. ⇒ The cell is now ready for editing.
3. Enter a structure member name in the cell and press the [Enter] key. ⇒ The content of the cell is updated.

TIP
The procedure for entering a structure member name and other parameters for an application instruction is similar to that for entering tag names, addresses, block names, and other parameters.

SEE ALSO
When you enter instruction parameters, if you press the [↓] key after entering a structure name and a period character, structure members are displayed and you can enter a structure member by selecting it with the cursor. For details, see E1.2.8, "Entering Structures Using Input Completion Function."

■ Array Index
An array index must always be enclosed within square brackets, like "data[1]" and "data[0].rel2."

■ Limitations for Structure Parameter Input
The following limitations apply to the input of structure parameters:
- You cannot change the name or I/O comments for a structure name or a structure in an Edit Block window. To change the I/O comment of a structure member, use structure type definition.
- The Address Assignment dialog box is not displayed when a structure member is entered.
F4.4.3 Instructions and Instruction Parameters

This section describes instructions provided for use with structures. It also describes how to use structure names as parameters of such instructions.

SEE ALSO
For details on instructions, see "Sequence CPU Instruction Manual – Instructions" (IM34M06P12-03E).

Instructions for Structures

- **STMOV (structure move instruction)**
  - **Use**
    Transfers structure data.
    In general, only a constant can be used in an instruction as an index modifier for structure members. To use a variable modifier for a structure array in a normal instruction, you can first use the STMOV instruction to move the structure array element to a non-array temporary structure and then use the members of the temporary structure as parameters in a normal instruction. With this, you can achieve the same result as using a variable index modifier. After execution of the normal instruction completes, you again use the STMOV instruction to return the resultant data to the original structure array.
  - **Parameters**
    The first and second parameters of the STMOV instruction are structure names. However, if the specified parameters are not of the same structure type, the instruction may not operate correctly and may inadvertently change the values of other devices.
    You can use index registers or macro index registers as array indices. For a constant index, enter a value between 0 to 2047 in decimal representation.

- **SCALL (structure macro call instruction)**
  In a SCALL instruction, the first parameter is a structure macro name, the second and third parameters are structure names to be passed as arguments to the macro.
  Only decimal constants are allowed as array indices. The constant must be a value between 0 and 32766.
  Index devices cannot be used as array indices.

- **STRCT (structure declaration instruction)**
  The STRCT instruction is always placed in the first line of a structure macro and can be up to 2 lines long.
  It parameterizes the type of a macro argument.
  The first parameter is a device (Q1 or Q2) used as a structure pointer within the macro.
  The second parameter is a structure type name.

Structure Member Parameters in Normal Instructions

Structure members can also be used as parameters in normal instructions.
Only decimal constants are allowed as array indices. The constant must be a value between 0 and 32766.
Index devices cannot be used as array indices.
F4.5 Structure Macros

Structure macros and normal macros have the same procedures for creation and file management but differ in structure-related operations. This section describes the differences between structure macros and normal macros.

The figure below shows how a structure macro instruction is called in a block and also how the parameters are handled at the structure macro end.

The parameters of SCALL instructions are structure names. The argument types are declared using the STRCT instructions.

Structure name is shown in Q1 and Q2.

Figure F4.8 Structure Macro Call
F4.5.1 Differences Between Structure Macros and Normal Macros

This section describes the differences between structure macros and normal macros.

- Comparison Table for Structure Macros and Normal Macros

<table>
<thead>
<tr>
<th>Item</th>
<th>Structure Macro</th>
<th>Normal macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>Macro call instruction</td>
<td>SCALL</td>
<td>MCALL</td>
</tr>
<tr>
<td>Number of macro arguments</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Macro argument type</td>
<td>Structure</td>
<td>Device</td>
</tr>
<tr>
<td>Argument type declaration instruction</td>
<td>STRCT (structure declaration instruction)</td>
<td>None</td>
</tr>
<tr>
<td>Coding of macro arguments within a macro</td>
<td>Q1.member, Q2.member</td>
<td>P1, P2, ....</td>
</tr>
<tr>
<td>Nesting</td>
<td>MCALL instructions can be coded within a macro, but not SCALL instructions</td>
<td>MCALL instructions can be coded within a macro, but not SCALL instructions</td>
</tr>
<tr>
<td>PARA instruction</td>
<td>9 to 16 allowed</td>
<td>4 to 16 allowed</td>
</tr>
</tbody>
</table>

- Representation of Structure Members

Structure names passed as parameter arguments to a structure macro cannot be used as-is in the macro. In a structure macro, structure name arguments are changed into “Q1” and “Q2” and represented as such.
F4.5.2 Setting Argument Types of A Structure Macro

Structure arguments can be used in structure instructions. The STRCT instruction is used for structure arguments. It requires two arguments: structure pointer and structure type name.

- **Structure Pointer (Q Device)**
  
  A Q device is a structure pointer device used within a macro.

  A Q device is a pointer to the given structure name, with its structure type name given by the second parameter.

  Structure members within a macro are coded as “Q1.member” and “Q2.member”.

- **Structure Type Name**

  The structure type name parameter specifies the structure type name of the macro argument.

  You can specify the name of any structure type definition file stored in the common folder as the structure type name parameter.

---

**CAUTION**

When calling a structure macro within a block, you must pass an argument of the same structure type as that specified in the STRCT instruction. Passing a structure argument of a different structure type may lead to unexpected behaviors.
F4.5.3 Structure Macro Programming

This section shows a sample program, which uses a structure in a structure macro.

- **Sample Program Codes for the Calling Block**

  ![Figure F4.10 Sample Program Codes for the Calling Block](image1)

- **Sample Program Codes for the Macro**

  ![Figure F4.11 Sample Program Codes for the Macro](image2)

- **Structure Type Definition**

  ![Figure F4.12 Structure Type Definition](image3)
The operations of the block and macro shown in figures F4.10, F4.11 and F4.12 are described below.

**Block**
- **A** Calls the SMOV macro. The SMOV macro requires 2 arguments.
- **B** Specifies structure object name “data” for argument “Q1”; “data” has structure type name “KOZO1”.
- **C** Specifies structure object name “kozo” for argument “Q1”; “kozo” has structure type name “KATA”.

**Argument Structure Type Declaration**
- **D** Declares macro argument “Q1” as having structure type “KOZO1”.
- **E** Declares macro argument “Q2” as having structure type “KATA”.

**Structure Type Definition**
- **F** Defines structure type (“KOZO1”) declared for “Q1” earlier.
- **G** Defines structure type (“KATA”) declared for “Q2” earlier.

**Sample Program**
- **H** “Q1.rel1” in the program refers to “data.rel1”.
- **I, L** “Q2.ListD” in the program refers to “kozo.ListD”.
- **J** “Q1.reg2” in the program refers to “data.reg2”.
- **K** “Q1.reg1” in the program refers to “data.reg1”.

F4.6 Structure-related Functions for Projects
This section describes functions that are provided for management of projects containing structures.

F4.6.1 Checking for Overlapping Structures
For each structure registered with objects in a block tag name definition or the common tag name definition, the Check for Overlapping Structures function checks whether the structure object uses addresses allocated to another structure or tag name in the tag name definition.

TIP
In the Common Tag Name Definition window and Block Tag Name Definition window, WideField3 automatically performs overlap checks for addresses allocated to tag names and displays the items in a different color. However, overlap check is not automatically performed for structure objects. You need to perform overlap checks manually.

To perform overlap checks, use the following procedure.

◆ Procedure ◆
(1) Confirm that the Common Tag Name Definition window or Block Tag Name Definition window is open, and select [Functions]–[Check for Overlapping Structures] from the menu bar.
⇒ Overlap check is performed and the result is displayed in the Results of Checking for Overlapping Structures dialog box.
(2) Click [Close].
⇒ The Results of Checking for Overlapping Structures dialog box closes.

TIP
The following items are displayed in the Results of Checking for Overlapping Structures dialog box.
- Structure object overlaps errors
- Errors where structure objects and addresses allocated to tag names overlap.
Overlap Range. For tag name overlaps, the name of the overlapping device is displayed. For structure name overlaps, the range of overlapping devices is displayed.
- Structure object name.
- Tag name.

CAUTION
Blocks referring to the common library or another project are outside of the target of the Check for Overlapping Structures function.
F4.6.2 Finding in Project and Replacing in Project

This section gives an overview of finding and replacing structure members and structure names over an entire project and describes the required procedures.

CAUTION

Blocks referring to the common library or another project are outside of the target of the Replace in Project function.

When [All Blocks in Project], [All Macros in Project] or [All Blocks/Macros in Project] is specified as a search condition, blocks referring to the common library or another project are outside of the target of the Find in Project function.

Overview

Overview of Find Function for Structure Members and Structure Names

Searching for structure members and structure names is essentially the same as searching for tag names.

In the Find dialog box, select the Search entire project tab and [Tag Name/Address] as the [Search Object], and enter a structure name, structure member name or a string of the form “structure name.*” in the [Search String] text box.

When searching for a structure member name or structure name, array values are considered in its search.

Structure member parameters and structure name parameters coded in the program are searched for the given string, in a similar way to searching for tag names.

SEE ALSO

For details on finding in structures in a project, see Section E4.4, "Finding and Replacing in Structures in the Project."

The table below lists the search results for some example search strings of structure members or structure names.

Table F4.8 Search Strings and Search Results

<table>
<thead>
<tr>
<th>Program Data</th>
<th>Search String</th>
<th>Search Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>dataS (tag name)</td>
<td>data</td>
<td>data</td>
</tr>
<tr>
<td>data (structure name)</td>
<td>data.reg1</td>
<td>data.reg1</td>
</tr>
<tr>
<td>data.reg1 (structure member)</td>
<td>data.rel1</td>
<td>data.rel1</td>
</tr>
<tr>
<td>data.*</td>
<td>data</td>
<td>data</td>
</tr>
<tr>
<td>data*</td>
<td>dataS</td>
<td>data</td>
</tr>
<tr>
<td></td>
<td>data.reg1</td>
<td>data.reg1</td>
</tr>
<tr>
<td></td>
<td>data.rel1</td>
<td>data.rel1</td>
</tr>
</tbody>
</table>

The table below lists the search results for some example search strings of structure members or structure names including arrays.

Table F4.9 Search Strings and Search Results

<table>
<thead>
<tr>
<th>Program Data</th>
<th>Search String</th>
<th>Search Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>data (structure name)</td>
<td>data2[1].reg1</td>
<td>First instruction with “data2[1].reg1” included within the used device range of its instruction parameters.</td>
</tr>
<tr>
<td>data2 (structure name) array index 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>data.rel1 (structure member)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>data2[1].rel1 (structure member)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
TIP
To search for structures including arrays, turn on the [Tag Name/Address] checkbox in [Search Object] and then turn on at least one of the other checkboxes in [Search Object].

● Overview of Replacement of Structure Member Names and Structure Names
Structure replacement applies to structure names used in parameters.
Turning on the [Replace in Tag Name Definitions too] checkbox in the Project Replace dialog box also replaces structure names in tag name definitions.

SEE ALSO
For details on the functions for finding in structures in the project, see Section E4.4, "Finding and Replacing in Structures in the Project."

● Find and Replace for Structure Arrays
- You cannot include array representation in search or replace strings. If a specified search or replace string contains an array, an error is generated. For example, "KOZO[V01]." is not a valid search string.
- Search or replacement of structure members or structure names searches all elements of arrays, as well as non-array structures. For example, search string “KOZO.rel1” matches “KOZO.rel1”, “KOZO[1].rel1” and the pattern “KOZO[*].rel1”.

Procedure for Finding in a Project
This section describes how to search for a structure member or structure name over a project. The relevant procedure is given below.

◆ Procedure ◆
(1) Select [Project]–[Find in Project] from the menu.
⇒ The Find in dialog box is displayed. The Search entire project tab is selected.
(2) Enter a search string in the [Search String] text box, turn on the [Tag Name/Address] option button in the [Search Object] group box and click [Find Next].
⇒ Search results are listed at the bottom area of the Find in dialog box.

TIP
Searching for a structure member name also searches for any address allocated to the structure member name; similarly, searching for an address also searches for any structure member name assigned to the address. For instance, if “St1.rew1=D00102” is defined in the tag name definition and [Tag Name/Address] in the [Search Object] group box is turned on, then searching for “St1.rew1” or “D00102” produces the same result.
Procedure for Replacing in a Project

This section describes how to replace a structure name over an entire project. The relevant procedure is given below.

Procedure

(1) Open the desired project and close all blocks and tag name definitions being edited.
(2) Select [Project]–[Replace in Project] from the menu.
   ⇒ The Replace dialog box is displayed. The Replace in entire project tab is selected.
(3) Enter the structure name to be replaced and the new structure name in the [Old Value] and [New Value] text boxes respectively, turn on the [Structure→Structure] option button in the [Replacement Method] group box and click [OK].

TIP

To replace structure members, turn on the [Tag Name→Tag Name] option button in the [Replacement Method] group box.

TIP

To also replace tag names in tag name definitions, turn on the [Replace in Tag Name Definitions too] checkbox.

⇒ Replacement begins and the Replace in Project–Progress dialog box is displayed with the progress of the project replacement.
(4) Click [Close].
⇒ The Replace in Project–Progress dialog box closes.
F4.6.3 Changing I/O Installation Position

The Change I/O Installation Position function does not change structure parameters in the program but only changes starting numbers for allocating input relays or output relays defined in structure object definitions in the common tag name definition or block tag name definitions.

⚠️ CAUTION ⚠️

Blocks referring to the common library or another project are outside of the target of the Change I/O Installation Position function.

SEE ALSO

For details on the change I/O installation position function, see Section D3.2.7, “Changing I/O Installation Position.”

◆ Procedure ◆

1. Check that the project to be changed is open and close all blocks and tag name definitions being edited.
3. Specify the slot number to be replaced, the new slot number and the number of slots to be replaced using the [Old Slot Number], [New Slot Number] and [Number of Slots to Replace] spin buttons respectively, and click [OK].

TIP

Turning on the [Specified Block] option button in the [Replacement Range] group box allows you to perform slot changes on selected blocks only.

⇒ A Change I/O Installation Position – Progress dialog box is displayed with replacement progress and results information.
F4.6.4 Structure-related Syntax Check Errors

This section describes structure-related errors that are checked by the program check function.

The syntax errors that are checked at the time of downloading and at the beginning when a program is transferred to ROM are the same.

All the error items belong to the severity class of errors.

Table F4.10 Syntax Check Errors

<table>
<thead>
<tr>
<th>Error Message</th>
<th>Possible Causes</th>
<th>Troubleshooting</th>
</tr>
</thead>
<tbody>
<tr>
<td>No address assigned to tag name</td>
<td>A structure object is yet to be defined.</td>
<td>Define a structure object for the structure name.</td>
</tr>
<tr>
<td></td>
<td>A structure member does not exist.</td>
<td>Check that the structure member used in the program is defined in the structure type definition.</td>
</tr>
<tr>
<td>Structure type differs</td>
<td>The S and D parameters of a STMOV instruction do not have the same structure type definition name.</td>
<td>Make sure that the S and D parameters of a STMOV instruction have the same structure type definition name.</td>
</tr>
<tr>
<td>STRCT instruction error</td>
<td>A STRCT instruction is coded incorrectly.</td>
<td>A STRCT instruction must be the first line of a macro and not longer than 2 lines. Q1 and Q2 must be set once and only once.</td>
</tr>
<tr>
<td>Cannot use this instruction parameter on selected CPU type</td>
<td>Structure type definition file specified in a structure type definition does not exist in the common folder.</td>
<td>Place the corresponding structure type definition file into the common folder.</td>
</tr>
<tr>
<td>Structure type definition file not found</td>
<td>CPU selection error</td>
<td>Select a CPU that supports structures.</td>
</tr>
<tr>
<td>Device is invalid or out of range</td>
<td>The number of device points used in a structure exceeds 2048. This error is generated when there are too many long word members in a structure.</td>
<td>Reduce the number of members in the structure type definition.</td>
</tr>
<tr>
<td></td>
<td>A constant specified for the array of a structure name is not within the range of 0 to 2047.</td>
<td>Use an index register instead, or change the value of the index.</td>
</tr>
</tbody>
</table>
F4.6.5 Used Devices List/Device List

The used devices list displays the usage of devices as parameters within a block. However, note that device ranges in structures, even if not used in program, are already reserved.

When a structure name or structure member is used in a program, all members are considered to be used in the program. Therefore, some of the devices displayed in the used devices list may actually not be used in the program but are listed because other members of the same structure name are used.

**CAUTION**

Blocks referring to the common library or another project are outside of the target of the used devices list and device list.

### Used Devices List Screen Example

![Used Devices List Screen Example](image)

All addresses between the start number and the end number for allocation are displayed as "structure", which indicates that the devices are used as structure members.

### Device List

The device list displays actual devices being actually used for structure members in a project as used devices.

**SEE ALSO**

For details on the functions of the device list, see Chapter E5, "Device List."
INDEX

A
access key ....................................................... A3-23
Action monitor ..................................................... J1-3
activating instructions ........................................ K2-8
Advanced function module register monitor .......... J3-19
Advanced function module relay monitor .......... J3-18
alarm display ...................................................... K3-3
application instruction ...... E1-23,J2-4,J2-14,J2-15

B
balloon comments ............................................ E1-77
balloon list window ........................................... A3-10
balloon monitor ................................................. J2-31
basic instructions ............................................. E1-21
Battery error/Memory check error............. K3-6,K3-11
block tag name definition............. A3-18,D2-4,D3-78,
E1-95,E2-3,E2-15,E2-26,E2-29,H9-3,J4-1
block............................................. D2-3,D4-2,D4-4,D4-7,D4-10,
D4-13,E1-37,E8-30,F2-4,F2-9,H2-9,H3-6,
H4-6,H13-4,H13-5,J2-2
built-in ROM ........................................................ J1-3

C
CADM3 ............................................. A3-12,D2-20,D4-13,H8-24
Cancel Forced Set/Reset .................................. K1-4
Cascade .............................................................. K4-6
cell.............................................................. E1-54,E1-57,E1-60,F4-13
Change I/O installation position.... B2-2,D3-69,F4-34
circuit comment............. E1-70,H2-5,H9-3,K2-18,K2-10
clear CPU properties .............. H5-13
clear devices ............................................... J1-11
Clear Log .................................................. K3-9,K3-17
clear program ................................................. J1-10
common library .............................................. B7-2
common tag name definition...... A3-4,A3-18,D2-4,
D3-79,E1-95,E2-3,E2-16,E2-28,E2-29,E2-33,
H2-5,H9-5,J4-1
Compact printing ................................. E8-31
compare .................................................. A1-6,E7-6,E7-16,H4-3,H4-6,
H4-8,H6-7,H8-19,H9-9
Compare Project ............................................... E7-6
Compare Window ............................................. E7-16
Comparing Files ........................................... D1-32,E7-1
Comparison Bar ........................................... A3-3,E7-3
concurrent execution .... A4-7,H7-4,H7-14,H8-7,K4-7
configuration ................................................. A3-4,A3-18,C1-1,D2-6,
D2-21,D3-9,E8-27,E8-45,H4-7,J1-5,K4-33
connection history ........................................ H11-20
connection line ............................................. B3-5,E1-46
constant definition ......................... E3-1
constant scan ............................................. D3-31
contact A ............................................. B3-5,B3-8,E1-9,J2-2
contact B ............................................. B3-5,B3-8,E1-9,J2-2
continuation line .......... A3-20,E1-4,E1-49
continuous type application instruction... E1-6,E1-24
conversion error ................................. E1-2,E1-103
convert .................................................. E1-2,E1-103
CPU properties .............................................. H5-1

D
Define Program Components .......... A3-5,D2-6
Device Manager .............. A1-7,H8-1,H8-3,H8-17
J3-1,J3-3,J3-8,J3-14,J3-21
disabling ............................................. K2-10
Display Data Change Identification .......... J2-23
display format ............ J2-15,J3-11,J4-1,J4-7,
K4-38,K4-42,K4-44,K4-47
Display style ............................................... J4-8
Display Operation Log ........... D1-28,D2-6,K3-21
Display User Log .................. K3-16
displaying details .................. J2-14
download .............. D1-20,H1-1,H2-1,H2-3,H2-9,
H5-9,H8-11,H9-8,H13-5,H13-6

E
E1-8,F4-23,F4-24
E-mail .............. C4-1,D1-30,H8-17,H10-11
environment setup .............. D1-3,H10-11
exclusive access control .............. J1-14
executable program .......... A3-18,D2-6,D2-7,D3-9
Export (Circuits comments/Subcomments) ... E1-104
Export (Cross reference) .............. E6-19
Export (Device files) .............. H8-22
Export (Device list) .............. E5-18
Export (Environment setup) .............. D1-34
Export (Key customization) .............. D1-38
Export (Tag name definition) .............. E2-34
Export Macro .................. F1-11
Index-2

F
FA Link error .......................... K3-6,K3-11
FA Link Tool ................................ H7-4,H7-12
FA-M3 Defender .......................... H11-1
Find Bar .................................. A3-2,E4-50
Find in Project .................... A3-5,E4-8,E4-15,E4-31
Find Instruction in Project ........ E4-15
Fixed Index Modification Monitor ..... J2-20
FL-net Refreshing ........................ D3-52
Font ........................................ B4-1,D1-19
Forced Reset ............................ J2-3,K1-2
Forced Set ................................ J2-3,K1-2
function key .................................. B3-5,E1-22

G
global device ................. B6-1,D3-35,E1-12,F2-8,
H14-2,J5-5
group tag name ....................... E2-31,F3-2,F3-5

H
Hide Circuits .............................. E1-85,J2-13

I
I/O comment ...................... A1-7,A3-20,D1-27,D3-80,
E1-74,E2-2,F4-8,J2-24,K2-19
I/O comparison error ................. K3-6,K3-11
I/O module error ........................ K3-6,K3-12
Image Copy ................................ E1-65,J2-35
Import (Circuits comments/Subcomments) ... E1-105
Import (Device files) .................. H8-23
Import (Environment setup) ......... D1-35
Import (Key customization) .......... D1-39
Import (Tag name definition) ........ E2-35
inactivating instructions ................ K2-8
index modification ..................... E1-39,E2-5,F4-6,J2-4
indirect specification ............. E1-40,E2-5,J2-18,J5-1
instruction error ........................ K3-6,K3-11
Instruction List ......................... E1-9,E1-23
instruction number .................... E1-10,E1-96,J2-13
instruction palette .................... A3-2,E1-9
Inter-CPU communication error ........ K3-5,K3-11
interrupt error .......................... K3-6,K3-11
invalid instruction found ................. K3-5,K3-11

K
Key customization ................................... D1-39

L
label ........................................ A3-20,E1-37,E1-51
Language Setup ........................ D1-39
local device ..................... B6-1,D3-35,D3-57,D4-10,
E2-4,E1-12,F1-4,F1-12,F2-1,F2-2,F2-7,F2-8,
H8-16,J3-9,J5-7
long word ......................... B3-9,E1-23,E1-28,E1-34,
J2-16,K1-6,
long-word index modification ........ E1-39

M
macro.................. A3-18,B5-1,D2-3,D4-2,D4-4,D4-7,D4-10,
D4-13,E8-30,F1-1,F1-7,H2-9,H3-6,H4-6
Macro monitor......................... H14-3,J2-2
macro tag name definition ........ A3-18,D2-3,E1-95,
E2-6,H2-10
Memory error/failure ................. K3-6,K3-11
Mnemonic editing ..................... E1-7,E1-91
MODE switch ............................. J1-3
momentary power failure ............. K3-5,K3-11
multiple connections ............... H1-22

O
offline balloon ......................... B1-5,H9-10
online balloon ......................... B1-5,H9-10
online editing ......................... A1-9,D3-80,
H13-9,H9-9,H13-9,K2-1,K2-5,K2-15,J2-34
operating mode ..................... A3-16,H1-23,J1-6
operation protection ............ A1-7,H11-5,H11-7,H11-12,
H11-14
Output window ..................... A3-10

P
power off .................................. K3-11
Program monitor ..................... A1-6,A1-7,A1-8,A1-9,J2-2,
J2-7,J2-30,J5-13,K2-5
project .................. A1-5,A1-6,A1-7,A1-9,A1-12,
A1-15,B2-4,D2-3,D2-7,D3-5,D3-61,E8-8,
E8-26,E8-44,F2-5,F2-10,F4-30,H1-23,H2-3,
H3-3,H4-3
project settings ................. J1-5
Project window ....................... A1-9,A3-4
protection .................. D3-26,D4-5,D4-12,H5-12
pulse ........................................... E1-23

R
Read Circuits ......................... E2-14,E3-10
Read mode ............................... D4-6
Register Macro ......................... F1-7,F1-14
Registered device monitor ........ A1-6,A1-7,A1-8,A1-9,
H13-2,J5-1,J5-3
replacing in project .................. D3-66,F4-31
Reset start .............................. H1-2,J1-12
Restart Refreshing .................... K1-14
Resume Monitoring .................. J2-30,J3-16
ROM pack error ...................... K3-5,K3-12

S
sampling trace .................. A1-8,K4-3,K4-5,K4-6,K4-35
scan time .................................. J1-3
scan timeout ......................... K3-6,K3-11
Search Window ....................... A3-11
security keyword ....................... H5-10
self-diagnosis error ................... K3-5
sensor control block (SCB) ........ C1-1,D3-10,D3-32,
J1-3,K3-6
Sequential Search ................... E4-21
set time ................................... J1-9
shared refreshing ................... D3-44
## Revision Information

**Document Name:** FA-M3 Programming Tool WideField3 - Offline User's Manual  
**Document No.:** IM 34M06Q16-02E

<table>
<thead>
<tr>
<th>Edition</th>
<th>Date</th>
<th>Revised Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>Jan. 2012</td>
<td>New publication</td>
</tr>
<tr>
<td>2nd</td>
<td>Sep. 2012</td>
<td>Supported for WideField3R2.03</td>
</tr>
<tr>
<td>3rd</td>
<td>May 2013</td>
<td>Supported for WideField3R2.04</td>
</tr>
</tbody>
</table>

Written by PLC Product Development & Engineering Department  
Control Instruments Business Division  
IA Platform Business Headquarters  
Yokogawa Electric Corporation

Published by Yokogawa Electric Corporation  
2-9-32 Nakacho, Musashino-shi, Tokyo, 180-8750, JAPAN

Printed by Kohoku Publishing & Printing Inc.