

Application Note



2500 Series® Programmable Automation Control System

Online Change with Janus Processors

Overview

The CTI Online Change facility allows you to change your application program "on the fly" while it is running in the target system. The facility allows a full download of a modified version of a project while the old version is still running. When download and project validation is complete, the old project is replaced with the downloaded project.

While the old project is being replaced, the old project is stopped and fieldbus connections are closed, except for I/O subsystems (Local I/O, Remote I/O, Profibus I/O, and Ethernet/IP Scanner and Adapter), which remain operational, maintaining the last state of all outputs. See Fieldbus Protocol Operation section below for restrictions.

Once the new project has been downloaded and validated, the current value of all variables existing in both the old and new projects are copied from the old project database to new project database. See Rules for Copying Variable Values section below. If the new project contains SFC program(s), SFC state information is copied to the new project database.



if the SFC charts match (indicating no changes) between old and new projects. See Rules for SFC Charts section below. All fieldbus drivers are restarted and logic processing for the new project begins.

The time to complete the CTI Online Change is dependent on project configuration – including code size, variables, function block instances, and fieldbus protocols. The main advantage is that it will execute with basically unlimited project changes while maintaining all database values at their last state throughout the process (bumpless transfer).

Advanced Capabilities

The CTI Online Change (OLC) facility provides the following advanced capabilities:

- Allows addition of new Programs and Function Blocks
- Allows addition and/or changes to user-defined structures (structure variables, Sub-Programs, and UDFBs)
- Allows addition and deletion of RETAIN variables.
- Allows changes to fieldbus configurations (i.e. add new configuration, Master/Port, Slave/Data Block, and/or mapped variables).
- Allows changes to Local I/O and/or Remote I/O fieldbus configurations without causing a state change (i.e. ‘bump’) to any existing I/O point. Communication to all existing modules and bases will remain active and maintain ‘last state’ through the entire OLC operation. *NOTE: does not apply to the JACP module, which does not support remote/local IO.*
- Profibus-DP Master fieldbus will close and cause an I/O ‘bump’ only when the Profibus configuration (Levels 1-3) is changed. Otherwise, all Profibus communications will remain active and I/O will maintain ‘last state’ through the entire OLC operation. *NOTE: does not apply to the JACP module, which does not support Profibus DP.*
- Ethernet/IP I/O Scanner will terminate, causing all specified “active” EIP Adapters to timeout and close the I/O connection, only when a change is made to either the EIP I/O Scanner or EIP Adapter configuration (Levels 1-3) in the Janus controller. Changes to variable mapping (Level 4) will not cause the connection to be closed.
- Ethernet/IP Adapter will terminate, causing all attached EIP I/O Scanners to timeout and close the I/O connection, only when a change is made to either the EIP I/O Scanner or EIP Adapter configuration (Levels 1-3). Changes to variable mapping (Level 4) will not cause the connection to be closed.
- Allows modifications to SFC charts. However, these changes will result in the termination and restart to ‘Initial Step’ for that SFC program and all ‘Child SFC Programs’ for which it is the parent. The user must be notified of this action in the dialog window described below and allowed to abort the OLC operation if this action is unacceptable. See Rules for Sequential Function Charts.
- Provides ‘OLC Confirmation’ dialog window before OLC in runtime is triggered. This window shows how the OLC operation will affect the running system.
- An online change can be triggered only to the “startup” project in Workbench. An active online change must be completed before OLC can be triggered on a different project.

Benefits

The CTI Online Change operation included in all Janus controllers provides the following benefits:

- Performance - application is paused for less than 500 msec in almost all cases.
- Maintains state of all unaltered variables and function block instances.
- Maintains connections for ‘unchanged’ Ethernet I/P I/O operations (including EIP I/O Scanner/Adapter and FlexIO) to prevent remote device timeouts
- Detailed reporting of how each application variable will be affected (Added, Deleted, or Copied)
- Reports all changes to User Defined Type (UDT) structures and function blocks



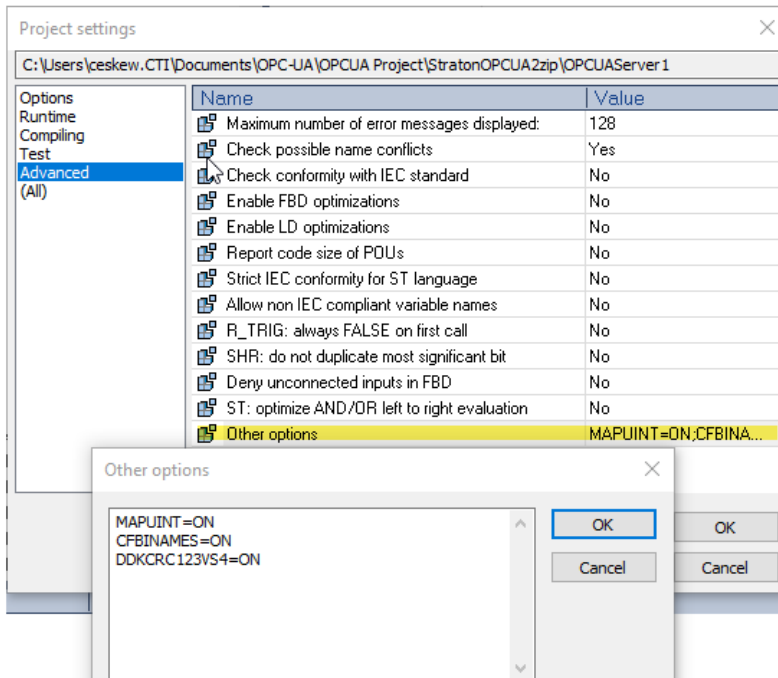
- Reports application version numbers of current and modified applications
- Provides detailed reporting of interruptions in execution of SFC programs that will result from the changes made to the SFC programs and/or SFC child programs
- Reports interruption in Profibus network operation that will result from changes made to the Profibus Master configuration.
- Reports changes to Ethernet/IP configurations that would cause an interruption in communications for all configured drivers. NOTE: A change to any EIP I/O Scanner, I/O Adapter, and/or FlexIO configuration will result in a reset of the EIP stack and interrupt all EIP drivers
- Offers user option to cancel Online Change after results of modifications are reported, without affecting the application running on the controller
- Provides comprehensive error detection - all processes are validated before execution and any error conditions result in termination of Online Change operation without affecting the application executing on the controller
- Provides ability to quickly restore development/monitoring environment to match application executing on the controller following user cancellation or error
- History of all Online Change operations recorded in 'Execution' history Log and Event Log

⚠ WARNING

When making Online Changes, you MUST ensure no other users are connected to the controller with Janus Workbench before proceeding with an online change. Having two or more users connected who are simultaneously making program changes and forcing variables can result in unexpected system operation.

Required Project Settings

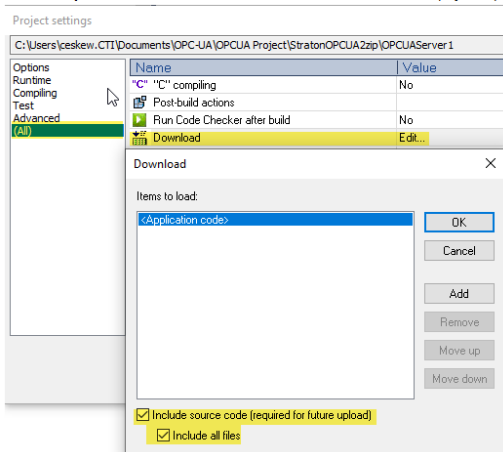
The following compile options must be set in 'Project Settings':



- MAPUINT=ON: Allows mapping of unsigned integers
- CFBINAMES=ON: Includes information for all Function Block Instances.

- DDKCRC123VS4=ON: This enables online change to detect where changes were made.

In addition, you will need to ensure that source code is downloaded with the project. Open the Project Settings dialog, select the "All" category, click on the Download item and check the *Include Source Code* box. See below.

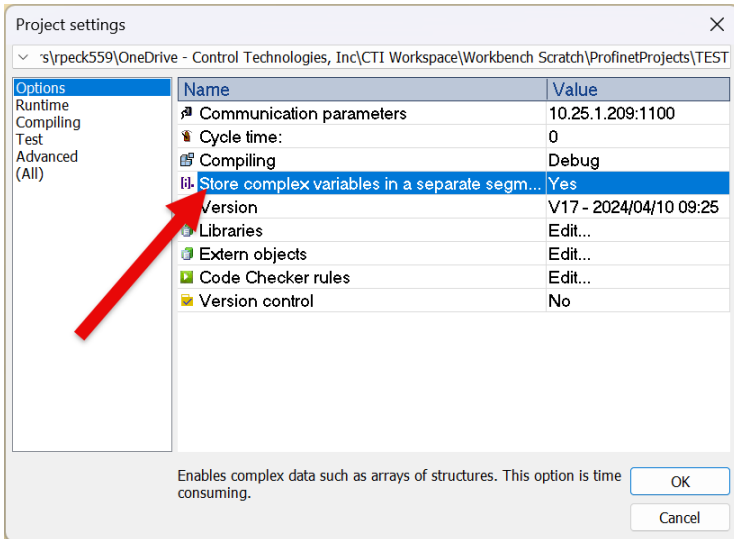


IMPORTANT NOTE: Online change is **NOT SUPPORTED** for 2500P-ACP1 applications. If you're using a coprocessor and need online change, use 2500P-JACP instead.

IMPORTANT NOTE:

-]P[and]N[contacts **must not** be used in any program where you intend to do online changes, as their states cannot be properly preserved through an online change. Instead, use the R_TRIG and F_TRIG function blocks.
- -(P)- and -(N)- coils **must not** be used in any program where you intend to do online changes, as their states cannot be properly preserved through an online change. Instead, use a standard coil preceeded by an R_TRIG or F_TRIG function block.

IMPORTANT NOTE: For proper operation on Online Change, the setting "Store complex variables in a separate segment" **MUST** be set to "Yes". When setting up the project in "CTI Product Options", this setting is automatically made. However, if the setting is changed to "No" by the user, online change will not work properly.

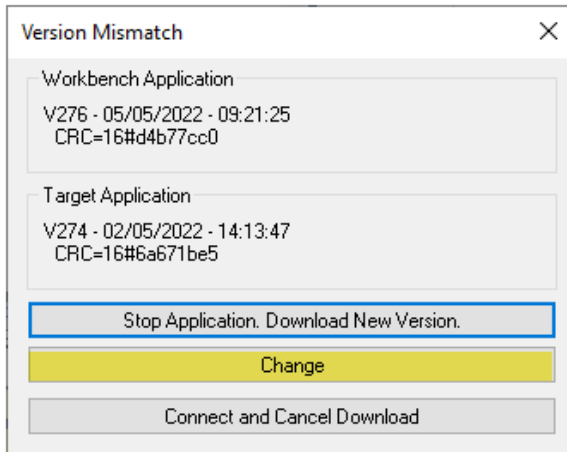


Online Change Procedure

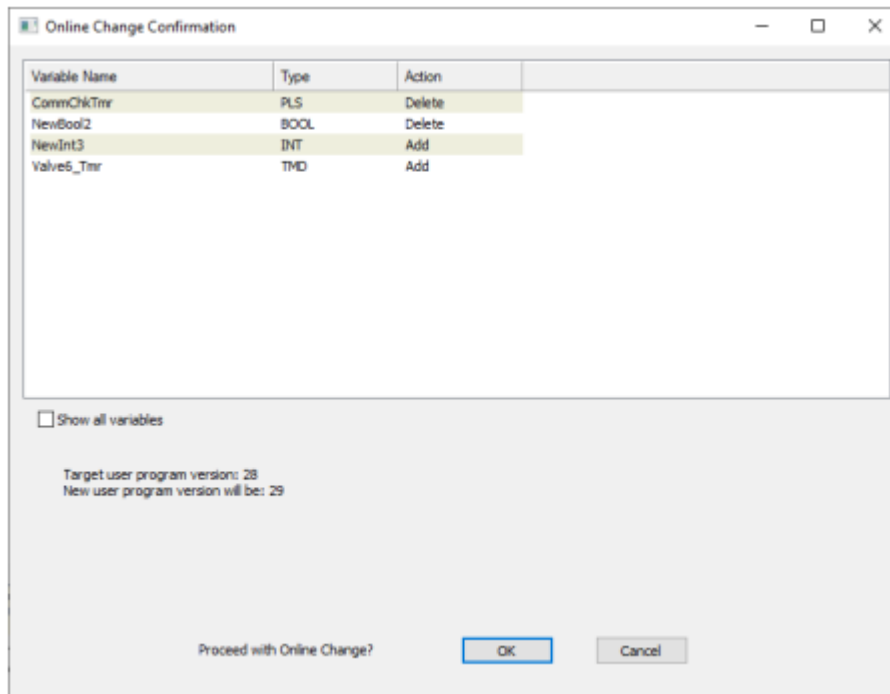
After successfully compiling and creating a new project version while offline (not connected) to the target, you can initiate Online Change by clicking on the 'On Line' button in main toolbar (shown below):



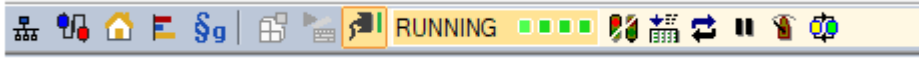
After the following pop-up dialog box is displayed, click on the Change button.



After initiating the download procedure, a confirmation window will be displayed that provides information regarding the effect of the online change operation. Based on this information, you can decide whether to proceed with online change or cancel the operation.



Selecting OK will download the new project executable and related files to the controller. The controller will then execute the Online Change operation. The results of Online Change are recorded in the 'Execution' history log, which can be viewed in the Output pane by selecting the Execution tab in the status bar at the bottom of the Workbench window. Successful operation is indicated by the 'RUNNING' status on the main toolbar as shown below.



Selecting Cancel will terminate the online change process. This will result in a Version Mismatch condition. See the next section for recovery procedures.

Recovery from Online Change Version Mismatch

User cancellation of Online Change or errors that may occur during the online change process will terminate the process while the old application continues in RUN state. When this occurs, the online status will be reported as 'Version Mismatch' since the application in Workbench does not match the one running in the controller (see below). This section describes how to recover from this condition.

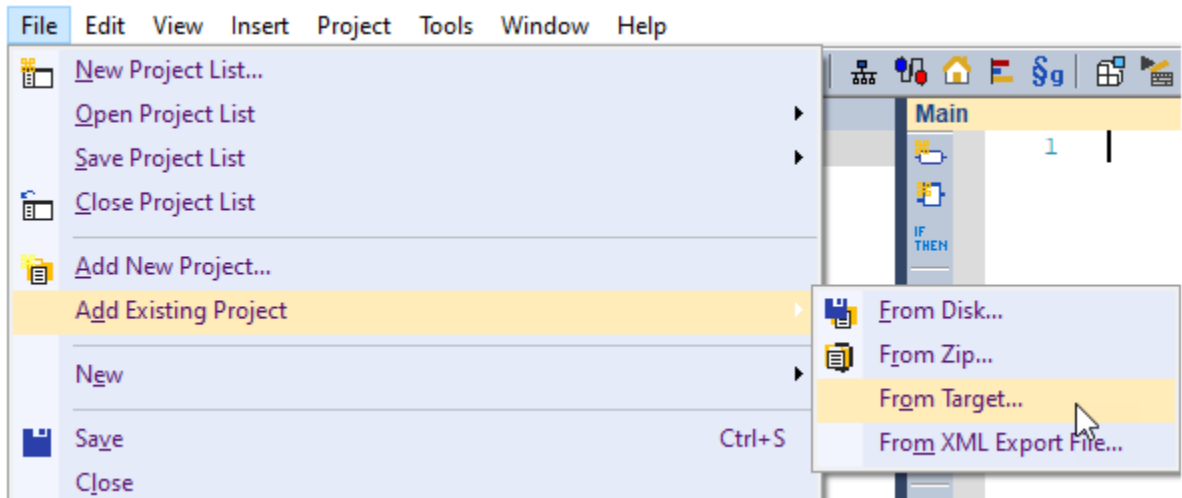


Errors detected by Workbench are reported in a pop-up dialog window. Errors detected by the controller firmware are reported in the 'Execution' history log (viewed by selecting the 'Execution' tab in the Status bar on bottom of Workbench). If any of these errors are detected, you must cancel Online Change by clicking on [Cancel] in the Online Change Confirmation dialog or acknowledging errors reported in pop-up windows.

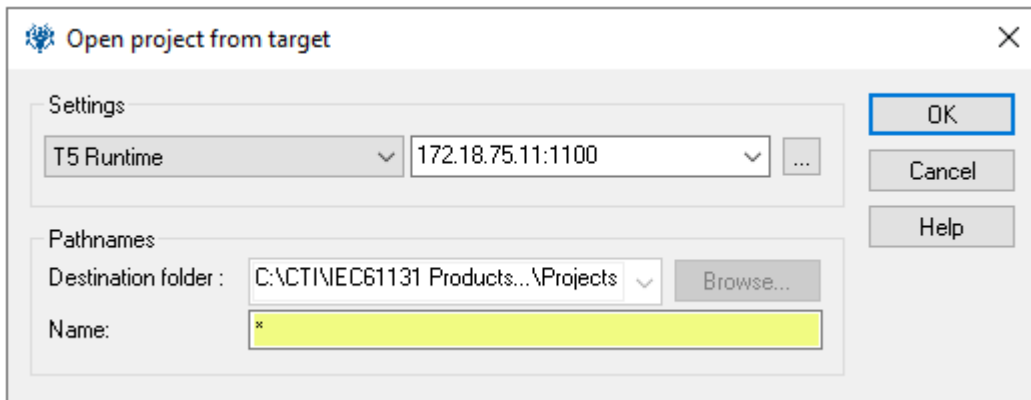
Janus Workbench can be restored to match the executing application through the following steps:

- Halt communications between CTI Workbench and the target by clicking on the 'Online' button in the main toolbar. This will transition Workbench to 'Offline' (or 'development') mode.
- Save the modified project by one of the methods below. *NOTE: Saving the modified project allows you to subsequently import it into the Workspace pane. Once imported, you can compare differences in the project versions and copy/paste changes between them. If you want to discard all changes, you can skip this step.*
 - Right-click on the project name in Workspace pane and choose 'Save Project / To Zip' options from the pop-up menu. A dialog box for entering the name of the zip file will be displayed. The default name contains the project name and version number.
 - Right click on the project name and choose and choose the 'Duplicate Project' option from the pop-up menu. A dialog box for entering a new project name will be displayed. Choosing this option will create a new project in the same workspace.
- Note Project Name and remove the modified project from the Workspace pane by right-clicking on the project name in the Workspace pane and selecting the 'Remove Project' option in the pop-up menu. *NOTE: This action does not delete the project folder or any internal files. It only removes the project from the Workspace pane.*

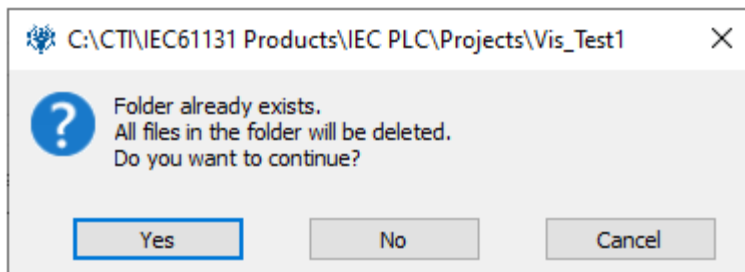
- Upload the source files for the original project. *Note: this action will overwrite the existing (modified) files in the project folder with the original project files. This is the reason the modified project was saved.*
- Select 'File / Add Existing Project/ From Target' from the main toolbar.



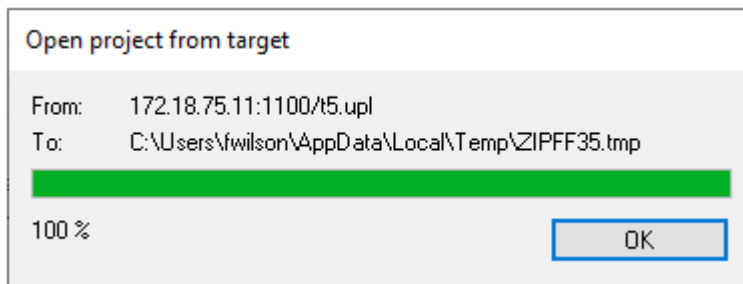
- A dialog box that allows you to enter a project name is displayed as shown below. *NOTE: The project name you enter must exactly match the original project name.*



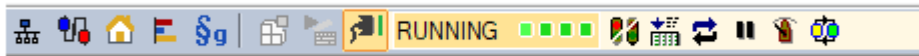
- A dialog box that allows you to enter a project name will be displayed as shown below. The project name you enter must exactly match the original project name. If the match is exact, selecting OK will open the following dialog box. Confirm overwrite by clicking the Yes button.



A progress dialog box will be displayed. After the Open process is complete, click on the OK button to close it.



- The original project (uploaded from the Janus controller) is now loaded onto the Workspace pane and selected as the startup (active) project. Go online to the target by clicking on the button in the main toolbar. The toolbar should indicate normal operation ("RUNNING").



Rules for Copying Variable Values

The following sections describe the conditions for copying variable values between the old project and the new project.

Rules for Basic Data Type Variable Values

A variable is uniquely identified by 'name' and 'data type'. A variable with a data type (BOOL, BYTE, SINT, USINT, WORD, INT, UINT, DWORD, DINT, UDINT, LINT, ULINT, REAL, LREAL, TIME, or STRING) is considered a basic variable.

A variable value is copied only if the 'name' exists in both projects and corresponding 'data type' is unchanged. A variable 'name' includes the variable scope, and therefore the scope must be the same in both projects. *This means a Global variable with name 'VarName1' is not the same as private UDFB variable named 'VarName1 or a local variable named 'VarName1'.*

NOTE: Changing a variable from Global to RETAIN or RETAIN to Global does not change the scope.

NOTE: Janus controllers with firmware version 1.26 and greater, and JACP modules with firmware version 1.05 and greater will preserve the Variable Lock state through the online change process. Previous versions will not maintain Variable Lock state.

STRING variables have two properties: maximum size and current size. Following are the rules for STRING variables.

- If the maximum size of a STRING variable is reduced in the new project:
 - If new 'maximum size' is less than 'current size' of the STRING in the old project, only number of characters that fit in the new maximum size are copied (starting with the first character of the STRING). The 'current size' of the STRING in the new project is reduced to new maximum.
 - If new 'maximum size' is greater than or equal to the 'current size' of the STRING in the old project, all characters in the STRING are copied and 'current size' of the STRING in the new project is unchanged.



- If the maximum size of a STRING variable is increased in the new project, all characters in the STRING are copied and 'current size' of the STRING in the new project is unchanged.

A single-element basic variable that is changed to an array by setting the dimension size greater than '1' is not copied across projects. This rule also applies when a basic data type array is changed to a single variable by setting the dimension size to '1'. None of the array elements are copied across projects.

Attention

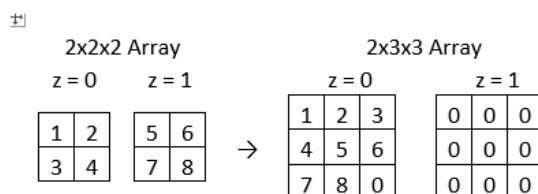
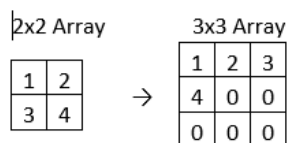
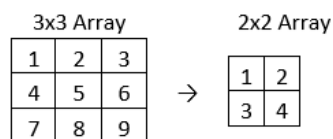
Single variables converted to arrays are not copied because the values are stored in different memory areas when 'Complex variables in a separate segment' option is selected in Project Settings. This option is automatically selected when the CTI Product Type is set to any Janus series product

Rules for Complex Variable Values

- Complex variables include the following items:
 - Variable arrays of basic data types (ie BOOL, INT, DINT, etc)
 - Standard Function Block instances
 - Variables of User Structure types
 - Instances of User Defined Function Blocks (UDFB)
 - Bitfields
 - Enumerations
 - Union variables
- A complex variable is uniquely identified by 'name' and 'data type'. A complex variable value is copied only if the variable 'name' exists in both projects and corresponding 'data type' is unchanged. This means that the variable (or instance) data for a user-defined type (structure or UDFB) is copied only if the 'name' exists in both projects and user-defined type (UDT) is unchanged.
 - A UDT is considered 'changed' if any of the following is detected:
 - UDT 'data type' is changed (i.e. name of structure, UDFB, etc.)
 - UDT member 'is added or deleted (See item (e) below for special rules for union data types)
 - A data property associated with any UDT member is changed: Member name, data type, string length, or dimension
 - Array dimension of any UDT member is changed (See item (d) below for special rules for arrays of basic data types)
- Special rules for UDFBs:
 - UDFBs are considered 'changed' if UDFB member is changed from 'Local Variable' to 'Input Parameter' or vice versa.
- Special rules for Enumerations:
 - Enumerations assign names to integer values based on the definition order.
 - Enumeration types are considered 'changed' when:
 - An existing value name is deleted
 - The order of value names is changed (i.e. associated value is changed)

- Special rules for Bitfields:
 - Bitfields assign names to bit positions within an 8, 16, 32, or 64-bit integer
 - Bitfields types are considered 'changed' when:
 - Bitfield data type is changed (example: changed from SINT to INT)
 - An existing value name is deleted
 - The order of value names is changed (i.e. associated bit position is changed)
- Special rules for Unions (added in Workbench V2.03):
 - A Union has multiple members that use the same memory location.
 - Union types are considered 'changed' when:
 - An existing member is deleted
 - Name or data type of an existing member is changed
 - (Addition of a new member to a union without changing existing members does not constitute a "change" and allows any associated variable values to be copied across projects)
- Special rules for Arrays:
 - If an array dimension size for a variable is reduced in the new project, only the values for elements zero through the new dimensioned size are copied.
 - If an array dimension size for a variable is increased in the new project, the values for elements zero through the old dimensioned size are copied. The remaining elements are set to their initialization value.
 - If the size of a multi-dimensional array with dimensions [z,y,x] is changed, the data is copied as follows:
 - The data copy starts at [0,0,0] and traverses the data starting with the x dimension (column), then the y dimension (row), and finally the z dimension (page). Straton only supports arrays with a maximum of 3 dimensions.
 - i.e., the traversal order is [0,0,0], [0,0,1], [0,0,2], ..., [0,1,0], [0,1,1], [0,1,2], ..., [1,0,0], [1,0,1], [1,0,2] ..., [1,1,0], [1,1,1], [1,1,2], ...
 - The data is copied to the destination array using the same order as the read. If the dimensions of the new array differ from the original, the data is either truncated to fit the array or padding zeros are added to the "new" array indices.
 - Multi-Dimensional Array Examples

Examples



Rules for User Defined Type Values

A complex variable is uniquely identified by 'name' and 'data type'. A complex variable value is copied only if the variable 'name' exists in both projects and corresponding 'data type' is unchanged. This means that the variable (or instance) data for a user-defined type (structure or UDFB) is copied only if the 'name' exists in both projects and user-defined type (UDT) is unchanged.

A UDT is considered 'changed' if any of the following is detected:

- UDT 'data type name' is changed
- UDT Member is added or deleted
- A data property (data type, string length, or dimension) associated with a UDT Member is changed. A change to the member "name" does not prevent variable data from being copied across projects as long as all data properties are unchanged.
- Array dimension of any UDT Member is changed
- The enumeration values or value order of an enumerated type change
- UDT Member array changed from 'Local Variable' to 'Input Parameter' or from 'Input Parameter' to 'Local Variable'.

Rules for Sequential Function Charts (SFC)

The following rules apply to Special Function Charts.

- The SFC POU 'Name' must exist in both old and new projects.
- The code within a step action block can be modified as desired.
- If the structure of the SFC chart (set of Steps and Transitions) for a Main SFC program is identical for both projects, the SFC program in the new project will resume operation at the point it left off in the old project.
- If the structure of the SFC chart is changed, then the SFC state for that SFC POU is reset to its initial state when the new project is started. All 'Child SFC' POUs of this parent will also be reset to their initialization step.

Note

After the **Online** Change procedure is triggered, a Confirmation window will be presented that includes information regarding how each SFC program will be affected by the **Online** Change operation. This allows you to decide whether or not to proceed.

Rules for WAIT and WAIT_TIME Statements

Main Programs and Sub-Programs written in ST language may contain statements 'WAIT' and/or 'WAIT_TIME', which suspend execution of that POU until the specified event becomes TRUE. These statements are extensions to the IEC standard and are not IEC 61131-3 compliant. The operational data associated with these instructions is program-specific and applicable only to a particular instance (version) of the program.

The CTI Online Change procedure will execute normally even if a "WAIT" or "WAIT_TIME" instruction is active. However, in this case, the expression associated with each instruction is set to FALSE when the new project is started. Therefore, all POUs will run from the first instruction.

Rules for Text Buffers

The Straton runtime includes a set of functions (known as 'Text Buffers') for managing collections of character-based data without length limitations imposed by STRINGS. Text Buffers can also be used for transmitting and/or receiving binary data in place of large SINT arrays. The use of Text Buffers requires that dynamic memory allocation be enabled within a fixed-size area managed by Straton memory management functions.

It is the responsibility of the user to save/restore data associated with Text Buffers when using Text Buffers. Data within the Straton fixed-sized memory area is not included in data managed by the CTI Online Change operation, and therefore is not copied from the old project to the new project. However, any variables holding the results of the associated Straton 'Text Buffer' management functions are copied across projects and can be used to help restore the current state of Text Buffer memory area following an Online Change.

Detecting Online Change in User Logic

The Straton GETSYSINFO function (with _SYSINFO_CHANGE_CYCLE parameter) can be used to provide indication of the PLC cycle immediately following an Online Change. This can be used to trigger any special processing required after an OLC operation. This function can be called from POU written in LD, ST, FBD, or IL language.

Example:

```
IF GETSYSINFO(_SYSINFO_CHANGE_CYCLE) <> 0 then
// OLC detected - special code goes here
END_IF;
```

Example for pStartup ApplyRecipeColumn function if variables are NOT to be initialized after an Online Change:

```
IF GETSYSINFO(_SYSINFO_CHANGE_CYCLE) = 0 THEN
ApplyRecipeColumn ('Initial values', 0);
END_IF;
```

Note

The Straton GETSYSINFO function (with _SYSINFO_CHANGE_CYCLE parameter) can be used to provide indication of the PLC cycle immediately following an **Online** Change. This can be used to trigger any special processing required after an OLC operation. This function can be called from a POU written in LD, ST, FBD, or IL language. See example below.

```
IF GETSYSINFO (_SYSINFO_CHANGE_CYCLE) <> 0 THEN
//OLC detected - special code goes here
END_IF
```

Fieldbus Protocol Operation

All I/O Subsystems (Local I/O, Remote I/O, and Profibus) remain operational during the online change procedure. This provides network communications to keep all I/O devices operational and maintains the last state of all outputs. The design of Local I/O and Remote I/O subsystems allow configuration changes, such as adding a module to Local I/O



or adding a Remote Base to Remote I/O, to be made without “bumping” the existing I/O. This is not possible with Profibus due to the constraint that the network configuration can be modified only while network operation is stopped. Therefore, Profibus outputs remain “bumpless” through the Online Change process only when the Profibus configuration is unchanged.

Local I/O

Any changes to the Local I/O configuration take effect when the new project is started.

All Input and Output points configured in the old project maintain their ‘last state’ values throughout the Online Change process. Newly created variables mapped to Local I/O points are set to their initialization values.

All mapped variables retain these values until changed by user logic or forced by operator (outputs) or updated by the I/O driver with data read from specified module (inputs).

Remote I/O

Any changes to the Remote I/O configuration take effect when the new project is started.

All Input and Output points configured in the old project maintain their ‘last state’ values throughout the Online Change process. Newly created variables mapped to Remote I/O points are set to their initialization values.

All mapped variables retain these values until changed by user logic or forced by operator (outputs) or updated by the I/O driver with data read from specified module (inputs).

Major changes, such as adding a new remote base or adding/changing modules within a remote base, require more time, since the remote base controller be reconfigured. Consequently, several I/O cycles will elapse before data can be written or read from that device. All variables retain their values during the reconfiguration process.

NOTE: If a previously configured Remote I/O Base is removed from the configuration during Online change while the base is connected, outputs of the modules installed in these bases will be disabled. However, the base will continue to display an 0 status code after the online change is complete until power is cycled or the RS-485 cable is removed from the base.

Profibus DP I/O

If the Profibus DP network configuration is not changed, the Profibus Master continues normal network communications with all slave devices through the Online Change process, and all variables mapped to I/O points maintain their ‘last state’.

Note

It is possible to set the CTI and/or Siemens Profibus RBC “Freeze” jumper and use the “Fail-safe” option to allow Profibus network reconfiguration without bumping outputs on existing slaves. Other Profibus slave devices may allow I/O points to be configured to hold last state when I/O communication is lost.

If the Profibus DP network configuration is changed (add or delete Slave device or change module configuration of a device), the Profibus Master must be reconfigured. This operation forces all network communications to be stopped (which results in all slave devices setting their outputs to ‘default’ state).

Attention

If the Profibus configuration is changed, the Profibus Master must stop all communications with slave devices while the revised Profibus network configuration is loaded. This will result in “bumping” the I/O unless fail-safe procedures for all slaves is implemented.



Ethernet IP Scanner/Adapter/Flex IO

These rules apply to the EIP I/O Scanner, EIP Adapter, EIP FlexIO, EIP Tag Client, EIP Tag Server, EIP Explicit Message Client, and EIP Explicit Message Adapter, these fieldbus types have dependencies regarding their operation during the Online Change process.

- If the Level 1 -3 configuration is changed for any of these fieldbuses, the drivers for all active EIP fieldbus drivers are reset (stopped and restarted) when the new project is started.
- If the Level 1-3 configuration is unchanged, the fieldbus drivers remain open through the Online Change process and the existing connection(s) are used by the new project. All existing TCP connections are maintained, and the I/O messages are temporarily suspended while the database is copied between projects. The I/O messages are restarted as soon as the new project is running. The Online Change operation should not cause I/O connection timeouts (4 X RPI) between Ethernet/IP devices when the RPI is set to 100 ms and above.
- Level 4 configuration changes will not cause the EIP fieldbus drivers reset.

Attention

*Due to the use common code for the initialization, the presence of EIP Tag Server or EIP Tag Client in the fieldbus configuration will cause the EIP I/O Scanner, EIP I/O Adapter, and EIP FlexIO drivers to reset during the **online** change process regardless of whether the configuration is changed The reset will cause the I/O to bump. The following rules assume that neither of these protocols are configured.*

Other Fieldbus Protocols

This applies to all other fieldbus client and server applications. During the online change operation, all connections are closed, the fieldbus drivers stopped, rebuilt, and restarted with the new project. All variables mapped to data blocks associated with these client protocols maintain 'last state' throughout this process.

