Tech Tips

Using Online Change in 2500 Series® Processors

The CTI 2500 Series® Processor allows you to edit the user program while the process continues to run. While this capability provides significant benefits in some process control applications, it must be approached with care.

WARNING

Use extreme care when performing run-time edits. Incorrect changes may cause the process to fail and could result in equipment damage and/or death or serious injury to personnel.

Carefully plan any run-time edits to an active process.
Avoid doing run-time edits to an active process if possible.

As the name implies, online edit allows changes to be made to the user program while the processor is in RUN mode. When you enter the first program change, the processor automatically enters a special EDIT mode. In EDIT mode, the process continues to be controlled by the original RLL program as it existed prior to entering the change. When you request a return to RUN mode after making all changes, the processor scan is extended while the new version of the program is compiled. Upon a successful compile of the new version, processor transitions to RUN mode and the process resumes with the new version in control.

WARNING

It is possible to enter program changes that will not compile and execute. If the new program will not compile successfully, the processor will enter PROGRAM mode with all outputs frozen at their last state. This could cause unpredictable operation resulting in equipment damage and/or death or serious injury to personnel. It is your responsibility to provide for safe recovery should this condition occur.

Always use the SYNTAX check function to validate all program changes
before setting the processor to RUN mode.

1.1 Overview

All CTI 2500 Series® and SIMATIC®/TI 505 PLCs include a feature called Online Edit that allow most operations in the PLC program to be modified while the PLC is in RUN mode without causing a "bump" to the I/O. Unless specifically noted, the operation of the CTI 2500 Series® PLC exactly duplicates the SIMATIC®/TI 505 processors. The following describes this Online Edit feature including any limitations of what can be modified and provides a procedure for steps to complete an Online Edit operation.

The following changes are permitted during an Online Edit:

a) Use any configured memory address: including Discrete/Word I/O, Word Memory (V,K), Control Relays, and Instructions (Timers/Counter, Drums, Shift Registers, Tables, One-Shots)

b) RLL program: Add, delete, or change RLL networks

c) SF Programs/Subroutines:
   - Add, delete, enable, or disable SP Programs/Subroutines
   - Add, delete, or change statements within an existing program (with some restrictions – see details below)

d) Analog Alarms:
   - Add, delete, enable, or disable Alarm control blocks
   - Change configuration of an existing Alarm control block

e) PID Loops:
   - Add, delete, enable, or disable Loop control blocks
   - Change configuration of an existing Loop control block

f) PLC Port Lockout state (Locked or Unlocked)

g) 505 I/O Configuration (includes Local Base 0 and Remote Bases 1-15):
   - Add, delete, enable, or disable Remote I/O bases
   - Change configuration of an existing base (slot configuration)

h) Profibus I/O Configuration:
   - Limited changes allowed (see details below)

Note also that an Online Edit will preserve a "force" that has been set on any memory address.

1.2 Preparation for Online Edits (in PROGRAM Mode)

1.2.1 *Use 505 WorkShop® PLC Programming Suite (V4.90 or later) if possible*

505 WorkShop performs an automatic "syntax check" of RLL networks during Online Edit operation and detects most errors that would result in a PLC Fatal Error.

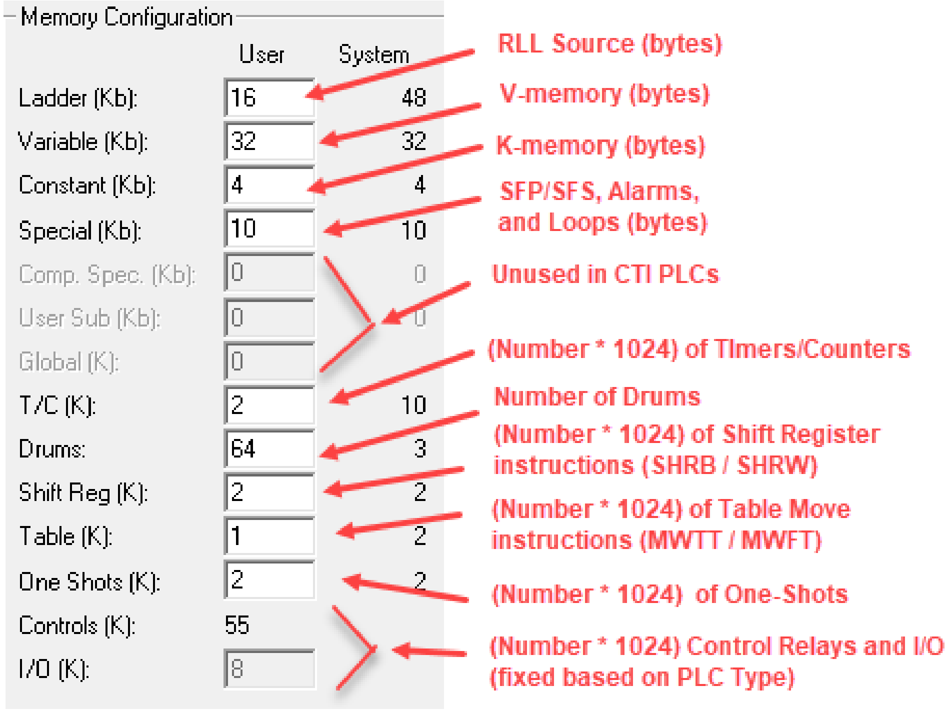
If using TISOFT™, you must use V7.1 (latest version) to ensure the operation of CTI 2500 Series® PLCs during Online Edit matches that of the SIMATIC®/TI 505 processors. Additionally, all diagnostic checks must be performed manually.

All examples in this document are shown using 505 WorkShop (V4.90 and later).

1.2.2 *Set PLC Memory Configuration to Allow Online Edits*

The PLC memory configuration must be arranged so that memory is available to add addresses, instructions, and control blocks to RLL program, SF Programs/Subroutines, Alarms, and/or Loops while a PLC program is in RUN mode.

The memory configuration sets the amount of PLC memory allocated to each of these areas. The following image shows a typical configuration and description PLC function that corresponds to each memory allocation entry.



Memory Configuration

	User	System
Ladder (Kb):	16	48
Variable (Kb):	32	32
Constant (Kb):	4	4
Special (Kb):	10	10
Comp. Spec. (Kb):	0	0
User Sub (Kb):	0	0
Global (K):	0	
T/C (K):	2	10
Drums:	64	3
Shift Reg (K):	2	2
Table (K):	1	2
One Shots (K):	2	2
Controls (K):	55	
I/O (K):	8	

RLL Source (bytes)

V-memory (bytes)

K-memory (bytes)

SFP/SFS, Alarms, and Loops (bytes)

Unused in CTI PLCs

(Number * 1024) of TImers/Counters

Number of Drums

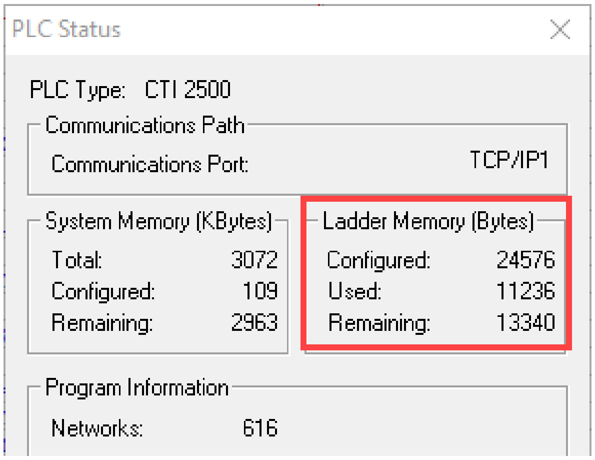
(Number * 1024) of Shift Register instructions (SHRB / SHRW)

(Number * 1024) of Table Move instructions (MWTT / MWFT)

(Number * 1024) of One-Shots

(Number * 1024) Control Relays and I/O (fixed based on PLC Type)

The Ladder (RLL Source) memory usage can be checked by selecting 'PLC Utilities' / 'PLC Status' in the main toolbar. See below.



PLC Status

PLC Type: CTI 2500

Communications Path

Communications Port: TCP/IP1

System Memory (KBytes)

Total:	3072
Configured:	109
Remaining:	2963

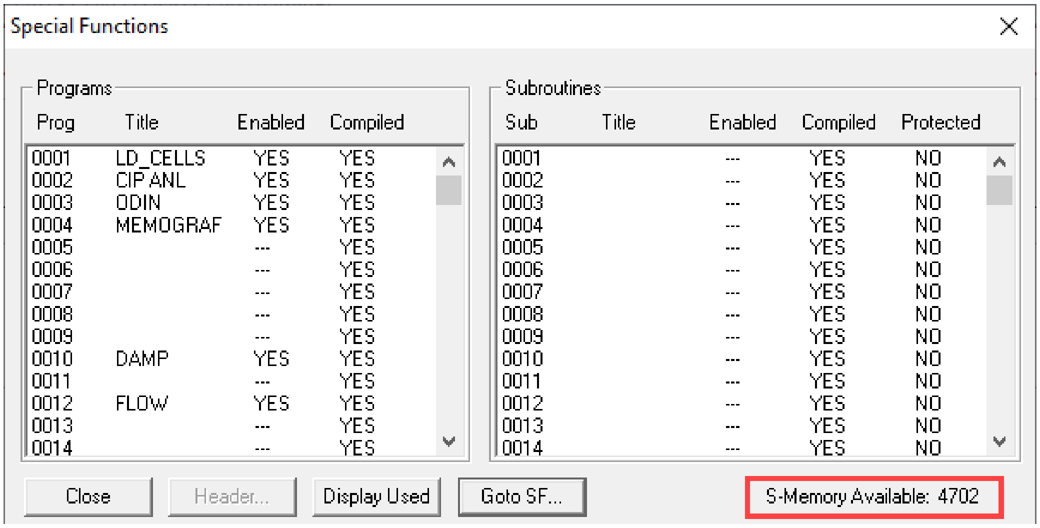
Ladder Memory (Bytes)

Configured:	24576
Used:	11236
Remaining:	13340

Program Information

Networks: 616

The S-Memory (Special memory used for SFP/SFS, Alarms, and Loops) usage can be checked by opening the SF Program list via the fSF button in the main menu.



Special Functions

Programs

Prog	Title	Enabled	Compiled
0001	LD_CELLS	YES	YES
0002	CIP ANL	YES	YES
0003	ODIN	YES	YES
0004	MEMOGRAF	YES	YES
0005		---	YES
0006		---	YES
0007		---	YES
0008		---	YES
0009		---	YES
0010	DAMP	YES	YES
0011		---	YES
0012	FLOW	YES	YES
0013		---	YES
0014		---	YES

Subroutines

Sub	Title	Enabled	Compiled	Protected
0001		---	YES	NO
0002		---	YES	NO
0003		---	YES	NO
0004		---	YES	NO
0005		---	YES	NO
0006		---	YES	NO
0007		---	YES	NO
0008		---	YES	NO
0009		---	YES	NO
0010		---	YES	NO
0011		---	YES	NO
0012		---	YES	NO
0013		---	YES	NO
0014		---	YES	NO

S-Memory Available: 4702

If using TISOFT, available Ladder and Special memory can be determined by accessing AUX 28 under ‘Diagnostics’.

We recommend setting each ‘User’ memory type to at least twice (2X) the amount that is currently used in the PLC program (if possible). Otherwise, allocate the available PLC memory across the memory areas most likely to be affected by Online Edits (such as Ladder memory, Variable memory, and Special Memory).

The PLC Memory Configuration can be changed only while the PLC is in PROGRAM mode.

1.2.3 *Set PLC Scan Time Configuration for Online Edit Processing*

CTI recommends that the PLC Scan Time configuration be set to maximize processing of “deferred” task codes used for most Online Edit operations.

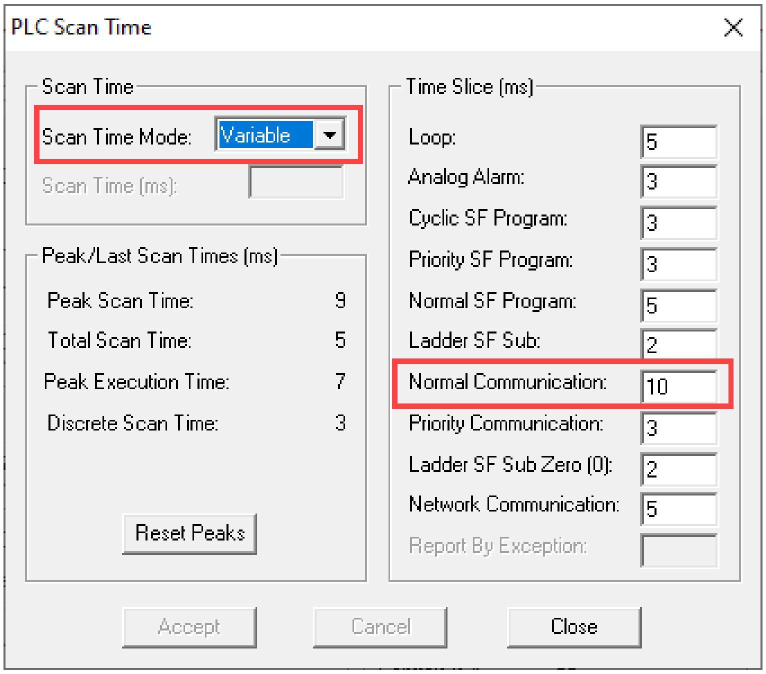
This is not a requirement and does not apply to special applications requiring ‘Fixed Scan Time’ or absolute minimum PLC cycle time under all conditions.

The following PLC Scan Time settings are recommended:

Scan Time Mode: ‘Variable’ or ‘Variable with Limit’

Normal Communication Time Slice: Based on S-Memory Usage: 5ms / 10K bytes
10 msec (minimum)

For example, if the PLC Program uses 40K of S-Memory, the ‘Normal Communication Time Slice’ should be set to 20 msec to ensure all SF Online Edits can be processed in a single PLC cycle.



Scan Time

Scan Time Mode: Variable

Scan Time (ms):

Peak/Last Scan Times (ms)

Peak Scan Time:	9
Total Scan Time:	5
Peak Execution Time:	7
Discrete Scan Time:	3

Time Slice (ms)

Loop:	5
Analog Alarm:	3
Cyclic SF Program:	3
Priority SF Program:	3
Normal SF Program:	5
Ladder SF Sub:	2
Normal Communication:	10
Priority Communication:	3
Ladder SF Sub Zero (0):	2
Network Communication:	5
Report By Exception:	

IMPORTANT NOTE
This is considered a "worst-case" setting. The CTI 2500 Series® PLC only uses time required to complete "pending" requests each PLC scan.
This setting will not increase the PLC scan time except during Online Edit operations. Even then, it is very rare for the 'Normal Communication' processing to exceed 3-4 msec.

1.3 Online Edits to PLC I/O Configurations

All Discrete I/O (X/Y) addresses and Word I/O (WX/WY) addresses supported by the 'PLC Type' (i.e. model) are automatically allocated in the CTI 2500 Series® processor. No user action is required to manage the I/O memory addresses.

1.3.1 *Local/Remote I/O*

The CTI 2500 Series® I/O system supports Online Edits (while the PLC is in RUN mode) without "bumping" the I/O for unchanged module slot positions.

The following operations are supported:

- Add or delete a Remote Base (Base 1-15)
- Enable or disable an existing Remote Base
- Modify I/O configuration of any Local/Remote Base

Add/Delete/Change:

- 'I/O Module Definition' for any module slot (1-16)
- 'I/O Address' mapped to any module slot
- "SF Designation" for any module slot

1.3.2 *Profibus I/O*

The CTI 2500 Series® Profibus-DP I/O system is compatible with SIMATIC®/TI 505 Profibus-DP I/O and allows exactly the same operations as Online Edits (while the PLC is in RUN mode).

Changes to Profibus-DP network configuration, such as Bus Parameters or Slave Configuration (including the addition, deletion, or change of exiting Slave configuration), is allowed while Profibus network is operating. However, this will cause a new Profibus configuration to be downloaded to the PLC and result in “bumping” I/O assignments associated with all Profibus slave devices during initialization of the new Profibus Master configuration.

Only the following changes to the Profibus I/O can be made without “bumping” the I/O without “bumping” the I/O for unchanged module slot positions:

- Change PLC ‘I/O Address’ mapped to a specific Module Slot for an existing Profibus-DP slave

If the CTI 2500 Series® (or SIMATIC®/TI 505) Profibus RBC is configured as Profibus-DP slave _*AND*_ the ‘Slave Parameters’ for the Profibus RBC includes ‘505 Mismatch Mode: Enable’, the following modifications can be made without “bumping” the I/O for unchanged slot positions:

- Insert/Delete/Change:
  - ‘Module’ assignment for any module slot (1-16)
  - ‘I/O Address’ mapped to any module slot

1.4 Performing RLL Online Edits

1.4.1 *General Operation*

The CTI 2500 Series® PLCs include an operational mode (called EDIT mode) that allows for multiple changes (Add, Delete, or Modify) to be integrated into the RLL program at the same time. This provides the capability to add/modify a complete segment of the RLL program without worrying about partial integration of multi-rung operations.

The Online Edit operations and EDIT mode are detailed below.

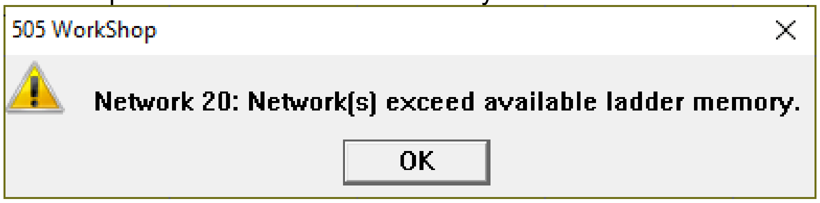
1.4.2 *Add an RLL Network*

Choose the rung immediately below where you wish to insert the new RLL Network.
Right-click and select ‘Insert’ from drop-down menu. Select ‘Network’ in pop-up window.
New ‘blank’ rung appears in the location selected.

Enter new RLL instructions to execute, and press ✓ (or [F8]) to accept the change.
Each RLL Network must be added and accepted individually.

505 WorkShop performs a verification check during each ‘Insert’ rung operation. This is done so an “Out of Ladder Memory” error can be detected and operation aborted before RLL Memory is corrupted. Any syntax errors are displayed only when ‘Transfer to RUN Mode’ is selected (see details below).

An example of an “Out of Ladder Memory” error is shown below:



505 WorkShop

Network 20: Network(s) exceed available ladder memory.

The time to complete this memory verification check is usually < 0.5 second for most RLL programs but can take several seconds for very large programs (25000 rungs). This time can be minimized by setting ‘Normal Communication’ time slice as recommended in Section 2.2.3.

1.4.3 *Delete RLL Network(s)*

Choose the RLL Network to be deleted. Right-click and select ‘Delete’ from drop-down menu. In pop-up window, verify ‘Network’ number(s) to be deleted. It is possible to delete a range of consecutive rungs if desired.

Press ✓ (or [F8]) to accept the change.

1.4.4 *Modify an existing RLL Network*

Select rung to be modified by double-clicking anywhere on the rung.
Rung is highlighted in the Ladder Editor.
Add, delete, or change instructions/addresses in that network.

Press ✓ (or [F8]) to accept the change.

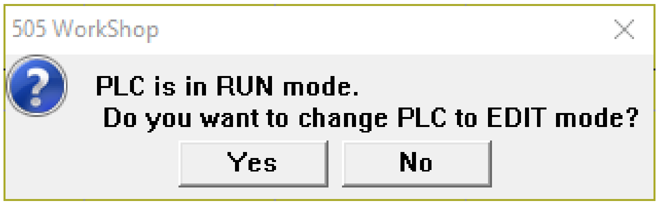
Online edits are accomplished by PLC Programing editor (WorkShop or TISOFT) sending a series of “task code” commands to the PLC as noted below:

- Transfer to EDIT mode.
- Read RLL Network being edited
- Delete Network (if RLL network is being deleted)
- Insert Network (if RLL network is being added)
- Modifying an existing rung (performed by issuing both Delete/Insert Network commands)
- Transfer to RUN Mode (when user transfers PLC to RUN mode)

Details for each of each operation are provided below.

- **Transfer to EDIT Mode:**

This action is triggered by selecting ‘Yes’ in the following pop-up dialog when first modification is “accepted” by the user:


505 WorkShop

PLC is in RUN mode.
Do you want to change PLC to EDIT mode?

IMPORTANT NOTE

It is very important to have an electronic or printed copy of the original RLL program before starting an RLL program Online Edit.

Once the PLC enters EDIT mode (after the first changed is accepted), there is no way to automatically “undo” the changes and revert back to original RLL Source. The only way to exit EDIT mode while the PLC is running is to transfer to RUN mode.as described below.

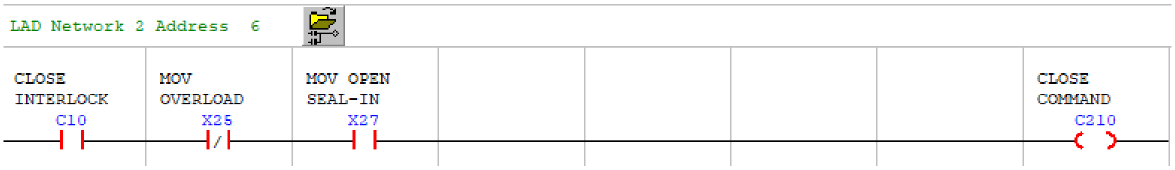
Consequently, if the situation arises where the PLC RLL program must be returned to the original RLL Source after the PLC is in EDIT mode (at least one change has been “accepted”), the only way to accomplish this is to individually “undo” each change and return each rung to its original state.

While in EDIT mode, the RLL Source Code is modified based on the RLL Program changes entered by the user. However, these changes are not yet compiled and executing in the PLC. The PLC is still running the original RLL Compiled Program (before Online Edit operation was started).

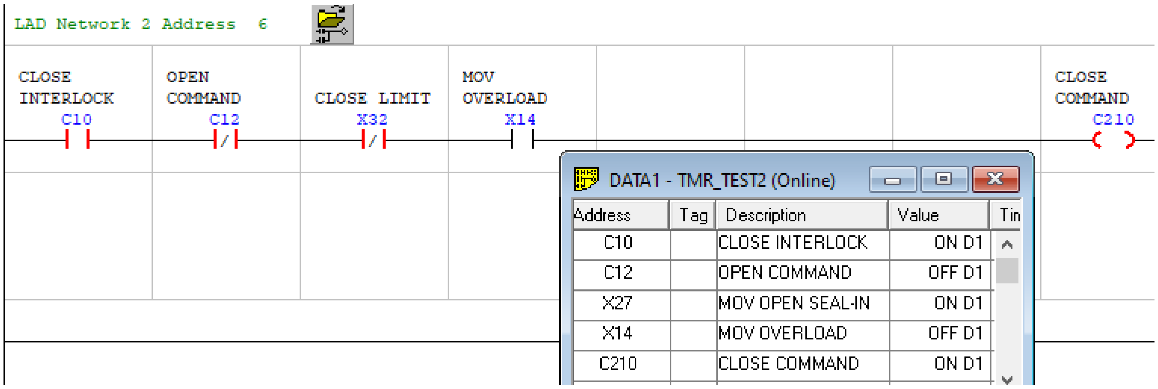
The WorkShop/TISOFT Ladder Editor displays rung information based on the RLL Source Code. Because of this, the status information provided in the Ladder Editor window can be confusing and misleading while the PLC is running in EDIT mode.

For example:

The following rung is in the original PLC program running in the PLC.



During Online Edit, it was changed as shown below and PLC transferred to EDIT mode. Now the Ladder Editor shows the following:



Address	Tag	Description	Value	Tin
C10		CLOSE INTERLOCK	ON D1	
C12		OPEN COMMAND	OFF D1	
X27		MOV OPEN SEAL-IN	ON D1	
X14		MOV OVERLOAD	OFF D1	
C210		CLOSE COMMAND	ON D1	

The rung operation appears to be incorrect because the Output C210 is ON even though input X14 is OFF. Actually, the PLC is still executing the original rung and operation is correct.

- **Read RLL Network being edited**

  This allows WorkShop/TISOFT to extract all RLL Network that is being edited from the RLL Program data file.

- **Delete Network (if RLL network is being deleted)**

  The RLL Network data (read above) is removed from the RLL Program by shifting remainder of the RLL program forward by the number of bytes in the rung that was deleted.

- **Insert Network (if RLL network is being added)**

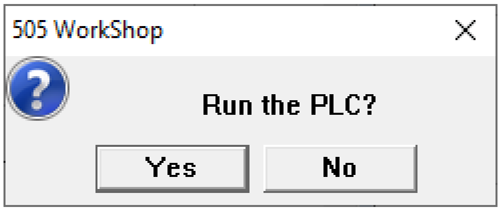
  The new rung is inserted starting at the file position where the edited RLL Network began, and the remainder of the RLL Program is shifted down by the number of bytes in the rung that was added.

- **Modifying an existing rung**

  This operation involves Deletion of the previous rung (RLL Network being edited) and Insertion of the new rung. Both of those operations are described above.

- **Transfer to RUN Mode**

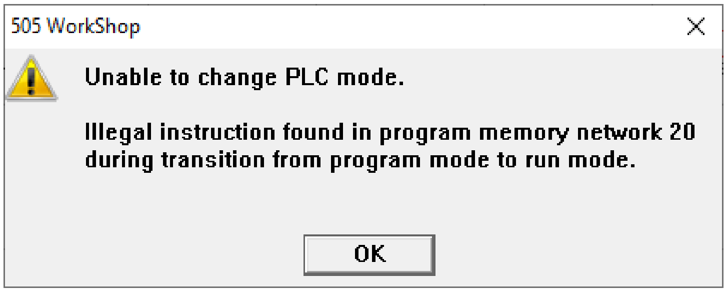
  This operation is triggered by choosing ‘Yes’ in the following pop-up dialog after user selected transfer to RUN mode via the Ladder Editor.


505 WorkShop

Run the PLC?

The following actions are then performed:

a) If syntax error was detected during last edit operation, 505 WorkShop prevents the transfer to RUN mode and displays a message indicating the RLL Network where error is located as shown below. This prevents a failed compile (and resulting Fatal Error) and allows the user to correct the error condition.


505 WorkShop

Unable to change PLC mode.

Illegal instruction found in program memory network 20 during transition from program mode to run mode.

NOTE: If using TISOFT, this 'Syntax Check' must be initiated manually.

b) Compile of PLC Program Source code is initiated. While the RLL Source is being compiled (can take up to 6-7 seconds for very large RLL programs), the PLC program execution (RLL, SFP/SFS, Alarms and Loops) and communications (Serial or TCP/IP) are suspended.

All Remote I/O bases/modules are held at 'last state' via "keep-alive" commands sent out by the PLC about every 350-400 msec.

Profibus-DP Master continues to run using last "output" data. "Input" data is not processed during the compile.

c) If the compile is successful, the PLC starts executing the new RLL compiled code and resumes normal processing at the start of the next PLC scan cycle. At this point, the Ladder Editor displays match the executing code.

1.4.5 *Potential Sources of Run-Time Edit Compile Errors*

Following are some conditions that will cause the RLL compile to fail, resulting in the processor being placed in PROGRAM mode with outputs frozen, if executed. Always execute a syntax check before attempting to go to RUN mode.

SKP Instruction without a Corresponding LBL

There must be a LBL statement associated with each SKP instruction and it must occur in the same program segment (SBR or TASK) as the SKP instruction.

SBR instruction without a terminating RTN

A subroutine must be terminated by an **unconditional RTN** instruction.

GTS, PGTS or PGTSZ without corresponding SBR

The subroutine referenced by a GTS, PGTS, or PGTSZ instruction must be defined before it can be referenced.

Use of unsupported features

Your RLL program must not use an instruction that is not supported by the firmware release installed in your processor or reference undefined or unconfigured data elements. This condition may not be detected by all versions of all programming software tools.

Exceeding L Memory

When you modify or add networks to an RLL program using the run-time edit function, it is possible for the edited program to exceed the amount of L-Memory that has been configured. If the configured L-Memory capacity is exceeded, one or more networks at the end of the program will be deleted

1.4.6 *Additional Considerations*

When you edit an existing network, Workshop or TISOFT will delete the existing network and then insert the edited network in its place. If the original network contains an instruction with retained state information and this instruction remains in the network after the edit, you may experience unexpected results when transferring to RUN mode. These unexpected results occur due to initialization of the state information for the "retained state" instruction.

For example, an existing network contains a One-Shot contact that passes power flow for one scan when detecting an OFF-to-ON input transition. If the One-Shot input condition has been TRUE for more than one scan, the output coil is turned OFF and will remain OFF until the input state goes FALSE and back TRUE. However, if the network is edited at this point, the "retained state" of the One-Shot will be lost and re-initialized when the program is compiled so that the output coil will turn ON for one scan immediately following the transfer to RUN mode.

WARNING

Take extreme care when performing a run-time edit on an existing network that contains one or more "retained state" instructions. When returning to RUN mode following the edit, these instructions are re-initialized during the program compilation. This may cause the network output coil(s) to temporarily change state.

You may experience unexpected results that could result in damage to equipment and/or death or serious injury to personnel. If you must edit a network containing one of these instructions, you must consider the effect upon the process caused by this initialization and ensure that the process state can safely handle this effect.

The instructions with retained state information are shown in the following table.

Operation of Retained-State Instructions in Networks affected by Run-Time Edits	
Instruction	***Initial Condition After Run-Time Edit***
CTR	Initialized to require OFF-to-ON transition of the count input. TCP (count preset) is set to the instruction's preset value and TCC (current count) is set to 0.
DCAT **MCAT**	TCP (time preset) and TCC (Time Remaining) are set to the Preset value in the DCAT/MCAT instruction. As a result, the Alarm Timer is restarted
DRUM	DSP (Preset Step) and DSC (Current Step) are set to the Preset Step specified in the DRUM instruction. DCC (Current Count) is set to the programmed count for this Preset Step. The process is now controlled by the Preset Step.
DSET	Initialized to require a OFF-to-ON transition of the input.
EDRUM **MDRMD** **MDRMW**	The Count Preset values for each of the Drum steps are copied from the EDRUM instruction to the corresponding DCP (Count Preset) variables. DSP (Preset Step) and DSC (Current Step) are set to the Preset Step specified by the instruction and DCC (Current Count) is set to the programmed count for this Preset Step. Finally,

Operation of Retained-State Instructions in Networks affected by Run-Time Edits	
Instruction	***Initial Condition After Run-Time Edit***
	the Jog Input is initialized to require OFF-to-ON transition. The process is now controlled by the Preset Step.
MWFT **MWTT**	The Table Pointer is set to the table base and the Move Count is set to 0.
OS	Initialized to set the Output on the first scan for which the Input is TRUE.
SHRB **SHRW**	Initialized to require an OFF-to-ON transition on the input.
TMR **TMRF**	TCP (Time Preset) and TCC (Time Remaining) are set to the Preset value in the TMR/TMRF instruction. As a result, the Timer is restarted.
TSET	Initialized to require an OFF-to-ON transition of the input.
UDC	Initialized to require an OFF-to-ON transition of the Count input. TCP (Count Preset) is set to the specified value and TCC (Current Count) is set to 0.

1.5 Performing SF Program Online Edits

This section details the Online Edit operations for the CTI 2500 Series® PLCs.

1.5.1 *Organization of SFPGM/SFSUB Memory*

All SF Programs are stored sequentially in S-Memory with SFPGMs first followed by SFSUBs. The lowest numbered is first (SFPGM1 is first if it exists) followed by the next higher number.

If a new SF Program is added, it is inserted into its proper position and all other programs are shifted as needed. The same situation applies if a SF Program is deleted or edited. Any edit operation that uses more or less memory causes a shift in SF Programs stored in subsequent S-memory positions (see NOTE below).

The memory organization provides very efficient program storage, but it also has the disadvantage of allowing an edit of one SF Program to cause all other SFPGMs/SFSUBs that follow in S-Memory structure to be disabled while the S-Memory area is being shifted.

IMPORTANT NOTE
Each Online Edit to a SF Program that includes a change to amount of memory used (add/delete line or add/delete a parameter in an expression) will result is S-Memory being shifted to accommodate the change.
This operation causes the SF Program being edited and all of the SFPGMs/SFSUBs that are stored in S-Memory following it to be disabled while memory is being shifted.
This operation usually can be completed within one PLC scan in most PLC programs. However, performing SF Online Edit on PLC programs containing a large number and/or size of SF Programs can result in extended disabled times for multiple SF Programs.
See recommended settings to minimize this issue in Section 2.5.4.

1.5.2 *Differences between CTI 2500 Series® and SIMATIC®/TI 505 SF Programs*

All SFPGM/SFSUB programs in the CTI 2500 Series® PLC run as “compiled” programs. This allows them to run much faster that interpreted programs, but it requires that each program meet very strict syntax requirements in order for the compile operation to succeed.

SIMATIC®/TI 505 PLCs with PowerMath™ option allow “compiled” SF Programs with the same requirements as CTI 2500 Series® PLCs. The difference is that these processors also allow “interpreted” SF Programs with less syntax rules.

1.5.3 *Online Edit Operation for SF Programs*

Online Edit of SF Programs is allowed, but it is not supported in the same way as done with the RLL program. Online Edit of RLL program includes a run-time EDIT mode that allows changes to multiple networks to be entered but not executed until the PLC is transferred to RUN mode.

The CTI 2500 Series® PLCs do not include a similar EDIT mode for SF Programs. Below are the rules for SF Program Online Edits:

- An SF Program Online Edit can consist of any of the following operations:
    - Add, delete, enable, or disable a SFPGM or SFSUB
    - Modify SF Program Header, including Title, Program Type, and Error Handling
    - Add or Modify one SF statement (line) of existing SFPGM/SFSUB
    - Delete one or range of consecutive SF statements
    - The PLC stores and compiles the entire SF Program as soon as it is “accepted” by the user. The SF Program is disabled during this operation.
        - If the SF Program was enabled at the start of the edit operation and the operation is successful, the SF Program is re-enabled.
        - If the compile fails (see explanation below), the SF Program is disabled and must be manually enabled via the SF Program Header dialog box.
    - Because each SF statement is compiled immediately after it is changed, there are cases that can result in the SF Program failing to compile. The insertion of any of the following statements during Online Edit will cause the SF Program to be disabled:
        - Add or Delete a ‘GOTO’ statement without a corresponding ‘LABEL’
          NOTE: This can be averting by adding the ‘LABEL’ or deleting ‘GOTO’ first.
        - Add or Delete ‘IF’, ‘ELSE’ , or ‘ENDIF’ without corresponding statements
          NOTE: Since it is not possible to add multiple statements on the same line, a new ‘IF-ENDIF’ segment cannot be added to a SF Program during Online Edit without disabling the program. It is possible to add an ‘ELSE’ statement to an existing ‘IF-ENDIF’ segment.
        - Addition of ‘FOR’ or ‘NEXT’ without corresponding statement
          NOTE: Since it is not possible to add multiple statements on the same line, a new ‘FOR-NEXT’ segment cannot be added to a SF Program during Online Edit without disabling the program.
        - Addition of ‘WHILE’ or ‘ENDWHILE’ without corresponding statement
          NOTE: Since it is not possible to add multiple statements on the same line, a new ‘WHILE’ segment cannot be added to a SF Program during Online Edit without disabling the program.
        - Addition of ‘’SWITCH’, ‘CASE’, or ‘ENDSWITCH’ without corresponding statements
          NOTE: Since it is not possible to add multiple statements on the same line, a new ‘SWITCH’ segment cannot be added to a SF Program during Online Edit without disabling the program.

1.5.4 *Recommendations for SF Program Online Edits*

If the Online Edit involves a minor change that can be accomplished within the "one statement at a time" restriction, it is possible to perform those modifications by a SF Program Online Edit.

For more complex changes requiring Add/Delete/Modify of multiple lines in a SF Program, we highly recommend that the SF Program be disabled (in SF Program Header) before modifying an existing SF Program. This will ensure that the SF Program is not executed while edits are in progress. The SF Program can then be manually 'Enabled' after edits are completed.

The time required to store, compile and re-enable the SF Programs after each edit depends on the number and size of SF Programs that exist. This time can be minimized by setting the 'PLC Scan Time' settings as described in Section 2.3.

If it is absolutely necessary to modify multiple statements in an existing SF Program without disabling it for an extended period, there is one work-around to this restriction by using the **[Load by Parts]** feature in WorkShop or TISOFT. This allows you to edit SF Program(s) off-line and then download the entire programs to the PLC while it is in RUN mode.

The SF Program is disabled only during the time while the PLC is overwriting the existing program. After the complete SF Program is received, it is automatically compiled and enabled (if compile is successful).

The drawback is that the **[Load by Parts]** function downloads all items in the specified group (SFPGMs or SFSUBs) that exist in the off-line program. You cannot specify an individual SF Program to download. This can result in the SF Program(s) being disabled for an extended time if many SF Programs exist in the PLC Program. However, there is also a work-around for this in the procedure described below:

Procedure to perform SF Program Online Edit via **[Load by Parts]** feature:

1. Save current PLC program to disk or use an existing offline copy of running program.
2. Open PLC program offline. Delete all SF Programs in SF group (SFPGMs or SFSUBs) except the ones that need modification.
3. Modify SF Program Header, statements, and/or Error Handling settings. The number of the SF Program must remain unchanged.
4. Connect to PLC and select **[Load by Parts]**.
5. Choose 'Load SF Programs' or 'Load SF Subroutines' and press [OK].

1.6 Online Edits to Alarm and Loop Blocks

The CTI 2500 Series® PLCs allow unrestricted changes to Analog Alarm and PID Loop control block configurations while the PLC is in RUN mode.

The following changes to Alarms and Loops are permitted during an Online Edit:

- New Alarm/Loop block can be created on the condition that that unused S-Memory is available.
  - Each Alarm block requires about 125 bytes of S-Memory
  - Each Loop block requires about 175 bytes of S-Memory
- Existing Alarm/Loop block can be deleted.
- Existing Alarm/Loop configuration can be modified, enabled, or disabled.

All changes to Alarm/Loop control blocks take effect at the end of the PLC cycle when the change was "accepted" (when [OK] button is pressed).