

Appendix F

Converting Ladder Programs Created Using an Older Version of AutoMax

You must convert any ladder program that was created using an AutoMax Executive earlier than V4.X before you can edit it using the V4.X ladder editor and the enhanced ladder instruction set. Once you convert the program, you cannot use it with an earlier PC editor.

You can convert ladder programs by doing any of the following:

- selecting a pre-V4.0 system from the System Configurator
- copying a system, section rack or program
- opening a program from the Editor

See “Opening Programs,” section 1.2.

The software prompts you to convert the system, which includes converting old ladder programs. Any conversion warnings are posted in a log file. If the programs are converted when a system, section, or rack is converted, the warnings are placed in the file UPDATE.LOG. If you convert an individual program from the Editor, the warnings are placed in the file *program_name*.LOG. You can find UPDATE.LOG in the system directory, and *program_name*.LOG in the rack directory.

At the dialog box prompting you to convert the program, choose Yes. The Editor does the following:

- replaces the old ladder instructions with the corresponding ones from the enhanced ladder instruction set
- converts any remark sequences into a rung consisting of one coil
The coil names are those of the keywords from the old remarks. These new variables are local variables. The remark text becomes the rung description for the new rung. Any tabs in the remark sequences are converted to single space characters.
- copies any variable description text to the new program and truncates the text to 40 characters
- renumbers the rungs starting at number one
- preserves any preset values for timers and counters
- creates new timer and counter data structures by using the variable name of the current timer or counter value as the name of the new timer or counter data structure

The EN input of the new counter instruction is connected to the power rail.

- converts the task time (in seconds) to the equivalent number of ticks as defined for the Processor within which the program is slated to run
- deletes any descriptions associated with individually defined array elements
These descriptions are not carried over to the converted program.
- converts shift instructions to a Shift Left Instruction. See “About How Shift Instructions Are Converted into the Ladder Editor” in this Appendix.

To save the converted file, choose the Save command. If you quit the Editor or close the converted program before saving the program, the original, unconverted program is unchanged.

The Editor creates a log file if any conversion error or warning messages are generated. See “Correcting Ladder Program Conversion Warnings and Errors” in this Appendix.

Once the Editor converts the program, it renames the original program file to *file.@PC* and renames the original remark *file.REM* to *file.@RE*. You can delete these files later if you wish.

F.1 About How Shift Instructions Are Converted into the Ladder Editor

Any shift register instructions present in the old ladder program are converted to a Shift Right (SR) instruction. The logic that drives the SHIFT and DATA parameters is converted to correspond with the SR instruction’s EN and BIT parameters, respectively. The following table describes how other conditions are handled.

If the:	Then:
old instruction had a shift register length that is less than 16	The shift register length for the corresponding SR instruction is 16 (an integer). Note that all the bits within the integer are shifted. A message is posted in the file UPDATE.LOG or <i>program_name</i> .LOG.
RESET input parameter was programmed in the old instruction	A MOVE block is added to the rung. This block is driven by the logic that was connected to the RESET parameter. The MOVE instruction resets the shift register integer when the reset coil is true.
DATA and SHIFT parameters were not programmed in the old instruction	The rung is converted into a MOVE instruction and a warning is posted in the file UPDATE.LOG or <i>program_name</i> .LOG.

F.2 Correcting Ladder Program Conversion Warnings and Errors

This section lists some of the warning messages generated when you convert an old ladder program to a V4.0 or later program and describes what you can do to fix them. The messages are posted in the file UPDATE.LOG or *program_name*.LOG.

Message:	What the message means:	Resolution:
In old Rung number <i>xx</i> (new Rung number <i>xx</i>), the old global preset variable is no longer used.	The variable assigned to the PN parameter in the old timer or counter instruction was a global integer variable.	To use a global variable as a timer or counter preset, define the timer or counter data structure as a global, double integer array by using the Variable Configurator. For more information about preset values, see "About Timer Variables" and "About Counter Variables."
In old Rung number <i>xx</i> (new Rung number <i>xx</i>), the old global current value variable is used.	The variable assigned to the CN parameter in the old timer or counter instruction was a global integer variable.	To use a global variable to store a timer's or counter's current value, define the timer or counter data structure as a global, double integer array by using the Variable Configurator. For more information about current values, see "About Timer Variables" and "About Counter Variables."
In old Rung number <i>xx</i> (new Rung number <i>xx</i>), a preset variable is used that was also used on another rung.	In older ladder programs, you could use the same preset variable on multiple timers or counters in the same program. Now, each timer or counter instruction in a program must have a unique data structure assigned to it.	To use the same preset variable on multiple timers or counters, add MOVE instructions to copy the preset value from one of the timer/counter data structures to another timer/counter data structure.