SERIAL2.EXE
Serialization Utility for TaskLink DOS

1. SERIAL2.EXE Serialization Utility for TaskLink DOS

1.1 Introduction

SERIAL2.EXE is a new External Serialization Program (ESP) created for TaskLink for DOS. The purpose is to support the many customer requests to have multiple serial numbers, at different addresses within one device. It allows individual control of each serial number and address for each serial number. This is provided by reading the serial data from a pre-generated list of serial numbers. The pre-generated list of serial numbers will have to be generated by the customer before the beginning of a job. The format of the pre-generated list of serial numbers is similar to (but not the same) as the .CSV file format used by TaskLink 9500 and TaskLink for Windows. The serial data is read from the .CSV file and converted into the SERIAL.DAT file format currently used by TaskLink for DOS and the SERIALIZ.EXE utility. In this way, no changes need to be made to TaskLink for DOS and compatibility with existing versions of TaskLink for DOS is provided.

1.2 Purpose

The purpose of this document is to provide the user with the information necessary to use the SERIAL2.EXE utility. It provides an overview of the requirements or capabilities of the program. It provides a general overview of the way serialization is accomplished using TaskLink for DOS. It details the format of the .CSV file and how to generate them. It provides information on how to set-up TaskLink and descriptions of the different error messages generated by the program.

All of the information for the SERIALIZ.EXE utility can be found in the TaskLink on-line help. TaskLink on-line help has not been updated to reflect the new SERIAL2.EXE utility. This document serves to provide the same level of detailed documentation as the on-line help does for the original utility.

This document also serves as a requirements document and design document during development of the utility.

1.3 Scope

This document is limited to providing information on how to use SERIAL2.EXE. It does not provide information on how to use the existing SERIALIZ.EXE program or how to write custom External Serialization Programs. It does not cover the operation of TaskLink except where it relates to the use of this program.

The Serialization Overview section of this document assumes the reader is familiar with the current SERIALIZ.EXE program and the concepts of serialization.

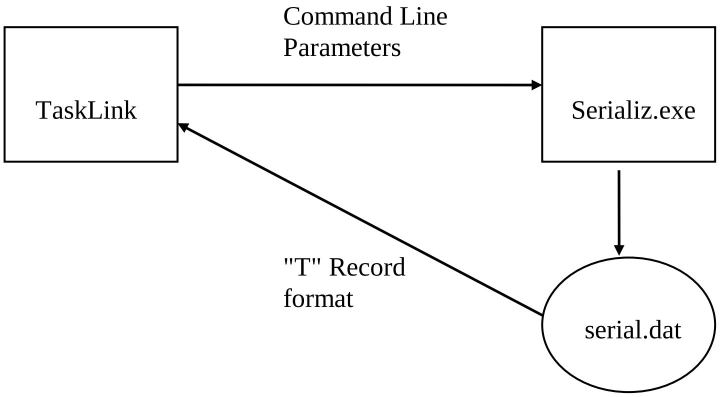
1.4 Definitions

.CSV	Comma Separated Variables
ESP	External Serialization Program

2. SERIAL2.EXE / Serialization Overview

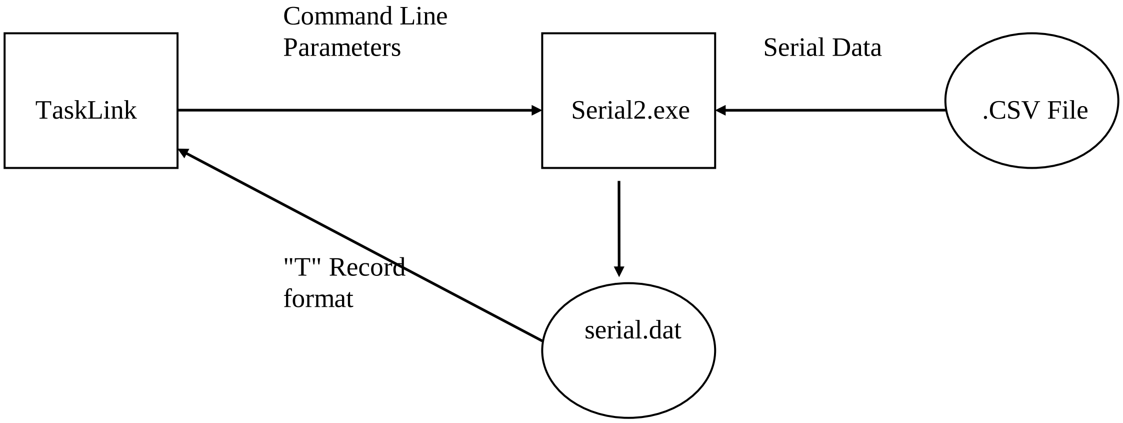
2.1 SERIALIZ.EXE Design (existing TaskLink DOS Method)

TaskLink DOS shells out to an external .EXE to generate the serial number. This is done for each device(s) programming cycle. The external serialization program (ESP) generates the serialization formation based on the command line parameters. The serialization information is passed back to TaskLink via a file named SERIAL.DAT. This file contains the information TaskLink needs to perform serialization of the device(s), in the "T" record format.



2.2 Serial2.exe Design (new Serialization Method)

TaskLink DOS shells out to an external .EXE to generate the serial number. This is done for each device(s) programming cycle. The external serialization program (ESP) reads the serialization formation from a .CSV file. The serialization information is converted and passed back to TaskLink via a file named SERIAL.DAT. This file contains the information TaskLink needs to perform serialization of the device(s), in the "T" record format.



2.3 SERIAL2.EXE and the .CSV file

SERIAL2.EXE and the .CSV file support up to 8 serial data fields per device for RAM serialization. Only 1 serial data field per device is supported for Label serialization. A single serial data entry can be used for either RAM, Labeling or Both.

The .CSV serial data file consists of a separate line entry for each serial number or group of serial numbers per device. Each line entry consists of 5 variables separated by commas on a single line. Each line should end with a new line character.

Blank lines can be used in the file to help separate serial number groups and make the file more readable. No headers or comments are allowed in the file. A line must either be blank or a properly formatted serial number entry. Spaces can be used within variables to allow for alignment of the commas and readability.

When using multiple serial numbers per device and the sumcheck calculation flag (-s), the sumcheck is calculated for all serial numbers used in the device. The serial number calculation is done in hexadecimal. Serial Number 0010 plus Serial Number 0010 = 20 in hex (32 in decimal), and will add the value 20 hex to the TaskLink base sumcheck. This feature is not recommended if one of the serial data fields is being used as a sumcheck compensation value.

When using multiple serial numbers per device on a gang programmer, pure-set mode (-m) must be used.

2.4 The .CSV file format

The format of each line in the .CSV file is composed of the following variables:

Index, Serial Data, Address, Data Length, Type

Index: A decimal number from 1 to 4294967295. Must be a decimal number and not hexadecimal. Must be in consecutive order.

Serial Data: A hexadecimal numbers limited to 20 bytes (40 ASCII characters for RAM data and 20 ASCII characters for label only data). If the serial data field is shorter than the length specified, then the serial data is left padded with zero's. Remember label data truly limited by the parameters set up in TaskLink, such as font size, label size, number of percent signs, etc…

Address: A hexadecimal number from 0 to FFFFFFFF. The address parameter is not used when the type field is set to label only.

Data Length: A decimal number representing the length of the serial data in bytes. There are two ASCII characters per byte, for RAM serialization. There are one ASCII characters per byte, for label only serialization. Must be a decimal number and not hexadecimal. The data length field is not used when the type field is set to label only. If the serial data field is shorter than the length specified, then the serial data is left padded with zero's.

Type: "R" indicates the data is for use in RAM serialization, "L" indicates the data is for use only on a label, "B" indicated the data is used in both locations. A type field must be specified.

2.5 .CSV Examples

Example 1 is a file that contains one serial number per device. The serial number is for use in RAM and Label serialization (this will be programming system dependent). Both the Label data and the RAM data are the same. The serial number is programmed at address 7FFFF. The length of the serial number is 2 bytes (4 ASCII characters). 4 ASCII characters will be printed on the label in place of the 4 percent signs %%%%.

Example 1:

```
1, 0001, 7FFFF, 2, B
2, 0002, 7FFFF, 2, B
3, 0003, 7FFFF, 2, B
4, 0004, 7FFFF, 2, B
```

Example 2 is a file that contains two serial numbers per device and separate label data. The first serial number is a 6 byte (12 ASCII Characters) Ethernet number programmed at address 70000. The second serial number is a 2 byte date code programmed at address 7FFFF. The label text is the last 3 bytes of the Ethernet number.

Example 2:

```
1, AABBCC112233, 70000,  6, R
1,         0697, 7FFFF, 2, R
1,       112233, , , L
(Blank Line)
2, AABBCC112234, 70000,  6, R
2,         0697, 7FFFF, 2, R
2,       112234, , , L
(Blank Line)
3, AABBCC112235, 70000,  6, R
3,         0697, 7FFFF, 2, R
3,       112235, , , L
(Blank Line)
4, AABBCC112236, 70000,  6, R
4,         0697, 7FFFF, 2, R
4,       112236, , , L
```

Example 3:

1, 1, 7FFFF, 2, B	Serial number is left padded with 3 zero's to the 2 byte length specified.
2, 0002, 7FFFF, 2, B	Serial number is exact length. No error, no padding.
3, 00003, 7FFFF, 2, B	Returns error because the serial data is too long.
4, 00004, 7FFFF, 2, L	Length is ignored, since the data is for label only. No error.

2.6 .CSV file validation

There are two sets of rules SERIAL2.EXE will use to validate the .CSV. The first set of rules will apply to each line of the file. If the file violates any of the following conditions, the error message "Invalid line format in line, xxx" will be displayed.

- All five fields must be present and separated by a comma.
- There must be a valid index field.
- There must be a valid serial data field.
- There must be a valid address field, if the serial data is used for RAM serialization.
- The address field can be left empty if the serial data is used for label only.
- There must be a valid length field, if the serial data is used for RAM serialization.
- The length field can be left empty if the serial data is used for label only.
- There must be a valid type field. The field must be a "B", a "R" or a "L".

The second set of rules will apply for gang programmers, where the SERIAL2.EXE may read more than one serial index at a time. If the file violates any of the following conditions, the error message "Serial data is not uniform" will be displayed.

- If one device contains label information then all devices must contain label information.
- The number of serial data fields per device must be the same for each serial index requested.

- The length of the serial data fields must be the same for each corresponding serial data field in each serial index. (Serial data field 1 for index 1 must be the same length as serial data field 1 for index 2. Serial data field 2 for index 1 must be the same length as the serial data field 2 for index 2, etc.)
- The address for the serial data fields must be the same for each corresponding serial data field in each index.

Valid	Invalid
1, AABBCC112233, 70000, 6, R	1, **AABBCC112233**, 70000, **6**, R
1, 0697, 7FFFF, 2, R	1, 0697, 7FFFF, 2, R
2, AABBCC112234, 70000, 6, R	2, **112234**, 70000, **3**, R
2, 0697, 7FFFF, 2, R	2, 0697, 7FFFF, 2, R

2.7 Generating a .CSV file

There are many ways a .CSV serialization file can be created. One of the easiest ways is to use an spreadsheet application such as Microsoft Excel or other equivalent program. Create a few sample rows by hand and then get the spreadsheet to create the remainder for you. Then save the spreadsheet as a text file (.CSV file). You are done.

Sample Spreadsheet:

1	AABBCC000001	70000	6	R
1	0001	7FFFF	2	R
=A1 + 1	=B1 + 1	=C1	=D1	R
=A2 + 1	=B2 + 1	=C2	=D2	R
=A3 + 1	=B3 + 1	=C3	=D3	R
=A4 + 1	=B4 + 1	=C4	=D4	R
=A5 + 1	=B5 + 1	=C5	=D5	R
=A6 + 1	=B6 + 1	=C6	=D6	R

1. Create the following example.
2. Highlight the sample rows and columns of data.
3. With the sample rows and columns of data highlighted place the mouse cursor over the little square, black box in the lower right corner of the main box surrounding the highlighted data. The cursor should change to a plus sign (+);
4. Hold down the left mouse button on the little black square box and drag it down the spreadsheet.
5. Release the mouse button.
6. If all goes well the spreadsheet will automatically fill in repeating the pattern it learned from the highlighted sample.
7. Verify data closely.
8. Save the data.
9. This example works for some simpler cases. You may have to get creative.

Note: **Remember to save a copy of the spreadsheet file. When the serial numbers run out and it is necessary to create more, you don't want to have to re-create this every time.**

3. Using with TaskLink

3.1 General Concept

In general SERIAL2.EXE works with TaskLink exactly the same way as SERIALIZ.EXE does with one exception. The concept of a serial number gets replaced with the concept of a serial index.

Instead of entering a starting and ending serial number you enter a starting and ending serial index. Instead of saving the next starting serial number in a Task, TaskLink will save the next starting index in a Task. The system status window and the session data log file will contain the serial index used instead of the serial number used.

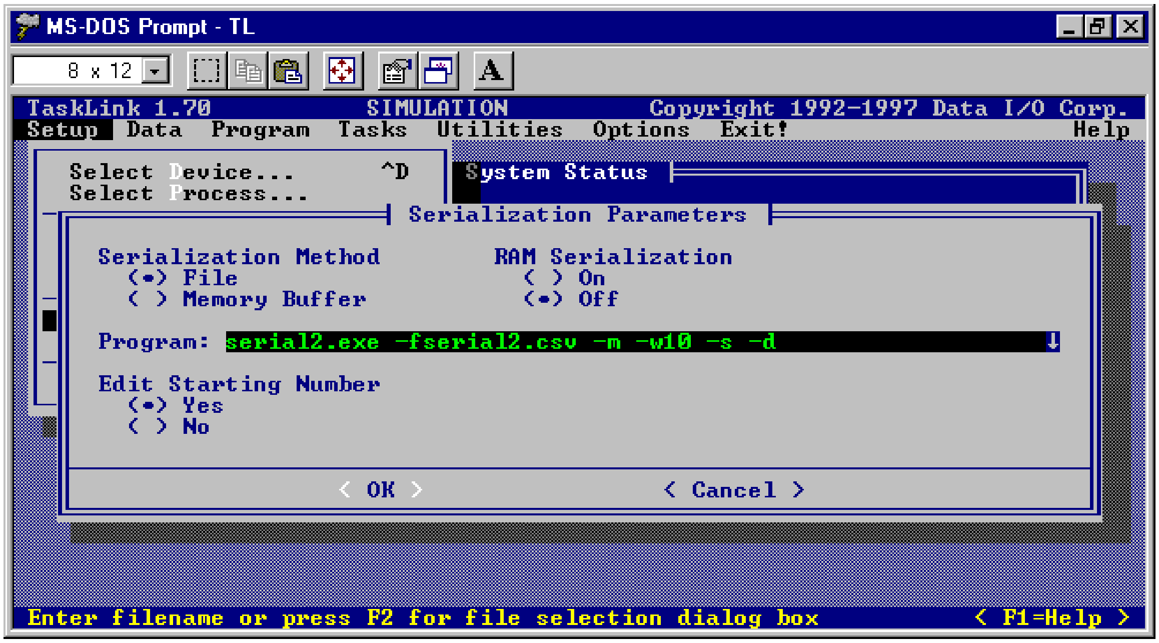
SERIAL2.EXE looks up serial number in the .CSV file by searching for the specified index, instead of generating the serial number directly.

3.2 Enabling serialization.

Enabling serialization in TaskLink is unchanged. There are three possible ways of enabling serialization.

1. RAM serialization is enabled from the Serialization Parameters Dialog under the Setup menu.
2. Graphics file serialization for Thermal and Laser Handler systems is enabled by checking the "Serialized Graphics" checkbox in the Handling/Labeling Parameters Dialog in the Setup menu.
3. Text serialization for all Handler systems enabled by placing two or more percent characters (%%%) in the label text field in the Handling/Labeling Parameters Dialog in the Setup menu. The percent characters are used as place holders, and at runtime TaskLink will replace the percent characters with the real serial data.

3.3 Serialization Command line parameters.



```
TaskLink 1.70              SIMULATION          Copyright 1992-1997 Data I/O Corp.
Setup  Data  Program  Tasks  Utilities  Options  Exit!                       Help

   Select Device...      ^D    System Status
   Select Process...
                       Serialization Parameters

     Serialization Method          RAM Serialization
       (•) File                      ( ) On
       ( ) Memory Buffer             (•) Off

     Program: serial2.exe -fserial2.csv -m -w10 -s -d

     Edit Starting Number
       (•) Yes
       ( ) No

                     < OK >                   < Cancel >

Enter filename or press F2 for file selection dialog box               < F1=Help >
```

The SERIAL2.EXE serialization program is used in the same manner as the SERIALIZ.EXE program. The program and its options are specified in the Serialization Parameters Dialog under the Setup menu. The -fFilename parameter must be used. The others are optional and shown for reference.

3.4 Customer Provided Command Line Parameters

There are two types of command line parameters passed to the serialization utility. Those specified by the user and must be in lower case characters. And those specified by TaskLink for DOS and must be in upper case characters. Since much of the information needed for serialization is now read from the .CSV file, the number of command line parameters specified by the user has been reduced.

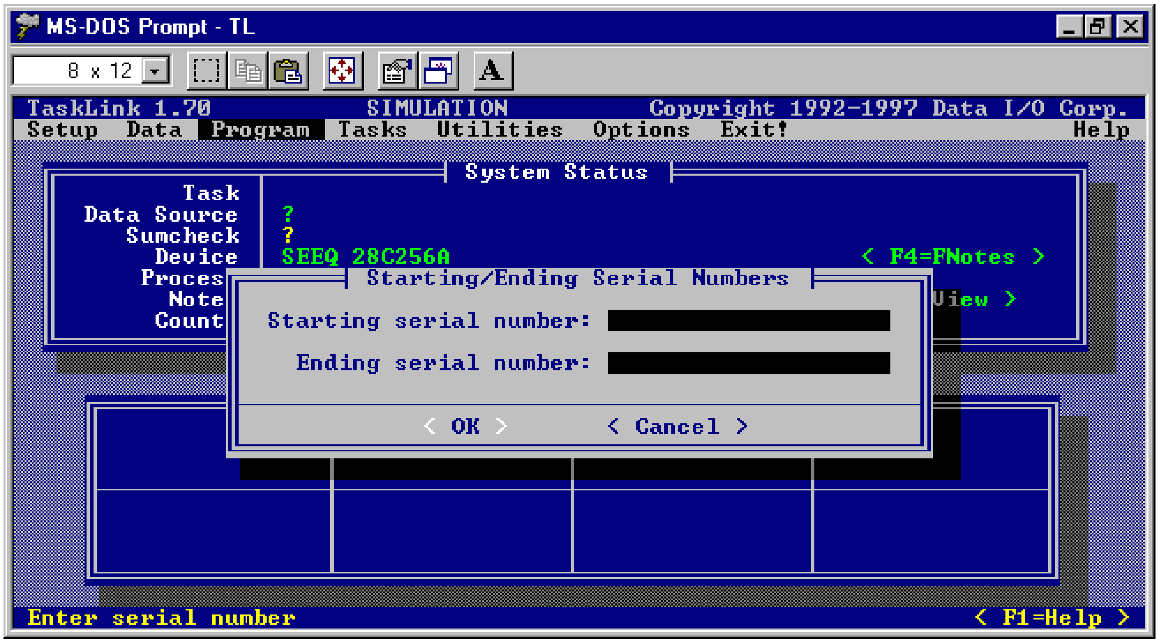
	SERIALIZ.EXE Utility		SERIAL2.EXE Utility	
Same for both:				
	-d	Delete serial.dat file	-d	Delete serial.dat file
	-m	Pure-set gang serialization	-m	Pure-set gang serialization
	-s	Sumcheck calculation flag	-s	Sumcheck calculation flag
	-w	Warning limit value	-w	Warning limit value
Different:				
	-f	**Format of the serial data**	**-f**	**Filename for the .CSV file**
SERIALIZ.EXE only:				
	-a	Address of the serial data in the device		
	-c	Compensation address for compensation value		
	-e	Ethernet mask value		
	-i	Increment value		
	-l	Length of the serial data		
	-o	Byte order of the serial data		
	-t	Label text format		
	-z	Ethernet mask length		

3.5 TaskLink Provided Command Line Parameters

There are two types of command line parameters passed to the serialization utility. Those specified by the user and must be in lower case characters. And those specified by TaskLink for DOS and must be in upper case characters. All of the command line parameters passed by TaskLink for DOS are accepted and used by the SERIAL2.EXE utility. The meaning of the -N and -E fields have changed from the current and ending serial numbers, to the current and ending serial index values. The meaning and usage of the remaining parameters has not been changed.

	SERIALIZ.EXE Utility		SERIAL2.EXE Utility
Different:			
	-N	**Indicates the current serial number**	**Indicates the current serial index**
	-E	**Indicates the ending serial number**	**Indicates the ending serial index**
Same for both:			
	-D	Indicates the device size in bytes	
	-F	Indicates first call to the ESP	
	-L	Indicates last call to the ESP	
	-Q	Indicates the quantity of serial numbers	
	-S	Indicates the base RAM sumcheck before serialization	
	-W	Indicates a warning message was received using T11 record.	

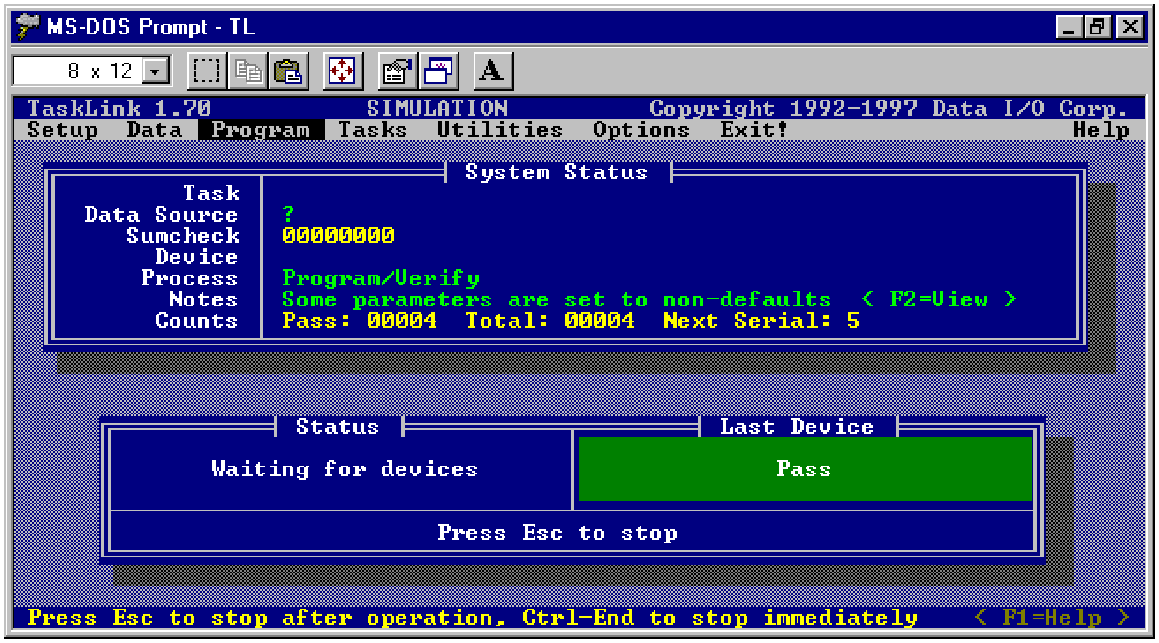
3.6 Entering the Starting and Ending serial index



```
TaskLink 1.70             SIMULATION          Copyright 1992-1997 Data I/O Corp.
Setup  Data  Program  Tasks  Utilities  Options  Exit!                      Help

                    System Status
        Task
 Data Source   ?
    Sumcheck   ?
      Device   SEEQ 28C256A                              < F4=FNotes >
     Proces  Starting/Ending Serial Numbers
       Note                                              View >
      Count   Starting serial number: 

                Ending serial number: 

                      < OK >          < Cancel >

Enter serial number                                              < F1=Help >
```

When processing devices and serialization is enabled, TaskLink prompts the operator to enter the starting and ending serial numbers. When TaskLink prompts for the starting and ending serial number, **you must enter the starting and ending serial indexes instead.**

3.7 TaskLink Displays and Logs



```
TaskLink 1.70             SIMULATION          Copyright 1992-1997 Data I/O Corp.
Setup  Data  Program  Tasks  Utilities  Options  Exit!                      Help

                    System Status
        Task
 Data Source   ?
    Sumcheck   00000000
      Device
     Process   Program/Verify
       Notes   Some parameters are set to non-defaults  < F2=View >
      Counts   Pass: 00004  Total: 00004  Next Serial: 5

            Status                      Last Device
       Waiting for devices                 Pass

                       Press Esc to stop

Press Esc to stop after operation, Ctrl-End to stop immediately  < F1=Help >
```

In the system status display, the next serial field shows the next starting index.

If the session log file feature is enabled, TaskLink will record the starting and ending serial indexes in the log file.

When running a Task, it is the starting index that will get saved in the Task.

3.8 Moving from SERIALIZ.EXE to SERIAL2.EXE

When moving from SERIALIZ.EXE to SERIAL2.EXE the following things should be taken into account when generating serial data files (.CSV):

Addresses: The ADDRESS field in no longer specified on the command line, it is now read from the file.

Increment value: It is the users responsibility to increment or decrement each serial number themselves in the .CSV file. TaskLink and SERIAL2.EXE will not increment or decrement any data fields.

Label text format: To have the label text in a different format than the RAM data, it is the users responsibility to create two entries in the .CSV file and specify each separately.

Byte order: It is the users responsibility to enter the characters into the correct byte order in the .CSV file. TaskLink and SERIAL2.EXE will place the characters in RAM exactly as they are read from the file.

Compensation: To have serial number compensation, it is the users responsibility to create two entries in the .CSV file. Treat the compensation data just like a second serial number. It will be the users responsibility to calculate the compensation value placed in the .CSV file.

Ethernet serial numbers: The serial data fields are long enough to hold a complete Ethernet addresses. No need to do anything special.

3.9 SERIAL2.EXE Error Messages

TaskLink may display the following error messages from the External serialization program.

No serial data filename specified
This error occurs if the -f command line parameter is not specified or is specified without a valid filename. Change the command line in the TaskLink serialization setup dialog and enter a valid .CSV filename with the -f flag. EX: -fserial2.csv. There is no space between the -f and the filename.

Serial data file not found
This error occurs if the filename specified by the -f flag on the serialization program command line in TaskLink is not found. Make sure the .CSV file is in the current directory where the SERIAL2.EXE program resides.

No starting serial number index specified
This error occurs if no starting serial index is specified in the serialization programs command line by TaskLink. Usually this will occur if no value is entered in the dialog when TaskLink prompts for the starting serial number, or the prompt for the starting serial number has been disabled in the Serialization Parameters Dialog under the setup menu.

Serial data number index not found
This error occurs when the serialization program has searched the entire .CSV file and the starting index was not found. Most likely possibilities are the wrong starting index was enter or the wrong .CSV file is being used.

Serial numbers requested will exceed ending serial index.
This error occurs when serializing with a gang programmer. When the next starting serial index plus the number of serial numbers needed exceeds the ending serial index specified by the operator, this message is displayed. Example: The serial index range specified at run-time was 1 to 100. The current starting index for the next operation is index 90. Using a PSX1000 TaskLink will request 20 serial numbers. The range is from 90 to 109 which exceeds the ending number of 100.

End of serial data file has been reached
This error occurs when serializing with a gang programmer. When the next starting serial index plus the number of serial numbers needed exceeds the number of serial numbers in the .CSV file, this message is displayed. Example: The .CSV file contains 100 serial numbers, 1 to 100. The current starting index for the next operation is index 90. Using a PSX1000 TaskLink will request 20 serial numbers and there are only 11 left.

Multiple label fields per device specified, line xxx
This error occurs if there are more than one serial data field per device is specified for label serialization.

Multiple serial numbers per device must use pure set mode.
When using multiple serial numbers per device on a gang programmer such as a PSX programmer, Pure-Set mode must be used (-m command line flag). In order for Gang-Set mode (default, no -m command line flag) to operate significant changes need to be made to TaskLink DOS. Gang-Set serialization mode can still be used when programming with one serial number per device.

Too many serial numbers per device, line xxx
This error occurs if there are more than 8 serial data fields per device specified for RAM serialization.

File error read on line, xxx
This error occurs if there is an error reading from the file. Usually this means the file is corrupted or there is a problem with the disk drive in the location where the file is stored.

Invalid line format at line, xxx
This error occurs when a line in the .CSV file does not meet one or more of the validation requirements. Review the section ".CSV file validation". Review the rules that apply to each line of the file. These rules are validated as each line is read.

Invalid address field line, xxx
This error occurs when an address field in a line in the .CSV file is not valid. Review the section ".CSV file validation". Review the rules that apply to each line of the file. These rules are validated as each line is read.

Data length does not match length field, line xxx
This error occurs when the length of the data in the serial data field is not the same length as that specified in the length field.

Invalid index in line xxx
This error occurs when SERIAL2.EXE encounters an index in the .CSV file that is not in consecutive order. The index field in the .CSV file must be in consecutive order.
Example 1, 2, 3, 4, 5…. is valid. 1, 2, 3, 5, 6, …. is invalid.

Error serial data is not uniform
This error occurs when the data in the .CSV file does not meet one or more of the validation requirements for gang programmers. Review the section ".CSV file validation". This error only applies to the section for gang programmers.

Invalid Parameter(s)

The following errors occur when a command line parameter is specified and does not contain a legal value, or has no value associated with it. Most of these errors are specified by TaskLink and should not be a problem under normal operation. See the section "command line parameters" for more information about each option.

Message	Parameter
TaskLink Specified:	
Invalid Device Size	-D
Invalid ending serial number index	-E
Invalid starting serial number index	-N
Invalid number of serial numbers	-Q
Invalid starting sumcheck	-S
User Specified:	
No serial data filename specified	-f
Invalid warning limit	-w

3.10 SERIAL.DAT Format

This is a brief summary of the "T" records used in the serial.dat file. A complete description can be found in the TaskLink on-line help topic titled "Writing a Serialization Program". They are included to show the design change for the T01 and T02 records. It changes from containing the current and next serial number, to containing the current and next serial index. All other "T" records are used in the same method in both programs.

SERIALIZ.EXE Utility		SERIAL2.EXE Utility	
Different (Mandatory):			
T01:	**Current Serial Number(s)**	**T01:**	**Current Serial Index(s)**
T02:	**Next Starting Serial Number**	**T02:**	**Next Starting Serial Index**
Same for both (Mandatory):			
T03:	The Data Translation Format	T03:	The Data Translation Format
T04:	Beginning of data	T04:	Beginning of data
Same for both (Optional):			
T00:	Indicates "Pure-Set" gang serialization mode (gang programmers)		
T05:	Fatal error message string		
T06:	Serial limit reached flag		
T07:	Hexadecimal address of the serial number in the device (gang programmers)		
T08:	Decimal field length of the serial data (gang programmers)		
T09:	RAM sumcheck field		
T10:	Label text field(s)		
T11:	Warning message string		