invensys
Foxboro

LIN
Recipe File Editor

User Guide

HA029968

October 2011 (Issue 2)

The manufacturer pursues a policy of continuous development and product improvement. The specifications in this document may therefore be changed without notice. The information in this document is given in good faith, but is intended for guidance only. The manufacturer will not accept responsibility for any losses arising from errors in this document.

Recipe File Editor Help Manual

Table of Contents

1 RECIPE FILE EDITOR HELP

The purpose of this document is to allow users quick access to information concerning a particular point of interest in the Recipe File Editor. The Recipe File Editor provides access to the Recipe File of an instrument installed across LIN based networks. This Windows™ based editor is used to create and configure the Recipe (.uyr) File via a simple user interface clearly displaying the relevant configuration details.

It also supports:

validation of the Block.Field references in the database.

protection of Intellectual Property.

1.1 OVERVIEW

The editor is divided into tabs that correspond to the different sections of the Recipe file.

A Recipe file is an interface between the LIN Database and the operation of the Recipe in each of the plant lines.

The editor is launched by selecting either

 Start > All Programs > ... > Recipe File Editor, where ... represents the installation path, or

Tools > Recipe File Editor from LINtools Menu

When launching the editor from LINtools it searches the instrument folder and applies the default database (.dbf) configuration. If launching it from the **Start** menu the user is asked to specify an existing database or to create a Recipe file that will not be associated with a database until the Recipe file is saved.

The editor shows the configured output via the **File Output** tab.

Note
Recipe files can be associated with a different database using the Database > Select Database command.

1.2 WHAT IS A RECIPE ?

A Recipe file contains a Recipe Set and Recipe details including a list of ingredients, instructions and/or values derived from the assigned LIN Database. It is controlled using a combination of the Recipe file, the RCP_SET block (used to group the RCP_LINE blocks into a specific recipe configuration) and one or more RCP_LINE blocks (used to control the downloading of a specific recipe from the Recipe file associated with the defined RCP_SET block) in the assigned database.

A Recipe Set contains up to 16 recipes with each Recipe including the same set of variables but with different assigned values (e.g. for different colour mixes). When configured, the selected Recipe is ‘downloaded’ to the LIN Database controlling one of up to 8 plant lines.

A Recipe is a set of LIN function block field names (variables) with a particular value assigned to each, e.g. if the recipe is for one or more identical paint-mixing lines, it would include setpoint values representing the amounts of each pigment in the mixture needed for a particular colour. In use, the recipe is ‘downloaded’ to the LIN Database controlling a particular plant line, i.e. each recipe value is copied to the corresponding Block.Field to achieve the required control.

Example
A bakery may have a basic dough recipe that lists all the ingredients required to make plain cookies and all the optional ingredients such as nuts, fruit, and chocolate chips, that can be added to the basic recipe to make various kinds of cookies, with each type being produced on a specific plant line.

1.3 CONFIGURING A RECIPE FILE

The following stages may overlap and their order can be varied especially when an existing strategy is being edited.

Note
A minimum of 1 recipe for each Recipe file is required.

For a new Recipe File, configure the File Header tab parameters.
The File Header tab parameters configure a 3-line, fixed format, description that provides most of the information about the Recipe file. It is specifically used to determine to relationship between the Recipe operation and the database.

Then, configure the Title Line tab parameters.
The Title Line tab parameters configure the number and identity of each Recipe Line (plant line) and Recipe associated with this Recipe file.

Configure the Variable Line tab parameters.
The Variable Line tab parameters are used to edit the Variables required by the Recipe. Each Variable Line allows one or more Recipe values to be associated with a particular Recipe Line element.

Inspect the File Output tab.
The File Output tab shows the Recipe File as displayed when opened using a text editor, e.g. Notepad.exe.

Tip!
Enable Intellectual Property Protection using LINtools to prevent malicious mis-use of your configured Recipe files.

The configured Recipe file is now complete and ready to download to the instrument.

1.4 WHAT IS INTELLECTUAL PROPERTY PROTECTION ?

This is the principle of preventing unauthorised access to or malicious use of configured strategy files. Protected files can only be opened by users who know the password and can only be used on authorised instruments.

Note
LINtools is used to enable you to protect your Intellectual Property (contents of the strategy files) by password protecting your files.

A protected file is encrypted using a key. This is a public identifier (text viewed by anyone) and a private password. A key must be specified to protect a file, but the same key can be used to protect multiple files. When opening a protected file a dialogue is displayed showing the key identifier, and asking for the password. The protected file will open if the correct password is used.

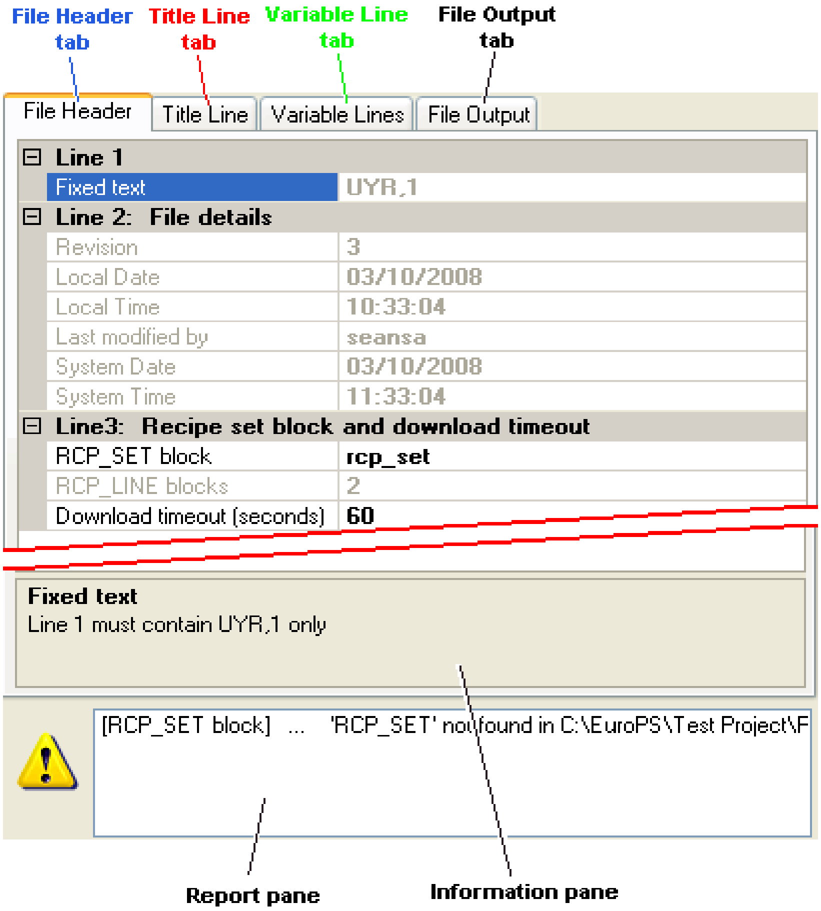
Note
Protection and storage of passwords is the responsibility of the user. Lost passwords cannot be recovered, and protected files cannot be opened without the appropriate password.

To use a protected file the instrument must be authorised with the correct key, as specified during instrument configuration. The same key can be used by many instruments. You can authorise an instrument with up to seven keys, which allows one instrument to use files protected with different passwords. A protected file cannot be used by an instrument that is not authorised with the correct key. This prevents users copying and using the strategy on different instruments without permission.

Note
The keys are managed using LINtools.

2 RECIPE FILE EDITOR WINDOW

The Recipe File Editor window, below, provides access to the parameters used to configure a Recipe file



⊟ Line 1	
Fixed text	UYR,1
⊟ Line 2: File details	
Revision	3
Local Date	03/10/2008
Local Time	10:33:04
Last modified by	seansa
System Date	03/10/2008
System Time	11:33:04
⊟ Line3: Recipe set block and download timeout	
RCP_SET block	rcp_set
RCP_LINE blocks	2
Download timeout (seconds)	60

Fixed text
Line 1 must contain UYR,1 only

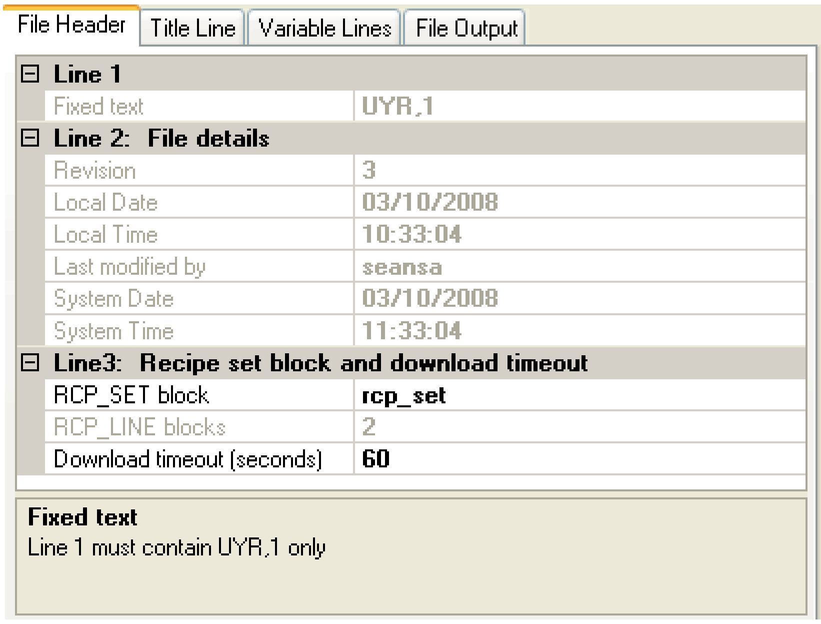
[RCP_SET block] ... 'RCP_SET' not found in C:\EuroPS\Test Project\F

The Editor also provides an Information pane which appears with the File Header tab and the Title Line tab, and a Report pane which appears only if configuration errors occur.

2.1 THE FILE HEADER TAB

The File Header tab is a 3-line, fixed format, description that provides general information contained in the Recipe file, see configure the File Header tab parameters.

The information configured on this tab is validated each time the tab is selected. Any configuration errors appear in the Report pane.

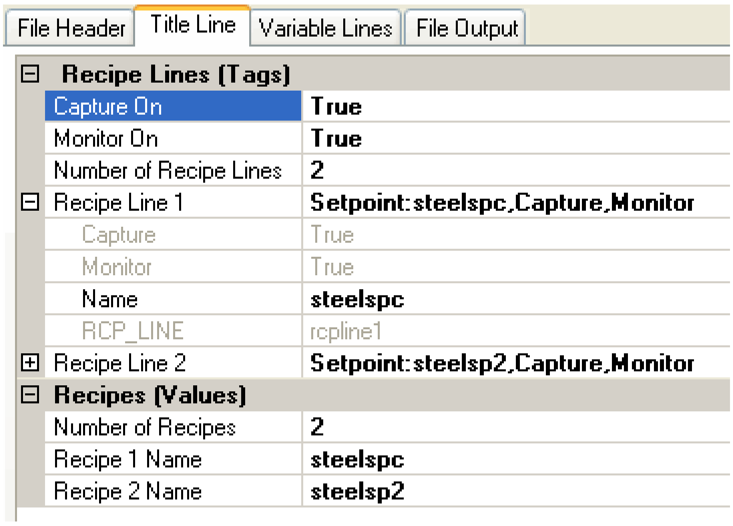


⊟ **Line 1**	
Fixed text	UYR,1
⊟ **Line 2: File details**	
Revision	3
Local Date	03/10/2008
Local Time	10:33:04
Last modified by	seansa
System Date	03/10/2008
System Time	11:33:04
⊟ **Line3: Recipe set block and download timeout**	
RCP_SET block	rcp_set
RCP_LINE blocks	2
Download timeout (seconds)	60

Fixed text
Line 1 must contain UYR,1 only

2.2 THE TITLE LINE TAB

The Title Line tab lists each of the Recipe Lines and Recipes that are configured in this file, see configure the Title Line tab parameters. Each recipe line is controlled by a RCP_LINE block in the assigned database.



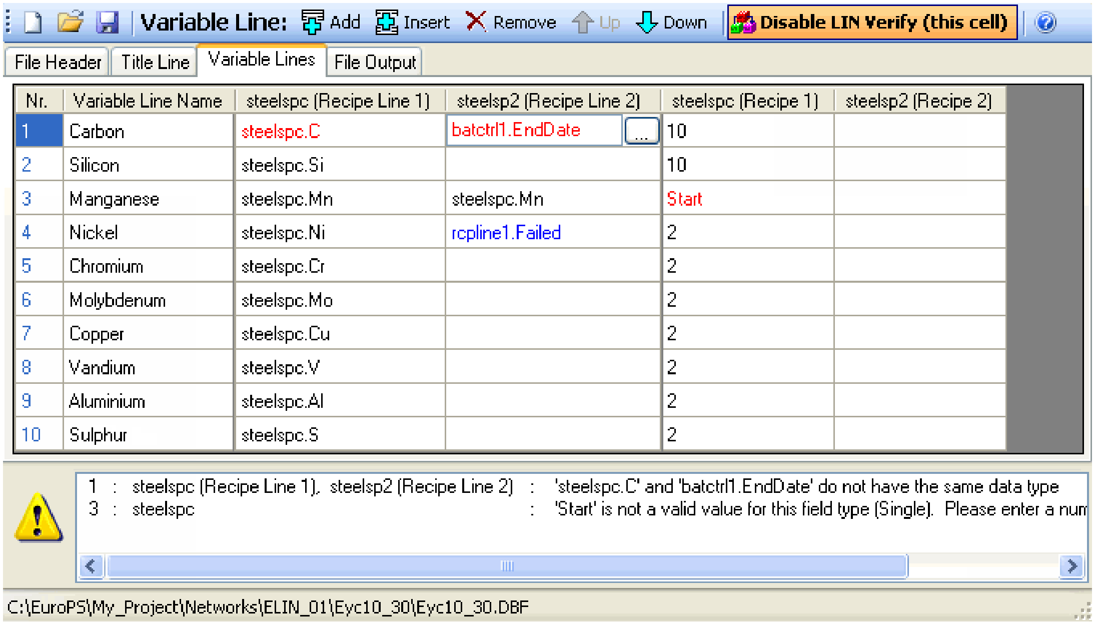
⊟ **Recipe Lines (Tags)**	
Capture On	True
Monitor On	True
Number of Recipe Lines	2
⊟ Recipe Line 1	Setpoint:steelspc,Capture,Monitor
Capture	True
Monitor	True
Name	steelspc
RCP_LINE	rcpline1
⊞ Recipe Line 2	Setpoint:steelsp2,Capture,Monitor
⊟ **Recipes (Values)**	
Number of Recipes	2
Recipe 1 Name	steelspc
Recipe 2 Name	steelsp2

2.3 THE VARIABLE LINES TAB

The Variable Line tab lists each of the Variables (Block.Field references in the assigned database) associated with the Recipe Lines and Recipes that are configured in this file, see configure the Variable Line tab parameters. This tab is used to configure Block.Field references and associated Recipe values for a maximum of 1000 rows.

The information configured on this tab is verified when the file is opened and when a Block.Field reference is configured. Verification checks that the configured Block.Field references are present in the assigned database and that the same data type is configured across all Block.Field references for each **Variable Line** row, i.e. row 1 contains an invalid data type configuration because steelspc.C is a Single-Precision floating point data type but batctrl1.EndDate is a Date data type. Successfully verified Block.Field references are shown in black and unsuccessfully verified Block.Field references are listed in the Report pane and shown in red. Parameters shown in blue indicate that verification of the Block.Field reference is disabled, i.e. verification of this Block.Field reference will not be attempted.

Tip!
When configuring each Variable use the filtering icons and drop-down menu on the picklist to show the required data type only. This will restrict the number of invalid data type errors that occur.



Nr.	Variable Line Name	steelspc (Recipe Line 1)	steelsp2 (Recipe Line 2)	steelspc (Recipe 1)	steelsp2 (Recipe 2)
1	Carbon	steelspc.C	batctrl1.EndDate	10	
2	Silicon	steelspc.Si		10	
3	Manganese	steelspc.Mn	steelspc.Mn	Start	
4	Nickel	steelspc.Ni	rcpline1.Failed	2	
5	Chromium	steelspc.Cr		2	
6	Molybdenum	steelspc.Mo		2	
7	Copper	steelspc.Cu		2	
8	Vandium	steelspc.V		2	
9	Aluminium	steelspc.Al		2	
10	Sulphur	steelspc.S		2	

```
1  :  steelspc (Recipe Line 1),  steelsp2 (Recipe Line 2)   :   'steelspc.C' and 'batctrl1.EndDate' do not have the same data type
3  :  steelspc                                               :   'Start' is not a valid value for this field type (Single).  Please enter a num
```

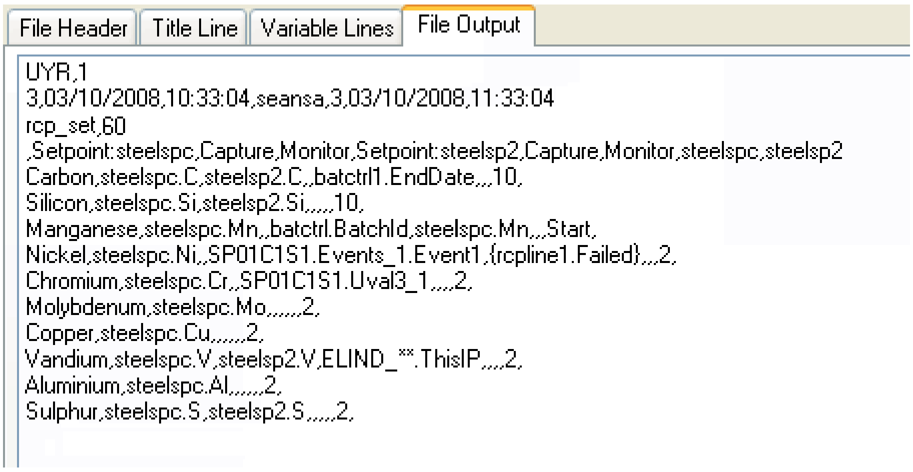
C:\EuroPS\My_Project\Networks\ELIN_01\Eyc10_30\Eyc10_30.DBF

2.4 THE FILE OUTPUT TAB

The File Output tab shows the Recipe File as displayed when opened using a text editor, e.g. ‘Notepad’.

> **Tip!**
> This editor is the recommended tool for configuring a Recipe file. Advanced users can still use a text editor, e.g. Notepad.exe, although additional constraints are applied.

This is the expected output from the configuration applied in the File Header tab, Title Line tab and Variable Lines tab.



```
UYR,1
3,03/10/2008,10:33:04,seansa,3,03/10/2008,11:33:04
rcp_set,60
,Setpoint:steelspc,Capture,Monitor,Setpoint:steelsp2,Capture,Monitor,steelspc,steelsp2
Carbon,steelspc.C,steelsp2.C,,batctrl1.EndDate,,,10,
Silicon,steelspc.Si,steelsp2.Si,,,,,10,
Manganese,steelspc.Mn,,batctrl.BatchId,steelspc.Mn,,,Start,
Nickel,steelspc.Ni,,SP01C1S1.Events_1.Event1,{rcpline1.Failed},,,2,
Chromium,steelspc.Cr,,SP01C1S1.Uval3_1,,,,2,
Molybdenum,steelspc.Mo,,,,,,2,
Copper,steelspc.Cu,,,,,,2,
Vandium,steelspc.V,steelsp2.V,ELIND_**.ThisIP,,,,2,
Aluminium,steelspc.Al,,,,,,2,
Sulphur,steelspc.S,steelsp2.S,,,,,2,
```

2.4.1 Constraints when using a text editor

If using a text editor such as 'Notepad' to configure a Recipe file, the following must be noted:

No line may exceed 512 characters, including line feed/carriage return instructions

Spaces are also counted as characters

String titles do not have to be unique, but extra characters are omitted when the file is saved if the string length is exceeded.

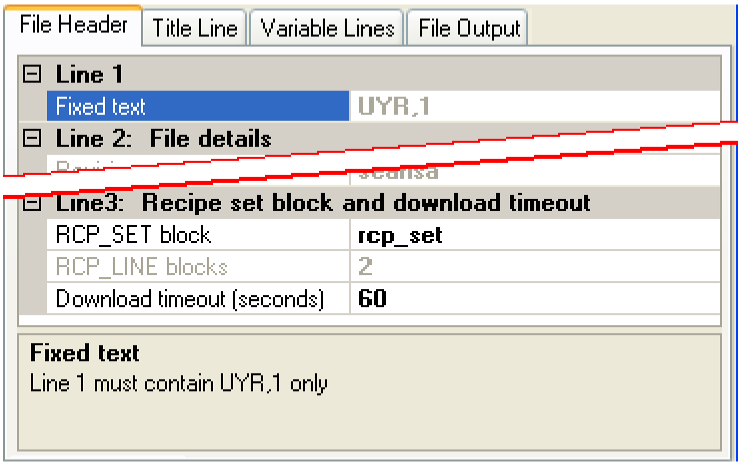
Trailing commas are treated as illegal characters unless otherwise stated

Commas (,), double quotes ("), single quotes ('), and equals (=) characters are illegal characters in fields

Non-printing characters (excluding <CR> and <LF>) are not permitted in the file

2.5 THE INFORMATION PANE

This pane displays additional information about the parameter selected in the File Header tab and the Title Line tab.



⊟ **Line 1**

Fixed text	UYR,1

⊟ **Line 2: File details**

	seansa

⊟ **Line3: Recipe set block and download timeout**

RCP_SET block	**rcp_set**
RCP_LINE blocks	2
Download timeout (seconds)	**60**

Fixed text
Line 1 must contain UYR,1 only

2.6 THE REPORT PANE

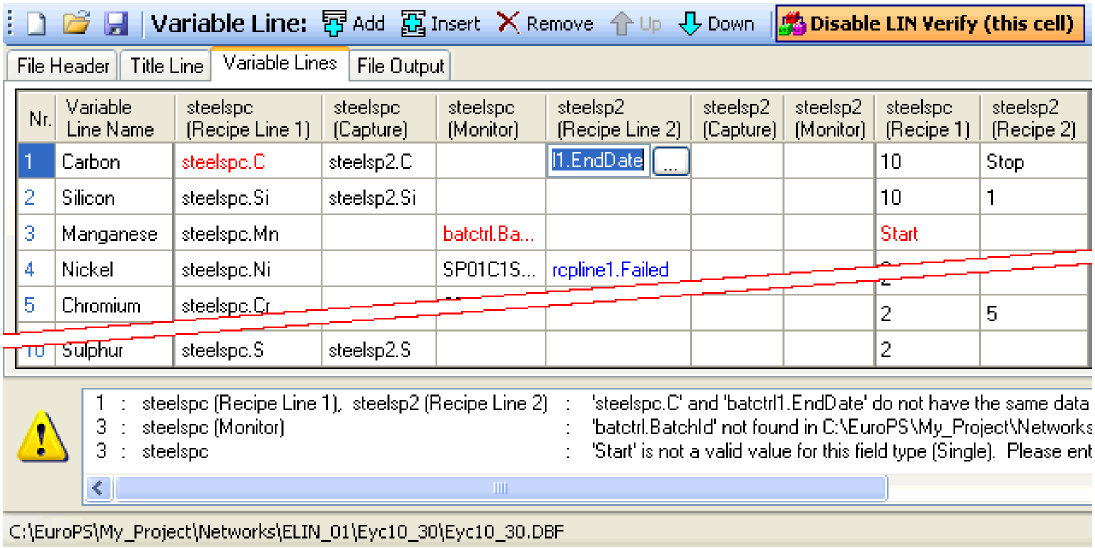
The Report pane displays information about any configuration errors, and appears only when one or more errors have occurred. The pane shows configuration errors with both the File Header tab and Variable Line tab.

Only the **RCP_SET block** parameter can be incorrectly configured on the File Header tab. This can only be caused after assigning a database to this file when the configured RCP_SET block does not exist in the database.

Errors configured in Variable Line tab are displayed in Variable Number order, i.e. 1, 3 and 3 as shown below, and includes an explanation of why the error has occurred.

Note

The number in the Report Pane corresponds to the Variable Nr row number containing an error (red). Configured parameters shown in blue indicate the Block.Field reference is not verified, i.e. the Block.Field reference may or may not exist in the assigned database.



Nr.	Variable Line Name	steelspc (Recipe Line 1)	steelspc (Capture)	steelspc (Monitor)	steelsp2 (Recipe Line 2)	steelsp2 (Capture)	steelsp2 (Monitor)	steelspc (Recipe 1)	steelsp2 (Recipe 2)
1	Carbon	steelspc.C	steelsp2.C		l1.EndDate ...			10	Stop
2	Silicon	steelspc.Si	steelsp2.Si					10	1
3	Manganese	steelspc.Mn		batctrl.Ba...				Start	
4	Nickel	steelspc.Ni		SP01C1S...	rcpline1.Failed			2	
5	Chromium	steelspc.Cr						2	5
10	Sulphur	steelspc.S	steelsp2.S					2	

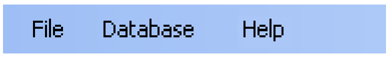
```
1  :  steelspc (Recipe Line 1),  steelsp2 (Recipe Line 2)  :  'steelspc.C' and 'batctrl1.EndDate' do not have the same data
3  :  steelspc (Monitor)                                    :  'batctrl.BatchId' not found in C:\EuroPS\My_Project\Networks
3  :  steelspc                                              :  'Start' is not a valid value for this field type (Single).  Please ent
```

C:\EuroPS\My_Project\Networks\ELIN_01\Eyc10_30\Eyc10_30.DBF

3 MENUS

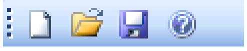
Menu bar

The Menu Bar is a special toolbar that contains pulldown command lists.



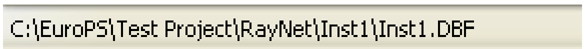
Toolbar

The Toolbar contains buttons that enable quick access to commands used to control this application.



Status Bar

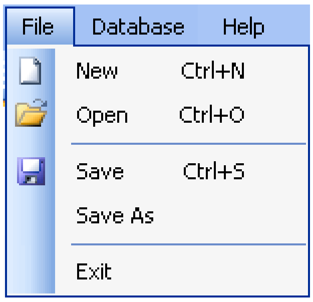
The Status Bar displays the file path to the database associated with this Recipe file.



C:\EuroPS\Test Project\RayNet\Inst1\Inst1.DBF

3.1 FILE MENU

The File menu enables the control of an individual Recipe file.



3.1.1 New

<Ctrl> + <N>

Click this to create a new Recipe file (.uyr). If the currently open Recipe file is already assigned to a database a dialogue appears, asking if the user wishes to continue using this database (.dbf file).

'Yes' continues with the existing database.

'No' creates a Recipe file that can be assigned to any other database when the Browse dialog is displayed. The Browse dialog is used to define an existing database or to create a Recipe file that is not yet associated with a database

Tip!
Use Database > Select Database command to change the assigned database.

3.1.2 Open

<Ctrl> + <O>

Click this to display the **Open** dialog. This permits the user to browse to an existing Recipe file.

3.1.3 Save

<Ctrl> + <S>

Click this to display the **Save** dialogue. Saves the active file with its current file name, location, and file format.

3.1.4 Save As

Click this to display the **Save As** dialogue. This permits the user to save the current file with a different filename and/or location.

If Intellectual Property Protection is enabled in LINtools the **Save as password encrypted file** checkbox is displayed in the **Save as** dialogue.

Set the **Save as password encrypted file** checkbox ☑ to save this as an encrypted file. This prevents the file from being opened by unauthorised users (i.e. users who do not know the password) or used by unauthorised instruments.

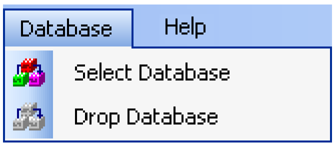
If the **Save as password encrypted file** checkbox is enabled, the Set encryption key dialogue is displayed when the Save button is pressed. This dialogue is used to set the password required to access this file. A password is used in conjunction with an identifier for the key. This is publicly available text used to identify files locked with this password.

3.1.5 Exit command

Click this to close this editor. If any unsaved changes exist, the user can cancel this command, save or discard the changes, and then repeat this command.

3.2 DATABASE MENU

The Database menu provides commands used to control the database assigned to the current Recipe file.



3.2.1 Select Database command

Click this to display the **Browse** dialogue. This permits the user to browse and assign an existing database file to the Recipe file currently open in this editor.

3.2.2 Drop Database command

Click this to remove the link to an already assigned database.

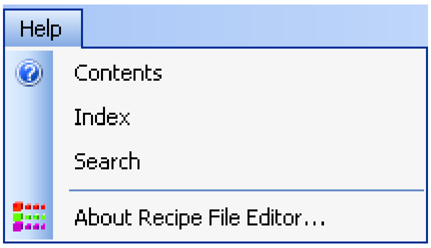
> **Note**
> Parameters that are used to define a specific block of a database become free format text fields. This allows the required information to be entered using block or block.field syntax.

3.3 HELP MENU

The Help menu enables the user to display details about this editor using the following commands. It is divided into 2 sections.

Documentation (including Contents, Index, and Search for Help on)

Editor (About)

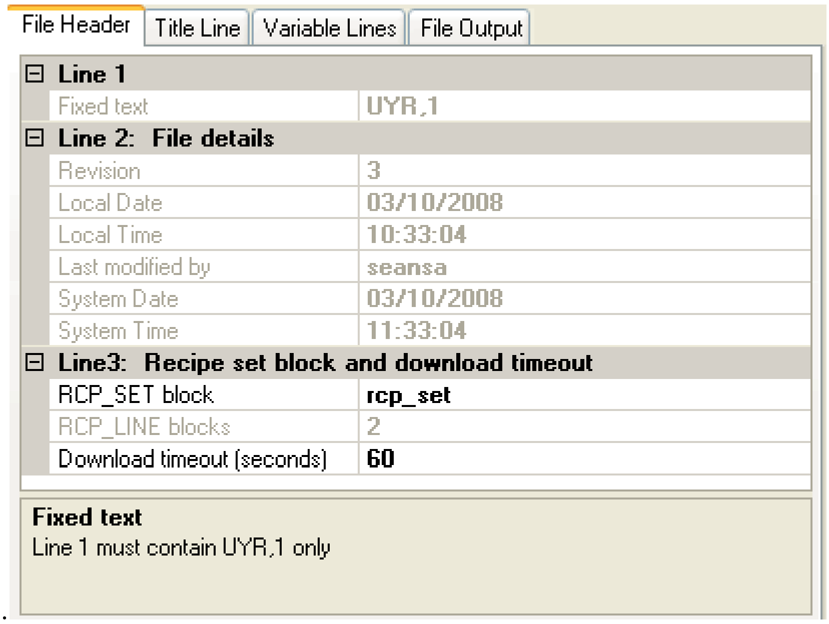


Contents Opens the 'Help' file, with 'Contents' as the priority.
Index Opens the 'Help' file, with 'Index' as the priority.
Search Opens the 'Help' file, with 'Search' as the priority.
About Recipe File Editor...
This displays a dialogue, showing the software version number.

4 HOW TO...

4.1 CONFIGURE THE FILE HEADER TAB PARAMETERS

These parameters are used to show the general information about this file and configure the interface between this file and specific blocks in a database



Line 1	
Fixed text	UYR,1
Line 2: File details	
Revision	3
Local Date	03/10/2008
Local Time	10:33:04
Last modified by	seansa
System Date	03/10/2008
System Time	11:33:04
Line3: Recipe set block and download timeout	
RCP_SET block	rcp_set
RCP_LINE blocks	2
Download timeout (seconds)	60

Fixed text
Line 1 must contain UYR,1 only

.

4.1.1 Line 1

This is a read-only parameter, and is shown on the File Output tab in the form as 'UYR,1'

4.1.2 Line 2

These are read-only parameters and display current revision details for this Recipe file.

This information is shown on the File Output tab as follows:

<Revision>,<Local Date>,<Local Time>,<Last Modified By>,<System Date>,<System Time>

Example
2,03/10/08,23:33:40,Fred Bloggs,04/10/08,00:33:40

where:

'2' is the revision level of this Recipe file. This value is incremented each time the Recipe file is saved. The value is reset to 0 (zero) if the 'Save as' command is used.

'03/10/08' is the local day/month/year that the last revision was saved.

'23:33:40' is the local hours:minutes:seconds that the last revision was saved.

'Fred Bloggs' is the name of the user to have last modified the file.

'04/10/08' is the system day/month/year that the last revision was saved.

'00:33:40' is the system hours:minutes:seconds that the last revision was saved.

HOW TO CONFIGURE THE FILE HEADER TAB PARAMETERS (Cont.)

4.1.3 Line 3

Line 3 parameters configure the link between the RCP_SET block and the corresponding Recipe file (.uyr). The Recipe file is used to apply the required values to the Batch.

This information is shown on the File Output tab in the format:

<RCP_SET>,<Timeout>

Example

RCP_SET1,30

Configure the ‘RCP_SET Block name’ parameter, i.e. <RCP_SET>. This defines the name of the block used to control the Recipe file.

Select the ‘RCP_SET block name’ to display ....

Press ... to display the picklist showing all the RCP_SET blocks available.

Select the required RCP_SET block from the picklist.

Note

Any RCP_SET block can be used if this parameter remains in the default configuration.

Tip!

Consult the LIN Block Reference Manual for block details.

2.Configure the’ Download timeout (seconds)’ parameter, i.e. <Timeout>. This defines a timeout period of up to a maximum of 60 seconds for downloading the Recipe file. If the file is not downloaded before this time period has expired the download is deemed to have failed.

iSelect the ‘Download timeout (seconds’) to enable a text cursor.

IiType the maximum time (seconds) allowed before it is assumed that the download has been unsuccessful. If this time is exceeded the Recipe shows a failed state.

4.2 CONFIGURE THE TITLE LINE TAB PARAMETERS

These parameters are used to define the Recipe Set using this file.

Each recipe line (Line N parameter) is associated with a single RCP_SET block in the database. Each RCP_SET block is associated with a maximum of 8 RCP_LINE blocks that are used control the recipes configured in this Recipe file.

This information is shown on the File Output tab in the following format:

,<Setpoint:Recipe Line Name>,<Capture>,<Monitor>,<Recipe Name>,-,-,-,<Recipe Name>

Example

,Setpoint:Cookies,Capture,PlainCookies,Fruit&NutCookies,ChocChipCookies

4.2.1 To configure the Title Line parameters

Note

When enabled the specified Block.Field references can be displayed on the Recipe Monitor page of the Visual Supervisor.

1. Configure the ‘Capture On’ parameter, i.e. <Capture>.
   This enables the optional Capture feature for each of the configured Recipe Lines. The Capture feature allows specified Block.Field reference values to be written to the Block.Field reference defined in the corresponding Recipe Line, i.e. Recipe Line values are overwritten with the value from the Block.Field reference specified in the Capture column.
2. Select Capture On and press ▾ to display the True/False options. Select as required.
   When True an additional Block.Field reference from the assigned database can be configured in the Variable Line tab parameters.
3. Configure the Monitor On parameter, i.e. <Monitor>.
   This enables the optional Monitor feature for each of the configured Recipe Lines. The Monitor feature allows values from the specified Block.Field reference to be monitored.
4. Select ‘Monitor On’ and press ▾ to display the True/False options. Select as required.
5. When True an additional Block.Field reference from the assigned database can be configured in the Variable Line tab parameters.
6. Configure the ‘Number of Recipe Lines’ parameter.
   This defines the number of identical plant lines (between 1 and 8 recipe lines) that are using this Recipe file.
7. Select the ‘Number of Recipe Lines’ to enable a text cursor.
8. Type the number of recipe lines using this Recipe file.
   The corresponding number of ‘Recipe Line N’ parameters are generated. Each ‘Recipe Line N’ parameter represents one recipe line (plant line, one RCP_LINE block per plant line) associated with a RCP_SET block (used to group the RCP_LINE blocks into a specific recipe configuration) in the database.

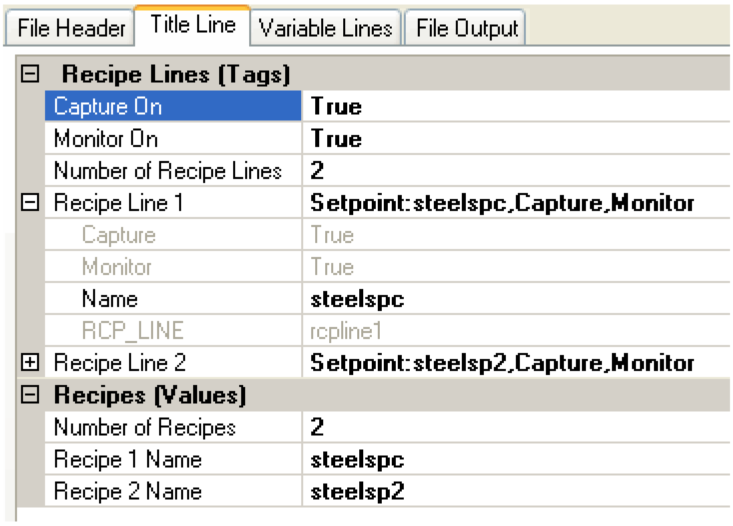
Note

A dialogue appears if the value defined in the Number of Recipe Lines parameter differs from the number of RCP_LINE blocks associated with the defined RCP_SET block.

4.2.1 HOW TO CONFIGURE THE TITLE LINE TAB PARAMETERS (Cont.)

1. Configure the ‘Recipe Line 1’ parameters, i.e. <Setpoint:Recipe Line Name,Capture,Monitor>. These parameters specify the properties of this recipe line.
2. Open the ‘Recipe Line 1’ parameter to display the additional parameters.
3. Observe that Capture, Monitor and RCP_LINE parameters are already defined. These parameters are automatically configured via the ‘Capture On’ parameter, ‘Monitor On’ parameter and the RCP_SET block already configured in the ‘RCP_SET Block name’ parameter on the File Header tab.
4. Select ‘Name’ to enable a text cursor and type the name of this recipe line, i.e. <Setpoint:Recipe Line Name>.
5. This information is used to identify this recipe line as shown on the Variable Line tab.
6. If necessary, open the remaining ‘Recipe Line n’ parameters as required to display the additional parameters and repeat the previous instructions.

Tip!
A number will be automatically applied if a Recipe Name is not defined.



⊟ **Recipe Lines (Tags)**	
Capture On	**True**
Monitor On	**True**
Number of Recipe Lines	**2**
⊟ Recipe Line 1	**Setpoint:steelspc,Capture,Monitor**
Capture	True
Monitor	True
Name	**steelspc**
RCP_LINE	rcpline1
⊞ Recipe Line 2	**Setpoint:steelsp2,Capture,Monitor**
⊟ **Recipes (Values)**	
Number of Recipes	**2**
Recipe 1 Name	**steelspc**
Recipe 2 Name	**steelsp2**

Configure the ‘Recipes (Values)’ parameter, i.e. <Setpoint:Recipe Line Name>.
These parameters define the details of a maximum of 16 recipes in this Recipe file.

Select the ‘Number of Recipes’ to enable a text cursor and type the number of recipes that will be configured in this Recipe file. The corresponding number of ‘Recipe N Name’ parameters are generated. Each parameter represents one recipe that is controlled by a single RCP_LINE block (used to control the downloading of a specific recipe from the Recipe file associated with the defined RCP_SET block).

Select ‘Recipe 1 Name’ to enable a text cursor and type the name of this recipe. This information is used to identify the recipe as shown on the Variable Line tab.

If necessary, name the remaining ’Recipe Name’ parameters.

4.3 CONFIGURE THE VARIABLE LINE TAB PARAMETERS

This is used to configure Block.Field references for a maximum of 1000 'Variable Nr' rows. The values from these Block.Field references are used to parameterise the Recipe.

This information is shown in the File Output tab as follows:

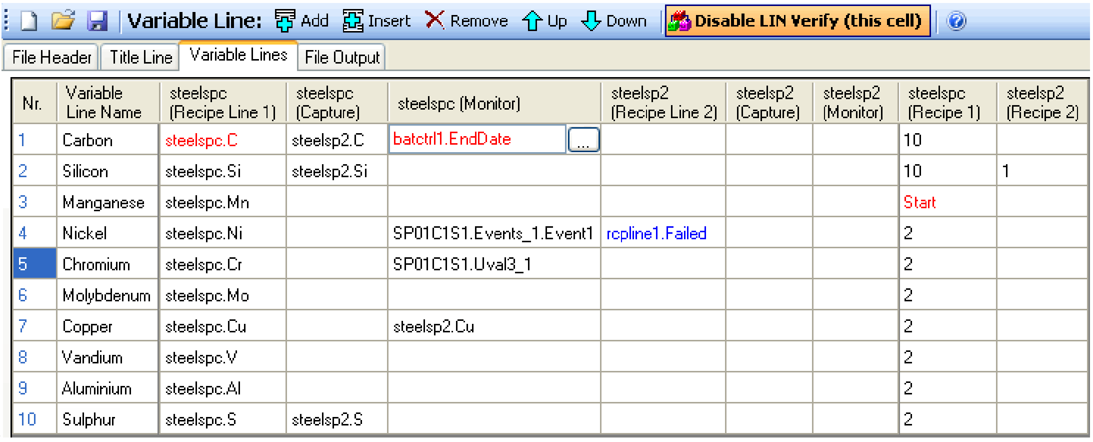
<Variable Line Name>,<Tagname>,<Value Captured from>,<Tagname Monitored>,-,-,-,<Value>,-

Example

Carbon,steelspc.C,steelsp2.C,SP01C1S1.Events_1.Event1,,,,1.203,3.5

4.3.1 To configure the Variable Line parameters

Observe that the parameters configured in the Title Line tab appear as the column titles on this tab.



Nr.	Variable Line Name	steelspc (Recipe Line 1)	steelspc (Capture)	steelspc (Monitor)	steelsp2 (Recipe Line 2)	steelsp2 (Capture)	steelsp2 (Monitor)	steelspc (Recipe 1)	steelsp2 (Recipe 2)
1	Carbon	steelspc.C	steelsp2.C	batctrl1.EndDate ...				10	
2	Silicon	steelspc.Si	steelsp2.Si					10	1
3	Manganese	steelspc.Mn						Start	
4	Nickel	steelspc.Ni		SP01C1S1.Events_1.Event1	rcpline1.Failed			2	
5	Chromium	steelspc.Cr		SP01C1S1.Uval3_1				2	
6	Molybdenum	steelspc.Mo						2	
7	Copper	steelspc.Cu		steelsp2.Cu				2	
8	Vandium	steelspc.V						2	
9	Aluminium	steelspc.Al						2	
10	Sulphur	steelspc.S	steelsp2.S					2	

.

Configure the number of Variables required. This should be equivalent to the number of ingredients in a product.

Press Add to add one recipe Variable. This adds one row to the end of the list of existing Variables (if any).

Press Insert to add one row before the selected Variable.

Press Remove to delete the selected Variable.

Tip!

Use Up and Down to change the order of the configured variables.

Configure the' '**Variable Line Name'** parameter. This is used to identify the specific ingredient or step in the recipe.
Type a title for variable in the corresponding Variable Name column. This value is displayed on the Recipe Monitor page of the Visual Supervisor.

Configure the '**Recipe Line 1'** parameter. This defines the Block.Field reference containing the value used to parameterise the corresponding '**Variable Line Name'**.
Observe the name configured in the '**Recipe N Name'** parameter on the Title Line tab is shown as the column header.

Select the required **Recipe Line N** parameter to display ....

Press ... to display the picklist and select the required Block.Field reference from the picklist.

If the Capture On parameter was enabled on the Title Line tab configure the Capture parameter. This defines the Block.Field reference that will be used to replace the value already configured in the corresponding Recipe Line N parameter.

Note

A recipe Capture can be performed via the Recipe Monitor page of the Visual Supervisor.

4.3.1 CONFIGURE THE VARIABLE LINE TABPARAMETERS (Cont.)

i.Select the required ‘Capture’ parameter to display [...].

ii.Press [...] to display the picklist and select the required Block.Field reference.

If the ‘Monitor On’ parameter was enabled on the Title Line tab configure the ‘Monitor’ parameter. This defines the Block.Field reference that will be monitored.

Note

Monitored values are displayed via the Recipe Monitor page of the Visual Supervisor.

i.Select the required ‘Monitor’ parameter to display [...].

ii.Press [...] to display the picklist and select the required Block.Field reference.

Tip!

Use **Disable LIN Verify (this cell)** to control the database verification of each Block.Field reference in the corresponding Variable Nr. Any errors are displayed in the Report pane.

If multiple Recipe Lines have been defined in the Title Line tab configure the **Recipe Line N, Capture On, and Monitor On** parameters as required.

Configure the **Recipe 1 to Recipe n** parameters as necessary, i.e. <Value>. This defines the default value of the Block.Field reference configured in the corresponding **Recipe Line N** parameter when the Recipe file is loaded.

If the Block.Field reference is an Enumerated data type the corresponding **Recipe 1** value provides a list of the enumerated values corresponding to the Block.Field reference defined in the **Recipe 1** parameter.

If the Block.Field reference is not an Enumerated data type the corresponding **Recipe 1** value supports free format text entry.

If the Block.Field reference is a Free format text entry it is verified to ensure the same data type is configured across all Variables in the row.

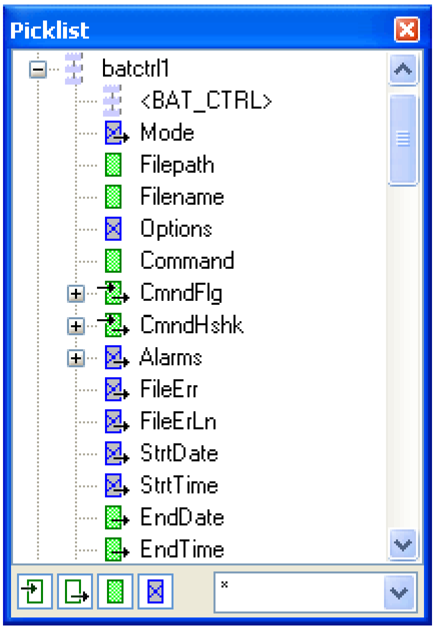
Tip!

The values configured in the Recipe N column will be replaced by any value written from the Block.Field reference configured in the corresponding Capture column.

5 USING THE PICKLIST

The picklist is a list of the Block.Field references available in the database assigned to this file. It contains icons and a drop-down menu used to show only the selected data type.

Tip!
Use the filtering icons and drop-down menu to ensure only the required data types are available when configuring Variable Line tab parameters.



Picklist

- batctrl1
  - <BAT_CTRL>
  - Mode
  - Filepath
  - Filename
  - Options
  - Command
  - CmndFlg
  - CmndHshk
  - Alarms
  - FileErr
  - FileErLn
  - StrtDate
  - StrtTime
  - EndDate
  - EndTime

*

To use the picklist

Press ⊞ to show the fields in the selected block.
The blocks are listed in alphabetical order, and can be filtered using the icons and drop-down menu. The icons can be used individually or combined to limit the Block.Field references shown. The drop-down menu supports additional data type filtering parameters that can also be used in conjunction with the icons.

Press the required filtering icon(s) to show only the Block.Field references that correspond to the selection.

Selecting ⇥ displays Block.Field references that can be wired to.

Selecting ↦ displays Block.Field references that can be wired from.

Selecting 🟩 displays writeable Block.Field references.

Selecting ☒ displays read-only Block.Field references.

Example
Selecting ⇥ and ↦ will display all Block.Field references that can be wired to, from or both.

5 USING THE PICKLIST (Cont.)

Select the required data type to show only the Block.Field references that correspond to the selection.

Tip!
Selecting * will display all Block.Field data types.

Boolean data types have True/False (Yes/No, On/Off, 1/0) values.

Byte data types have signed 8-bit (values from -128 to 127) values.

Integer data types have signed 16-bit (values from -32768 to 32767) values.

Long data types have signed 32-bit (values from -2147483648 to 2147483647) values.

Single data types have signed 32-bit (Single-Precision floating point values, i.e. an integer and a succession of decimal numbers plus an exponent, e.g. 1.374E+2) values.

String data types have textual (8-character string values) values.

Enumeration data types have menu (list of optional values) values.

BlockRef data types have block name (other block names values) values.

UnsignedByte data types have Unsigned 8-bit (values from 0 to 255) values.

UnsignedInt data types have Unsigned 16-bit (values from 0000 to FFFF) values.

UnsignedLong data types have Unsigned 32-bit (values from 0 to 4294967295) values.

Alarm data types have alarm (values indicating an alarm status and wired as a Boolean data type) values.

Date data types have date (day, month and year values) values.

Time data types have time (hour, minute and second values) values.

Subfield8 data types have 8-bit subfield (8-bit Boolean data type values wired individually or as one set) values.

Subfield16 data types have 16-bit subfield (16-bit Boolean data type values wired individually or as one set) values.

Note
The following data types are not permitted in the Recipe file.

Source data types have textual (maximum 78 character expression corresponding to EXPR.Expr of the database) values.

Normalised data types have calculated UCHAR block (UCHAR.X1 to UCHAR.X14 and UCHAR.Y1 to UCHAR.Y14 input co-ordinates in normalised 16-bit format distributed from 0 to 65535, monotonically increasing. Calculated from values configured in UCHAR.X0 and UCHAR.X15, and UCHAR.Y0 and UCHAR.Y15 respectively) values.

NormalLow data types have low UCHAR block (minimum input co-ordinate in normalised 16-bit format configured in UCHAR.X0 and UCHAR.Y0) values.

NormalHigh data types have high UCHAR block (maximum input co-ordinate in normalised 16-bit format configured in UCHAR.X15 and UCHAR.Y15) values.

WideString data types have textual (free format character string values) values.

Example
Selecting ⇥ and Boolean will display all Boolean data type Block.Field references that can be wired to.

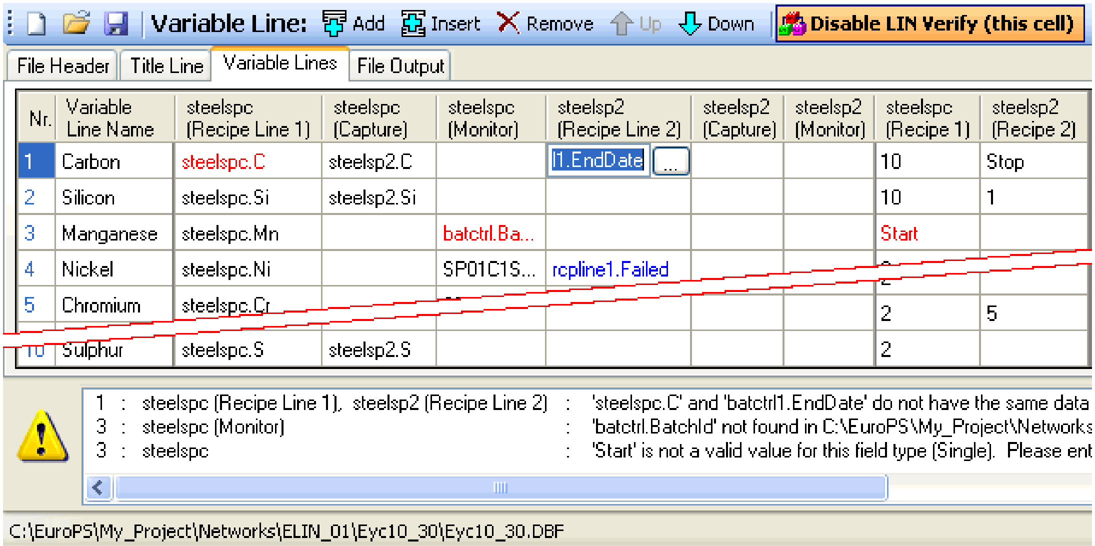
Select the required Block.Field reference to confirm configuration.

6 CONTROL VERIFICATION

This allows each Block.Field reference to be individually verified. Verifying a Block.Field reference checks that the same data type is configured across all Variables in the row, i.e. steelspc.Si is a Single-Precision floating point value so all other Block.Field references in this row must also be Single-Precision floating point values.

Tip!
Use the filtering icons and drop-down menu on the picklist to show the required data type only.

Successfully verified Block.Field references are shown in black and unsuccessfully verified Block.Field references shown in red. Parameters shown in blue indicate that verification of the Block.Field reference is disabled, i.e. verification of this Block.Field reference will not be attempted.



Nr.	Variable Line Name	steelspc (Recipe Line 1)	steelspc (Capture)	steelspc (Monitor)	steelsp2 (Recipe Line 2)	steelsp2 (Capture)	steelsp2 (Monitor)	steelspc (Recipe 1)	steelsp2 (Recipe 2)
1	Carbon	steelspc.C	steelsp2.C		l1.EndDate ...			10	Stop
2	Silicon	steelspc.Si	steelsp2.Si					10	1
3	Manganese	steelspc.Mn		batctrl.Ba...				Start	
4	Nickel	steelspc.Ni		SP01C1S...	rcpline1.Failed			2	
5	Chromium	steelspc.Cr						2	5
10	Sulphur	steelspc.S	steelsp2.S					2	

```
1  :  steelspc (Recipe Line 1),  steelsp2 (Recipe Line 2)  :  'steelspc.C' and 'batctrl1.EndDate' do not have the same data
3  :  steelspc (Monitor)                                   :  'batctrl.BatchId' not found in C:\EuroPS\My_Project\Networks
3  :  steelspc                                             :  'Start' is not a valid value for this field type (Single).  Please ent
```

C:\EuroPS\My_Project\Networks\ELIN_01\Eyc10_30\Eyc10_30.DBF

A Block.Field reference with verification disabled is shown between '{' and '}' on the File Output tab in the form of,

<Variable Line Name>,<Tagname>,<Value Captured>,<{Tagname Monitored}>,-,-,-,<Value>,-

Example
Nickel,steelspc.Si,,{BAT_CTRL.BatchId},,,1.203,

6.1 TO CONTROL THE DATA TYPE VERIFICATION

Select the Block.Field reference required.
If verification is disabled, the Block.Field reference is blue and the button will appear as **Enable LIN Verify (this cell)**. Press to enable database verification.
If the Block.Field reference is successfully verified it will be shown in black.
If the Block.Field reference is unsuccessfully verified it will be shown in red.
The button can then be used to disable verification of this Block.Field reference, i.e. **Disable LIN Verify (this cell)**.

If verification is enabled, the Block.Field reference will be black if verification is successful or red if verification is unsuccessful and the button will appear as **Disable LIN Verify (this cell)**. Press to disable database verification.
The Block.Field reference will be shown in blue indicating it will not be verified.
The button can then be used to enable verification of this Block.Field reference, i.e. **Enable LIN Verify (this cell)**.

7 MANAGE DATABASE SELECTION

Each Recipe file can be initially configured with or without an assigned database.

7.1 TO MANAGE THE SELECTION OF A DATABASE

If a Recipe file is not yet assigned to a database, use the ‘Database > Select database’ command from the Menu.

This launches a dialogue that allows the user to browse to the required instrument folder and select the database used to control this Recipe file. When the database is selected, all Block.Field references are verified, and a warning dialogue appears if the Block.Field references cannot be successfully verified.

Tip!
If the Recipe file is already assigned to a database, the path is displayed in the Status bar. <no DBF loaded> is displayed if the Recipe file is not already assigned to a database and verification is disabled.

If a Recipe file is already assigned to a database, use the ‘Database > Drop database’ command from the Menu. This removes the link to the already assigned database and any invalid recipe variable configuration is ignored, i.e. configuration errors do not apply because a database is not assigned. However, any Block.Field parameter that is not to be verified remains highlighted.

When a Recipe file is not assigned to a database the parameters that are used to define a specific block of a database become free format text fields. This allows the required database information to be configured using block or block.field syntax as appropriate.

8 PROTECT YOUR INTELLECTUAL PROPERTY

Recipe files can be password protected if Intellectual Property Protection is enabled using LINtools. LINtools allows database files, sequence files and action files to be password protected and permits the user to manage the passwords.

8.1 TO PASSWORD PROTECT FILES:

Enable Intellectual Property Protection in LINtools, see LINtools Help file.

Note
Intellectual Property Protection remains enabled until the Offer option to encrypt files on Save As is disabled (□).

In the Recipe editor select ‘File > Save as’ command to display the Save as dialogue. The ‘Save as password encrypted file’ checkbox is now available.

Beware...
Protection and storage of passwords is the responsibility of the user. Lost passwords cannot be recovered and protected files cannot be opened without the correct password.

Set the ‘Save as password encrypted file’ checkbox ☑ and press Save. This launches the’ Set encryption key’ dialogue which is used to configure the identifier and password for this file.

Enter the key identifier and Password. A key identifier can be a keyword (or memorable word) that is used to help associate the encrypted file with the corresponding password. It is also used by the instrument to identify which password is required to access the protected file. A new key identifier can be configured or an existing key identifier can be used. When the ‘Use a new encryption key’ is selected, enter the required key identifier. The configured password must then be confirmed to ensure that the password used is as intended.

If ‘Use an existing encryption key’ is selected, all existing key identifiers that have already been used this session are available from the drop-down list. This automatically assigns the corresponding password.

Tip!
The same key can be used by multiple files. Files in the strategy can be encrypted individually with different keys but a single instrument is limited to storing 7 keys.

Press **OK** to accept the key identifier and corresponding Password configuration.

9 WHAT IS?

9.1 WHAT IS EI-BISYNCH ?

EI-Bisynch is a proprietary protocol based on the ANSI X3.28-2.5 A4 standard for message framing. Despite its name, it is an ASCII based asynchronous protocol. Data is transferred using 7 data bits, even parity, 1 stop bit.

Address

Each instrument has a configurable address consisting of two digits, the first being a 'group' number 0 to 9, and the second a 'unit' number 0 to 9. There are, in principle, therefore 100 different addresses that may be used, 00 to 99, although in some instruments, address 00 is reserved for use in configuration mode, leaving addresses 01 to 99 available.

Mnemonics

EI-Bisynch identifies parameters within an instrument using what are known as 'mnemonics'. These are usually two letter abbreviations for a given parameter, for example PV for Process Variable, OP for Output, SP for Setpoint, and so on.

Channels

EI-Bisynch provides for 'r;channel' data. This would be used, for example, when a single physical unit contains several independent control loops, each having their own Process Variable, Setpoint, and Output Power. In this case, the values for each loop are obtained by specifying different channel numbers: '1', '2', etc.

The channel number, if used, is encoded as a single ASCII character preceding the mnemonic, for example 1PV.

9.2 WHAT IS INTELLECTUAL PROPERTY PROTECTION ?

This is the principle of preventing unauthorised access to or malicious use of configured strategy files. Protected files can be opened only by users who know the password and such files can be used only on authorised instruments.

Note
LINtools is used to enable you to protect your Intellectual Property (contents of the strategy files) by password protecting your files.

A protected file is encrypted using a key consisting of 'public identifier' (text viewed by anyone) and a private password. A key must be specified to protect a file, but the same key can be used to protect multiple files. When opening a protected file a dialogue is displayed showing the identifier, and asking for the password. The protected file opens if the correct password is used.

Beware...
Protection and storage of passwords is the responsibility of the user. Lost passwords cannot be recovered, and protected files cannot be opened without the appropriate password.

To use a protected file the instrument must be authorised with the correct key, as specified during instrument configuration. The same key can be used by many instruments. You can authorise an instrument with up to seven keys, which allows one instrument to use files protected with different passwords. A protected file cannot be used by an instrument that is not authorised with the correct key. This prevents users copying and using the strategy on different instruments without permission.

Note
The keys are managed using LINtools.

9.3 WHAT IS LIN?

Local Instrument Network or LIN is a communications protocol providing a token-passing, masterless Network which allows peer-to-peer communications and file transfer between instruments. It is supported via Ethernet, Arcnet and Serial communications connections.

A LIN Database groups data into blocks of related data. For example, a function block can represent an input, an output, a controller and so on. The LIN configuration tool (LINtools) and display packages (e.g. the User Screen Editor) recognise different types of function block, and handle them appropriately.

Communications between LIN and Modbus instruments require a Gateway. This is an interface between the LIN communications protocol and the Serial communications protocol, generally provided by blocks in the database. It involves the mapping of data from the LIN Database to the Modbus registers and digitals.

9.4 WHAT IS MODBUS?

Modbus is a communications protocol. Instruments that communicate via the Modbus communications protocol require a Gateway, that can be configured either as a Modbus master or a Modbus slave. It is supported via **Serial** and **TCP** communications interfaces.

The Modbus communication interface is built around messages. The format of these Modbus messages depends on the type of physical interface used. The same messages are used on both **EIA232** and Modbus/TCP over Ethernet. The same protocol can be used regardless of the connection type. This permits a device to communicate with several Modbus nodes at once, even if they are connected with different interface types, without the need to use a different protocol for every connection.

When using more versatile network systems like TCP/IP over Ethernet, Modbus messages are embedded in packets with the format necessary for the physical interface. In that case Modbus and other types of connections can co-exist at the same physical interface at the same time. Although the main Modbus message structure is peer-to-peer, Modbus is able to function on both point-to-point and multidrop networks.

9.4.1 Modbus Messages

Four basic elements are present in each message.

Device address	This is the address of the instrument receiving the data.
Function code	This is a code number defining message type.
Data	This is a block of data with additional information.
Error check	This is a Numeric check value to test for communication errors.

The sequence of these elements is the same for all messages, to make it easy to parse the content of the Modbus message. The instrument configured to operate as the master in the Modbus network always initiates communications. It sends a message and, depending of the contents of the message, a slave takes action and responds to it. Addressing in the message header is used to define which device should respond to a message.

Note

All other nodes on the Modbus network ignore the message if the address field does not match their own address.

The instruments use Modbus registers and bits to structure the instrument data. The Modbus registers and bits are simply a list of data points. Generally, there is no pre-defined structuring of these points into blocks or loops, etc., and most implementations define the allocation of registers individually.

Communications between LIN and Modbus instruments require a Gateway. This is an interface between the LIN communications protocol and the Serial communications protocol, generally provided by blocks in the database. It involves the mapping of data from the LIN Database to the Modbus registers and digitals.

9.5 WHAT IS PROFIBUS?

Profibus is an industry standard open network communications protocol supporting a maximum of 126 devices, including Profibus Master and Profibus Slaves. This protocol allows fast, cyclic and acyclic transfer of time critical data from intelligent devices such as temperature controllers, I/O units, drive, etc. to a PLC or PC based controller, with a scan time of around 10mS.

Refer to
www.profibus.com for full details of installation, technology and application.

9.5.1 Profibus Messages

Profibus messages use cyclic and acyclic data exchange functionality to transfer Input data or Output data. This is configured in the appropriate LIN Reference field on the Input data page or the Output data page.

Output data is a value that is written to the Profibus Slave from the Profibus Master.

Input data is a value that is read by the Profibus Master from the Profibus Slave.

9.6 WHAT IS REVIEW ?

Review software allows the display and printing of archived Data Recording (.uhh) files from suitable recorders, data acquisition units, etc. Review imports the recorded data from the .uhh files into its own database.

Recorded data is stored together in the database, so that data can be viewed across files.

The archived data files can be viewed and printed as though produced on a paper chart. For the purposes of Review, a 'chart' is a definition of a set of points, and a variety of setup options, e.g. number of grid divisions. The chart definition is stored in a Chart Group file, .cgp, see Review Help file, RM027962.

9.7 WHAT IS A BATCH ?

A Batch is the resulting product of a system derived from an individual Phase or a combination of Phases produced from a recipe controlled by an instrument.

A Batch file can be parameterised using a Recipe file of the same filename, e.g. Mixers.uyr and Mixers.uyb.

Example
A bakery may have a basic dough recipe that lists all of the ingredients required to make plain cookies and all of the optional ingredients such as nuts, fruit, and chocolate chips, that can be added to the basic recipe to make various kinds of cookies. In a steel mill, a recipe might be a collection of machine setup parameters. For batch processors, a recipe can be used to describe the various steps in the batch process.

9.8 WHAT IS A LIN DATABASE ?

A **LIN Database** (.dbf) is a database that runs in a LIN instrument providing data and rules, also known as a 'Strategy'. It is loaded by the LIN Instrument at runtime and allows the instrument software to control and monitor signals from sensors in the plant/system, e.g. an industrial plant, and then output the signals back to actuators.

The cycle of signal input, signal processing, and signal output to the entity is repeated continuously while the database is run in the instrument.

More than one LIN instrument can be involved in controlling a single entity, but only one LIN Database can run in a single LIN instrument at a time.

A LIN Database can be configured as a standard LIN Database or blended LIN Database, both operating in conjunction with one or more LIN Sequences running in the LIN instrument. It can also make use of LIN Actions stored in action files in the LIN instrument.

In LINtools, a LIN Database is represented and configured graphically as an arrangement of connected LIN function blocks, where input to output links, are represented as wires between blocks.

9.9 WHAT IS A LIN SEQUENCE?

A LIN Sequence is a program that runs in a instrument, in conjunction with a Database. More than one Sequence can be running at once in the instrument.

A Sequence interacts with the associated Database by writing new values to specified Database fields, in response to changes in the values of other specified Database fields.

A Sequence is employed when the process being controlled by the Database can adopt several distinct states - e.g. 'Starting Up', 'Full Running', 'Shutting Down', etc.

The purpose of a Sequence is to determine

1.The initial state(s) adopted by the process at start-up.

2.The conditions triggering state-changes (events).

3.The new state(s) adopted when changes are triggered.

4.The way the Database controls the process in each of the possible states.

9.10 WHAT IS A LIN FUNCTION BLOCK ?

LINtools uses a block-structured approach to configuring a control strategy, where a variety of ready-made **function blocks** perform the processing required in the LIN Database. Any strategy can use a combination of LIN block, Control Module block and Application block to create a strategy suitable for the LIN instrument.

A function block is an instance of a reusable module of program code, called a template, dedicated to a particular type of processing operation - e.g. the ADD2 template adds two numbers. In general, function blocks take in analogue and/or digital signals via their inputs, process them in a variety of ways, and then pass the results on via their outputs. You 'wire' the function blocks together so that the signals can flow between them to execute the control strategy.

Note

For ease of use, some blocks divide the fields into pages. Fields in blocks with paged parameters are referenced using a block.page.field.subfield.

9.11 WHAT IS A RECIPE ?

A Recipe file contains a Recipe Set and Recipe details including a list of ingredients, instructions and/or values derived from the assigned LIN Database. It is controlled using a combination of the Recipe file, the RCP_SET block (used to group the RCP_LINE blocks into a specific recipe configuration) and one or more RCP_LINE blocks (used to control the downloading of a specific recipe from the Recipe file associated with the defined RCP_SET block) in the assigned database.

A Recipe Set contains up to 16 recipes with each Recipe including the same set of variables but with different assigned values (e.g. for different colour mixes). The selected Recipe can then be ‘downloaded’ to the LIN Database controlling one of up to 8 plant lines.

Example

A bakery may have three Recipe Lines producing plain cookies, fruit and nut cookies and chocolate chip cookies. Each type of cookie will be manufactured using a specific plant line (recipe line).

A Recipe is a set of LIN function block field names (variables) with a particular value assigned to each, e.g. if the recipe is for one or more identical paint-mixing lines, it would include setpoint values representing the amounts of each pigment in the mixture needed for a particular colour. In use, the recipe is ‘downloaded’ to the LIN Database controlling a particular plant line, i.e. each recipe value is copied to the corresponding Block.Field to achieve the required control.

Example

A bakery may have a basic cookie dough recipe that lists all of the ingredients required to make plain cookies and all of the optional ingredients such as nuts, fruit, and chocolate chips, that can be added to the basic recipe to make various kinds of cookies. In a steel mill, a recipe might be a collection of machine setup parameters.

9.12 WHAT IS A USER SCREEN ?

A User Screen is a page that has been configured by the user to display specific information and control specific functions, i.e. Batch phases, and can be displayed by a Visual Supervisor.

Note

User Screen files are configured using the User Screen Editor and must be downloaded to the instrument with the associated database, see LINtools Engineering Studio.

10 INDEX

International sales and service

Invensys Operations Management
5604 Granite Parkway, Suit 1000
Plano
Texas 75024
469-365-6400

Invensys Canada – Ontario
Invensys Operations Management
5050 South Service Road
Suite 200
Burlington ON L7L 5Y7
Canada
1-905-333-1311

Invensys Canada – Montreal
Invensys Operations Management
4 Lake Road
Dollard-des-Ormeaux
Quebec H9B 3H9
Canada
1-514-421-4210
Tel: +9714 8074700
Fax: +9714 807477

Latin America
Invensys Systems Argentina
Nuñez 4334
(1430) Buenos Aires
Argentina
Tel: +54-11-6345-2100

Middle East
Jebel Ali Free Zone
P.O. Box 61495
Dubai
United Arab Emirates
Tel: +9714 8074700
Fax: +9714 807477

Asia Pacific
IPS (S) Pte Ltd
15 Changi Business Park Central
Singapore 486057
Tel: +65 6829 8888
Fax: +65 6829 8401

Europe and Africa
Invensys Systems France
S.A.10
Avenue du Centaure B.P. 8255 Cergy
95801 Cergy Pontoise Cedex
France
Tel: +33 1 34 43 25 25
Fax: +33 1 34 43 25 00

United Kingdom – Worthing
Invensys Eurotherm Ltd
Tel: +44 1903 268500
Fax: +44 1903 265982
Email: info.uk@eurotherm.com
www.eurotherm.co.uk

Invensys Systems, Inc pursues a policy of continuous development and product improvement. The specifications in this document may therefore be changed without notice. The information in this document is given in good faith, but is intended for guidance only. Invensys Systems, Inc will accept no responsibility for any losses arising from errors in this document.

invensys
Foxboro