Eurotherm®

by **Schneider** Electric

LIN Instruments
Batch File Editor

Table of Contents

1   Batch File Editor – Overview

> **Note**
> Throughout this User Guide, reference is made to 'User Screen'. This is NOT applicable to T2750 instruments.

A Batch (.uyb) file is an interface between the <u>LIN Database</u> and the Batch process equipment and consists of a list of parameter values that are used to produce a product <u>batch</u>.

The Batch File Editor provides an environment allowing Batch files to be inspected and configured. The editor operation is divided into tabs that correspond to the different sections of the Batch file.

> **Note**
> Each Batch Phase can be controlled using a <u>LIN Sequence</u>, <u>Setpoint Program</u> or a <u>User Screen</u> linked to a database.

To launch the Batch File Editor

The editor is launched by selecting, ▦ Start > All Programs > ... > Batch File Editor, where … represents the default installation path, or Tools > Batch 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 you will be asked to specify an existing database or create a Batch file that will not be associated with a database until the Batch file is saved.

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

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

1.1   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 recipe is a list of ingredients or steps used to create the product and is configured using the Recipe File Editor.

> **Example**
> *A bakery may have a basic cookie dough recipe that lists all 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.*

1.2   Configuring a Batch File - Overview

The following stages may overlap, and their order can be varied especially when you are editing an existing strategy.

To configure a Batch File, follow these stages.

   1   For a new Batch File, <u>configure the File Header tab parameters</u>.

The File Header tab parameters configure a 6-line, fixed format, description that provides most of the information contained in the Batch file. It is specifically used to determine to relationship between the Batch operation and the database.

> **Note**
> A LIN Database always requires a Header block (Configuration category) to determine the strategy instrument type. For the T2750, this is the 'TACTICIAN' block.

   2   Then, <u>configure the Batch Phases tab parameters</u>.

The Batch Phases tab parameters configure each phase in the Batch associated with this Batch file.

   3   Inspect the File Output tab.

The File Output tab shows the Batch File as displayed when opened using a text editor, i.e. Notepad.exe.

Your configured Batch file should now be complete and ready to download to the instrument

1.3   Batch File Editor Window

1.3.1 Batch File Editor window

The Batch File Editor window provides access to the parameters used to configure a Batch file. It is divided into three tabs:

■ <u>File Header tab</u>

■ <u>Batch Phases tab</u>

■ File Output tab

The Editor also provides:

■ Information pane

This appears only in the Header Text tab display.

■ Report pane

This pane appears if configuration errors occur.

1.3.2 What is the File Header tab?

The File Header tab is a 6-line, fixed format, description that provides general information contained in the Batch file, see configure the File Header tab parameters.

The information configured on this tab is validated each time the tab is selected. Each invalid configuration is listed in the Report pane.

UYB,1 (Eycon) File Header tab	UYB,2 (T2750 File Header tab
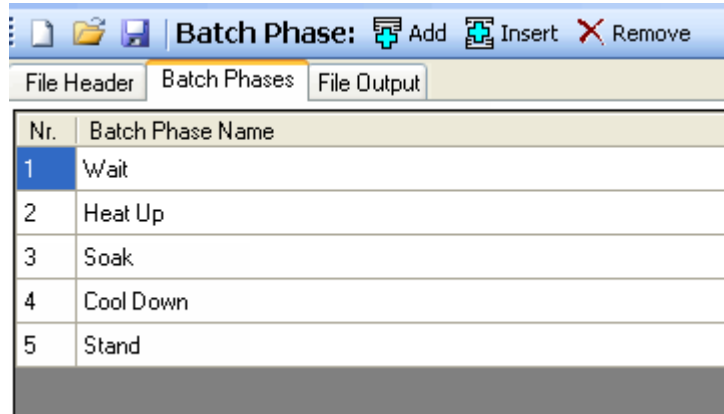
1.3.3 What is the Batch Phases tab?

The Batch Phases tab lists each of the phases that will be used in the Batch, see <u>configure the Batch Phases tab parameters</u>.

Each phase is the result of the Phase logic controlled by the Phase Controller, Sequence (SFC) Setpoint Program or User Screen.

> **Note**
> If the Batch file is controlled using a Setpoint Program, any configured Batch Phase Name will be replaced with the corresponding Segment name, i.e. Batch Phase number 1 (Wait) is replaced with the name from Segment 1 of the Setpoint Program.

1.3.4 What is the File Output tab?

The File Output tab shows the Batch File as displayed when opened using a text editor, i.e. Notepad.

> **Tip!**
> *This editor is the recommended tool for configuring a Batch file. Advanced users can still use a text editor, i.e. Notepad, although <u>additional constraints</u> are applied.*

This is the expected output from the configuration applied in the <u>File Header tab</u> and <u>Batch Phases tab</u>.

```
File Header   Batch Phases   File Output

UYB,1
3,03/10/2008,10:33:04,raypy,03/10/2008,11:33:04
BATCH1,0,60,2,Bat:R
0,,,#U16:#W1:10,#U17:#W2:10,#U18:#W3:10,#U19:#W4:10,,,
L Group,,

Loading Bin ->M1
Bin in M1
M1-Prewash
M1-Treatment
M1-Drain
M1-Drying
Loading Bin ->M2
Bin in M2
Spare1
M2-Prewash
M2-Treatment
M2-Drain
M2-Drying
Loading Bin ->M3
Bin in M3
Spare2
M3-Drying
Unloading Bin
19
20
```

1.3.5 Constraints when using a text editor

If using a text editor (e.g. Notepad) to configure a Batch file, the following constraints must be noted,

- No line may exceed 512 characters, including line feed/carriage return instructions
- Spaces are counted as characters
- 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

1.3.6 What is the Information pane?

This pane displays additional information about the parameter selected in the File Header tab.
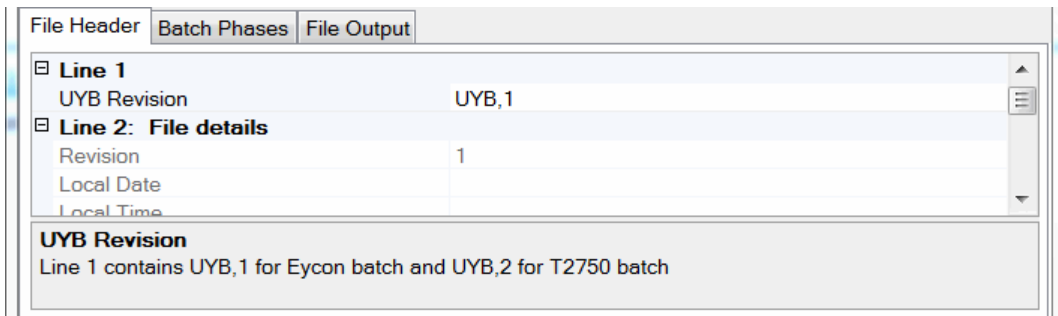
1.3.7 What is the Report pane?

The Report pane displays information about each configuration error, but only appears when errors have occurred.
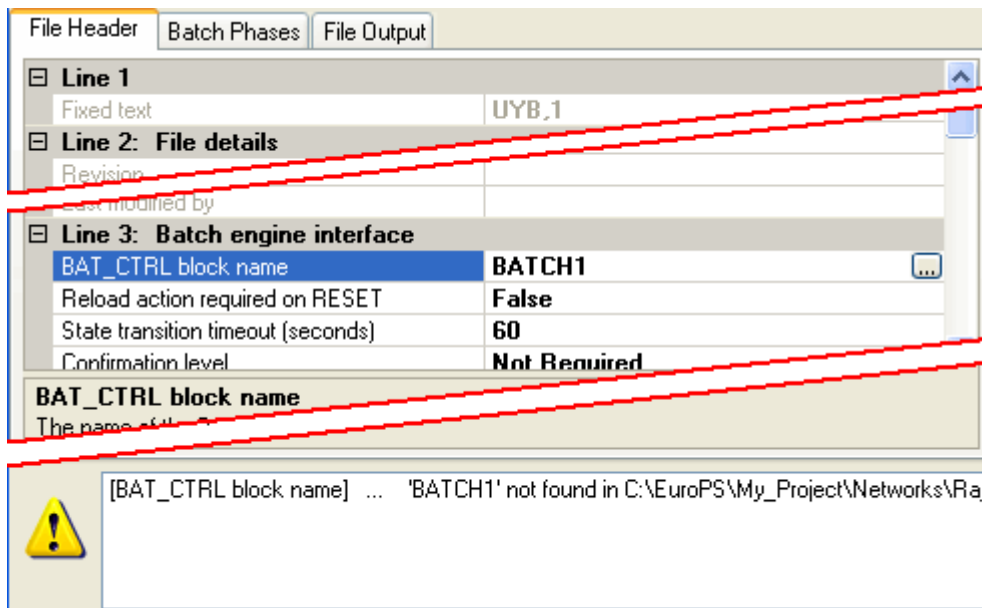
2 Menus
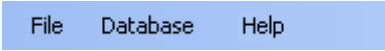
This section provides a summary of the functions of all toolbar buttons and menus displayed in this editor.

It is intended to be a quick reference to support button or menu item functions.

Note
A 'tooltip' will appear when first locating the cursor over a toolbutton.

Menu bar

The Menu Bar is a special toolbar that contains the pulldown commands. Each pulldown displays a further list of commands.

Toolbar

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

Status Bar

The Status Bar displays the filepath to the database assigned with this Batch file.

2.1 File menu

The File menu enables the control of an individual LINtools application file using the following commands.
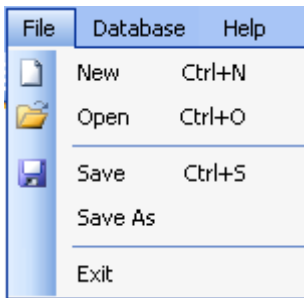
2.1.1 New Program file command

<Ctrl> + <N>

Click this to create a new Batch file (.uyb).

If the currently open Batch file is already assigned to a database a dialog asking if you want to continue using this database (.dbf file) is displayed.

- Select Yes to continue to use the existing database.
- Select No to create a Batch 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 Batch file that is not yet associated with a database

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

2.1.2 Open file command

<Ctrl> + <O>

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

2.1.3 Save command

<Ctrl> + <S>

Click this to display the **Save** dialog. Click this to automatically save the active file with its current file name, location, and file format.

2.1.4 Save As file command

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

2.1.5 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.

2.2 Database menu

The Database menu provides commands used to control the database assigned to Batch file currently open in this editor.
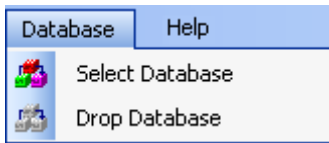
2.2.1 Select Database command

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

2.2.2 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.

2.3 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)

Any of the first 3 options opens the on-line help documentation with the selected as the priority command.
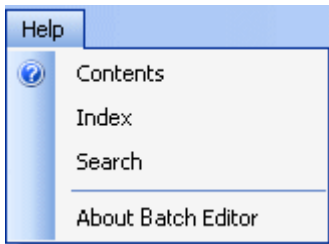
2.3.1 Help Topics command

Click this to open the help documentation.

- Select Contents to open the 'Help' file, displaying the 'Contents' page as the priority.
- Select Index to open the 'Help' file, displaying the 'Index' page as the priority.
- Select Search to open the 'Help' file, displaying the 'Search' page as the priority.

2.3.2 About command

Click this to display a dialog, showing the software version number.

3   How to...

3.1   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.

To configure the File Header parameters

1   Observe the parameters in Line 1 and Line 2.

> **Note**
> Line 1 specifies the batch file type, type 1 being for Eycon and type 2 for the T2750 target instrument.
>
> The parameters shown in Line 2 are read-only and provide revision data about the file (Revision number, Local and System Date and Time, and Modified by parameters).

2   Configure the Line 3 parameters.

These parameters are used to define the Batch engine interface. This configures the control of the standard Batch state machine.

> **Note**
> Configuration of Line 4 parameters, Line 5 parameters and Line 6 parameters is optional.
>
> From Line 4, only the user item parameters apply to the T2750.

3   Configure the Line 4 parameters.

These parameters are used to define the Batch display interface. This configures the link between the blocks in the database that define the display details for a configured User Screen, and/or a Batch start screen in the instrument.

> **Tip!**
> *Some Line 3 parameters and Line 4 parameters can be used by Review.*

4   Configure the Line 5 parameters.

These parameters are used to define the Batch log interface. This configures the link between the blocks in the database used to configure the values that will be logged and generate batch reports and the associated Report file, .uyf.

5   Configure the Line 6 parameters.

These parameters are used to define the Batch recipe interface. This configures the link between the Recipe interface block, RCP_SET block, RCP_LINE block and corresponding Recipe file (.uyr), and the Batch file. This Recipe file is used to parameterise the Batch.

3.2   How to configure the Line 1 parameters

This sets the strategy instrument for the batch, and is shown on the File Output tab in the form of,

<p align="center"><code>UYB,1</code> or <code>UYB,2</code></p>

■ UYB Revision

Set this value by choosing the appropriate value from the drop-down list:

`UYB,1` for Eycon batch files, `UYB,2` for T2750 batch files.

3.3   How to configure the Line 2 parameters

These are read-only parameters and display current revision details of this Batch file.

This information is shown on the File Output tab in the form of,

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

Example:

03,03/10/08,10:33:04,Fred Bloggs,03/10/08,11:33:04

where:

'`03`' is the revision level of this Batch file. This value is incremented each time the Batch 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.

'`10:33:04`' 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.

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

'11:33:04' is the system hours:minutes:seconds that the last revision was saved.

3.4   How to configure the Line 3 parameters

These parameters are used to define the Batch engine interface. This configures the control of the standard Batch state machine.

This information is shown on the File Output tab in the form of,

    <Batch Control block>,<End Action>,<Timeout>,<Confirm Level>,<BatchID Prefix>:R

> **Example**
> BATCH1,0,60,2,ABC:R

1   Configure the **Batch block name** parameter, i.e. `<Batch Control block>`.

This defines the name of the block used to run the Batch.

> i   Select the **Batch block name** to display ▣.
>
> ii   Press ▣ to display a picklist showing all the BAT_CTRL (UYB,1) or BatchControl (UYB,2) blocks available.
>
> iii   Select the required block from the picklist.

> **Tip!**
> *Consult the LIN Block Reference Manual for details*

2   Configure the **Reload action required on RESET** parameter, i.e. `<End Action>`.

This defines the action applied on RESET.

> i   Select the **Reload action required on RESET** to display a drop-down arrow.
>
> ii   Select **True** to unload the Batch file on a RESET and require it to be re-loaded to run the next Batch, or **False** allow the Batch to restart without reloading the Batch file.

3   Configure the **State transition timeout (seconds)** parameter, i.e. `<Timeout>`.

This defines a timeout period of up to a maximum of 60 seconds for state transitions. If the transition from one state to the next does not occur before this time period has expired to transition is considered as a failure.

> i   Select the **State transition timeout (seconds)** to enable a text cursor.
>
> ii   Type the maximum time (seconds) allowed for the transition from one state to the next to occur.

4   Configure the **Confirmation level** parameter, i.e. `<Confirm Level>`.

This defines the level of confirmation required when starting a Batch from the instrument panel.

> **Note**
> This is not applicable to the T2750. It is set to 'Not Required' and greyed out for a UYB,2 Batch File.

> i   Select the **Confirmation level** to display a drop-down arrow.
>
> ii   Select the level of confirmation required.
> - **Not Required** (default) specifies that confirmation is not required.
> - **Confirmation Dialog Box** specifies that *OK/Cancel* confirmation is required
> - **Password Required** specifies that re-entry of the *Password* is required

> **Note**
> If the use of electronic signatures has been configured, requested authorisation of the higher level will be required, i.e. the Batch cannot continue until two different signatures have been entered if the Batch file is set to Verify but electronic signatures in the instrument is set to Sign & Authorise.

5   Configure the **Batch ID prefix** parameter, i.e. `<BatchID Prefix>`.

> **Note**
> The **Batch ID** is used to identify each Batch from the Batch Selector dialog of <u>Review</u>.

This optional parameter can be used in conjunction with the **Batch ID read-only** parameter (UYB,1 only), to define a prefix for the associated Batch. The `<BatchID Prefix>` overwrites the BatchID, from the left, with the characters configured in the `<BatchID Prefix>`.

> **Example**
> *A BatchID of* `00000014` *with a Prefix of* `RKN` *would result in a BatchID of* `RKN00014`.

> i   Select the **Batch ID prefix (Optional)** to enable a text cursor.

ii   Type the text to be used as a prefix for the BatchID.

6   Configure the **Batch ID read-only** parameter (UYB,1 only), i.e. `:R`.

This is used to control the editing of the BatchID from the instrument panel, and is only available if the **BatchID prefix (Optional)** parameter is configured.

i   Select the **Batch ID read-only** to display a drop-down arrow.

ii   Select **True** to prevent any editing of the BatchID from the instrument (HMI for the T2750) (BatchID is read-only), or **False** to allow the BatchID to be edited when the Batch file is accessed in the instrument.

7   Configure the **Batch ID Modifier** parameter (UYB,2 only), i.e. `:R`.

This is used to define whether the Batch ID can be changed once it has been automatically set when the batch is loaded.

i   Select the **Batch ID Modifier** to display a drop-down arrow.

ii   Select **ReadWrite** to make the Batch ID writeable (it can also be changed, but doesn't have to be), **ReadOnly** (the Batch ID cannot be changed), or **MustWrite** (the Batch ID *must* be changed).

> **Note**
> Configuration of Line 4 parameters, Line 5 parameters and Line 6 parameters is optional.

3.5 How to configure the Line 4 parameters

> **Note**
> The first three parameters (User Screen page number, Display Group block name and Message Filter block name) are not applicable to the T2750 instrument and are thus greyed out for this instrument type.

These parameters are used to define the Batch display interface. This configures the link between the blocks in the database that define the display details for a configured User Screen, and/or a Batch start screen in the instrument.

This information is shown on the File Output tab in the form of,

```
<User Screen>,<Display Group>,<Message Filter>,<User Item1>,...,<User Item6>
```

> **Example**
> `0,group01,BAT_CTRL,#U16:#W1:10,#U17:#W2:10,#U18:#W3:10,#U19:#W3:10,,`

1   Configure the **User Screen page number (Optional)** parameter, i.e. `<User Screen>`.

This defines the number of the user screen page associated with the Batch, or 0 if none.

i   Select the **User Screen page number** to enable a text cursor.

ii   Type the number of the user screen page associated with the Batch.

2   Configure the **Display Group block name (Optional)** parameter, i.e. `<Display Group>`.

This defines the name of the GROUP block (used to group blocks and channels into a specific recording configuration) associated with the Batch or blank if none.

i   Select the Display Group block name (Optional) to display .

ii   Press to display the picklist showing all the GROUP blocks available.

iii   Select the required GROUP block from the picklist.

> **Tip!**
> *Consult the LIN Block Reference Manual for details.*

3   Configure the **Message Filter block name (Optional)** parameter, i.e. `<Message Filter>`.

This defines the name of the block used to give a context for the message associated with this Batch or blank if none. It may be the name of the BAT_CTRL block or a GROUP block containing the BAT_CTRL block or possibly some other block of interest. This controls the display of the message bar on the instrument panel Batch status display and is only applied if PNL_MSG blocks are used to generate operator messages associated with the Batch.

i   Select the Message Filter block name (Optional) to display .

ii   Press to display the picklist showing all the relevant blocks available.

iii   Select the required block from the picklist.

> **Tip!**
> *Consult the LIN Block Reference Manual for details.*

4 Configure the **User Item 1 (Optional) to User Item 7 (Optional)** parameters, i.e. `<User Item1>`.

These parameters are used to configure the start-up values of an individual Batch shown on the Batch start screen dialog of the instrument.

This specifies the Title and Value to be displayed on each Batch start screen, i.e. `<Title>`, `<Value>`, `<Width>`, `<Initial Value>`, `<Attributes>`. The value may be edited prior to starting the Batch.

i   Open the **User Item 1 (Optional)** parameter to display the additional parameters.

- Select **Title** to enable a text cursor.
- Type the User dictionary file reference in the form of #`<Dict>` (U = User dictionary, W = Writeable dictionary, B = Batch dictionary)`<Index>`, e.g. `#U12`. For UYB,2 batch files, this can also be plain text.
- Select **Value** to enable a text cursor.
- Type the Value for the Writeable dictionary file reference, e.g. `#W12` or a database block field that will be written to this parameter, e.g. `[LOOP_PID.SL]`. For UYB,2 batch files, this can also be plain text.
- Select **MinimumWidth** to enable a text cursor.
- Type the minimum number of characters to be used to display the value.
- Select **Width** to enable a text cursor.
- Type the number of characters used to display the value.

> **Example**
> *A Width of `3` would result in a value of `12389` being shown as `123`.*

- Select the **Initial value** to enable a text cursor.
- Type the required dictionary file reference to be used as an initial value shown in **Value** when a Batch is loaded, e.g. `#U13` (user dictionary file reference). This value must be another dictionary file reference and applied at load. For UYB,2 batch files, this can also be plain text.
- Select the **Attributes** to enable a text cursor.
- Define the process for editing the value configured in the **Initial value** parameter. Type `W` to specify that the value must be written. This implies that data entry must occur after loading and before starting the Batch. It doesn't mean that the value must necessarily be *changed*, merely that it must be written to. Type `M` to specify that the value must be modified. This applies to Writeable dictionary file references only and specifies that the value configured in **Initial value** must be edited after the Batch is loaded and before it is started.

ii   If necessary, open the **User Item 2 (Optional), User Item 3 (Optional), User Item 4 (Optional), User Item 5 (Optional), User Item 6 (Optional), User Item 7 (Optional)** parameters as required to display the additional parameters and repeat. Each of these items is only available if the previous item has had values defined.

3.6 How to configure the Line 5 parameters

These parameters are used to define the Batch log interface. This configures the link between the blocks in the database used to configure the values that will be logged and generate batch reports and the associated Report file, .uyf.

This information is shown on the File Output tab in the form of,

<LOGROUP>,<[Block.Field]>,<Report>

Example
BAT_LOG,[BATCH1.Id],BAT_REPT

Note
The first two parameters (Batch log LGROUP name and LIN database field) are not applicable to the T2750 instrument and are thus greyed out for this instrument type.

1   Configure the **Batch log LGROUP block name (Optional)** parameter, i.e. <BAT_LOG>.

This defines the name of the LGROUP block (this block groups data from DR_DGCHP, DR_ANCHP, DR_REPRT and DR_ALARM blocks for archiving) used to group Batch data logging.

Tip!
The default state of this block is ON (LGROUP.State is ON). This must be set OFF (LGROUP.State is OFF) to ensure that Batch data logging is controlled by the Batch.

   i   Select the Batch log LGROUP block name (Optional) to display 🔲.

   ii   Press 🔲 to display the picklist showing all the LGROUP blocks available.

   iii   Select the required LGROUP block from the picklist.

2 Configure the **LIN database field (Optional)** parameter, i.e. <[BATCH1.Id]>.

This defines the block and field used as the first 2 characters of the Filename.

   i   Select the LIN database field (Optional) to display 🔲.

   ii   Press 🔲 to display the picklist showing all the blocks available.

   iii   Select the required block and then field from the picklist.

3 Configure the **Batch Report block (Optional)** parameter, i.e. <BAT_REPT>.

This defines the name of the DR_REPRT block used to generate Batch reports. If a Batch report is to be generated, three reports (Start = 1, Stop = 2, Abort = 3) are required in the Report file (.uyf). This file must have the same filename as the Batch file (.uyb), i.e. sample.uyb and sample.uyf.

3.7 How to configure the Line 6 parameters

These parameters are used to define the Batch recipe interface. This configures the link between the Recipe interface block, RCP_SET block, RCP_LINE block and corresponding Recipe file (.uyr), and the Batch file (.uyb). This Recipe file is used to parameterise the Batch.

This information is not shown on the File Output tab.

4   Configure the **Recipe interface block (Optional)** parameter.

This defines the name of the RCP_LINE block used to parameterise the Batch.

Tip!
If a Batch file (.uyb) is configured use the corresponding Recipe file (.uyr), e.g. Mixers.uyb and Mixers.uyr.

   i   Select the **Recipe interface block (Optional)** to display 🔲.

   ii   Press 🔲 to display the picklist showing all the RCP_LINE blocks available.

   iii   Select the required block from the picklist.

## 3.8 How to Configure the Batch Phases tab parameters

These parameters are used to configure the identification of each Phase in the Batch.

Each phase of a Batch is can be controlled via a sequence or User Screen.

Tip!
A maximum of 40 phases for each Batch is permitted.

3.9 How to configure the Batch Phases parameters

1   Press [Add] to add one Phase to the Batch Phases tab.

   If Phases already exist on the Batch Phases tab, this will add one row to the end of the list of existing Phases.

   - Alternatively, press [Insert] to add one row before the selected Phase.

2   Type a title for Batch Phase in the corresponding Batch Phase Name column.

   This is used to identify the Batch Phase in the configured Phase Controller sequence (SFC) or User Screen.

File Header	Batch Phases	File Output

Nr.	Batch Phase Name
1	Loading Bin ->M1
2	Bin in M1
3	M1-Prewash

3   Press [Remove] to delete the selected Phase.

3.10 Manage database selection

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

3.11 To manage the selection of a database

■   If a Batch file is not yet assigned to a database, use the **Database > Select database** command from the Menu.

   This will launch a dialog that allows you to browse to the required instrument folder and select the database required to control this Batch file. When the database is selected, the Report pane will display all the configuration errors that exist.

> **Tip!**
> *If the Batch file is already assigned to a database the Status bar will specify the path. It will display <no DBF loaded> if the Batch file is not already assigned to a database.*

■   If a Batch file is already assigned to a database, use the **Database > Drop database** command from the Menu.

   This will remove the link to the already assigned database and any invalid Batch configuration is ignored, i.e. configuration errors do not apply because a database is not assigned.

   When a Batch 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.

4   What is

4.1   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 'r;group' number 0 to 9, and the second a 'r;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 'r;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: 'r;1', 'r;2', etc.

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

4.2   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 (i.e. User Screen Editor) recognise different types of function block, and handle them appropriately.

Communications between LIN and <u>Modbus</u> 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 <u>LIN Database</u> to the Modbus registers and digitals.

4.3   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 interface.

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 **RS232** 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 relate to 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, the 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.

Modbus Messages

Each Modbus message has the same structure. 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 acts 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 differently.

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.

4.4   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.

Profibus Messages

Profibus message use cyclic and acyclic data exchange functionality to transfer Output data or Input data.

- 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.

This is configured in the appropriate LIN Reference field on the Input data page or the Output data page.

4.5   What is Review?

The Review software allows the display and printing of archived Data Recording files, .uhh, 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.

4.6   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.

4.7 What is a LIN Sequence?

A **LIN Sequence** is a program that runs in an instrument, in conjunction with a <u>LIN Database</u>. 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.

You employ a Sequence 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

- The initial state(s) adopted by the process at start-up.
- The conditions triggering state-changes (events).
- The new state(s) adopted when changes are triggered.
- The way the Database controls the process in each of the possible states.

4.8 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. Any strategy can use a combination of LIN block, Control Module block and Application block to create a strategy suitable for the 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 the <u>LIN Database</u>. 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*.

4.9 What is a Setpoint Program?

A **Setpoint Program** is a combination of two program files (Program Template file (.uyw) and Program File (.uyy)) that run in a instrument, in conjunction with a <u>LIN Database</u>.

The purpose of a Setpoint Program is to control and manage the target value that an automatic control system, e.g. PID controller, will aim to reach.

> **Example**
> *A boiler control system might have a temperature Setpoint, i.e. a temperature the control system aims to attain in the system.*

4.10 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.

5   Index

Scan for local contents

HA029961 Issue 3 (CN36284)