

F.A.B.I.A.N.
**(Fast Access Booking
Information and
Notification)**
Control System Link

FabianLink32.exe

Written by: Simon Dowson, Richard Kerry.

Contents

Contents	1
1 Overview	4
1.1 Description	4
1.2 Device firmware version.....	4
1.3 BNCS configuration	4
1.3.1 The v1/v2 Location	4
1.3.2 The V3 Location	5
1.3.3 The V4 Location	5
1.4 Info driver compatibility	5
1.5 CSI version	5
1.5.1 Automatic Shutdown on CSI Closing.....	5
1.6 Messaging System – Bncslf32.dll.	5
1.7 Driver PC requirements	6
1.8 Resilience and redundancy.....	6
1.9 Device interface description	6
1.10 Device limitations	6
1.11 Device setup	6
2 Driver setup	6
2.1 Ini file settings	6
2.1.1 [Configuration]	6
2.1.2 [Fabian_Server_n].....	12
2.1.3 Note 1. Packet-Mode.....	16
2.1.4 Device_nnn Configuration	16
2.2 Revertive Tally Cache	20
2.2.1 Poll On Connect	20
2.2.2 Per-server	21
2.2.3 Per-device	22
2.3 Command Line Overrides.....	23
2.4 Setting-Up a Link.....	24
2.5 Status InfoDriver	24
2.6 Active InfoDriver	24
2.6.1 Link Active – Slots 1 to 1000	24
2.6.2 Connection Override – Slots 1001 to 2000.....	25
2.6.3 Active Address – Slots 2001 to 3000.....	25
2.6.4 Test – Slots 3001 to 4000.....	25
2.7 Dynamic Database Updating	25
2.8 Bi-Directional Links	26
2.9 Active Link Switching	27
2.9.1 FM - Fabian Message	27
2.10 Active Device Switching.....	27
2.11 Revertive Filtering	28

2.12	Traffic Management of F-Series Messages	28
2.12.1	Compact Comments	29
2.12.2	Spread Responses	29
2.12.3	Packed Messages	29
2.12.4	Spread Close Messages	30
2.12.5	Packed Close Messages	30
2.13	Sample Ini.....	30
3	Program GUI.....	34
3.1	File Menu.....	35
3.1.1	Exit.....	35
3.2	Window Menu.....	35
3.2.1	Clear.....	35
3.3	Diagnostics Menu.....	35
3.3.1	On.....	35
3.3.2	Send RUTs	36
3.3.3	Show RUTs.....	36
3.3.4	Logging.....	36
3.4	Help Menu	36
3.4.1	About	36
3.5	Title Line	36
4	Resilience and redundancy	36
5	Logging	36
6	Documents referenced	36
7	Notes.....	37
7.1.1	[Configuration]	37
8	Version history.....	37
8.1	Software Version	37
8.2	Document version	42

1 Overview

1.1 Description

The Fabian Control System Link module FabianLink32 enables a Broadcast Networked Control System/Colledia Control (BNCS/CC) installation to control a remote BNCS installation over a TCP/IP link. FabianLink32 becomes a client of a Fabian Server (FabianServer32). It forwards commands, interrogation messages and database change notifications on the local BNCS network and passes them on to the server if it is configured to process them. The Fabian Server is a BNCS client on the remote BNCS network. Fabian Link registers as a BNCS driver with the local CSI.

FabianLink32 can maintain active connections with up to 128 Fabian servers. Remote device ID's can be translated into different local ID's if there is a conflict between the two areas or the system administrator wishes to logically group the areas.

In the event that the network connection to the server is lost, FabianLink32 will attempt to reconnect, progressively increasing the time between retries, until it attempts every 30 seconds. Connection from FabianLink32 to FabianServer32 will be restored automatically when the link itself is restored

FabianLink32 can be configured to connect to an alternative FabianServer32. Once connection is lost to the server, either because the link has failed or the server itself has failed, then FabianLink32 will attempt connection to the 'Alternative' server. The address of the alternative server is set in the FBCSLINK.INI file. This connection will remain and no attempt will be made to reconnect to the 'first' server, until such time that the link or alternative server itself is lost.

The server requires a user name and password before it will allow access to either booking or control functions. FabianLink32 looks after the logging on procedure and if successful can be made to run a local application.

1.2 Device firmware version

FabianLink32.exe has been tested with the following versions of FabianServer32:

V4.07.02, V4.07.04, V4.09.13

This program is a revised version of v3fbblink, which in turn was based on earlier programs. Most references to FabianLink32 apply equally to v3fbblink.

If a version number less than four is used, then v3fbblink (or an earlier instance) is meant.

Similarly references to FabianServer32 may apply to v3fbserve or earlier.

1.3 BNCS configuration

The program is able to find the dev.ini file in any of the conventional locations:

1.3.1 The v1/v2 Location

Configuration files are in **C:\Windows** or **C:\Winnt**, according to the value of the WINDIR environment variable, as also seen using the GetWindowsDirectory function.

Log files are written in **C:\bnclslogs**.

1.3.2 The V3 Location

The configuration files are in a folder specified in **C:\bncs_config.ini**. The location of the config files is given by the ConfigPath key.

Log files are written in **C:\bncslogs**.

1.3.3 The V4 Location

The configuration files are in a folder specified using environment variables CC_ROOT and CC_SYSTEM. The config files are found in

%CC_ROOT%\%CC_SYSTEM%\config\system.

Log files are written in **%CC_ROOT%\%CC_SYSTEM%\logs**.

1.4 Info driver compatibility

FabianLink32 has a direct connection to an InfoDriver for filtering; also it registers as one to provide its own monitoring and control interface (see the descriptions of StatusInfoDriver, ActiveInfoDriver and RevertiveFilterInfoDriver in section 2.1.1 [Configuration] of the Driver setup section, and also section 2.5 Status InfoDriver and section 2.6 Active InfoDriver).

It is known to work with the following InfoDriver versions :

InfoDriver32 - 1.0.0.7, 1.0.0.16.

1.5 CSI version

Known to work with the following:

Csi32 – V1.0.30.xx. V1.1.18. V1.2.19. V1.2.32. V1.2.48.

V3csi – V3.04.03.

It is presumed to work with many earlier and later versions, although there is no specific information preserved on earlier compatibility.

1.5.1 Automatic Shutdown on CSI Closing

If FabianLink32 has any device numbers configured then it will automatically close when CSI closes.

If it has no registrations, which will typically happen the first time it is run, when it has automatically generated its own default ini file, it will not automatically shut-down with CSI. This is due to there being no device registrations with CSI which are what CSI normally uses to achieve this.

It is possible that future development of CSI might remove this limitation.

1.6 Messaging System – BncsIf32.dll.

Versions up to 4.6 used WM_COPYDATA for communicating with CSI.

Versions from 4.7 onward use BBC_COPYDATA, also known as “DLL messaging”, via the BncsIf32.dll library.

In the course of changing FabianServer32 to DLL messaging a flaw was found with this library, necessitating a new version being produced. Like FabianServer32, FabianLink32 requires BncsIf32.dll to be at least version 4.3.

If you are using the wrong version you'll get a message dialog to the effect that: The procedure entry point BNCS32TxMessage2 could not be found in the dynamic link library BNCSIF32.dll.

1.7 Driver PC requirements

There are no specific requirements other than for standard BNCS.

1.8 Resilience and redundancy

The Fabian Link/Server system is able to support a variety of resilience options. See sections 2.9 Active Link Switching and 0 Active Device Switching for details.

1.9 Device interface description

FabianLink32 connects to a Server using Telnet on a TCP/IP connection.

1.10 Device limitations

1.11 Device setup

See FabianServer32 documentation for details on setting up the server.
See also section 2.4 Setting-Up a Link.

2 Driver setup

When FabianLink32 is started it looks for a file in the configuration directory called FBCSLINK.INI. If not found the file is created.

2.1 Ini file settings

2.1.1 [Configuration]

The [Configuration] section will contain the following default values.

Item	Value	Comment
DebugMode	0	If this is set to '1' FabianLink32 will produce more verbose output both during initialisation and normal operation. It can be turned on and off using the 'Diagnostics' option on the menu bar.

Item	Value	Comment
Send RUTS	1	<p>If set to '1' FabianLink32 will send a regular RUT (aRe yoU There) message to the servers. If no RUT replies are received (RUR) after five attempts the connection will be broken and re-established. This setting is mainly intended for Dial-Up, ISDN or Telnet proxy situations where regular RUTs will prevent the connection from timing out and the line dropping.</p> <p>Note: See the ExpectRUTReply parameter in the server configuration section.</p>
Show RUTS	0	If set to '1' FabianLink32 will display the RUT messages in the debug window
InfoDriver	999	<p><i>Obsolete entry. This has been replaced by the 'StatusInfoDriver' entry below, which is more descriptive of its function. The 'InfoDriver' parameter will be copied to the new 'StatusInfoDriver' parameter and this old entry labelled 'OBSOLETE'. If this entry does not exist in the INI file when the application starts, i.e. it has been removed, then no action is taken.</i></p>
StatusInfoDriver	0	<p>If set to a valid device Id FabianLink32 will report the state of each server connection. One slot is used per server. The connection states are : -</p> <p>0 = NOT_CONNECTED 1 = CONNECTED 2 = LOGGEDON 3 = INACTIVE 4 = ACTIVE</p>

Item	Value	Comment
ActiveInfoDriver	0	If set to a valid device Id FabianLink32 will allow server connections to be independently disabled or enabled by writing a 0 or 1 respectively to the appropriate slot. See section 2.6 Active InfoDriver for more details.
RevertiveFilterInfoDriver	0	If set to a valid device Id FabianLink32 will use a slot per device Id to maintain a comma delimited list of assertive indexes to filter out. See section 2.11 Revertive Filtering.
AlertOnError	0	If set to '1' a message box will pop-up when a successful connection is made or a disconnection occurs. If set to '0' the message appears in the scrolling window.
QuitOnLinkFail	0	If set to 1 FabianLink32 will close when connection to the server is lost.
CacheEnable	1	Cache 1=ON 0=OFF. FabianLink32 has its own revertive cache. This is entirely separate to any cache that CSI might be maintaining. Enabling the Fabian Link cache reduces link traffic by satisfying poll commands locally.
CacheFilePath	C:\	<i>This entry is obsolete and if found will be marked as such.</i> <i>Tally data is no longer held in files.</i>

Item	Value	Comment
PollServerDatabases	0	<p>If set to 1, the FabianLink32 will regularly poll the remote servers for database names for all devices registered for redirection. The names returned from the server are compared with the local database names provided by CSI. If the names differ then a message is sent to the local network to update the name.</p> <p>Any ApplCore Database Change (ADC) notifications received from the server will be forwarded as Router Database 'RD' commands via CSI to the local network.</p> <p>Note : Up until V3.03.10 this parameter was called 'UpdateDatabases'. Any existing parameter with this name will be marked 'OBSOLETE'.</p>
DatabaseUpdateDelay	1000	<p>This sets the rate in milliseconds at which the database update system checks each name. The default is 1000ms.</p>
CheckLinksByConnectionNumber	1	<p>When set to zero link checks are performed using the IP address of the server. When set to 1 the checks use the index, or entry number, of the server from the FBCSLINK.INI file. See section 2.9 Active Link Switching for details.</p> <p>Note: From V3.03.10 the default is now '1'.</p>
AlwaysRegisterMaxIndices	0	<p>If set to 1 this parameter results in any registrations with devices on the remote server being requested with a maximum index of 4096, rather than using the 'DatabaseSize_1' entry from each 'DEV_xxx.INI' file</p>

Item	Value	Comment
PollOnConnect	NO (default) DATABASE MAXIMUM SELECTIVE-NO SELECTIVE-DATABASE SELECTIVE-MAXIMUM	<p>Sets globally whether a poll will be sent immediately after connecting to the Server and registering for a device.</p> <p>Options are :</p> <p>NO – No poll will be done.</p> <p>DATABASE – A poll will be done using the value of DatabaseSize_1 for the device.</p> <p>MAXIMUM – A poll will be done using the maximum allowable value (ie 4096)</p> <p>SELECTIVE-NO SELECTIVE-DATABASE SELECTIVE-MAXIMUM</p> <p>Same as the above except enable the Selective functionality.</p> <p>See section 2.2.1 Poll On Connect for full details.</p>
DualLink	1	<p>DualLink should be set to 1 when you have a dual headed link, that is when two or more instances of FabianLink32 are connected to the same server. This mode activates messaging on the local network (FI, FA and FO – Inquire, Active & Object) so that only one link actively passes data.</p> <p>Note: From V3.03.10 the default is '1'.</p>
<i>SynchronousMode</i>	1	<p><i>This entry is obsolete and if found will be marked as such.</i></p> <p><i>FabianLink32 must always be operated synchronously through internal FIFO buffering; otherwise asynchronous re-entrancy will randomly cause corrupted data and possibly other side effects.</i></p>
ProxyAddress	NONE	<p>If FabianLink32 needs to connect to a server via a firewall proxy server then this address needs to point to that server. If this parameter is set to 'NONE' then the connection is configured to be direct.</p>

Item	Value	Comment
ProxyPort	23	The IP port on which to connect to the proxy server.
ProxyPrompt	NONE	If a proxy server is being used then this parameter is a prompt or keyword to look for coming from the proxy server inviting a command to initiate a connection to the remote server. For example 'Login' or 'Ready'. This is dependent upon the proxy server used and may be established by using a Telnet application to connect to the proxy server to see what it sends.
ProxyCommand	NONE	If a proxy server is being used then this parameter is the command that FabianLink32 should use in response to the 'ProxyPrompt' from the proxy server. This is dependent upon the proxy server used and may be established by using a Telnet application to connect to the proxy server to see what it requires.
ConnectRetry_Seconds	15 (default value)	<p>Delay (in seconds) added to the connection retry queue system after a connection fails.</p> <p>The permitted range is 1 to 300 (5 minutes).</p> <p>If a value less than 1 is provided it will be changed to 1 and written into the config file.</p> <p>If a value greater than 300 is provided it will be changed to 300 and written into the config file.</p>
NumberOfHosts	64	<p>Number of server connections.</p> <p>The maximum allowed is 128.</p> <p>This number determines the number of connections expected and the number of Fabian_Server_n sections to be read from the configuration file.</p>
CompactFMessageComments	0 (off – default) Or 1 (on)	<p>Selects the compact form for F-series network messages, ie the format without comments.</p> <p>See section 2.12.1 Compact Comments.</p>

Item	Value	Comment
SpreadFMessageResponses_ms	0	Selects the time interval between successive F-message packets generated during the link check. See section 2.12.2 Spread Responses.
PackFMessageResponses	0 (off – default) Or 1 (on)	Indicates that multiple F-series messages should be packed into a single network message, rather than being sent using one network message per F-series message. See section 2.12.3 Packed Messages.
SpreadFCloseMessages_ms	0	Selects the time interval between successive FC packets generated during closedown. See section 2.12.2 Spread Responses.
PackFCloseMessages	0 (off – default) Or 1 (on)	Indicates that multiple FC messages should be packed into a single network message, rather than being sent using one network message per F-series message. See section 2.12.3 Packed Messages.
LinkCheckDelay_ms	60000	Sets the interval for link checking (in milliseconds). See sections 2.9 Active Link Switching and 2.12 Traffic Management of F-Series Messages.

2.1.2 [Fabian_Server_n]

After the configuration section there will be a number of sections labelled [Fabian_Server_n] where 'n' is an integer between 1 and NumberOfHosts. Each section has the following entries:-

Item	Value (Example or Default)	Comment
------	----------------------------	---------

Item	Value (Example or Default)	Comment
Name	NONE	The name of the remote server or site. This is used in messages to the user regarding the status of a link. This name will be automatically updated with the name of the remote server when a connection is established.
Address	NONE	The IP address of the remote server, from version 3.02.00 this can be a WINS or FQDN name.
AltAddress	NONE	This is the IP address of the Alternative FabianServer32 to which this application will connect to in the absence of a connection to the default FabianServer32. This will not be used if it set to 'NONE'.
<i>ProxyAddress</i>	<i>OBSOLETE</i>	<i>This parameter has been moved to the [Configuration] section as it common to all server connections</i>
<i>ProxyPrompt</i>	<i>OBSOLETE</i>	<i>This parameter has been moved to the [Configuration] section as it common to all server connections</i>
<i>ProxyCommand</i>	<i>OBSOLETE</i>	<i>This parameter has been moved to the [Configuration] section as it common to all server connections</i>
Port	23	The default port address is the Telnet port (23). If this is already in use the recommended alternative is 401
UserName	User	Your user name on the remote server. No limit on the number of valid characters used is enforced by Fabian Link. Fabian Server currently allows up to 24 characters.
UserPass	Password	Your password on the remote server. No limit on the number of valid characters used is enforced by FabianLink. Fabian Server currently allows up to 24 characters.

Item	Value (Example or Default)	Comment
ControlPass	Password	<p>A secondary password required before the remote server will allow you access to control functions.</p> <p>No limit on the number of valid characters used is not enforced by FabianLink.</p> <p>Fabian Server currently allows up to 24 characters.</p>
LogonExec	NONE	This is an application that you wish to run as soon as FabianLink32 has successfully connected to the remote server.
ExpectRUTReply	0	When FabianLink32 sends a RUT message it will expect a reply if this parameter is set to 1. If no reply is received after 5 RUTs have been sent then the connection is broken and an attempt made to re-establish.
BNCSRevertiveMode	NORMAL PACKET	<p>Commands are always sent from FabianLink32 to the server as ASCII text. When set to NORMAL this parameter ensures that reverts from the server are also returned as ASCII text. If set to PACKET the reverts are returned in a more compact hexadecimal form which gives a greater packet density and hence greater throughput of data.</p> <p>See Note 1. Packet-Mode</p>
LinkActive	1 (for servers 1 to 4) 0 (for servers 5 to NumberOfHosts)	If set to '1' FabianLink32 will attempt to connect to the remote server. If set to '0' it will wait until the link is enabled via one of the ActiveInfoDriver slots. This feature enables links to be turned on and off from user panels and client applications.
Devices	16	The number of remote devices that you wish to have access to. The default of 16 also sets the number of default entries that FabianLink32 will create.

Item	Value (Example or Default)	Comment
LockTimeToRemoteSystem	0	If set to '1' this will enable the sending of 'Get System Time' (GST) messages to the remote server. The response from the server will cause FabianLink32 to set the time on the workstation on which it is running. If the CSI on the same workstation is running with its time server enabled then all workstations on the local network will have their time locked to the remote system. If FabianLink32 is connected to multiple remote systems then only one link, if any, should have this option enabled.
BiDirectionalLink	0	This parameter toggles the processing of 'ApplCore Database Change' (ADC) messages received from a remote server. If set to '0', which is the default, then ADC messages carrying notification of a database change on the remote network will be forwarded by FabianLink32 to the local network as 'Router Database' (RD) messages. If set to '1' the ADC messages are not forwarded to the local network. Note : See section 2.8 Bi-Directional Links.
UseProxy	0	If set to '1' then attempt to make a connection will be made via the proxy server detailed in the [Configuration] section.
ConnectionOverride	0	This value controls the use of the Alternative server address, if specified. If the value is 0 and an alt address is configured, then Fabian Link will automatically switch to use the other address on any failure to connect. If the value is 1, only the main address will be used. If the value is 2, only the alternate address will be used. No other values are recognized. See section 2.6 Active InfoDriver for further details.

Item	Value (Example or Default)	Comment
Device_nnn (where nn is a number from 001 to the value of "Devices")		Device connection configuration. See section 2.1.4 Device_nnn Configuration for details.
PollOnConnect_Server	(no default)	Per-server poll-on-connect setting. See section 2.2.1 Poll On Connect for full details.
PollOnConnect_Device_nnn	(no default)	Per-device poll-on-connect setting. See section 2.2.1 Poll On Connect for full details.

Where a value is described as a string with no limit on length there is actually a limit of 256 characters.

2.1.3 Note 1. Packet-Mode

If Packet-mode is required the server is informed using the "SUM 1" command when logging-in. When it finds any revertives it wishes to send to the client it does so using packet-mode rather than normal-mode. What this means within the server is that when it receives a Universal Revertive (UR) message from CSI it sends a packet-mode message to the Fabian Link, rather than sending individual normal-mode messages for each individual revertive message received from CSI. A packet-mode message may contain multiple messages but the whole UR message is encoded and sent down the Telnet link. FabianLink will then decode this and pass the revertives to its CSI.

Note however that currently the received revertives are sent to CSI as separate revertive messages and not as a single UR message. This is to allow the re-mapping of received device numbers. It may be noticed (in CapLog) that the number of revertives per UR message on the Link's network is less than the number per UR message on the Server's network.

2.1.4 Device_nnn Configuration

With the number of device specified in the Devices parameter being the default of 16 the following entries will be added to each [FabianServer_n] section :-

```
Device_001=GRD,000,000,ALL,ALL
Device_002=GRD,000,000,ALL,ALL
Device_003=GRD,000,000,ALL,ALL
Device_004=GRD,000,000,ALL,ALL
Device_005=GRD,000,000,ALL,ALL
Device_006=GRD,000,000,ALL,ALL
Device_007=GRD,000,000,ALL,ALL
Device_008=GRD,000,000,ALL,ALL
Device_009=GRD,000,000,ALL,ALL
Device_010=GRD,000,000,ALL,ALL
Device_011=GRD,000,000,ALL,ALL
Device_012=GRD,000,000,ALL,ALL
```



```
Device_013=GRD,000,000,ALL,ALL
Device_014=GRD,000,000,ALL,ALL
Device_015=GRD,000,000,ALL,ALL
Device_016=GRD,000,000,ALL,ALL
```

(If the "Devices" parameter was absent, for example if the ini file is being generated for the first time, the first entry will be written as:

Example: Device_001=Device Id (ie GRD, GPID or ID), Local Id, Remote Id

This is meant as an indication of the format to use.

Provided a value of "Devices" is found, the conventional form shown above is written.)

When this section has been configured with a valid remote server address FabianLink32 will attempt to process the list of devices the next time it is run. For every device in the 'Devices' entry an entry of the form:-

'Device_xxx=GRD,xxx,yyy,NONE,NONE'

will be created. It is assumed that all the remote devices are router drivers; if this is not the case 'GRD' should be replaced with 'GPID' or 'INFO' where appropriate. ('ID' can be used in place of 'INFO'). Mapping is achieved by the using the second number:-

Device_001=GRD,001,001...

...would map remote system GRD 001 onto the local system as GRD 001 on the remote system.

Device_001=GRD,001,999 ...

...would map remote system GRD 001 onto the local system as GRD 999 on the remote system.

The entries should be modified to include mapping (if any) in the list of devices
For example

Device_001=GRD,001,001	GRD	No mapping
Device_002=GRD,009,009	GRD	No mapping
Device_003=GPID,099,099	GPI	No mapping
Device_004=GRD,007,007	GRD	No mapping
Device_005=INFO,100,100	InfoDriver	No mapping
Device_006=ID,101,501	InfoDriver	Mapped Remote system 501, Local system 101
Device_007=GRD,011,511	GRD	Mapped Remote system 511, Local system 11
Device_008=GRD,012,512	GRD	Mapped Remote system 512, Local system 12
Device_009=GRD,013,513	GRD	Mapped Remote system 513, Local system 13
Device_010=GRD,014,514	GRD	Mapped Remote system, 514 Local system 14

FAQ

Q. Some devices in my devices list seem to register ok, but others don't. What might be wrong?

A. Check that you have DEV_nnn.INI files available for the devices that won't register.

2.1.4.1 Database Update Configuration

From V3.06.03 onwards two extra parameters have been added which determine how the database updating mechanism is configured for each device. The first of the new parameters (forward) holds the configuration for the forward direction, in other words, which database changes are passed across the connection by Fabian Link. The second of the new parameters (reverse) determines whether changes in the remote databases are allowed through to change the local database, if any.

2.1.4.1.1 Database Forward Update Flag

The database forward update flag controls which local database changes are passed across the connection to change the remote database.

When Fabian Link receives a Router Modify (RM) message from CSI, if the forward database update flag is set for the given database number, and the relevant host connection is active, a corresponding ARM message is sent to the Fabian Server hosting the device.

If the database number is 1 (ie destination) then it is checked for validity against the known size of that database (or the maximum possible size, if so configured) and the command is not sent to the server if it is not valid. Other database numbers' sizes are not checked in this way.

2.1.4.1.2 Database Reverse Update Flag

The database reverse update flag controls whether changes in the remote databases are allowed through to change the local database, if any.

2.1.4.1.2.1 ARN

When an ARN command is received, if the host connection is active and the connection to CSI is OK, and PollServerDatabases is set, and the reverse database update flag is set for the given database number, then the local database entry will be updated correspondingly, via CSI.

If the new value is the same as the existing value no change is sent to CSI.

If the name value received is "!!!" no change is made as this value is used to indicate invalid or absent names. In this case an error message is shown.

2.1.4.1.2.2 ADC

When an ADC command is received, if the host connection is active, and the reverse database update flag is set for the given database number, then the local database entry will be updated correspondingly, via CSI.

2.1.4.2 Device Configuration String

'Device_nnn=GRD,000,000,NONE,NONE' is the default when a new FBCSLINK.ini is being created and indicates that no name changes will be passed by FabianLink32 for that device, on that link.

Replacing 'NONE' with 'ALL' would cause FabianLink32 to pass all name changes in the appropriate direction for that device.

Replacing 'NONE' with a group of numbers (e.g. '01459') would allow through only name changes for those particular databases, in that particular direction, for that particular device on that particular link.

To Summarise:-

NONE	<i>(default) No Name changes in that direction.</i>
ALL	<i>Name changes for ANY database WILL be passed in that direction.</i>
0	<i>Only name changes for Database 0 will be passed in that direction.</i>
160	<i>Only name changes for Databases 0,1 and 6 will be passed in that direction.</i>
23945	<i>Only name changes for Databases 2,3,4,5 and 9 will be passed in that direction.</i>
0123456789	<i>The same as 'ALL'.</i>

FabianLink32 will look for GRD_XXX.INI, GPID_XXX, ID_XXX.INI or DEV_XXX.INI files for all the devices in the list. If the file for a device is not found, a warning message will be given indicating that the device will be ignored for the current session. The INI files are required so that the maximum number of destinations, inputs/outputs or info slots can be read. FabianLink32 needs this information so that it can maintain an internal tally table for each device.

2.1.4.3 Additional Device Options

From version 4.07.12 an option has been added to allow further configuration to be added for individual devices. This is activated by adding "DeviceOptions" to the Device_nnn line.

For example:

Device_001=ID,401,401,NONE,ALL,DeviceOptions

Once this has been done another line should be added using the key DeviceOptions_nnn, with the same nnn as the Device line. This line should list all the options required, separated by commas if more than one. The current selection is as follows.

SplitSizes	<p>If the PollServerDatabases option is enabled then an ADP command is sent to the Fabian Server to retrieve its database sizes for databases 0 and 1. The lesser of the local and remote size values is then used for all commands that require database sizes to be used.</p> <p>If the SplitSizes option is not present the lesser of the local and remote database 1 size values is also used as the size of the device</p>
-------------------	---

	<p>(ie number of slots or destinations) for all commands that need this.</p> <p>If the SplitSizes option is present the device size will always be the DatabaseSize_1 value from the device's local ini file. It will not be re-set based on a value read from the server.</p> <p>If PollServerDatabases is not set then this option has no effect as no size information is read from the server. In this case the database 1 size read from the local ini file is always used for both the database 1 size and the device size.</p>
RegisterMaxIndex	<p>If this option is enabled the device will register for all indices, and the maximum index value (4096) will be used as the maximum size by all device commands that need it.</p> <p>This setting is essentially the same as the global AlwaysRegisterMaxIndices setting, except that it applies to a specific device.</p>

For example:

DeviceOptions_001=SplitSizes,RegisterMaxIndex

DeviceOptions_001=SplitSizes

2.2 Revertive Tally Cache

FabianLink32 caches revertive data in a similar way to CSI, maintaining an internal tally table for each remote device. This means that once the **cache** for a remote device has been filled further requests from panels will be satisfied from the cache.

When a panel polls for the first time after the link is established FabianLink32 may be configured to poll the remote driver via the Fabian server. The revertives are stored and subsequent polls will be met by FabianLink32 from its internal tables. The error corrected TCP/IP link should ensure that the local tables are kept up to date and data traffic on the link is kept to a minimum.

2.2.1 Poll On Connect

When a connection is made to a Fabian Server all relevant device registrations are then done.

If it is considered necessary the Fabian Link may also be configured immediately to poll these devices. This is done using the various PollOnConnect entries in the configuration file.

Poll-on-connect may be controlled at three levels – global, per-server or per-device. This allows particular servers, or particular devices to have different poll-on-connect settings.

For example a server on a slow link could have poll-on-connect disabled, or a device with a wide but sparsely used address range could poll only the necessary addresses.

There are three main options available:

NO	No Poll will be done. This is the default setting at the global level.
DATABASE	A Poll will be done using the device's size as configured by the DatabaseSize_1 entry in its dev.ini file.
MAXIMUM	A Poll will be done using the maximum possible value of devices size, ie 4096.

(These settings are all case-insensitive.)

Note that a large poll will, if done using Normal mode, take an appreciable period of time to complete. This may be speeded up by using Packet mode. Or a selective mode may be applicable to control the server or device's action.

To use the per-server or per-device modes the global setting needs to include the "Selective" flag. SELECTIVE-NO, SELECTIVE-DATABASE and SELECTIVE-MAXIMUM correspond to the settings listed above and enable the selective modes.

SELECTIVE-NO	By default no Poll will be done.
SELECTIVE-DATABASE	By default a Poll will be done using the device's size as configured in DatabaseSize_1 entry in its dev.ini file.
SELECTIVE-MAXIMUM	By default a Poll will be done using the maximum possible value of devices size, ie 4096.

In all these cases per-server or per-device settings may override this.

If these Selective modes are not used any per-server or per-device setting will be ignored.

2.2.2 Per-server

The poll-on-connect behaviour may be changed on particular servers by using per-server configuration. This is controlled using the PollOnConnect_Server entry in the Fabian_Server_n section.

This setting has no default value. FabianLink32 will not write any value if none is found.

The setting is optional. If it is absent a per-device setting may still be used.

The setting may have the following values:

NO	No Poll will be done on any devices on this server.
DATABASE	A Poll will be done on all devices on this server using the device's size as configured in DatabaseSize_1 entry in its dev.ini file.
MAXIMUM	A Poll will be done on all devices on this server using the maximum possible value of devices size, ie 4096.
SELECTIVE-NO	By default no Poll will be done.
SELECTIVE-DATABASE	By default a Poll will be done using the device's size as configured in DatabaseSize_1 entry in its dev.ini file.
SELECTIVE-MAXIMUM	By default a Poll will be done using the maximum possible value of devices size, ie 4096.

These settings are all case-insensitive.

2.2.3 Per-device

The poll-on-connect behaviour may be changed for particular devices by using per-device configuration. This is controlled using the `PollOnConnect_Device_<nnn>` entries in the `Fabian_Server_<n>` section. Replace `<nnn>` by a number, which should be in “%03d” format, ie three digits with leading zeroes if less than three digits are needed. There may be any number of such entries but they must be numbered consecutively starting with 001. When a consecutive number is absent, the program will stop reading and no higher numbered settings will be read.

This setting has no default value. FabianLink32 will not write any value if none is found.

The setting is optional. If it is absent a per-device setting may still be used.

The setting is in the form:

`PollOnConnect_Device_<nnn>=<devices>:<mode>`

or

`PollOnConnect_Device_<nnn>=<devices>:<mode>:<indices>`

The devices section may be a single device number, a set of numbers, a range of numbers or a set of ranges. Sets are comma-separated. Ranges use “-” to indicate the limits.

If a device list references a device not on this server the devices not on this server are ignored and those that are local are actioned as if the others weren’t listed.

The mode may have the following values:

NONE	No Poll will be done on the listed device(s).
DATABASE	A Poll will be done on the listed device(s) using each device’s size as configured by the DatabaseSize_1 entry in its dev.ini file.
MAXIMUM	A Poll will be done on the listed device(s) using the maximum possible value of device size, ie 4096.
EXPLICIT	The device(s) will be polled according to <indices>.

These settings are all case-insensitive.

If Explicit mode is used, then an indices section must be provided. For the other modes, there must be no indices section.

The indices section may be a single device number, a set of numbers, a range of numbers or a set of ranges. Sets are comma-separated. Ranges use “-”.

The indices section may also use the keywords “maximum” or “database” to indicate the maximum index (4096) or any given device’s database-1 size. These may only be used at the end of the section, not in the middle of it, nor in the devices section.

Some examples of device poll-on-connect configuration strings:

- `PollOnConnect_Device_001=815:Maximum`

- Poll device 815. 1 to 4096.
- PollOnConnect_Device_002=50:None
 - Don't poll device 50. Even if this server is otherwise being polled.
- PollOnConnect_Device_003=801:Database
 - Poll device 801 according to its database-1 size.
- PollOnConnect_Device_004=900:Explicit:1-5,12-15,19,21,100-Maximum
 - Poll device 900. Slots 1 to 5, 12 to 15, 19, 21, 100 to 4096.
- PollOnConnect_Device_005=900:Explicit:1-5,12-15,19,21,100-Database
 - Poll device 900. Slots 1 to 5, 12 to 15, 19, 21, 100 to database-1 size.
- PollOnConnect_Device_006=900:Explicit:1-5,12-15,19,21,100-102
 - Poll device 900. Slots 1 to 5, 12 to 15, 19, 21, 100 to 102.
- PollOnConnect_Device_007=900-903:Explicit:1-5,12-15,19,21,100-102
 - Poll devices 900 to 903. Slots 1 to 5, 12 to 15, 19, 21, 100 to 102.
- PollOnConnect_Device_008=900,905:Explicit:1-5,12-15,19,21,100-102
 - Poll devices 900 and 905. Slots 1 to 5, 12 to 15, 19, 21, 100 to 102.
- PollOnConnect_Device_009=900,905:Explicit:1-5,12-15,maximum,21,100-102
 - This is an error – "maximum" may only be used at the end of the list.
- PollOnConnect_Device_010=900,905:Explicit:1-5,12-database,19,21,100-102
 - This is an error – "database" may only be used at the end of the list.
- PollOnConnect_Device_011=900,905:Explicit:1-5,12-15,19,21,database-102
 - This is an error – "database" may only be used at the end of the list.
- PollOnConnect_Device_012=900,database:Explicit:1-5,12-15,19,21,100-102
 - This is an error – "database" may only be used in the indices section.
- PollOnConnect_Device_013=900,database:Whatever:1-5,12-15,19,21,100-102
 - This is an error – unknown mode "Whatever".
- PollOnConnect_Device_014=900,905::1-5,12-15,19,21,100-102
 - This is an error – mode missing.
- PollOnConnect_Device_015=900,905:Explicit:1-5,12-15,19:21,100-102
 - This is an error – Extra colon in the indices section.

2.3 Command Line Overrides

The device Ids for Status, Active and Revertive Filter InfoDrivers are taken, by default, from the FBCSLINK.INI file. However where it is desired to use the same INI file on more than one workstation these Id's can be overridden by supplying them as command line parameters, in that order.

For example starting FabianLink32 like this :-

FabianLink32.exe 101 102 103

... will result in the Status, Active and Revertive Filter InfoDrivers having Ids of 101, 102 and 103 respectively.

2.4 Setting-Up a Link

Establish the TCP/IP link before running FabianLink32 which will then make up to five attempts to contact each of the remote servers. Connections should occur on the first try, but occasionally some settling time is required after the TCP/IP link is established.

Note: If Alternative Servers have been specified in the FBCSLINK.INI then the application will attempt to connect to the main server first, and if connection cannot be obtained will attempt to connect to the alternative server. If the alternative server cannot be found then FabianLink32 will try once again to connect to the main server. This process will continue until a successful connection is obtained.

2.5 Status InfoDriver

The Status InfoDriver specified by the **StatusInfoDriver** parameter in the **[Configuration]** section has a slot showing the connection status for each corresponding server.

For example, slot 7 will reflect the status of the server whose configuration can be found in the [Fabian_Server_7] section. The slots contents are numeric and have the following meanings: -

0 = NOT_CONNECTED

1 = CONNECTED

2 = LOGGEDON

3 = INACTIVE

4 = ACTIVE

2.6 Active InfoDriver

The Active InfoDriver is specified by the **ActiveInfoDriver** parameter in the **[Configuration]** section, or by the second command-line parameter. It is used for controlling the connections to servers. There are three groups of slots.

2.6.1 Link Active – Slots 1 to 1000

Slots 1 to 1000 control the Active state of a link.

Writing a **1** to a slot will instruct FabianLink32 to connect to the server corresponding to that slot number. Writing a **0** to the slot will disconnect the link. For example, writing a **1** to slot **7** will instruct FabianLink32 to connect to the server whose configuration can be found in the **[Fabian_Server_7]** section of the FBCSLINK.INI file. Writing a **0** to slot **7** will disconnect the link.

The slot contents are also reflected in the **LinkActive** parameter in the INI file so the setting will be persistent. The upper limit of a 1000 allows more than the currently allowed number of servers.

2.6.2 Connection Override – Slots 1001 to 2000

Slots **1001** to **2000** control whether FabianLink32 connects to the main or alternate address for the server connection corresponding to (slot number – 1000).

There are three options. Writing to any of these will break any existing connection prior to reconnecting:

Writing a **0** to a slot will attempt a reconnection to the server address that was **not** used last. If a repeated disconnection should occur FabianLink32 will alternate between the main and alternative servers.

Writing a **1** to a slot will force a reconnection to the **main** server. If a repeated disconnection should occur FabianLink32 will repeatedly connect to the main server.

Writing a **2** to a slot will force a reconnection to the **alternative** server. If a repeated disconnection should occur FabianLink32 will repeatedly connect to the alternative server.

The slot contents are also reflected in **ConnectionOverride** parameter in the INI file so the setting will be persistent. The upper limit of 2000 allows more than the currently allowed number of servers.

2.6.3 Active Address – Slots 2001 to 3000

Slots 2001 to 3000 reflect the currently active address for the server connection corresponding to (slot number – 2000).

A **1** in a slot means the main address is being used whereas a **2** means the alternate address is in use.

2.6.4 Test – Slots 3001 to 4000

Slots 3001 to 4000 are for test purposes and are not documented. Any functionality found here should not be relied upon for operational use. Nor should it be expected to be consistent.

2.7 Dynamic Database Updating

FabianLink32 is capable of automatically keeping databases up to date. The default is off (PollServerDatabases=0). If enabled (ie. set to '1') then FabianLink32 will take each redirected driver Id in turn and query an index once per second. If there is a discrepancy between the string returned and the one in the local system's database then the local system will be updated.

When FabianLink32 logs on to a server it issues an ADP command for each device that is configured to redirect commands for. The ADP command requests a device profile from the server which takes the form of a single line of information. The information contains, amongst other things, a comma delimited list of the databases that exist for that device. FabianLink32 uses this list to determine which databases exist on the remote system for a given device and will only make automatic database name requests for those that exist.

The size of databases 2 – 9 on a remote system are potentially unknown and subject to change. They may well be not under the control of the person administering the BNCS system on the FabianLink32 side. Rather than interrogate all possible 4096 entries for each device FabianLink32 uses the following simple rules: -

- If a database exists in the device profile list for a device then the automatic database update routine will always query entries up to the maximum size of database 1, even if there are invalid entries.

- Entries above the maximum size of database 1 will then be scanned until an invalid entry is returned, at which point FabianLink32 will move on to the next database or device.

Note 1: To enable dynamic database updating the **PollServerDatabases** parameter in the **Configuration** section of the FBCSLINK.INI file should be set to **1**. By default it is **0**.

Note 2: For automatic updating of databases 2 through 9 the remote system needs to use CSI 1.18.07 and Fabian Server 1.11.11 or later.

Database updates will always be processed. What is configurable is whether there is an automated process to ensure the databases are synchronized from start-up.

2.8 Bi-Directional Links

The different ways in which Fabian Links are used gives rise to many configuration options. Some options interact and their behaviour changes as a result. Over the years some parameters have been added and others retired. One area that deserves explanation is that of the 'BiDirectionalLink' setting.

This setting was introduced to prevent a database howl-round when Fabian Links are used in both directions at the same time. Commands and revertives are unidirectional. Commands are passed from FabianLink32 to FabianServer32 and cannot pass in the other direction. Similarly revertives pass from FabianServer32 to FabianLink32 and do not pass the other way.

When database change commands were introduced into V2 BNCS they were neither commands nor revertives, but simply system messages. Each CSI on the network received them and where appropriate passed notification up to the client and driver applications it was hosting.

The appearance of a Router Modify (RM) message on the local network causes FabianLink32 to send an 'ApplCore Router Modify' (ARM) message to FabianServer32 which results in similar RM messages on the remote network. The remote FabianServer32 not only passes on the message from FabianLink32, but also receives subsequent notification from CSI of a database change. FabianServer32 does not intelligently track whether the change request originally came from FabianLink32 or from a client on its local network, so it passes the database change back to FabianLink32 as an 'ApplCore Database Change' (ADC) message. When FabianLink32 receives an ADC message it passes it on to the local network as a 'Router Database' (RD) message. Unlike the RM message which CSI sends to both clients and drivers, the RD is a message that CSI only sends to clients. Because FabianLink32 is a CSI client and only redirects commands it does not get sent RD messages. Therefore any possibility of the database change being passed, yet again, to FabianServer32 is avoided. If an RM message had been used instead of an RD message then this would have resulted in a database howlround.

If a link exists in both directions then there is a possibility that a howl-round can occur. Theoretically this can only occur if the device mapping is incorrect, in which case a command/revertive howl-round is also likely. However, a database howl-round was observed at a critical time and the cause was not known. The BiDirectionalLink parameter was, for expediency, added to 'break' the loop. The cause was not revisited. In the meantime fixes to device mapping have been applied which may well have cured the original problem. This parameter is therefore redundant and could be retired.

2.9 Active Link Switching

It is possible for two or more instances of FabianLink32 to be used to connect to remote servers. Regular communication between the FabianLink32 applications, using the FA, FI and FO messages (Active, Inquire and Object), enables them to decide which one has an active connection to a particular server. Only one can be active at a time.

When **CheckLinksByConnectionNumber=0** in the FBCSLINK.INI file the checks are performed using the IP address of the remote server to determine whether another instance of FabianLink32 has an active link to the same address.

When **CheckLinksByConnectionNumber=1** then the checks use the Fabian_Server_X section entry number where X is between 1 and NumberOfHosts. All instances of FabianLink32 must be set to check links by connection number. Using this method no two instances of FabianLink32 can have an active link using parameters from the same section.

Checking link status using the connection number is intended to cater for dual redundant links where two instances of FabianLink32 are involved in connections to two Fabian servers. Each [Fabian_Server_X] section in FBCSLINK.INI contains the details for a connection to a remote 'system'. This system might have a second, reserve server. Each [Fabian_Server_X] section has entries for two server addresses. FabianLink32 can have a valid connection with whichever of the two servers provides a connection. Checking active connections simply using the server address no longer works as the following example illustrates.

Consider two instances of FabianLink32, A and B. These have 'Address' and 'AltAddress' configured to connect to servers C and D. All is well if both FabianLink32s connect to either server C or server D, since the FA, FI and FO messaging will be comparing the same server address. But if FabianLink32 A connects to server C and FabianLink32 B connects to server D then the two FabianLink32s are comparing different addresses and both will go active, resulting in double commands and quadruple revertives. This happens because a command on the local system gets passed down each active link to the remote network and those commands will be actioned. This will result in two revertives. Both revertives will be passed back down both links, where each FabianLink32 will send two revertives to the local network. This will mean a total of four revertives on the local network from just one command.

Checking link status by comparing the connection number, ie the [Fabian_Server_X] section number, means there cannot be two active connections involving the servers 'Address' and 'AltAddress'.

This method in some ways supersedes comparing addresses, but is more complicated to understand and unnecessary when two FabianLink32 are only connecting to a single server.

2.9.1 FM - Fabian Message

As well as the FA, FO, FI and FC messages the application will also generate FM messages. This means Fabian Message and the contents of the message are purely for information. Other Fabian Links don't respond to them.

2.10 Active Device Switching

In addition to its own link switching mechanism FabianLink32 also employs the same TxRx/RxOnly switching scheme as normal BNCS drivers, using the NA, NI and NO messages (Active, Inquire and Object). Because FabianLink32 is masquerading as a real driver it must behave just like a real driver.

Normal drivers make decisions as to whether or not accept a request to go TxRx/RxOnly based on the status of the connection to the equipment they are driving. FabianLink32 behaves in a similar manner by basing its decision on the active/inactive state of its link for the device in question.

The basic rules are as follows

- If the link for a driver is **active** then FabianLink32 will not accept a request to go **RxOnly**.
- If the link for a driver is **inactive** then FabianLink32 will not accept a request to go **TxRx**.
- If a link is **inactive** and becomes **active** FabianLink32 will send messages on the local network requesting that it be allowed to go **TxRx** for all the devices configured for that link.
- If a link is **active** and becomes **inactive** FabianLink32 will send messages on the local network to say that all the devices configured for that link are closing.

2.11 Revertive Filtering

This mechanism offers protection to the local network from rapidly changing revertive information coming across a Fabian link. The filtering occurs after any device Id mapping has taken place.

If 'RevertiveFilterInfoDriver' in the 'Configuration' section of FBCSLINK.INI is set to a valid Id, or one is supplied as a command line override, then FabianLink32 will attempt to connect to an InfoDriver with that Id.

Each slot from 1 to 999 in the InfoDriver represents a device Id as it would be presented on the local network. The content of a slot is a delimited list of indices to filter. For example if slot 101 contains the string '73,199,1088', then any tallies from slots 73, 199 or 1088 from device 101 will be filtered and not passed immediately onto the local network.

Slots 1001 to 1999 also represent device Id's 1 to 999 and dictate how the filtering will be applied.

If a slot contains '-1' then no revertives will ever reach the local network for the indices in the list described above. Revertives for indices not in the list are presented to the network as normal. This feature allows a changing index to be silenced when either the rate of change or the nature of the data itself mean that it is of no benefit and should be discarded.

If the value in the slot is a positive integer then this is used as the number of seconds to inhibit any tallies from the filtered indices. For example if slot 1101 contains '5' then revertives from the filtered slots on device 101 would only be presented every 5 seconds. Revertives for indices not in the list are not subject to any delay.

2.12 Traffic Management of F-Series Messages

Prior to version 4.5, all FA and FI messages generated during the regular link check have been sent in individual network message packets.

It has been noted that where there are many links this will cause large amounts of network traffic and this appears to be able to interfere with reliable operation of the network.

From version 4.5 there are three new options that may be selected to alleviate aspects of this problem.

These options may be used individually or in any combination.

2.12.1 Compact Comments

Normally F-series messages are combined with explanatory comments.

Setting "CompactFMessageComments" will prevent this.

For example, with the full comment, an FA message will be something like :

"FA 101 192.168.1.42 4 – Link active from WS 101 connection 4 to TestStation (192.168.1.42)"

With the comment compacted the message will become:

"FA 101 192.168.1.42 4"

In isolation this will have no effect on message traffic as each message will still be sent in a separate network message. When used in conjunction with packing of messages, this option will allow more messages to be packed into a single network message.

This option affects all F messages, whether generated as part of the regular link check, or by other means.

2.12.2 Spread Responses

Normally all the FA and FI messages generated during the regular link check are sent in a group in very quick succession.

If there are a large number of connections this will cause a network traffic spike.

The spreading option will cause a delay between each of these messages being sent.

The parameter is "SpreadFMessageResponses_ms", and sets the delay, in milliseconds, between each message being sent.

The main link checking is done once per minute, so this should be set to a value such that all the messages will be sent before the next main link check starts.

2.12.3 Packed Messages

Normally each FA or FI message is sent in a network packet of its own.

The "PackFMessageResponses" setting instructs the program to put as many messages into one packet as possible, up to a maximum of 511 characters. This will reduce the total number of network messages sent.

Note that two characters have specific behaviour when this feature is used and should be avoided in server names.

2.12.3.1 Colon (:)

A colon (:) is used to separate the individual F-series messages from one-another.

2.12.3.2 Single Quote, or Apostrophe (')

A single quote, or apostrophe (') is interpreted by CSI as starting (or ending) a string within the message. If it is found singly within a server name CSI will ignore the next colon and hence the next F-message.

It is therefore advised that this character is not used in any Fabian Server name.

2.12.4 Spread Close Messages

Normally all the FC messages generated during close-down are sent in a group in very quick succession.

If there are a large number of connections this will cause a network traffic spike.

The spreading option will cause a delay between each of these messages being sent.

The parameter is "SpreadFCloseMessages_ms", and specifies the delay, in milliseconds, between each message being sent. A value of zero means no spreading; no additional delay – send the messages as quickly as possible.

2.12.5 Packed Close Messages

Normally each FC message is sent in a network packet of its own.

The "PackFCloseMessages" setting instructs the program to put as many messages into one packet as possible, up to a maximum of 511 characters. This will reduce the total number of network messages sent.

Note that the same two characters have specific behaviour when this feature is used and should be avoided in server names – See 2.12.3.1 Colon (:) and 2.12.3.2 Single Quote, or Apostrophe (').

2.13 Sample Ini

The following is based on part of an FBCSLINK.INI used during testing.

```
[Configuration]
DebugMode=1
Send RUTS=1
Show RUTS=1
InfoDriver=OBSOLETE
ActiveInfoDriver=951
RevertiveFilterInfoDriver=952
AlertOnError=0
QuitOnLinkFail=0
CacheEnable=1
UpdateDatabases=OBSOLETE
CheckLinksByConnectionNumber=0
AlwaysRegisterMaxIndices=1
DualLink=1
SynchronousMode=OBSOLETE
StatusInfoDriver=953
PollServerDatabases=0
```

DatabaseUpdateDelay=1000
ProxyAddress=testproxy.reith.bbc.co.uk
ProxyPort=23
ProxyPrompt=proxy->
ProxyCommand=c
LogMode=1
ConnectRetry_Seconds=15
NumberOfHosts=32
CompactFMessageComments=0
SpreadFMessageResponses_ms=50
PackFMessageResponses=0
SpreadFCloseMessages_ms=1000
PackFCloseMessages=0

[Fabian_Server_1]

Name=Richard's Laptop
Address=192.168.1.50
AltAddress=NONE
ProxyAddress=OBSOLETE
ProxyPrompt=OBSOLETE
ProxyCommand=OBSOLETE
Port=23
UserName=Richard
UserPass=Pass
ControlPass=PassCon
LogonExec=NONE
ExpectRUTreply=0
BNCSRevertiveMode=NORMAL
BiDirectionalLink=0
LinkActive=1
ConnectionOverride=0
LockTimeToRemoteSystem=1
Devices=16
Device_001=Example: Device_001=Device Id (ie GRD, GPID or ID), Local
Id, Remote Id
Device_002=ID,900,900,ALL,01
Device_003=ID,801,901,ALL,01
Device_004=ID,803,903,ALL,01
Device_005=ID,805,905,ALL,01

Device_006=ID,808,908,ALL,01
Device_007=ID,809,909,ALL,01
Device_008=ID,810,910,ALL,01
Device_009=ID,811,911,ALL,01
Device_010=ID,815,115,ALL,01
Device_011=GRD,050,050,ALL,01
Device_012=GRD,000,000,ALL,01
Device_013=GRD,000,000,ALL,01
Device_014=GRD,000,000,ALL,01
Device_015=GRD,000,000,ALL,01
Device_016=GRD,000,000,ALL,01
UseProxy=0
[Fabian_Server_2]
Name=The Garden Shed
Address=18.194.127.254
AltAddress=NONE
Port=3696
UserName=ShedUser
ProxyAddress=OBSOLETE
ProxyPrompt=OBSOLETE
ProxyCommand=OBSOLETE
UserPass=Password
ControlPass=Password
LogonExec=NONE
ExpectRUTreply=0
BNCSRevertiveMode=NORMAL
BiDirectionalLink=0
LinkActive=1
ConnectionOverride=0
LockTimeToRemoteSystem=0
Devices=16
Device_001=Example: Device_001=Device Id (ie GRD, GPID or ID), Local
Id, Remote Id
Device_002=GRD,123,123,ALL,01
Device_003=GRD,000,000,ALL,01
Device_004=GRD,000,000,ALL,01
Device_005=GRD,000,000,ALL,01
Device_006=GRD,000,000,ALL,01
Device_007=GRD,000,000,ALL,01

Device_008=GRD,000,000,ALL,01
Device_009=GRD,000,000,ALL,01
Device_010=GRD,000,000,ALL,01
Device_011=GRD,000,000,ALL,01
Device_012=GRD,000,000,ALL,01
Device_013=GRD,000,000,ALL,01
Device_014=GRD,000,000,ALL,01
Device_015=GRD,000,000,ALL,01
Device_016=GRD,000,000,ALL,01
UseProxy=1
[Fabian_Server_3]
Name= Local Radio & TV
Address=19.27.280.3
AltAddress=NONE
Port=23
UserName=User
UserPass=Password
ControlPass=Password
LogonExec=NONE
ExpectRUTreply=0
UseProxy=0
BNCSRevertiveMode=NORMAL
BiDirectionalLink=0
LinkActive=1
ConnectionOverride=0
LockTimeToRemoteSystem=0
Devices=1
Device_001=ID,401,401,NONE,ALL
[Fabian_Server_4]
Name=NONE
Address=NONE
AltAddress=NONE
ProxyAddress=OBSOLETE
ProxyPrompt=OBSOLETE
ProxyCommand=OBSOLETE
Port=23
UserName=User
UserPass=Password
ControlPass=Password

```
LogonExec=NONE
ExpectRUTreply=0
BNCSRevertiveMode=NORMAL
BiDirectionalLink=0
LinkActive=0
ConnectionOverride=0
LockTimeToRemoteSystem=0
Devices=16
Device_001=Example: Device_001=Device Id (ie GRD, GPID or ID), Local
Id, Remote Id
Device_002=GRD,000,000,ALL,01
Device_003=GRD,000,000,ALL,01
Device_004=GRD,000,000,ALL,01
Device_005=GRD,000,000,ALL,01
Device_006=GRD,000,000,ALL,01
Device_007=GRD,000,000,ALL,01
Device_008=GRD,000,000,ALL,01
Device_009=GRD,000,000,ALL,01
Device_010=GRD,000,000,ALL,01
Device_011=GRD,000,000,ALL,01
Device_012=GRD,000,000,ALL,01
Device_013=GRD,000,000,ALL,01
Device_014=GRD,000,000,ALL,01
Device_015=GRD,000,000,ALL,01
Device_016=GRD,000,000,ALL,01
UseProxy=0
```

3 Program GUI

When running the program looks like Figure 1: FabianLink32.

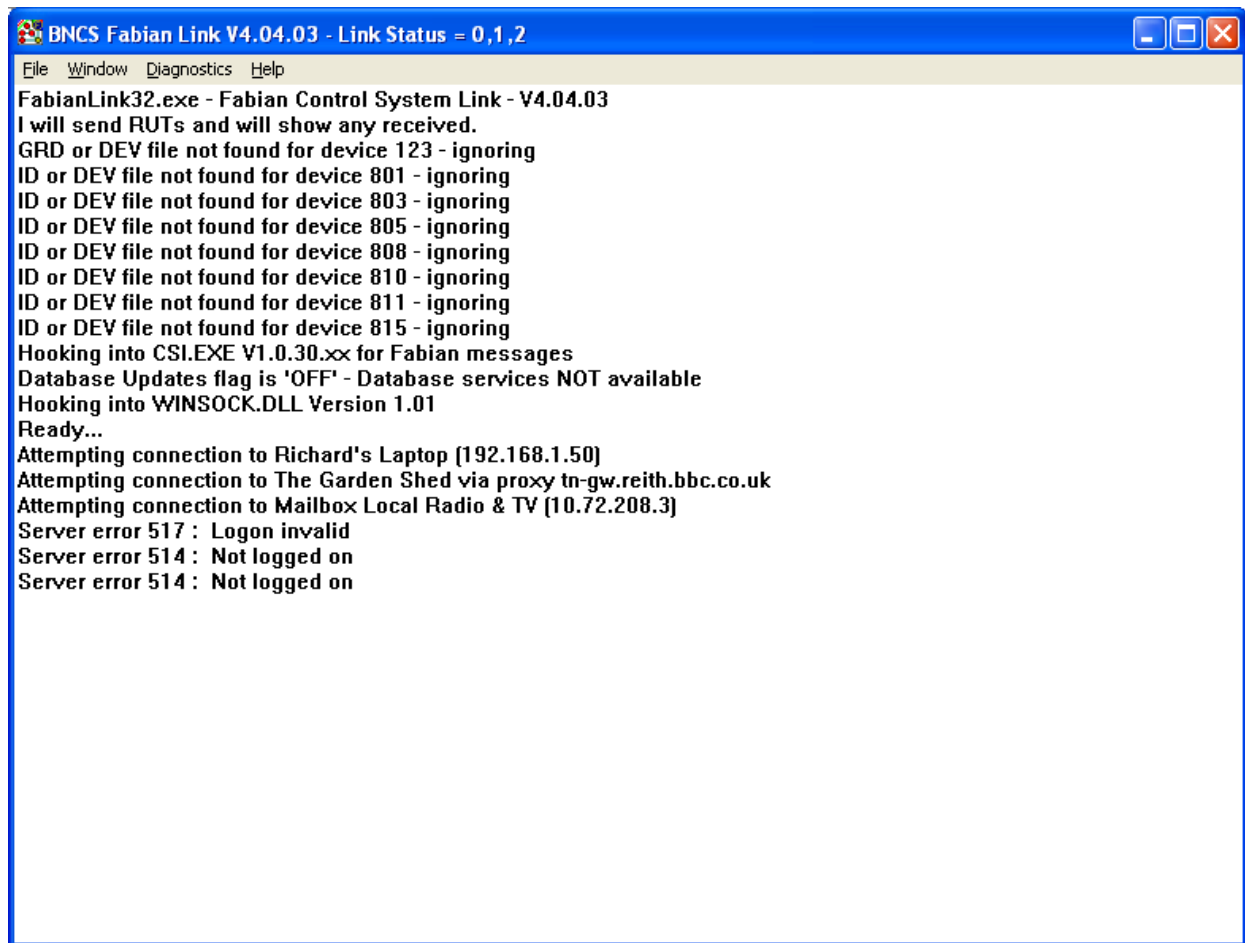


Figure 1: FabianLink32

There are a few controls available from the menu.

3.1 File Menu

3.1.1 Exit

This menu item terminates the program.

3.2 Window Menu

3.2.1 Clear

This menu item clears the screen.

3.3 Diagnostics Menu

3.3.1 On

This menu item is a toggle button which turns on or off writing of messages to the screen.

3.3.2 Send RUTs

Selects whether to send RUT (aRe yoU There) messages.

3.3.3 Show RUTs

Selects whether to show any RUT (aRe yoU There) messages sent or received.

3.3.4 Logging

This menu item is a toggle button which turns on or off writing of messages to the log-file.

3.4 Help Menu

3.4.1 About

This menu item displays a dialog showing information about the instance of the application running – version number, build date and time, author, etc. Press the OK button to close it.

3.5 Title Line

The title line includes a "Link Status" string. In the above example it is "0,1,2".

This represents the connection status of all the configured servers. The numbers are connection status values as listed above in the StatusInfoDriver description under section 2.1.1 [Configuration] and section 2.5 Status InfoDriver.

They are displayed up to the highest non-zero value. In the above example there may be servers configured above number 3, but any servers above server 3 are not connected.

4 Resilience and redundancy

The Fabian Link/Server system is able to support a variety of resilience options.

See sections 2.9 Active Link Switching and 0 Active Device Switching for details.

5 Logging

The driver will write debug messages to its list box, if enabled. This is controlled by the Debug|On item in the menu, and the DebugMode entry in the dev_nnn.ini file.

It will also write the messages to a log-file, if enabled. This is controlled by the Log|Enabled item on the menu. See section 1.3 BNCS configuration for the location.

The file is called yyyyymmdd_FBLink.log, where yyyy is the year, mm is the month and dd the day of the month.

6 Documents referenced

FabianServer32 documentation.

7 Notes

There are some "secret" parameters allowed in the FBCSLINK.INI file.

7.1.1 [Configuration]

Item	Value	Comment
FilterDebug	0 or 1	If this is set to '1' FabianLink32 will produce extra diagnostic output to show the operation of the Revertive Filtering system. It is not anticipated that this will be of any use in normal operation – it's just for tracing certain problems with filtering.
PerSessionLogging	0 or 1	If this is set to '1' and V4 locations are being used then the log-file will be written in a folder within the 'logs' folder named according to the date and time at which FabianLink32 was started. For example : FBLink-2014-06-05-14-29-41 Hence every run of the application will cause a new log file to be created.
TimeMessagePeriod_minutes	Integer	If this is present and set to a positive integer then the system will generate an FM message announcing the current date/time at the given interval, in minutes.

These parameters are "secret" in the sense that no default value is written into the dev.ini file if they are not present. They are only there if manually added.

8 Version history

8.1 Software Version

Version No	Date	Details	Name
4.04.03	14 Oct. 09	Version current at time of generation of this document.	Richard Kerry
4.04.07	22/7/2010	Fixes occasional crash due too late callback during closedown.	Richard Kerry
4.04.08	12/8/2010	Fixes issue using external InfoDriver to control revertive filtering.	Richard Kerry
4.04.09		Prevented stream of filter-related revertives during startup.	Richard Kerry
4.04.10	08/09/2010	Arranged for packet-mode to work.	Richard Kerry

4.04.11	19/10/2010	Increased maximum possible number of connections and made actual number configurable.	Richard Kerry
4.04.12	09/12/2010	Fixed packet-mode revertives. Fixed use of V4 folder structure. Added configurable delay before retrying connect after a failure. Still immediate on start-up or when Activated. Don't immediately re-queue on disconnection if link is not Active.	Richard Kerry
4.04.13	18/01/2011	Warning if F message arrived but Dual-link not set. F message comments now consistent – always address and name.	Richard Kerry
4.05.01	12/05/2011	Handling for reducing F-message traffic. Packing several into one message packet. Selectively not including the comment. Spreading the messages in time. Database handling, if always-register-max is set no longer needs dev.ini files to be present.	Richard Kerry
4.05.02	19/05/2011	Fixed crash on revertive received if no filter ID set. Enlarged F-message packing size to 511.	Richard Kerry

4.06.01	10/10/2011	<p>Added poll-on-connect system</p> <p>Fixed implementation of packet-mode revertive handling.</p> <p>GPI driver connections are now set-up internally with the size from the DatabaseSize_1 entry. This is a change from before when 4096 was always used. It is possible this will affect existing users.</p> <p>It affects (for GPI devices only):</p> <ul style="list-style-type: none"> • Registration sizes if AlwaysRegisterMax is not used. • The ADP command. • Possibly some database operations. • Any command handling that needs to check poll limits: GL, GS, GP, IF, IH, IL, IW, IP, RC, RF, RL, RM, RP. <p>Note that the previous fixed limit of 4096 applied from version 2.3.1.</p> <p>Prior to 2.3.1 the limit was 1280.</p> <p>Prior to 2.1.4 the limit was 582.</p> <p>Prior to this version the registration was always done using the ARR command, irrespective of the type of the device. From this version the correct registration command will be used, ARR, AIR or AGR according to the device type configured. The poll is also done using the correct command: ARP, AIP or AGP.</p> <p>InfoDriver revertive caching no longer uses files.</p> <p>Prior to this version if a Poll command was received which referred to a mixture of cached and non-cached data, a poll for the whole range of the data would be sent to the Server. From this version the server will only be polled for data that is not available from the cache.</p>	Richard Kerry
4.06.02	26/01/2012	Bug fixes – registering max, or device type mismatch in configuration.	Richard Kerry
4.06.03	16/05/2012	<p>Extended poll-on-connect to allow per-server and per-device configuration.</p> <p>Removed length restriction on passwords and user names. Let the Server enforce them.</p>	Richard Kerry
4.06.04	17/05/2012	Fix crash if no poll-on-connect configured.	Richard Kerry
4.06.05	24/05/2012	Fix crash on state change.	Richard Kerry

4.06.06	19/06/2012	Added diagnostics to help track down crash starting the name database scan on certain systems.	Richard Kerry
4.06.07	21/06/2012	Fixed crash starting the name database scan on certain systems. Fixed crash when poll command received doesn't match the type that the device is configured as.	Richard Kerry
4.06.08 mistakenly also marked as 4.6.7	23/07/2012	Added FC message blocking and spreading. Fixed some loop coverage issues, eg stopping all hosts being checked during F-message handling. Fixed multiple instance check on startup. Fixed Active and Status ID handling to allow any address to be polled.	Richard Kerry
4.06.09	20/08/2012	Version number changed to avoid any confusion due to duplication of previous number.	Richard Kerry
4.07.01	10/12/2012	Fixed problem of clipping of long strings in revertives from InfoDriver slots. Changed to DLL messaging.	Richard Kerry
4.07.02	23/04/2013	Fixed fault where instance failed to become Active when partner disappeared without sending FC, in "spreading" mode.	Richard Kerry
4.07.03 and 4.07.04	08/10/2013	Fix to device::get() to ensure that out of range device number doesn't return crap - previously it would flag the issue and carry on regardless returning something from an array with a negative index	Dave Yates
4.07.05	06/02/2014	Some additional diagnostic messages to aid diagnostics of get() range issues.	Richard Kerry
4.07.06	05/06/2014	Addresses issues with diagnostic messages when '%' characters are encountered. Adds configuration of link check interval. Improvements to handling of Tx/Rx status. Now lists drivers as TxRx when connected, RxOnly if Inactive, RxBroken if no connection. Adds option for per-session log files.	Richard Kerry
4.07.07	10/06/2014	Further fixes to issues with diagnostic messages when '%' characters are encountered.	Richard Kerry
4.07.08	14/08/2014	Fixes crash when AlwaysRegisterMaxIndices is set.	Richard Kerry

4.07.09	18/11/2014	<p>Where an error condition arises which is signalled using a message box, the message box now has properties set so that it will always be on the top of all other dialogs.</p> <p>Fixed an issue whereby when a value was written to an Active InfoDriver slot to swap the addresses, or force use of one, a revertive would be issued from the wrong slot in the address indication range (starting at 2000).</p> <p>Some uses of C strings changed to STL strings to reduce opportunities for buffer addressing related problems.</p>	Richard Kerry
4.07.10	15/12/2014	<p>Further uses of C strings changed to STL strings to reduce opportunities for buffer addressing related problems.</p> <p>Due to certain instances of smaller buffers having being found and replaced during this process it is believed that this might fix the "close on crash" issue.</p> <p>Added option to get an FM message issued showing the current time/date at a specified interval.</p>	Richard Kerry
4.07.11	25/02/2015	<p>Further uses of C strings changed to STL strings to reduce opportunities for buffer addressing related problems.</p> <p>In particular this fixes the situation where a very long string (more than 1080 bytes) sent in via the TCP port would cause a crash.</p> <p>Note that the protocol works based on a series of bytes separated by CRs (ie value 13 (d)). It will not respond if a command is buried in a longer string, only if it starts straight after the CR.</p>	Richard Kerry
4.07.12	04/02/2016	<p>Added extra controls of the handling of device and database sizes. This requirement has arisen due to unforeseen interaction between the database sizes used by database commands and the device size used for checking during handling of Write and Poll commands.</p>	Richard Kerry

4.07.13	08/02/2017	<p>If AltAddress is blank for any server it is ignored and will not be used (ie no automatic swapping between Main and Alternate will be done). This has always also been disabled if it is set to "NONE".</p> <p>When '0' is written to the Active infoDriver to disconnect an Active instance, an FC is now sent out to kick any other instance into becoming Active immediately.</p>	Richard Kerry
4.07.14	Not released		
4.07.15	20/02/2017	<p>Fixed buffer overflow relating to generation of F messages, which was causing problems in Huxley.</p> <p>Gist is (in void CsiSup::FMessageResponder::AccumulateMessage): if (iAllLen + iThisLen > MAX_F_MSG_LEN) becomes if ((iAllLen + iThisLen) >= MAX_F_MSG_LEN) otherwise string gets to be 512 bytes long which when sent by string-version of the message sender has a term added and becomes 513 bytes which then doesn't get trapped by CSI</p>	David Yates
4.07.16	20/03/2017	<p>Fixed so that Active Address reporting is as documented – 1 is main address, 2 is alt address.</p> <p>Reviewed use of various buffers and strings to reduce possibility of overruns, or attempt to give notice if a problem is found while running.</p> <p>Fixed issue whereby Send/Show RUTs settings were always written to fbcslink.ini within C:\Windows. Should have been the one used for everything else, which it now is.</p>	Richard Kerry
4.07.17	14/12/2017	Fixed fault in window caption. Should show 'V' indicating a Release version but was showing another character.	Richard Kerry

8.2 Document version

Version No	Date	Details	Name
	Feb 09	Updated template	A Atkin
	29 Oct. 09	First version using revised document template.	Richard Kerry

	22/7/2010	Added note on shutdown on CSI closing.	Richard Kerry
	08/09/2010	Minor revisions following fixes to packet-mode.	Richard Kerry
	19/10/2010	Revised following changes to allowed number of connections.	Richard Kerry
	09/12/2010	Removed Duplicate mention of LinkActive parameter	Richard Kerry
	10/10/2011	Updated to match changes to software.	Richard Kerry
	16/05/2012	Updated to match changes to software. Changed branding following change from Siemens to Atos.	Richard Kerry
	20/03/2013	Clarified operation of database update configuration.	Richard Kerry
	16/02/2016	Clarification relating to Device Options settings and database polling.	Richard Kerry
	08/02/2017	Added note on the meaning of FM (Fabian Message) messages.	Richard Kerry

Atos IT Services Limited
4 Triton Square
Regent's Place
London NW1 3HG, UK
<http://uk.atos.net/>

BNCS
4 Triton Square
Regent's Place
London NW1 3HG, UK
collediacontrol.it-solutions.gb@atos.net