

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

**FabianServer32.exe**

Written by: Simon Dowson, Richard Kerry.

# Contents

<b>Contents .....</b>	<b>2</b>
<b>1 Overview .....</b>	<b>6</b>
1.1 Description .....	6
1.2 Terminology .....	6
<b>2 Server Overview .....</b>	<b>6</b>
2.1 TCP/IP Link .....	6
2.2 Database Engine.....	6
2.3 Resource Manager .....	6
2.4 Message Handler .....	7
2.5 Mirror Servers .....	7
2.6 Device firmware version.....	7
2.7 BNCS configuration .....	7
2.7.1 The v1/v2 Location.....	7
2.7.2 The V3 Location .....	7
2.7.3 The V4 Location .....	7
2.7.4 Log-file Locations.....	7
2.7.5 The “Daily Log” .....	8
2.8 Driver compatibility .....	8
2.9 CSI version.....	8
2.10 Messaging System – BncsIf32.dll. ....	8
2.11 Driver PC requirements.....	9
2.12 Resilience and Redundancy .....	9
2.13 Device interface description.....	9
2.14 Device limitations .....	9
2.15 Device setup.....	9
2.16 Status Reporting InfoDriver .....	9
<b>3 Driver setup .....</b>	<b>9</b>
<b>4 Server Configuration .....</b>	<b>10</b>
4.1 Setting Up .....	10
4.1.1 Status InfoDriver Configuration .....	10
4.2 Ini file settings .....	10
4.2.1 The FBServer.ini file .....	10
4.2.2 [Configuration] Section .....	10
4.2.3 [User_nnn] Section .....	14
4.2.4 Access Control. Basic syntax. ....	17
4.2.5 [Database] Section .....	18
4.2.6 [Mirror] Section .....	19
<b>5 Resource Files.....</b>	<b>20</b>

5.1	Directory PATH listing .....	20
<b>6</b>	<b>Server Access .....</b>	<b>21</b>
6.1	Dedicated clients .....	21
6.2	Raw TCP Interface .....	21
6.3	Establishing a TCP Session .....	21
6.3.1	Logging In .....	21
<b>7</b>	<b>Command Usage .....</b>	<b>22</b>
7.1	Database Manipulation.....	22
7.2	Resource Manipulation.....	23
7.3	Messaging .....	25
<b>8</b>	<b>System Notification Messages.....</b>	<b>25</b>
8.1	Notification Overview .....	25
8.2	Response to SNM's.....	25
8.3	Database Booking Change .....	26
8.4	Resource List Change .....	26
8.5	ApplCore Database Change .....	26
<b>9</b>	<b>Control System Interface .....</b>	<b>26</b>
9.1	Control Overview .....	26
9.2	Sample Ini.....	26
<b>10</b>	<b>Infodriver .....</b>	<b>29</b>
10.1	Status Infodriver .....	29
10.1.1	Overall Status.....	29
10.1.2	Connection Status .....	29
10.1.3	Updates and Revertives .....	30
<b>11</b>	<b>Filtering of BNCS Access.....</b>	<b>30</b>
11.1	Filtering .....	30
11.2	Address Filter Configuration .....	31
11.2.1	Wildcard Address Filtering .....	32
11.2.1	Control of parameters – None, blank and comment .....	32
11.3	Data Filtering.....	32
11.4	Data Types .....	33
11.4.1	String Parameters.....	33
11.4.2	Integer Parameters.....	34
11.4.3	Real Parameters .....	34
11.4.4	Floating-point resolution. ....	35
11.4.5	Multiple Parameters .....	36
11.4.6	Examples .....	36
11.5	The Get System Access Command (GSA).....	38
<b>12</b>	<b>Named Filters .....</b>	<b>38</b>
12.1	Filter Sections .....	38
12.2	[Filter_<name>] .....	39

12.2.1	Absolute Filters .....	39
12.2.2	Relative Filters .....	40
12.2.3	Offsettable Filters .....	40
<b>13</b>	<b>Named Data Filters .....</b>	<b>41</b>
13.1	Compact or Extended Syntax.....	42
13.2	Compact.....	42
13.3	Integer.....	43
13.3.1	There is only one set .....	43
13.3.2	Numbered keys .....	44
13.4	Real .....	44
13.5	String .....	45
13.6	Examples .....	46
<b>14</b>	<b>Driver GUI.....</b>	<b>48</b>
14.1	File Menu .....	49
14.1.1	Exit.....	49
14.2	Users Menu .....	49
14.2.1	Manage Users... ..	49
14.3	Diagnostics Menu .....	50
14.3.1	On .....	50
14.3.2	Show RUTs .....	50
14.3.3	Logging .....	50
14.3.4	Show Revertives Received .....	50
14.3.5	Show Revertives Sent .....	51
14.3.6	Show Commands Received.....	51
14.4	Window Menu .....	51
14.4.1	Clear .....	51
14.5	Help Menu .....	51
14.5.1	About .....	51
<b>15</b>	<b>Resilience and Redundancy .....</b>	<b>51</b>
<b>16</b>	<b>Logging .....</b>	<b>51</b>
16.1	Standard Log .....	51
16.2	Fabian Server Specific Log (Daily Log) .....	52
<b>17</b>	<b>Regular Expression Syntax .....</b>	<b>52</b>
17.1	Wildcard .....	52
17.2	Anchors .....	52
17.3	Marked sub-expressions.....	52
17.4	Non-marking grouping .....	52
17.5	Repeats .....	52
17.6	Non greedy repeats.....	54
17.7	Back references.....	54
17.8	Alternation.....	54
17.9	Character sets .....	55

17.9.1	Single characters .....	55
17.9.2	Character ranges .....	55
17.9.3	Negation .....	55
17.9.4	Character classes .....	55
17.9.5	Collating Elements .....	55
17.9.6	Equivalence classes .....	56
17.9.7	Escapes .....	56
17.9.8	Combinations .....	56
<b>17.10</b>	<b>Escapes .....</b>	<b>56</b>
17.10.1	Escapes matching a specific character .....	56
17.10.2	"Single character" character classes .....	57
17.10.3	Character Properties .....	57
17.10.4	Word Boundaries .....	58
17.10.5	Buffer boundaries .....	58
17.10.6	Continuation Escape .....	58
17.10.7	Quoting escape .....	58
17.10.8	Unicode escapes .....	58
17.10.9	Any other escape .....	59
<b>17.11</b>	<b>Perl Extended Patterns .....</b>	<b>59</b>
17.11.1	Comments .....	59
17.11.2	Modifiers .....	59
17.11.3	Non-marking grouping .....	59
17.11.4	Lookahead .....	59
17.11.5	Lookbehind .....	59
17.11.6	Independent sub-expressions .....	60
17.11.7	Conditional Expressions .....	60
<b>17.12</b>	<b>Operator precedence .....</b>	<b>60</b>
<b>17.13</b>	<b>What gets matched .....</b>	<b>60</b>
<b>17.14</b>	<b>Modifiers .....</b>	<b>61</b>
<b>18</b>	<b>Documents referenced .....</b>	<b>61</b>
<b>19</b>	<b>Notes .....</b>	<b>62</b>
19.1.1	[Configuration] .....	62
<b>20</b>	<b>Version history .....</b>	<b>62</b>
<b>20.1</b>	<b>Software Version .....</b>	<b>62</b>
<b>20.2</b>	<b>Document Version .....</b>	<b>66</b>

# 1 Overview

## 1.1 Description

The name FABIAN comes from its original intended function as a **F**ast **A**ccess **B**ooking **I**nformation **A**nd **N**otification **S**ystem supplying regional booking information and real time messaging facilities to assist in the distribution of programme material. Its main purpose now is to act as a means of linking BNCS control system networks.

## 1.2 Terminology

Throughout this section of the document :- **<CR>** means press carriage return. Text in ***bold italics*** or **bold** is either output or input from the user screen

# 2 Server Overview

The FABIAN Server is the hub of the system and performs 4 basic functions.

TCP/IP link

Database Engine

Resource Manager

Message Handler

## 2.1 TCP/IP Link

The server can manage clients wishing to access to the BNCS network (cluster) from another BNCS cluster or simple terminal emulation program. Device Id mapping can be used to prevent clashes between local and remote devices.

Filtering using addresses or data contents is available to control the access available to any client.

## 2.2 Database Engine

The database engine performs basic storage and retrieval of database records and fields. It has a capacity to store and manipulate a whole year's bookings and associated information. There can be up to 999 bookings per day with up to 99 amendments per booking. All amendments are retained.

A database record consists of 128 character fields. Each field can be up to 128 characters in length.

In addition to the fixed fields the database engine can store and retrieve text files up to 32k in length.

## 2.3 Resource Manager

The resource manager is a system for managing up to 26000 named resources and allocating them to bookings. Possible conflicts between resources and bookings can be prevented. Temporary resources are allowed.

## 2.4 Message Handler

A client can subscribe to up to 99 message notification groups. These are communication channels for inter-client messaging. Some channels are dedicated to particular functions such as AMS data, others are entirely user configured.

## 2.5 Mirror Servers

A server can optionally be set up to mirror another server. It logs on to the remote server and receives database change information which it uses to interrogate the remote machine. The local server overwrites bookings in the local database or inserts them at a predefined offset. The configuration of the mirroring process is detailed in the a later section.

## 2.6 Device firmware version

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

FabianLink32

FabianExec32 – V4.05.04

FabianScheduler32 – V4.05.07

FabianSunLibrarian32 – V4.05.04

Though it may be assumed to work with various other versions, both older and newer. And with the corresponding v3 applications.

## 2.7 BNCS configuration

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

### 2.7.1 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.

Debug log files are written in **C:\bnclslogs**.

### 2.7.2 The V3 Location

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

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

### 2.7.3 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**.

### 2.7.4 Log-file Locations

Log files may be written in folders below the folder names listed above.

If MayCreateLogFolders is set then any missing folders within these paths will be created automatically.

#### **2.7.4.1 All Logs in Named Folder**

If AllLogsInNamedFolder is set the log files will be written in a folder called FBServer, within the folder listed above.

#### **2.7.4.2 Per-Session Logging**

If PerSessionLogging is set the log file will be written in a folder called **FBServer-yyyy-mm-dd-hh-mm-ss** within the folder listed above.

Where yyyy-mm-dd-hh-mm-ss is the date and time at which the program was started.

#### **2.7.4.3 Log File per Hour**

If LogFilePerHour is not configured, log files will be named `yyyymmdd_FBServer.log`.

If LogFilePerHour is configured, log files will be named `yyyymmddhh_FBServer.log`.

Where yyyy is the current year, mm is the month, dd is the day and hh is the hour.

#### **2.7.5 The "Daily Log"**

The "daily log" file was present in Fabian software before the current scheme for diagnostic messages and logging was added. The files are also used by some other applications, eg FabianExec. Therefore it is still present and its configuration is separate from the debug log file.

The daily log file is written in the folder configured by the [Configuration]DataPath key in the fbserver.ini file.

It is named `yyyymmdd.txt`, where yyyy is the current year, mm is the month and dd is the day.

### **2.8 Driver compatibility**

FabianServer32 has no direct (or external) connection with any BNCS Drivers.

It can communicate with Drivers (InfoDrivers, router or GPI drivers) as a BNCS client.

No information has been preserved about compatibility with any drivers though it is assumed that it is able to communicate with all drivers supporting BNCS client control.

### **2.9 CSI version**

Known to work with the following:

Csi32 – V1.0.30, V1.1.18, V1.1.27, V1.2.48, V1.2.60

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.

### **2.10 Messaging System – BncsIf32.dll.**

Versions up to 4.7 used WM\_COPYDATA for communicating with CSI.



Versions from 4.8 onward use BBC\_COPYDATA, also known as "DLL messaging", via the BncsIf32.dll library.

In the course of changing to DLL messaging a flaw was found with this library, necessitating a new version being produced. FabianServer32 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.

## **2.11 Driver PC requirements**

There are no specific requirements other than for standard BNCS.

## **2.12 Resilience and Redundancy**

The Fabian Link/Server system is able to support a variety of resilience options. See section 15 Resilience and Redundancy for details.

## **2.13 Device interface description**

Communication with any client is done using over a TCP/IP network connection.

## **2.14 Device limitations**

There are no known limitations.

## **2.15 Device setup**

As an automation program there is no device set-up required for FabianServer32.

## **2.16 Status Reporting InfoDriver**

FabianServer32 may be run with an external InfoDriver to indicate its status and the status of any clients connected.

See Section 10.1 Status Infodriver for details.

# **3 Driver setup**

FabianServer32 controls BNCS Drivers as a BNCS Client.

If this is required then CSI must be running before FabianServer is started. Also BNCSAccess must be set in the ini file.

FabianServer32 may be used for resource or booking control without BNCS networking being available. In that case this parameter need not be set.

# 4 Server Configuration

## 4.1 Setting Up

Assign TCP/IP protocol to your Network Card and specify an IP address. (This will have to be given by your system coordinator)

If you want the server to act as a TCP/IP BNCS link you may (if you're using NBF rather than NBT) need to install a NetBIOS transport protocol stack in addition to the TCP/IP.

It is a good idea to set up a client machine with the above software at this time so that you can test the basic connectivity of the network using the TCP/IP 'ping' utility.

When you are happy that the machines are configured correctly and are able to 'ping' each machine from the other you can then proceed with the rest of the installation.

Install BNCS on the PC you wish to run FabianServer32 on. Locate FabianServer32.exe. Run CSI and then run FabianServer32 which will then create FBserver.ini file in the config directory (See section 2.7 BNCS configuration). You will now need to edit FBServer.ini, as detailed below.

### 4.1.1 Status InfoDriver Configuration

If a Status reporting InfoDriver is to be used, the appropriate InfoDriver needs to be running before FabianServer32 is started. When FabianServer32.exe is run the device number of the InfoDriver must be provided as a command-line parameter.

## 4.2 Ini file settings

The configuration file for FabianServer32 is called FBSERVER.INI.

See section 2.7 BNCS configuration for details of its location.

### 4.2.1 The FBServer.ini file

The INI file is split into several sections.

### 4.2.2 [Configuration] Section

The Configuration section contains general configuration information :-

Item	Default Value	Comment
ServerName	Anonymous	Server name - displayed by any client or TCP sessions e.g. Exeter Local Radio

Item	Default Value	Comment
ServerPort	23	<p>This is the TCP/IP port that clients must use in order talk to the server. A PC may have several TCP/IP services of different types running and each one has its own unique port number. The Fabian server uses port 23 by default, which is also the Telnet port number. If this is already in use on the PC you wish to run the Fabian server on then a good alternative is port 401.</p> <p>It is also possible to supply a connection port number only as part of the User specification. In this case the global port number may be set to -1.</p>
LoginRequired	1	If set to '0' no username or password is required.
BookingDatabaseEnable	0	If set to '1' the booking system engine is enabled. By default it is set to '0'
DataPath	C:	The path to the "daily" log files and files used for database and resource handling.
CloseWithCSI	1	Closes when CSI closes
DisableCSICache	1	Forces CSI cache off so that poll requests are always routed on to the network.
EnableLogging	1	<p>Writes daily log files of link activity.</p> <p>This is not the same as the log mode enabled using LogMode. Files are written in the folder selected by DataFile. Files are named yyyyymmdd.txt, where yyyy is the current year, mm is the month and dd the day.</p>
DebugMode	0	<p>When set to '1' the server displays more verbose information about its activities.</p> <p>This flag is toggled by the 'Diagnostics' option on the main menu.</p>

Item	Default Value	Comment
LogMode	0	<p>When set to '1' the server writes more verbose information about its activities to a log file.</p> <p>See 2.7 BNCS configuration for the location of this file.</p> <p>This flag is toggled by the 'Logging' option on the main menu.</p>
DebugShowRevertivesReceived	0	Messages specifically about each revertive received from CSI are written to the screen and/or log-file according to the Debug and Log Mode settings.
DebugShowRevertivesSent	0	Messages specifically about each revertive sent to the TCP clients are written to the screen and/or log-file according to the Debug and Log Mode settings.
DebugShowCommandsReceived	0	Messages specifically about each command received from a client are written to the screen and/or log-file according to the Debug and Log Mode settings.
Show RUTS	0	<p>When set to '1' the incoming 'aRe yoU There' messages from clients will be displayed in the diagnostic information. This is useful to see if clients are configured to send RUT messages.</p>
SystemCode	0	<p>A numeric system password to allow remote supervisor access to certain system commands.</p> <p>This is currently used to restrict access to the SSN and SST commands, for setting the system name and date/time.</p>
BNCSAccess	1	If set to '1' the server will act as a BNCS control system host and allow 'ApplCore' style commands to be used. Note : A second password is required via the SUC command in order gain access and CSI must be running prior to running the server.
SchedNotifyGroup	1	Used with Fabian Scheduler to set schedule group notification of changes.

Item	Default Value	Comment
MaximumUser	16 (default) 64 (maximum)	Maximum number of user configuration sections.  These will be populated with default values at start-up.  This refers to the number of different users which may be configured at any time. The number of client connections that may be active at any time is not necessarily the same, though at the moment it is 64.
DefaultWriteFilters	4 (default)	Number of DevWriteAccess_nn lines populated by default.
DefaultReadFilters	4 (default)	Number of DevReadAccess_nn lines populated by default.
DefaultCombinedFilters	2 (default)	Number of DevAccess_nn lines populated by default.
LoggingMillisecondDisplay	0	Enables inclusion of milliseconds in the times written to the daily log file (see EnableLogging).
AllLogsInNamedFolder	0	If this is set then all log files are created within a folder called "FBServer", within the main logging folder.  See Section 2.7 BNCS configuration
LogFilePerHour	0	If this is set then the name used for any log file will include digits representing the hour.  If it is not set the name will only include the date.
MayCreateLogFolders	1	Enables creation of any folder required by the value of DataPath, or by the selected debugging log-file location, if it doesn't exist when the application starts.

There are some "secret" parameters available for use in the Configuration section.

These parameters are "secret" only in the sense that they do not get written to the file if not already present. They are only there if manually added.

Item	Value	Comment
PerSessionLogging	0 or 1	If this is set to '1' and V4 locations are

		<p>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 FabianServer32 was started.</p> <p>For example : FBServer-2014-06-05-16-29-26</p> <p>Hence every run of the application will cause a new log file to be created.</p> <p>It is envisaged that this setting will only be of use during development.</p>
--	--	--

#### 4.2.3 [User\_nnn] Section

The User sections contain user details for up to 64 users or automated clients. The sections are named User\_001 to User\_064. The actual number in use is set by the [Configuration]MaximumUser setting. This also determines the number of sections that will have default values written during start-up.

Each of the sections has the same contents :-

Item	Default Value	Comment
UserName	---	<p>User name.</p> <p>This may be up to 24 characters.</p> <p>Note that the user-name and passwords are not case-sensitive.</p>
UserPass	---	<p>User password.</p> <p>This may be up to 24 characters.</p> <p>Note that the user-name and passwords are not case-sensitive.</p>
ControlPass	---	<p>Control password.</p> <p>This may be up to 24 characters.</p> <p>Note that the user-name and passwords are not case-sensitive.</p>
BNCSAccess	<p>READONLY</p> <p>WRITEFILTERED</p> <p>WRITEALL</p> <p>Or other values as listed in Section 11.1 Filtering</p>	<p>May be READONLY, WRITEFILTERED or WRITEALL or other values as specified in Section 11.1 Filtering.</p> <p>If WRITEFILTERED is specified, the device access table below is used to specify the device(s) &amp; index range to be controlled.</p> <p>If WRITEALL is specified then read and write(Control) access is given to all indices on all devices on the network</p> <p>READALL gives read access to all indices but no write access.</p>

Item	Default Value	Comment
Fields	1,2,3,4,5	Fields=1,2,3,4,5 The Server database entries this user wishes to use, in order - see Database below.
<i>SendRUTs</i>	<i>OBSOLETE</i>	<i>Sends RUT's (aRe yOu There) to clients every 60 seconds</i>
ExpectRUTreply	0	Expects RUT replies from clients - Disables the link if replies are lost - set to 0 for a BNCSlink client login.
WorkstationOverride	0	The default is '0', which means the workstation number appended to command packets on the network will be the workstation Id from CSI.INI. If this parameter is set then it will override the CSI.INI value
AllowDatabaseChanges	1	The default is '1'. If set to '0' then the server will reject and Router modify commands from this client.
UserServerPort	-1	If a different port number is required for connections by this user then it should be entered here.  If this is -1, the port set by [Configuration]UserPort will be used.
UserActive	1	A flag to enable or disable this user from log-ins.  Also controls whether the port corresponding UserServerPort is considered for opening.
DevWriteAccess_01	NONE	Table of devices and indices that may be written to if Filtered write access is selected.  See Section 4.2.4 Access Control. Basic syntax. for full details of syntax.  Also Section 11 Filtering of BNCS Access for details of the extended filtering available.
DevWriteAccess_02	NONE	
etc ..up to DevWriteAccess_16	etc.	A number (DefaultWriteFilters) of such entries are written by default.  When a log-in occurs, any number of such settings may be used, provided the numbers are consecutive.

Item	Default Value	Comment
DevReadAccess_01	NONE	Table of devices and indices that may be read from if Filtered read access is selected.  See Section 4.2.4 Access Control. Basic syntax. for full details of syntax.  Also Section 11 Filtering of BNCS Access for details of the extended filtering available.
DevReadAccess_02	NONE	
etc ..up to DevReadAccess_16	etc.	A number (DefaultReadFilters) of such entries are written by default.  When a log-in occurs, any number of such settings may be used, provided the numbers are consecutive.
DevAccess_nn	NONE	Table of devices and indices that may be both read from and written to.  A number (DefaultCombinedFilters) of such entries are written by default.

Note that the data section for a user is read when that user logs-in. If the contents of the file are changed the user will not pick-up any changes until the next log-in.

However, this is not the case if UserName, UserServerPort or UserActive is changed. Any changes to these values may immediately cause ports to be opened or closed.

There are some "secret" parameters available for use in the User\_nnn section.

These parameters are "secret" only in the sense that they do not get written to the file if not already present. They are only there if manually added.

Item	Value	Comment
MayCreateDatabaseFolders	0 - disable 1- enable  Default is 0 but is not written to the file.	Allows this user automatically to create the Year/Month/Day folder structure used for databases and schedules.  Previously all users of the database system have always been able to create any necessary files but the folder structure needed already to exist. This flag allows this user also to create the folders necessary to create any necessary file. Only the folders leading to the required file will be created, if not already present.  Use Fabian Scheduler to generate larger groups of folders (ie for a whole year).



CompatibilityRegistrations	0 - disable 1- enable Default is 0 but is not written to the file.	<p>The changes to revertive handling starting in version 4.9.20 have included stricter use of registrations.</p> <p>Previously packed messages would include revertive information where a slot was not registered for, although the device had registrations for other slots. The normal behaviour now is for the registration to be obeyed taking account of both slot and device numbers.</p> <p>Previously packed messages would include all database information irrespective of whether a slot or device was registered. The normal behaviour now is that database commands will be actioned, and responses will be sent, provided there is any registration for the device, irrespective of the range of indices registered for.</p> <p>If the older usage is required, this parameter should be set to 1.</p>
CompatibilityPackedMessageLength	0 - disable 1- enable Default is 0 but is not written to the file.	<p>While the changes to packed message generation were being implemented it was noted that there was a problem in Fabian Link prior to version 4.7.12.</p> <p>In earlier versions the buffer used to receive this message was not big enough to fit the maximum length that Fabian Server could send. Due to the messages being built based on network messages received directly from CSI it was not in fact possible for Fabian Server to generate messages of this full length. The new method for generating packed messages means that the full length may be use, which has been seen to crash Fabian Link.</p> <p>If this fault is seen and it isn't possible to upgrade Fabian Link then this item should be configured which will ensure the message length is kept within the length accepted by Fabian Link.</p>

There are believed to be very few users of the Fabian database system, or likely to be affected by the compatibility issues, so these controls will not contribute to "clutter" in the configuration files of users for whom they are not relevant.

#### 4.2.4 Access Control. Basic syntax.

The basic syntax for DevWriteAccess, DevReadAccess and DevAccess entries is :

<device number>:<list of indices>

The list of indices is any combination of single integers, or ranges of integers, separated by commas. Ranges are indicated by two integers separated by '-'.

Some examples of the DevWriteAccess\_nn or DevReadAccess\_nn entries follow.

042:6,89-90,22-23	
200:1-16,21-36	Device 200, indices 1-16 & 21-36 (GPI's)
201:1-26	Device 201 indices 1-26 (Infodriver)
5:16	Device 5 Destination 16 (Router )
103:9,11,13	...
104:1-128	...

See Section 11 Filtering of BNCS Access for information on the extended filtering available. DevAccess\_nn may also be used, to set read and write controls simultaneously.

Although there is no limit on the number of devices that may be filtered, nor on the number of indices within a line, there is a limit of 256 characters on the lines of data read from the configuration file.

If a filter specification needs more characters to define it will be necessary to use a Named Data Filter - See section 13 Named Data Filters.

#### 4.2.5 [Database] Section

The Database section is the database configuration and determines the field names and sizes for up to 128 fields. The first 7 fields of a typical configuration is shown below.

Item	Example Value	Comment
Field_001	Field_001, 0	Each entry consists of the field name and its size.
Field_002	Field_002, 0	
Field_003	Field_003, 0	
Field_004	Field_004, 0	
Etc... to		
Field_127	Field_127, 0	
Field_128	Field_128, 0	
etc....		

Some examples of Field\_nnn settings follow.

Field_001	Booking,3
Field_002	Amendment,2
Field_003	LineUp,4
Field_004	Start,4
Field_005	End,4
Field_006	Status,2
Field_007	Title,64

#### 4.2.6 [Mirror] Section

The Mirror section is for the optional configuration of the server mirroring mechanism. This allows you to mirror bookings database information

Item	Value	Comment
MirrorEnable	0	'1' enables server mirroring, '0' turns it off.
Address	NONE	The IP address of the remote server to be mirrored
Port	23	The service port on the remote server. The default is 23
UserName	USER	Username to use on the remote server
UserPass	PASSWORD	Password to go with the above user name.
RemoteFields	1,2,3,4,5	The fields in the remote database that you are interested in.
LocalFields	5,4,3,2,1	The fields in the local database where the booking information is to be placed. For example data in remote field 3 would be put into local field 6.
Mode	INSERT OVERWRITE (default is INSERT)	There are two <b>Mode</b> options. In OVERWRITE mode bookings in the local database are overwritten by those obtained from the remote database. In INSERT mode the bookings are inserted as the next free booking starting at the offset described below.
InsertOffset	600	In INSERT mode the local server will look for a free entry in the local database starting at <b>InsertOffset</b> and insert the booking there.

# 5 Resource Files

This section relates to the files used by the resource management system. These may be automatically created and set-up by Fabian Server.

When you are happy with the configuration of the INI file, the files and folders need to be created. If MayCreateDatabaseFolder is set then these will all be created as necessary if they are not already present. If it is not set they will need to be created manually, or using Fabian Scheduler.

12 directories are created, 1 for each month of the year. Within each of these a separate directory is created for each day of the month. For each day of the month two further subdirectories are created, RESOURCE.BKG and RESOURCE.NAM. Below is a view of the overall directory structure as provided by Windows' "TREE" command :-

## 5.1 Directory PATH listing

```
C:\FBSEVER\  
+---01  
| +---01  
| | +---RESOURCE.NAM  
| | +---RESOURCE.BKG  
| +---02  
| | +---RESOURCE.NAM  
| | +---RESOURCE.BKG  
| +---03  
| | +---RESOURCE.NAM  
| | +---RESOURCE.BKG  
| +---04  
| | +---RESOURCE.NAM  
| | +---RESOURCE.BKG  
| +---05  
| | +---RESOURCE.NAM  
| | +---RESOURCE.BKG  
| +---06  
| | +---RESOURCE.NAM  
| | +---RESOURCE.BKG  
|  
| etc, etc...  
|  
| +---29  
| | +---RESOURCE.NAM  
| | +---RESOURCE.BKG  
| +---30  
| | +---RESOURCE.NAM  
| | +---RESOURCE.BKG  
| +---31  
| | +---RESOURCE.NAM  
| | +---RESOURCE.BKG  
+---02  
| +---01  
| | +---RESOURCE.NAM  
| | +---RESOURCE.BKG  
+---02
```

```
| | +---RESOURCE.NAM  
| | +---RESOURCE.BKG  
| | +---03  
| | +---RESOURCE.NAM  
| | +---RESOURCE.BKG  
|  
| etc, etc...
```

## 6 Server Access

The normal means of access to a Fabian server is via a dedicated client. However, raw TCP access is used for diagnostic purposes and system familiarisation. Direct user interaction with the server is described in the TCP section below.

### 6.1 Dedicated clients

A dedicated client may be an application that allows a user to interact with the server for the purposes of manipulating bookings or it may be an automatic application. An automatic client application might do connect two or more servers together, perform statistical analysis or do monthly backups.

### 6.2 Raw TCP Interface

Fabian Server has a TCP terminal interface providing a bidirectional interactive communications facility. Note that this is not Telnet, which allows for certain control messages to be used and implies certain properties of the connection.

It provides the most basic text based interactive interface, but is useful for diagnostic purposes. All the commands that are available to dedicated clients are available via the TCP interface, in fact, the server does not inherently know or need to know the type of client it is talking to.

### 6.3 Establishing a TCP Session

Establishing a TCP session involves connecting to the host computer running the Fabian Server. Fabian uses the default Telnet port 23. If the host already has a Telnet or other service on that port then the recommended alternative is port 401.

If the UserServerPort fields are used to specify ports by user then other port numbers may be encountered.

Once connected, a banner is displayed showing the location of the server and other details.

#### 6.3.1 Logging In

To log in enter your user name and password using the SUN and SUP commands, as follows

SUN Joe <CR>

SUP Bloggs <CR>

or

SUN Joe:SUP Bloggs <CR>

The SUC command may then be used to enter the control password, if control functionality is required.

## 7 Command Usage

You can enter one command per line or multiple commands separated by colons. All the commands in the Command Reference document can be used via TCP, though it should be noted that this method of interacting with a Fabian server is recommended only for rudimentary testing and diagnostics.

At the time you log in the server sets your 'current' variables such as month and day to reflect the present date. You can use **GCD** and **GCM** to prove this. i.e. in response to :-

GCD:GCM <CR>

The server should respond with something similar to :-

GCD 24

GCM 5

which in this case indicates the 25th of May. When you use a command that requires a response the information is prefixed by your original command as shown above. If you use a command that does not require a response then the lack of an error message means that your command has been accepted.

**GDI <CR>** will return the number of bookings in the database for the current day. **GSU** will return a list of other users on the system. **HELP <CR>** or **GSH <CR>** will display the servers help page of which there may be several.

Note that Fabian Server uses the <CR> (13) character to terminate a line of one or more commands. <LF> (10) and <null> (0) characters are completely ignored and will not be passed on for further processing. The <Back> (8) character may be used for simple editing of commands, in particular when being entered manually. A small number of VT100 command codes are also recognized for editing, all of which start with <escape>[ (<escape> is character 27). A,B,C,D,H,K for Up, Down, Left, Right, Home, End respectively. These are not normally entered manually but most terminals or emulators can generate them.

### 7.1 Database Manipulation

A database record can consist of up to 128 fields. It is very unlikely that the system you are logged on to will use anywhere near that many so you use the Set User Fields command to 'filter' the fields contained in any records that you read or write. For example :-

SUF 1,2,3,4,5,6,7 <CR>

will tell the server that it should only return those fields whenever you request a database record.

Now use the GDI command to make sure there is at least one booking for today. Then type the following :-

SCB 1:SCA 0:GNB <CR>

This tells the server to set the current booking number to 1 and current amendment to 0. The GNB command asks the server to find the next booking in sequence. The response will be similar to :-

GNB 147 3

this indicates that the next booking after booking 1 is 147 and the first amendment is number 3.

Now type :-

SCB 147:SCA 3:GDR <CR>

This sets your current booking and amendment variables to point next booking. The GDR command tells the server to retrieve the database record. The response will depend upon the configuration of the server you are connected to, but is likely to look similar to :-

GDR 102,01,1615,1630,1745,00,PARLIAMENTARY FEED NORTH

To get the next booking in sequence use the GNB command again. The server will use your current booking and amendment as a starting point and return the next one it finds. Then use the SCB, SCA and GDR commands to retrieve the database record. Instead of using GNB to get the next booking you could use GNA to get the next amendment.

Although long winded from a terminal based users point of view this kind of simple access allows powerful manipulation of the database. A dedicated client can go the process above very rapidly to retrieve a whole day's schedule.

Writing database records is a very similar process and the combination of simple commands gives flexible access to a large database.

Messages about database changes are sent to all interested clients (see the GDU and SDU commands) and are similar to :-

DBC 24/5 147(3)

meaning that a database record for booking 147, amendment 3 for the 24th of May has been added or modified.

## 7.2 Resource Manipulation

The server maintains resource lists of up to 26,000 resources. The resource list is split into 26 different files from RESOURCE.\_A\_ to RESOURCE.\_Z\_. When the server receives an SRN command to add a new resource to the list it uses the first letter of that resource to determine which file to place it in. To add a new resource use the Set Resource New (SRN) command :-

SRN ANDOVER <CR>

The server will place the resource name ANDOVER into file RESOURCE.\_A\_. Checks are made for duplicates and an error message generated if found. To get a list of resources use Get Resource List (GRL) as follows :-

GRL A <CR>

using the first letter of the resource file as an argument. The response will be similar to :-

GRL

ALFRED

ANDREW

APPLCORE

ANDOVER

/GRL

Resource commitments are not held in the fields of the main database. Instead resource commitment files are kept for each day for each resource. To commit a resource to a booking first use SCB to indicate which booking you are going to commit the resource to. Then use the 'Set Resource Commitment' (SRC) command to commit the resource as follows :-

```
SCB 147:SRC ADD ANDREW 1100 1200 <CR>
```

This will commit the resource ANDREW to booking 147 between 1100 and 1200. Now commit ALFRED and APPLCORE :-

```
SRC ADD ALFRED 1115 1200 <CR>
```

```
SRC ADD APPLCORE 1100 1230 <CR>
```

There are now three resources committed to booking 147. The times that the resources are allocated for are in no way connected to the booking times. This means that resources need only be committed to a booking for the period they are actually required. Alternatively a resource may be committed for longer than a booking so that preparation and/or tidyup time can be included.

To see all the commitments for resource ANDREW type :-

```
GRC ANDREW <CR>
```

To filter commitments for a certain time period use :-

```
GRC ANDREW 0800 1600 <CR>
```

The response will be a list of all commitments that lie within or overlap those times, similar to the following :-

```
GRC ANDREW____.____
```

```
147,1100,1200
```

```
192,1415,1500
```

```
111,0830,0900
```

```
/GRC
```

The representation of the resource name is shown as the file that contains the list. Each line contains the booking number, start time and end time. If no commitments exist the response will be :-

```
GRC ANDREW____.____
```

```
NONE
```

```
/GRC
```

This can be taken as an indication that the resource is available to be booked between those times.

To list all the resources committed to a particular booking first select the booking and then use the 'Get Resource Booking' (GRB) command :-

```
SCB 147:GRB <CR>
```

The response will be similar to

```
GRB 147
```



ALFRED,1100,1200  
ANDREW,1100,1145  
APPLCORE,1115,1200  
ANDOVER,1130,1300  
/GRB  
If no resources have been added to the booking the response will be :-  
/GRB 147  
NONE  
/GRB

## 7.3 Messaging

There are 99 message channels. A message channel can be used for communication between 2 or more clients on the server. These may be clients that provide a human interface or those which are automated and need to communicate with one another.

To subscribe to a channel use the Set User Groups (SUG) command as follows :-

SUG 1,5,27 <CR>

will enable you to receive messages from groups 1, 5 and 27. To unsubscribe from all groups use the SUG command without any arguments.

A message coming in will appear in the form :-

SGM 27 The next meeting will be at 1530 on 23rd.

To send a message use the Send Group Message (SGM) command as follows :-

SGM 14 They're having another meeting! <CR>

If your message spans several lines just repeat the procedure.

# 8 System Notification Messages

## 8.1 Notification Overview

There are several System Notification Messages (SNM) despatched by the server supplying important information about the status of the database or resource manager.

## 8.2 Response to SNM's

A client application will most likely want to act upon SNMs that indicate a change in the database or resources by contacting the server for updated information. To prevent all clients requiring attention from the server at the same time the clients should wait a random period of time between 2 and 10 seconds before requesting information from the server.

### 8.3 Database Booking Change

When a database record is changed the server send a Database Booking Change (DBC) message of the form :-

DBC Month/Day Bkg(Amend) eg :-

DBC 5/23 147(6)

### 8.4 Resource List Change

The resource lists are contained in 26 files. The first letter of a resource determines which file it will go in. When a resource name is added or deleted from a resource list file the server sends out a Resource List Change (RLC) message of the form :-

RLC <FirstLetter\$>

<FirstLetter\$> points to the resource file that has changed and the file contents can be retrieved using Get Resource List and the letter returned by RLC.

### 8.5 ApplCore Database Change

If a FABIAN server is acting a BNCS host and any of the databases is modified an 'ApplCore Database Change' message is sent out to all clients. It has the form :-

ADC <Router%> <Switch%> <Index%> <Name\$>

<Router%> is the router which has had a database change database. <Switch%> is '0' if the change is a source and '1' if it is a destination. <Index%> is the source or destination number that has changed and <Name\$> is the new name.

## 9 Control System Interface

### 9.1 Control Overview

A Fabian Server is able to act as a BNCS control system host. Router, GPI and InfoDriver control is possible via the same text based interface used for database manipulation. The commands are almost identical to those used in ApplCore panels, but prefixed with the letter 'A'. For example '**RC 27 10 55**' becomes '**ARC 27 10 55**'. The command reference document details the commands that are available.

To be able to use the ApplCore commands a second password has to be sent to the server with Set User Control :-

SUC <Password2\$>

This prevents authorised users of the booking system functions from gaining access to the control functions.

More information on the setting up the server as a BNCS control host is given in the 'Server Configuration' section.

### 9.2 Sample Ini

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

```
[Configuration]
ServerName=Anonymous
ServerPort=23
LoginRequired=1
DatabaseEnable=OBSOLETE
DataPath=C:\CollediaControl\Core_System\logs
SynchronousMode=OBSOLETE
CloseWithCSI=1
DisableCSICache=1
EnableLogging=1
DebugMode=0
Show RUTS=0
SystemCode=0
BNCSAccess=1
SchedNotifyGroup=1
BookingDatabaseEnable=0
LogMode=0
[User_001]
UserName=Tester
UserPass=Pass
ControlPass=PassCon
BNCSAccess=READONLY
Fields=1,2,3,4,5
ExpectRUTreply=0
DevWriteAccess_01=NONE
DevWriteAccess_02=NONE
DevWriteAccess_03=NONE
```

... to ...

```
DevWriteAccess_15=NONE
DevWriteAccess_16=NONE
WorkstationOverride=0
AllowDatabaseChanges=1
[User_002]
UserName=---
UserPass=---
ControlPass=---
BNCSAccess=READONLY
Fields=1,2,3,4,5
ExpectRUTreply=0
```

```
DevWriteAccess_01=NONE
DevWriteAccess_02=NONE
... to ...
DevWriteAccess_15=NONE
DevWriteAccess_16=NONE
WorkstationOverride=0
AllowDatabaseChanges=1
```

... to ...

```
[User_064]
UserName=---
UserPass=---
ControlPass=---
BNCSSAccess=READONLY
Fields=1,2,3,4,5
ExpectRUTreply=0
DevWriteAccess_01=NONE
DevWriteAccess_02=NONE
... to ...
DevWriteAccess_15=NONE
DevWriteAccess_16=NONE
WorkstationOverride=0
AllowDatabaseChanges=1
[Mirror]
MirrorEnable=0
Address=NONE
Port=23
UserName=USER
UserPass=PASSWORD
RemoteFields=1,2,3,4,5
LocalFields=5,4,3,2,1
Mode=INSERT
InsertOffset=600
```

# 10 Infodriver

## 10.1 Status Infodriver

FabianServer32 can connect to an external InfoDriver to indicate its status and that of any connected clients.

The data written to the InfoDriver is in blocks of ten slots.

### 10.1.1 Overall Status

The first ten slots, 1 to 10, show the state of the program itself.

Slot number	Contents
1	"1" if the program is running. "0" if the program is not running and shut down cleanly. This value will not be set to "0" if the program does not shut down properly, eg if it has crashed.
2	"Running" if the program is running. "Stopped" if the program is not running and shut down cleanly. This value will not be set to "Stopped" if the program does not shut down properly, eg if it has crashed.
3	The number of clients currently connected.

### 10.1.2 Connection Status

Slots from 10 to 489 are in groups of ten slots, and provide information about each connection.

Slots 11 to 19 show the first connection.

Slots 21 to 29 show the second connection.

And so on, up to slots 481 to 489 for the 48<sup>th</sup> connection.

Slot offset	Contents
1	"1" if this connection is active. "0" if the connection is not active. This is normally only seen briefly while the log-in is in progress.
2	User-name logged-in.
3	Address from which the user is logged-in.
4	Port number from which the user is logged-in.

5	The number of devices for which this client has registered for reverts.

### 10.1.3 Updates and Reverts

The data is normally only written to the InfoDriver (causing a revert to be issued) at program start-up or when the value changes. If it is required that a revert is required at any other time, then any value may be written to a slot. If a write is attempted then a set of reverts will be produced for all the slots in that block of ten. Ie writing to slot 3 will elicit reverts for the Overall Status block, slots 1 to 10; writing to slot 10 (or any slot between 10 and 19) will elicit reverts for client number 1, slots 21 to 29 for connection number 2 etc.

## 11 Filtering of BNCS Access

A facility is available to control (or filter) the data that is transferred between a user (ie a user tcp channel) and the BNCS network. It is possible to separately configure the filtering both write (Control) and read (Monitor) data. It is also possible to control access based on the contents of the data.

### 11.1 Filtering

Filtering is enabled or disabled with the BNCSAccess entry in the [User\_nn] section.

This may have any of the following values listed in the table below. These allow the selection of All, None or Filtered access for either read or write.

Read	Write	Name
None	None	NOACCESS
None	Filtered	WRITEONLYFILTERED
None	All	WRITEONLYALL
Filtered	None	READONLYFILTERED
Filtered	Filtered	READFILTEREDWRITEFILTERED
Filtered	All	READFILTEREDWRITEALL
All	None	READONLYALL or READONLY This is the default setting.
All	Filtered	READALLWRITEFILTERED or WRITEFILTERED

All	All	READALLWRITEALL or WRITEALL
-----	-----	-----------------------------

The names READONLY, WRITEFILTERED and WRITEALL are available for compatibility with earlier versions.

Filtering of writes means that an attempt to write to a particular slot or index, using AIW, ARC or AGS, will only be passed to the BNCS network if the filter lists that address/slot number/index.

Filtering of reads means that if Fabian Server is registered for reverts for a particular slot and sees a revert for that slot, then the filter will be checked to determine whether the data may be passed to the client. It doesn't affect polls initiated by the client. A poll may refer to an address that is not enabled by the filter. In this case the poll will take place but data in reverts from such addresses won't be passed to the client.

From version 4.9.21 these settings also apply to database entries and device information.

If a user is permitted to read or write a particular index/slot/destination then they are also permitted to read or write all databases for that index. There is no data filtering for database entries, and the database number is ignored.

If a user is permitted to read or write any index on a device then that device will be visible when using the ADP or ADL commands.

## 11.2 Address Filter Configuration

If filtering by address is selected using one of the entries above, it is then necessary to configure the level of filtering required.

Filtering by address is configured using the DevWriteAccess\_nn, DevReadAccess\_nn and DevAccess\_nn entries in the [User\_nn] sections in the configuration file.

The basic syntax for DevWriteAccess, DevReadAccess and DevAccess entries is :

<device number>:<list of indices>

The list of indices is any combination of single integers, or ranges of integers, separated by commas. Ranges are indicated by two integers separated by '-'.

Some examples of the DevWriteAccess\_nn or DevReadAccess\_nn entries follow.

```
042:6,89-90,22-25  Device 42, slots 89 to 90 (ie 89 and 90), 22 to 25 (ie 22, 23, 24, 25).
200:1-16,21-36    Device 200, indices 1 to 16 and 21 to 36
201:1-26          Device 201 indices 1 to 26
5:16              Device 5 Destination 16
103:9,11,13 ...
104:1-128 ...
```

### 11.2.1 Wildcard Address Filtering

If it is required to specify write access to all addresses of a device the "wildcard" or "all" syntax may be used. This uses "\*" instead of numeric references to indices.

eg.

200:\*            Device 200, all indices

This is functionally the same as

200:1-4096

But the implementation is different, such that less memory is needed and no address search is needed.

### 11.2.1 Control of parameters – None, blank and comment

Entries will be read until either a break in consecutive numbering or the value "NONE" is read.

An entry of NONE will stop the configuration being read. Any entries with higher numbers will be ignored.

A blank entry will be ignored. Further entries will be read.

An entry where the first character of the value is ; (semi-colon) will be ignored. Further entries will be read.

The values of DefaultWriteFilters, DefaultReadFilters and DefaultCombinedFilters determine the number of empty (NONE) entries which will be written into the configuration file if they are missing at start-up.

## 11.3 Data Filtering

If filtering is enabled for read or write it is also possible to filter based on the contents of the data, ie to require that only certain values may be written to or read from the BNCS network.

In general it is necessary to specify which indices a data filter applies to, so its configuration is an addition to the syntax for configuring address filters. The data filter is put in parentheses after the relevant address filter spec. It comprises a type keyword, followed by a colon, followed by a parameter name and value.

The type of data may be specified as integer, floating-point number (called "real") or string.

Following this type indication, a number of further parameters may be used, to specify various collections of allowed values. In some cases multiple parameters are allowed, which should be separated by commas.

The general forms for the data filter are:

(type:name=value)

For a single parameter, and:

(type:name1=value1,name2=value2)

For two.

A specific example might be:



(string:minlength=4,maxlength=12)

Which could be encountered in an access control line such as the following:

DevWriteAccess\_01=926:1(string:minlength=4,maxlength=12)

See the examples below for the meaning of this.

## 11.4 Data Types

Data filters may be one of three types : integer, real, string

Integer or real may be used in isolation, without further parameters. This means that the data, if a string, must be formatted as an integer or floating point number respectively. Although string may be used in isolation it has no meaning if used in this way as all strings are strings.

### 11.4.1 String Parameters

String parameters are described in the following table.

Name	Format	Description
minlength	string:minlength=n	Specifies that the string must be at least a minimum length.
maxlength	string:maxlength=n	Specifies that the string must be at most a maximum length.
set	string:set=a,b,c or string:set=/a/b/c/	Specifies that the string must have one of the given values. These should normally be separated by commas.  If it is necessary to allow commas within any of the selections, then the parameter value should begin and end with "/" and "/" should also be used to separate the individual entries.
regexp	string:regexp=re	Specifies that the string must match the given Regular Expression.  Details of the Regular Expression syntax used may be found in Section 17 Regular Expression Syntax.  Note that the parsing of the regular expression parameter from the configuration file is not able to properly validate its syntax. As a result of this the regexp parameter will take the entire contents of the line up to the final ").  Thus a line such as :  DevWriteAccess_02=927:51(string:regexp=(source dest)[0-9]+),10-13(integer:25-46)  does not, as might be expected, generate two address filters:  address = 51, regular expression = "(source dest)[0-

		<p>9]+)"</p> <p>and</p> <p>address = 10 to 13, integer range 25 to 46.</p> <p>In fact it generates only one address filter :</p> <p>address = 51, regular expression = "(source dest)[0-9]+),10-13(integer:25-46"</p> <p>For the former functionality, the two filters should be given on separate lines.</p>
--	--	---

#### 11.4.2 Integer Parameters

Integer parameters are described in the following table.

Name	Format	Description
min	integer:min=n	Specifies that the integer must not be less than the given value.
max	integer:max=n	Specifies that the integer must not be greater than the given value.
range	integer:range=a-b	<p>Specifies that the integer's value must be between the two given values.</p> <p>Note that if no parameter name is given the parameter is assumed to be of this type.</p> <p>eg. (integer:25-46)</p> <p>Note that the range must be given in order, with the lower value first.</p> <p>46-25 will not create a valid range.</p>
set	integer:set=a,b,c	Specifies that the integer must have one of the given values. These should be given separated by commas, or ranges may be used, with the range limits separated by "-". Note that "-" may be found meaning both "negative number" and "range" in the same line.
simpleset	integer:simpleset=a,b,c	<p>Specifies that the integer must have one of the given values. These should be separated by commas.</p> <p>Ranges may not be used in simplesets.</p>

#### 11.4.3 Real Parameters

Real number parameters are described in the following table.

Name	Format	Description
------	--------	-------------

min	real:min=n	Specifies that the real must be no less than the given value.
max	real:max=n	Specifies that the real must be no greater than the given value.
range	real:range=a-b	<p>Specifies that the number's value must be between the two given values.</p> <p>Note that if no parameter name is given the parameter is assumed to be of this type.</p> <p>eg. (real:25.4-46.9)</p> <p>Note that the range must be given in order, with the lower value first.</p> <p>46.9-25.4 will not create a valid range.</p>
set	real:set=a,b,c	<p>Specifies that the floating point number must have one of the given values. These should be given separated by commas, or ranges may be used, with the range limits separated by "-". Note that "-" may be found meaning both "negative number" and "range" in the same line.</p> <p>Note that as floating point numbers may not normally be accurately compared for equality there is a range of acceptable values where a single value is listed in a set. The size of that range is currently hard-coded as 0.001, ie "2.5" means "2.499" to "2.501".</p>
simpleset	real:simpleset=a,b,c	<p>Specifies that the the floating point number must have one of the given values. These should be given separated by commas.</p> <p>Ranges may not be used in simplesets.</p> <p>Note that as floating point numbers may not normally be accurately compared for equality there is a range of acceptable values where a single value is listed in a set. The size of that range is currently hard-coded as 0.001, ie "2.5" means "2.499" to "2.501".</p>

#### 11.4.4 Floating-point resolution.

The floating-point numbers used are the C++ "float" type, with range 3.4E +/- 38 (7 digits).

At certain points within the program the "double" type is used where there is no alternative.

Where the program is explicitly comparing floating-point numbers, in the "set" or "simpleset" parameters, there is an explicit range for matching, as indicated in the above table.

Where other comparisons are done, in "min", "max" or "range", the issue of accuracy is not clear. Inaccuracy may be caused in the process of converting the string representations of the numbers in the config file to the program's internal representations.

#### 11.4.5 Multiple Parameters

Note that data filters are processed by exclusion. That's to say the program will work through all the given filters and if any one fails the filter as a whole will fail – the data will not be read/written.

This means that not all filters may be used in combinations. Only min and max, or minlength and maxlength, may be used in pairs. Any other filter, if used in a combination, is likely to fail.

#### 11.4.6 Examples

Some examples follow. These all use DevWriteAccess, though are equally applicable to DevReadAccess or DevAccess.

Note that "may have written to them any integer" means "may have written to them any value which is of the form of an integer" if the device is an InfoDriver. If it is a router or GPI driver then only integers may be written at any time.

Likewise "may have written to them any floating point number" means "may have written to them any value which is of the form of a floating point number " if the device is an InfoDriver. If it is a router or GPI driver only integers may be written at any time, therefore any filter type other than integer is likely to cause no value to be written to the device.

```
DevWriteAccess_01=926:1(string:minlength=4,maxlength=12),3(string:set=4:3,16:9,14:9),50
```

Device 926, index 1 is a string with a length between 4 and 12 characters.

Device 926, index 3 is a string which is one of the following values - "4:3", "16:9", "14:9".

Device 926, index 50 may have any value written to it.

```
DevWriteAccess_02=927:51(string:regexp=(source|dest)[0-9]+)
```

Device 927, index 51 may have any string written to it which matches the regular expression "(source|dest)[0-9]+".

```
DevWriteAccess_02=927:10-13(integer:25-46)
```

Device 927, indices 10 to 13 may have written to them any integer from 25 to 46.

```
DevWriteAccess_03=928:41-43(integer:max=12)
```

Device 928, indices 41 to 43 may have written to them any integer up to a maximum of 12. This includes zero and all negative values.

```
DevWriteAccess_04=929:20(integer:min=-16),21(integer:-46--25),10-13(integer:25-46)
```

Device 929, index 20 may have written to it any integer from -16. This includes zero and all positive values.

Device 929, index 21 may have written to it any integer from -46 to -25.

Device 929, indices 10 to 13 may have written to it any integer from 25 to 46.

DevWriteAccess\_05=930:22-25(integer:set=-19--15,-5,-2,3-5,12,15)

Device 930, indices 22 to 25 may have written to them any of the following integer values : -19, -18, -17, -16, -15, -5, -2, 3, 4, 5, 12, 15.

Note that "-19--15" means -19 to -15 ie the "-" symbol means both "negative number" and "range".

DevWriteAccess\_06=931:23(real:min=-2.5,max=4.45),25(integer)

Device 931, index 23 may have any floating point number written to it between -2.5 and 4.25. Index 25 may have any integer written to it.

DevWriteAccess\_07=932:24(real:set=1.6-1.9,2.5,3.4-5.9),26(real)

Device 932, index 24 may have any floating point number written to it from the following set : 1.6 to 1.9, 2.5, 3.4 to 5.9. Index 26 may have any floating point number written to it.

Note that as floating point numbers may not normally be accurately compared for equality there is a range of acceptable values where a single value is listed in a set. The size of that range is currently hard-coded as 0.001, ie "2.5" means "2.499" to "2.501".

DevWriteAccess\_08=933:27(real:12.6-14.7),28(real:-273.15--100.8)

Device 933, index 27 may have any floating point number between 12.6 and 14.7 written to it.

Index 28 may have written to it any floating point number between -273.15 and -100.8.

DevWriteAccess\_09=934:29(integer:range=1-10),30(real:range=1.5-10.5)

Device 934, index 29 may have written to it any integer from 1 to 10.

Device 934, index 30 may have written to it any floating point number from 1.5 to 10.5.

DevWriteAccess\_11=930:31(integer:max)

This is not valid. "max" needs a value. Device 930 index 31 will not be writable.

DevWriteAccess\_12=935:22-25(integer:simpleset=-5,-2,12,15)

Device 935, indices 22 to 25 may have written to them any of the following integer values : -5, -2, 12, 15.

DevWriteAccess\_13=936:24(real:simpleset=1.6,2.5,3.4)

Device 936, index 24 may have written to them any of the following floating point values : 1.6, 2.5, 3.4.

DevWriteAccess\_14=937:31(string:set=/1,2/3,4/5,6/)

Device 937, index 31 is a string which is one of the following values - "1,2", "3,4", "5,6".

## 11.5 The Get System Access Command (GSA)

The Get System Access command has previously read the access settings for the current user from the configuration file and displayed them. This information may not be correct for the actual logged-in instance of the current user.

The GSA command now gives a number of options, accessed by adding a numeric parameter.

GSA option	Information source	Data filters
GSA 0	Displays the access settings in use for the current user session.	No data filter settings are shown.
GSA 1	Displays the access settings in use for the current user session.	Data filter settings are shown.
GSA 2	Displays the settings for the current user in the config file.	No data filter settings are shown.
GSA 3	Displays the settings for the current user in the config file.	Data filter settings are shown.

# 12 Named Filters

It is possible to include access control filter information in named sections and to refer to them by name from the User configuration sections, or from other named filter sections.

The named filter sections are called the filter definitions.

References to them from the User sections or from other named filter sections are called the filter references.

## 12.1 Filter Sections

The named filter sections should have names of the form "[Filter\_<name>]".

For example : [Filter\_filter1], [Filter\_NormalUser], [Filter\_Axon\_ARC20]

The filter section may contain a list of DevWriteAccess\_nn, DevReadAccess\_nn or DevAccess\_nn entries. These should be numbered consecutively starting at one. A break in the consecutive sequence marks the end of the block.

## 12.2 [Filter\_<name>]

The filter section must contain a FilterType entry. This determines whether the filter is an Address or Data filter.

ADDRESS      It's an Address filter.

DATA          It's a Data filter. See section 11.3 Data Filtering for more information.

The filter section must contain a BlockType entry. This determines how it is referred to by the User section or by other named filter sections.

BlockType	
ABSOLUTE	Filter information includes device numbers and indices. This is as found in User sections when named filters are not being used.
RELATIVE	Filter information includes indices only. Device number must be provided in the reference.
OFFSETTABLE	Filter information includes offset indices only. Device number and "anchor" indices must be provided in the reference.

### 12.2.1 Absolute Filters

An Absolute filter definition specifies the device number and indices.

An Absolute filter is referenced using its name only.

An absolute filter may include references to several device numbers.

#### 12.2.1.1 Examples

```
[User_008]
UserName=User8
BNCSAccess=READFILTEREDWRITEFILTERED
DevWriteAccess_01=filter2

[Filter_filter2]
; An absolute filter.
; The filter definition specifies both indices and device.
; It is referenced with no further parameters.
; eg. DevWriteAccess_nn=filter2
BlockType=ABSOLUTE
DevWriteAccess_01=928:31,33,35,40-43,71
```

In this case the result will be that device 928 can write to indices 31, 33, 35, 40, 41, 42, 43, 71

### 12.2.2 Relative Filters

A Relative filter definition specifies indices but does not specify the device number.

A Relative filter is referenced using its name and the device number.

It is envisaged that Relative filters will be used where it is useful to define the access control pattern for a type of device and then to indicate that device type exists at several device numbers.

A relative filter may only be applied to one device number at a time.

#### 12.2.2.1 Examples

```
[User_008]
UserName=User8
BNCSAccess=READFILTEREDWRITEFILTERED
DevWriteAccess_02=filter1:926

[Filter_filter1]
; A relative filter.
; The filter definition specifies the indices but not the device.
; The reference must provide the device.
; eg. DevWriteAccess_nn=filter1:042
BlockType =RELATIVE
DevWriteAccess_01=21,23,25,30-33,61
```

In this case the result will be that device 926 can write to indices 21, 23, 25, 30, 31, 32, 33, 61.

### 12.2.3 Offsettable Filters

An Offsettable filter definition specifies index offsets but does not specify the device number or the actual indices.

An Offsettable filter is referenced using its name and the device number and one or more "anchor" indices. This has the effect of expanding the filter repeatedly with all of its offset numbers added to the given "anchor" values.

It is envisaged that Offsettable filters will be used where it is useful to define the access control pattern for a type of device and then to indicate that devices exist at several offset index positions within any particular device number. This arises for devices such as the Axon Synapse family where the driver controls devices starting at a number of different offset slot numbers.

An Offsettable filter may only be applied to one device number at a time.

The anchor values may be specified in the specific "anchor" style, or using the conventional single value or range notation.



Type	Pattern	Examples	Meaning
Anchor	n-	42- 101-	The offsettable filter will be expanded starting at the given anchor value with no upper limit. The pattern represents a range with no upper limit.
Range	n-n	1-20 201-210	The offsettable filter will be expanded starting at the lower value up to the upper limit.
Single	n	1 301	The offsettable filter will be expanded only if an offset is zero. In that case it will apply at the value given here. The pattern represents a range where the upper limit equals the lower.

### 12.2.3.1 Examples

```
[User_008]
UserName=User8
BNCSAccess=READFILTEREDWRITEFILTERED
DevWriteAccess_01=filter3:927:120-,150-,180-190,200

[Filter_filter3]
; An offsettable filter.
; The filter specifies indices but needs both device number and
; anchor(s).
; The reference must provide the device and anchor indices.
; eg. DevWriteAccess_nn=filter3:042:23-,123-
BlockType =OFFSETTABLE
DevWriteAccess_01=0,1,3,5,10-13,20
```

In this case the result will be that device 927 can write to indices 120, 121, 123, 125, 130, 131, 132, 133, 140, 150, 151, 153, 155, 160, 161, 162, 163, 170, 180, 181, 183, 185, 190, 200.

## 13 Named Data Filters

A data filter may also be specified by name, using a named data filter.

To access this, the filter name should be specified in the data filter part of the general address filter syntax.

Eg.

```
DevWriteAccess_01=926:1(MyFilter)
```

Means that writes to index 1 of device 926 will be checked using a filter called MyFilter.

Named Data Filters are defined in the config file in sections named [Filter\_<name>] as are named address filters. However, they should include FilterType=DATA.

### 13.1 Compact or Extended Syntax

A named data filter may either be defined using the compact form or the extended form.

The compact form simply allows the data filter syntax defined in section 11.3 Data Filtering to be used.

The extended syntax separates the description of the filter into several separate lines.

Key	Value	Description
Type		Select compact syntax or data type for extended syntax.
	COMPACT	Compact syntax. Use the Definition key for the actual filter definition.
	STRING	Extended syntax. String data.
	INTEGER	Extended syntax. Integer data.
	REAL	Extended syntax. Floating-point number data.

### 13.2 Compact

This is declared using Type=COMPACT, and defining the filter using a "Definition" key.

Key	Value	Description
Definition		Filter definition, using standard data filter definition syntax as in section 11.3 Data Filtering.

The filter spec may optionally be enclosed in parentheses.

Eg.

```
Type=COMPACT
```

```
Definition=(string:minlength=4,maxlength=12)
```

Or

Definition=string:minlength=4,maxlength=12

## 13.3 Integer

Type=INTEGER

Name	Value and example	Description
Minimum	Single integer -25	Specifies a minimum acceptable value.
Maximum	Single integer -46	Specifies a maximum acceptable value.
Range	Range of integers separated by '-' -46--25	Specifies a range of acceptable values.
SetMultipleItem_nn	Set description -19--15,-5,-2	Specifies a number of items for a set.  Items may be single integers or ranges.
SetSingleItem_nn	Single integer -5	Specifies a single integer item to be included in the set.
SetRangeItem_nn	Range of integers separated by '-' -19—15	Specifies a range item to be included in the set.
SetRangeItemMinimum_nn	Single integer -19	In conjunction with the corresponding SetRangeItemMaximum entry, specifies a range item to be included in the set.
SetRangeItemMaximum_nn	Single integer -15	In conjunction with the corresponding SetRangeItemMinimum entry, specifies a range item to be included in the set.

### 13.3.1 There is only one set

Note that all the "Set" entries contribute to a single set, so:

SetMultipleItem\_01=-19--17,-5,-2

SetMultipleItem\_02=3-5,12,15

SetSingleItem\_01=-10

```

SetSingleItem_02=-12
SetSingleItem_03=1
SetSingleItem_04=7
SetRangeItem_01=-8--6
SetRangeItem_02=18-20
SetRangeItemMinimum_01=-29
SetRangeItemMaximum_01=-27
SetRangeItemMinimum_02=25
SetRangeItemMaximum_02=27

```

would generate a single set containing

-19, -18, -17, -5, -2, 3, 4, 5, 12, 15, -10, -12, 1, 7, -8, -7, -6, 18, 19, 20, -29, -28, -27, 25, 26, 27.

### 13.3.2 Numbered keys

All the items listed with names including "nn" may be given using a series of consecutive numbers starting at 01.

A break in the sequence will terminate the reading of this set of keys. Otherwise there is no upper limit.

## 13.4 Real

Type=REAL

Name	Value and example	Description
Minimum	Single floating-point number -2.5	Specifies a minimum acceptable value.
Maximum	Single floating-point number 4.45	Specifies a maximum acceptable value.
Range	Range of Single floating-point numbers separated by '-' -2.5-4.45	Specifies a range of acceptable values.
SetMultipleItem_nn	Set description 1.6-1.9,2.5,3.4-5.9	Specifies a number of items for a set. Items may be single integers or ranges.

SetSingleItem_nn	Single floating-point number 2.5	Specifies a single integer item to be included in the set.
SetRangeItem_nn	1.6-1.9	Specifies a range item to be included in the set.
SetRangeItemMinimum_nn	Single floating-point number 1.6	In conjunction with the corresponding SetRangeItemMaximum entry, specifies a range item to be included in the set.
SetRangeItemMaximum_nn	Single floating-point number 1.9	In conjunction with the corresponding SetRangeItemMinimum entry, specifies a range item to be included in the set.

As with integers there is only one set even if it is built using a selection of "Set" keys. See 13.3.1 There is only one set.

The process of reading keys including "nn" is the same as for integers. See 13.3.2 Numbered keys.

## 13.5 String

Type=STRING

Name	Value and example	Description
MinimumLength	Single integer 4	Specifies a minimum acceptable length.
MaximumLength	Single integer 12	Specifies a maximum acceptable length.
SetMultipleItem_nn	Set description [4:3,16:9,14:9]	Specifies a number of items for a set.  Items should be separated by ', ' (comma) if there is no SetMultipleItemSeparator specified, or by the character specified by SetMultipleItemSeparator.

SetSingleItem_nn	Single string [4:3	Specifies a single string item to be included in the set.
RegularExpression	Regular expression (source dest)[0-9]+	
SetMultipleItemSeparator	/	Specifies the separator character for SetMultipleItem entries.  If this is given with more than one character then the first character of the string will be used.

As with integers there is only one set even if it is built using a selection of "Set" keys. See 13.3.1 There is only one set.

The process of reading keys including "nn" is the same as for integers. See 13.3.2 Numbered keys.

## 13.6 Examples

```
Type=STRING
SetSingleItem_01=[4:3
SetSingleItem_02=16:9
SetSingleItem_03=14:9]
```

Results in a set containing "[4:3", "16:9", "14:9]"

```
Type=STRING
SetMultipleItem_01=[4:3,16:9,14:9]
```

Results in a set containing "[4:3", "16:9", "14:9]"

```
Type=INTEGER
SetMultipleItem_01=-19--17,-5,-2
SetMultipleItem_02=3-5,12,15
```

Results in a set containing -19, -18, -17, -5, -2, 3, 4, 5, 12, 15.

```
Type=INTEGER
SetSingleItem_01=-5
SetSingleItem_02=-2
SetSingleItem_03=12
SetSingleItem_04=15
SetRangeItem_01=-19--17
SetRangeItem_02=3-5
```

Results in a set containing -5, -2, 12, 15, -19, -18, -17, 3, 4, 5.

```
Type=INTEGER
SetSingleItem_01=-5
SetSingleItem_02=-2
SetSingleItem_03=12
SetSingleItem_04=15
SetRangeItemMinimum_01=-19
SetRangeItemMaximum_01=-17
SetRangeItemMinimum_02=3
SetRangeItemMaximum_02=5
```

Results in a set containing -5, -2, 12, 15, -19, -18, -17, 3, 4, 5.

```
Type=REAL
SetMultipleItem_01=1.6-1.9, 2.5, 3.4-5.9
```

Results in a set containing 1.6 to 1.9, 2.5, 3.4 to 5.9.

```
Type=REAL
SetSingleItem_01=2.5
SetRangeItem_01=1.6-1.9
SetRangeItem_02=3.4-5.9
```

Results in a set containing 1.6 to 1.9, 2.5, 3.4 to 5.9.

```
Type=REAL
SetSingleItem_01=2.5,
SetRangeItemMinimum_01=1.6
SetRangeItemMaximum_01=1.9
SetRangeItemMinimum_02=3.4
SetRangeItemMaximum_02=5.9
```

Results in a set containing 1.6 to 1.9, 2.5, 3.4 to 5.9.

```
Type=INTEGER
SetMultipleItem_01=-5, -2, 12, 15
```

Results in a set containing -5, -2, 12, 15

```
Type=INTEGER
SetSingleItem_01=-5
SetSingleItem_02=-2
```

```
SetSingleItem_03=12
```

```
SetSingleItem_04=15
```

Results in a set containing -5, -2, 12, 15

```
Type=REAL
```

```
SetMultipleItem_01=1.6,2.5,3.4
```

Results in a set containing 1.6, 2.5, 3.4

```
Type=REAL
```

```
SetSingleItem_01=1.6
```

```
SetSingleItem_02=2.5
```

```
SetSingleItem_03=3.4
```

Results in a set containing 1.6, 2.5, 3.4

```
Type=STRING
```

```
SetMultipleItemSeparator=/
```

```
SetMultipleItem_01=/1,2/3,4/5,6/
```

Results in a set containing "1,2", "3,4", "5,6".

```
Type=STRING
```

```
SetSingleItem_01=1,2
```

```
SetSingleItem_02=3,4
```

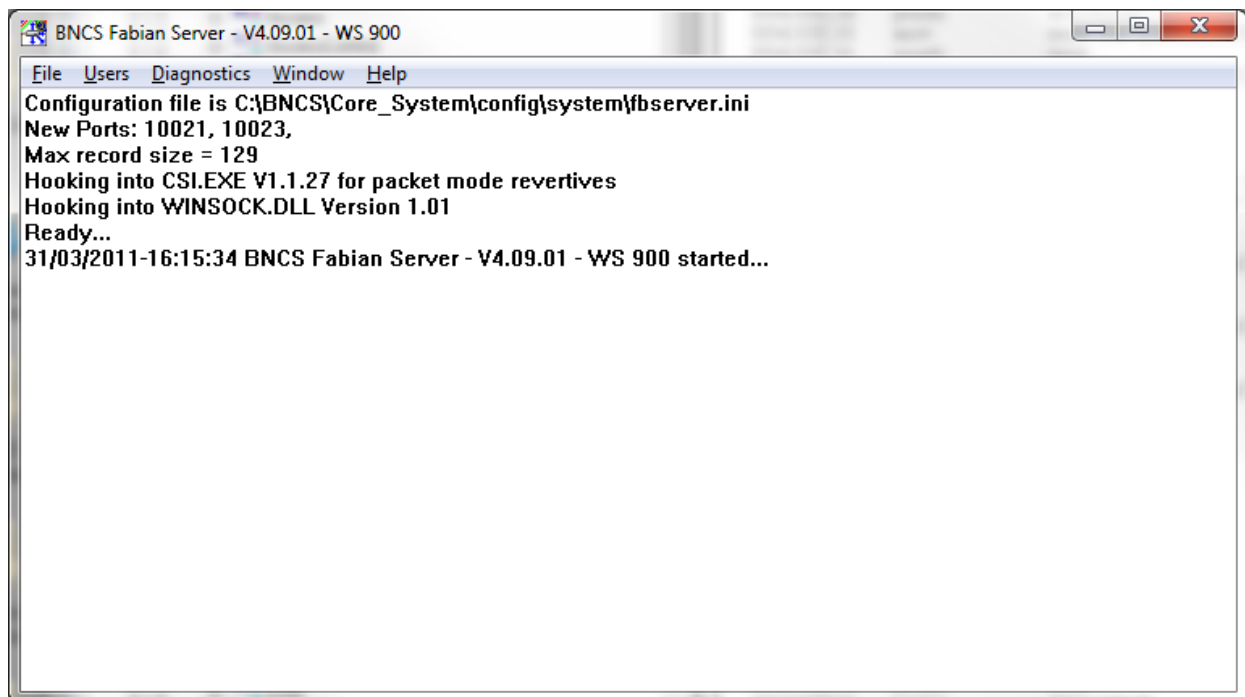
```
SetSingleItem_03=5,6
```

Results in a set containing "1,2", "3,4", "5,6".

## 14 Driver GUI

When running the program looks like Figure 1: FabianServer32 .





**Figure 1: FabianServer32**

There are a number of controls available from the menu.

## 14.1 File Menu

### 14.1.1 Exit

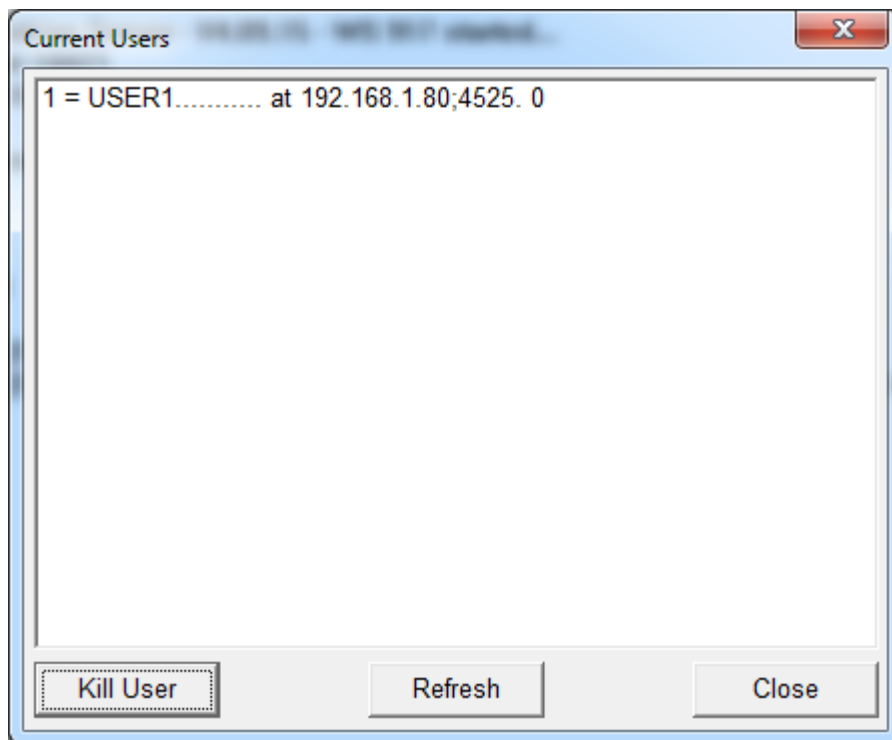
This menu item terminates the program.

## 14.2 Users Menu

### 14.2.1 Manage Users...

(The Users first-level menu entry in some versions)

This item lists any current active connections in a pop-up dialog - Figure 2: Current Users Dialog.



**Figure 2: Current Users Dialog**

Use Close to close the dialog. The dialog won't disappear until Close is selected.

To forcibly close any connection - select it and select Kill User.

The Refresh button will repopulate the list of users. It should be used if there is reason to believe clients have connected or disconnected while this dialog is showing.

## 14.3 Diagnostics Menu

All the items set by this menu are preserved in the ini file.

### 14.3.1 On

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

### 14.3.2 Show RUTs

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

### 14.3.3 Logging

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

### 14.3.4 Show Revertives Received

Selects whether diagnostics relating to revertive messages received will be written to the log-file and/or the screen.

#### 14.3.5 Show Revertives Sent

Selects whether diagnostics relating to revertive messages being sent will be written to the log-file and/or the screen.

#### 14.3.6 Show Commands Received

Selects whether diagnostics relating to commands received will be written to the log-file and/or the screen.

### 14.4 Window Menu

#### 14.4.1 Clear

This menu item clears the screen.

### 14.5 Help Menu

#### 14.5.1 About

This menu item will cause a dialog to be displayed giving some information about the version of the software. Press the OK button to remove the dialog.

## 15 Resilience and Redundancy

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

All the actual redundancy-related functionality is in FabianLink – see it's documentation for full details.

Instances of FabianServer32 running on a BNCS network have no knowledge of one-another; they don't know how many others are there, whether they are independent or are redundant pairs.

## 16 Logging

### 16.1 Standard Log

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

It will also write the messages to a log-file, if enabled. This is controlled by the Diagnostics|Logging item on the menu.

From version 4.9.20 the enabling of Debug and Log writing are separate. Prior to this version Debug needed to be enabled as well as Logging for anything to be written to the Log file. From version 4.9.20, logging is available without Debug being active.

See Section 2.7 BNCS configuration for the location.

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

## 16.2 Fabian Server Specific Log (Daily Log)

There is another log file written by FabianServer32, which is handled in the same way as in earlier versions (including v3fbserv). This is written to the location specified by the DataPath parameter. Only messages specifically sent to it are written, unlike the other log where everything written to the diagnostics display is also written if enabled.

It is named yyyyymmdd.TXT, where yyyy is the year, mm is the month and dd the day of the month.

Its use is enabled by setting EnableLogging in fbserver.ini.

Times within this file may include milliseconds if LoggingMillisecondDisplay is set in fbserver.ini.

# 17 Regular Expression Syntax

The regular expression syntax used for data filters is based on that used by the programming language Perl.

In Perl regular expressions, all characters match themselves except for the following special characters:

`. [ { ( ) \ * + ? | ^ $`

## 17.1 Wildcard

The single character '.' when used outside of a character set will match any single character.

## 17.2 Anchors

A '^' character shall match the start of a line.

A '\$' character shall match the end of a line.

## 17.3 Marked sub-expressions

A section beginning '(' and ending ')' acts as a marked sub-expression. Whatever matched the sub-expression is split out in a separate field by the matching algorithms. Marked sub-expressions can also be repeated, or referred to by a back-reference.

## 17.4 Non-marking grouping

A marked sub-expression is useful to lexically group part of a regular expression, but has the side-effect of spitting out an extra field in the result. As an alternative you can lexically group part of a regular expression, without generating a marked sub-expression by using '(?: and )', for example '(?:ab)+' will repeat "ab" without splitting out any separate sub-expressions.

## 17.5 Repeats

Any atom (a single character, a marked sub-expression, or a character class) can be repeated with the '\*', '+', '?', and '{}' operators.

The \* operator will match the preceding atom zero or more times, for example the expression  $a^*b$  will match any of the following:

b  
ab  
aaaaaaaab

The + operator will match the preceding atom one or more times, for example the expression  $a+b$  will match any of the following:

ab  
aaaaaaaab

But will not match:

b

The ? operator will match the preceding atom zero or one times, for example the expression  $ca?b$  will match any of the following:

cb  
cab

But will not match:

caab

An atom can also be repeated with a bounded repeat:

$a\{n\}$  Matches 'a' repeated exactly n times.

$a\{n,\}$  Matches 'a' repeated n or more times.

$a\{n, m\}$  Matches 'a' repeated between n and m times inclusive.

For example:

$^a\{2,3\}\$$

Will match either of:

aa  
aaa

But neither of:

a  
aaaa

It is an error to use a repeat operator, if the preceding construct cannot be repeated, for example:

a(\*)

Will raise an error, as there is nothing for the \* operator to be applied to.

## 17.6 Non greedy repeats

The normal repeat operators are "greedy", that is to say they will consume as much input as possible. There are non-greedy versions available that will consume as little input as possible while still producing a match.

\*? Matches the previous atom zero or more times, while consuming as little input as possible.

+? Matches the previous atom one or more times, while consuming as little input as possible.

?? Matches the previous atom zero or one times, while consuming as little input as possible.

{n,}? Matches the previous atom n or more times, while consuming as little input as possible.

{n,m}? Matches the previous atom between n and m times, while consuming as little input as possible.

## 17.7 Back references

An escape character followed by a digit n, where n is in the range 1-9, matches the same string that was matched by sub-expression n. For example the expression:

^(a\*).\*\1\$

Will match the string:

aaabbbaaa

But not the string:

aaabba

## 17.8 Alternation

The | operator will match either of its arguments, so for example: abc|def will match either "abc" or "def".

Parenthesis can be used to group alternations, for example: ab(d|ef) will match either of "abd" or "abef".

Empty alternatives are not allowed (these are almost always a mistake), but if you really want an empty alternative use (?:) as a placeholder, for example:

"|abc" is not a valid expression, but  
"(?:)|abc" is and is equivalent, also the expression:  
"(?:abc)?" has exactly the same effect.

## 17.9 Character sets

A character set is a bracket-expression starting with [ and ending with ], it defines a set of characters, and matches any single character that is a member of that set.

A bracket expression may contain any combination of the following:

### 17.9.1 Single characters

For example [abc], will match any of the characters 'a', 'b', or 'c'.

### 17.9.2 Character ranges

For example [a-c] will match any single character in the range 'a' to 'c'. For Perl regular expressions, a character x is within the range y to z, if it collates within that range; this results in locale specific behaviour.

### 17.9.3 Negation

If the bracket-expression begins with the ^ character, then it matches the complement of the characters it contains, for example [^a-c] matches any character that is not in the range a-c.

### 17.9.4 Character classes

An expression of the form [[:name:]] matches the named character class "name", for example [[:lower:]] matches any lower case character. See character class names.

### 17.9.5 Collating Elements

An expression of the form [[:col.]] matches the collating element col. A collating element is any single character, or any sequence of characters that collates as a single unit. Collating elements may also be used as the end point of a range, for example: [[:ae.]]-c matches the character sequence "ae", plus any single character in the range "ae"-c, assuming that "ae" is treated as a single collating element in the current locale.

As an extension, a collating element may also be specified via it's symbolic name, for example:

[[:NUL.]]  
matches a NUL character.

### 17.9.6 Equivalence classes

An expression of the form `[[=col=]]`, matches any character or collating element whose primary sort key is the same as that for collating element `col`, as with collating elements the name `col` may be a symbolic name. A primary sort key is one that ignores case, accentation, or locale-specific tailorings; so for example `[[=a=]]` matches any of the characters: `a`, `à`, `á`, `â`, `ã`, `ä`, `å`, `À`, `À`, `Á`, `Â`, `Ã`, `Ä` and `Å`. Unfortunately implementation of this is reliant on the platform's collation and localisation support; this feature cannot be relied upon to work portably across all platforms, or even all locales on one platform.

### 17.9.7 Escapes

All the escape sequences that match a single character, or a single character class are permitted within a character class definition, except the negated character classes (`\D` `\W` etc).

### 17.9.8 Combinations

All of the above can be combined in one character set declaration, for example: `[[[:digit:]]a-c[.NUL.]]`.

## 17.10 Escapes

Any special character preceded by an escape shall match itself.

The following escape sequences are also supported:

### 17.10.1 Escapes matching a specific character

The following escape sequences are all synonyms for single characters:

Escape	Character
<code>\a</code>	<code>'\a'</code>
<code>\e</code>	<code>0x1B</code>
<code>\f</code>	<code>\f</code>
<code>\n</code>	<code>\n</code>
<code>\r</code>	<code>\r</code>
<code>\t</code>	<code>\t</code>
<code>\v</code>	<code>\v</code>
<code>\b</code>	<code>\b</code> (but only inside a character class declaration).
<code>\cX</code>	An ASCII escape sequence - the character whose code point is <code>X % 32</code>
<code>\xdd</code>	A hexadecimal escape sequence - matches the single character whose code point is <code>0xdd</code> .



<code>\x{dddd}</code>	A hexadecimal escape sequence - matches the single character whose code point is 0xdddd.
<code>\0ddd</code>	An octal escape sequence - matches the single character whose code point is 0ddd.
<code>\N{name}</code>	Matches the single character which has the symbolic name name. For example <code>\N{newline}</code> matches the single character <code>\n</code> .

### 17.10.2 "Single character" character classes

Any escaped character `x`, if `x` is the name of a character class shall match any character that is a member of that class, and any escaped character `X`, if `x` is the name of a character class, shall match any character not in that class.

The following are supported by default:

Escape sequence	Equivalent to
<code>\d</code>	<code>[:digit:]</code>
<code>\l</code>	<code>[:lower:]</code>
<code>\s</code>	<code>[:space:]</code>
<code>\u</code>	<code>[:upper:]</code>
<code>\w</code>	<code>[:word:]</code>
<code>\D</code>	<code>[^:digit:]</code>
<code>\L</code>	<code>[^:lower:]</code>
<code>\S</code>	<code>[^:space:]</code>
<code>\U</code>	<code>[^:upper:]</code>
<code>\W</code>	<code>[^:word:]</code>

### 17.10.3 Character Properties

The character property names in the following table are all equivalent to the names used in character classes.

Form	Description	Equivalent character set form
<code>\pX</code>	Matches any character that has the property <code>X</code> .	<code>[:X:]</code>
<code>\p{Name}</code>	Matches any character that has the property <code>Name</code> .	<code>[:Name:]</code>

\PX	Matches any character that does not have the property X.	[^[:X:]]
\P{Name}	Matches any character that does not have the property Name.	[^[:Name:]]

#### 17.10.4 Word Boundaries

The following escape sequences match the boundaries of words:

\<	Matches the start of a word.
\>	Matches the end of a word.
\b	Matches a word boundary (the start or end of a word).
\B	Matches only when not at a word boundary.

#### 17.10.5 Buffer boundaries

The following match only at buffer boundaries: a "buffer" in this context is the whole of the input text that is being matched against (note that ^ and \$ may match embedded newlines within the text).

\`	Matches at the start of a buffer only.
\'	Matches at the end of a buffer only.
\A	Matches at the start of a buffer only (the same as \`).
\z	Matches at the end of a buffer only (the same as \').
\Z	Matches an optional sequence of newlines at the end of a buffer: equivalent to the regular expression \n*\z

#### 17.10.6 Continuation Escape

The sequence \G matches only at the end of the last match found, or at the start of the text being matched if no previous match was found.

#### 17.10.7 Quoting escape

The escape sequence \Q begins a "quoted sequence": all the subsequent characters are treated as literals, until either the end of the regular expression or \E is found. For example the expression: \Q\\*+\Ea+ would match either of:

\\*+a

\\*+aaa

#### 17.10.8 Unicode escapes

\C	Matches a single code point: this has exactly the same effect as a "." operator.
----	--

<code>\X</code>	Matches a combining character sequence: that is any non-combining character followed by a sequence of zero or more combining characters.
-----------------	--

#### 17.10.9 Any other escape

Any other escape sequence matches the character that is escaped, for example `\@` matches a literal '@'.

## 17.11 Perl Extended Patterns

Perl-specific extensions to the regular expression syntax all start with (?.

#### 17.11.1 Comments

(`?# ...`) is treated as a comment, its contents are ignored.

#### 17.11.2 Modifiers

(`?imsx-imsx ...`) alters which of the perl modifiers are in effect within the pattern, changes take effect from the point that the block is first seen and extend to any enclosing `)`. Letters before a '-' turn that perl modifier on, letters afterward, turn it off.

(`?imsx-imsx:pattern`) applies the specified modifiers to "pattern" only.

#### 17.11.3 Non-marking grouping

(`?:pattern`) lexically groups "pattern", without generating an additional sub-expression.

#### 17.11.4 Lookahead

(`?=pattern`) consumes zero characters, only if "pattern" matches.

(`?!pattern`) consumes zero characters, only if "pattern" does not match.

Lookahead is typically used to create the logical AND of two regular expressions, for example if a password must contain a lower case letter, an upper case letter, a punctuation symbol, and be at least 6 characters long, then the expression:

```
(?=.*[[:lower:]]) (?=.*[[:upper:]]) (?=.*[[:punct:]]) .{6,}
```

could be used to validate the password.

#### 17.11.5 Lookbehind

(`?<=pattern`) consumes zero characters, only if "pattern" could be matched against the characters preceding the current position ("pattern" must be of fixed length).

(`?<!pattern`) consumes zero characters, only if "pattern" could not be matched against the characters preceding the current position ("pattern" must be of fixed length).

#### 17.11.6 Independent sub-expressions

(?>pattern) "pattern" is matched independently of the surrounding patterns, the expression will never backtrack into "pattern". Independent sub-expressions are typically used to improve performance; only the best possible match for "pattern" will be considered, if this doesn't allow the expression as a whole to match then no match is found at all.

#### 17.11.7 Conditional Expressions

(?(condition)yes-pattern|no-pattern) attempts to match "yes-pattern" if the "condition" is true, otherwise attempts to match "no-pattern".

(?(condition)yes-pattern) attempts to match "yes-pattern" if the "condition" is true, otherwise fails.

"Condition" may be either a forward lookahead assert, or the index of a marked sub-expression (the condition becomes true if the sub-expression has been matched).

### 17.12 Operator precedence

The order of precedence for of operators is as shown in the following table:

Collation-related bracket symbols	[==] [::] [..]
Escaped characters	\
Character set (bracket expression)	[]
Grouping	()
Single-character-ERE duplication	* + ? {m,n}
Concatenation	
Anchoring	^\$
Alternation	

### 17.13 What gets matched

If you view the regular expression as a directed (possibly cyclic) graph, then the best match found is the first match found by a depth-first-search performed on that graph, while matching the input text.

Alternatively:

the best match found is the leftmost match, with individual elements matched as follows;

Construct	What gets matched
-----------	-------------------

AtomA AtomB	Locates the best match for AtomA that has a following match for AtomB.
Expression1   Expression2	If Expression1 can be matched then returns that match, otherwise attempts to match Expression2.
S{N}	Matches S repeated exactly N times.
S{N,M}	Matches S repeated between N and M times, and as many times as possible.
S{N,M}?	Matches S repeated between N and M times, and as few times as possible.
S?, S*, S+	The same as S{0,1}, S{0,UINT_MAX}, S{1,UINT_MAX} respectively.
S??, S*?, S+?	The same as S{0,1}?, S{0,UINT_MAX}?, S{1,UINT_MAX}? respectively.
(?>S)	Matches the best match for S, and only that.
(?=S), (?<=S)	Matches only the best match for S (this is only visible if there are capturing parenthesis within S).
(?!S), (?<!S)	Considers only whether a match for S exists or not.
(?(condition)yes-pattern   no-pattern)	If condition is true, then only yes-pattern is considered, otherwise only no-pattern is considered.

## 17.14 Modifiers

The perl "smix" modifiers can either be applied using a (?smix-smix) prefix to the regular expression.

# 18 Documents referenced

Perl 5.8 Regular Expression syntax is documented at <http://perldoc.perl.org/perlre.html>.

# 19 Notes

19.1.1 [Configuration]

## 20 Version history

### 20.1 Software Version

Version No	Date	Details	Name
4.07.02	14 Oct. 09	Version current at time of generation of this document.	Richard Kerry
4.07.04	08 Sep 2010	Packet-mode revertive handling fixed for use with CSI32 with heartbeat messages.	Richard Kerry
4.08.01	14 Jan 2011	Changed messaging system. Added Status InfoDriver.	Richard Kerry
4.09.01	31 March 2011	Added per-user port specification. Added filtering by address on read. Added filtering by data contents on both write and read.	Richard Kerry
4.09.11	24 Nov 2011	Added handling of blank and comment while reading access control specs. Fixed length limits in ALR command.	Richard Kerry
4.09.12	16/05/2012	Fixed length issue with passwords. Now definitely 24. This affects both passwords and user-name. Fixed bug in length of data being sent. Unnecessary extra byte was being sent which confused V3BncsLink.	Richard Kerry
4.09.13	28/05/2012	Enabled vertical scrolling within Users dialog.	Richard Kerry
4.09.14	05/06/2014	Addresses issues with diagnostic messages when '%' characters are encountered. Adds option for per-session log files.	Richard Kerry
4.09.15	10/06/2014	Further fixes to issues with diagnostic messages when '%' characters are encountered.	Richard Kerry

4.09.16	24/11/2014	<p>Quantity of diagnostic messages reduced when switched off. Note – not reduced to zero, but should ensure that the system doesn't show a message for every revertive handled.</p> <p>Current Users dialog changed so it remains visible while killing users, ie until Close is done manually.</p> <p>There is now also a Refresh button to allow the display to be redisplayed if the list has changed while the dialog is showing.</p> <p>The number of possible simultaneous connections has been increased to 64.</p> <p>Added option of displaying milliseconds in the times in the daily log file.</p>	Richard Kerry
4.09.17	10/02/2015	<p>Users dialog may now be resized. (Bug 2630)</p> <p>Some uses of C strings changed to STL strings as required by changed common functions recently modified during Fabian Link work.</p>	Richard Kerry
4.09.18	26/03/2015	<p>Reinstated writing of incoming A-series commands to the "old" log file.</p> <p>Outgoing revertive commands now also written to the "old" log file.</p>	Richard Kerry
4.09.19	14/04/2015	<p>Added diagnostics to aid finding the occasional crash when closed by Workstation Manager. Also fixed an issue which may have been the cause of this problem.</p> <p>Some changes to formats of messages in the various log files. These are mainly to remove duplication of times, dates and addresses.</p>	Richard Kerry

4.09.20	16/03/2016	Removes old URL from "HELP" commands	DGY
		<p>Speeded up logging to file and writing diagnostic messages to screen. This has been done by using a separate thread to do the writes to file and screen. Also there are some extra menu controls and configuration entries to control whether diagnostics relating to commands and revertives are written.</p> <p>Stopped using network messages direct from CSI to generate packed revertive messages. These are now generated from the regular revertive messages, as always used to generate the normal format revertive messages. This means that they properly take account of the results of any packet recovery actions.</p> <p>Along with this change the use of registration information has been tightened up. Previously packed messages and database change messages could contain information relating to slots which had not been registered for (although other slots on the same device had). If the original behaviour is still required there is a compatibility option that may be enabled.</p> <p>There is also a compatibility option relating to the maximum length of packed messages that may be generated. Fabian Link prior to version 4.7.12 has a smaller buffer for this than is expected, and the method Fabian Server now uses can fill the buffer to an extent that has been seen to crash Fabian Link. If this occurs there is a compatibility setting that may be used to stop it.</p> <p>Bugs affected: 2765, 2496, 2701.</p>	Richard Kerry
4.09.21	27/04/2016	<p>It is now possible for all folders required for logging (or the database and resource management systems) to be created automatically. (Bug 2788)</p> <p>Extended use of the Access Control system to driver database related commands and device profile commands. (Bug 2794)</p> <p>Commands affected are: ADL, ADP, ALR, ARN, ARI, ARM, ACD. Also responses: ADC, UR.</p>	Richard Kerry



4.09.22	09/05/2016	<p>Extended use of the Access Control system to a number of commands that didn't do so previously. Commands affected are: ARF, ARP, AIF, AIP, AGF, AGP.</p> <p>Changed ACD to provide no response at all if the Access Control system does not provide read access to the device queried.</p> <p>(Bug 2794)</p>	Richard Kerry
4.09.23	30/06/2016	<p>Commands and responses relating to router databases now ignore indices of registrations and per-index access control.</p> <p>If any index is registered for on a device and access is permitted, then a database command relating to any index in that device will work.</p> <p>Registration compatibility mode amended with this in mind.</p> <p>Whether read or write Access is checked depends on the nature of the command or response.</p> <p>(Bugs 2794, 2799, 2800)</p>	Richard Kerry
4.09.24	03/10/2016	<p>The SUN command will now log-out any logged-in user (but not close the TCP socket). This is to prevent any privileges available to the old user being made available to a new user for whom they may not be appropriate.</p> <p>(Bug 2829)</p> <p>The GSF command has been disallowed as it has been discovered that it can allow access to any file. (Bug 2830)</p>	Richard Kerry
4.09.25	10/10/2016	<p>The SUP command, if used at any time when a user is validly logged in, will log-out the user (but not close the TCP socket).</p> <p>(further work on Bug 2829)</p>	Richard Kerry
4.09.26	18/10/2016	<p>Corrected which parts of data are cleared during new automatic log-offs in SUN and SUP. This is to correct the information that is shown by GSU and in the Manage Users dialog.</p> <p>(further work on Bug 2829)</p>	Richard Kerry
4.09.27	24/10/2016	<p>Corrected GSU command which should only show actual logged-in user sessions.</p> <p>(further work on Bug 2829)</p>	Richard Kerry

## 20.2 Document Version

Version No	Date	Details	Name
	Feb 09	Updated template	A Atkin
	04 Nov. 09	First version using revised document template.	Richard Kerry
	08 Sep 2010	Some misleading information or incomplete sections marked or removed.	Richard Kerry
	09 Dec 2010	Clarified that there is no Resilience functionality here.	Richard Kerry
	14 Jan 2011	Added note about change to messaging system.	Richard Kerry
	14 Jan 2011	Added Status InfoDriver.	Richard Kerry
	31 Mar 2011	Added per-user port specification. Added filtering by address on read. Added filtering by data contents on both write and read.	Richard Kerry
	24 Nov 2011	Added details of NONE, blank and comment while reading access control specs.	Richard Kerry
	16/05/2012	Updated corresponding to latest features in application. Changed branding following change from Siemens to Atos.	Richard Kerry
	26/03/2015	Change of PDF generation tool allows bookmarks to be generated, but no longer watermark.	Richard Kerry
	17/11/2015	Clarified usage of Telnet (it isn't) and some issues relating to command entry.	Richard Kerry
	12/07/2016	Removed reference to test server as no longer available.	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](mailto:collediacontrol.it-solutions.gb@atos.net)