GXIO.INI Sections and Settings

1. Introduction

The GXIO.INI file contains only the settings recognised by the Protocol Module, GXIO.EXE. For most configurations only a single GXIO.INI file will be present. However, it is possible to override the fixed name "GXIO.INI" with variable name files:

GXIO.INI	Default INI file for GXIO.EXE;
gxio_ini_file	Optional host-specific override file that may be defined in the GXHOSTS.INI Customisation File Group. Settings in here override the same setting (if present) in GXIO.INI;
gxio.xxx	Optional, free-format Hosts file specified by the /I command line argument. A setting in *gxio.xxx* will override the same setting (if present) in GXIO.INI.

Important Note: The GXIO.INI file override algorithm is different from the GX.INI file override algorithm. Only one GXIO.INI override file can be passed on to GXIO.EXE. If **both** the /I command line option **and** a HostID% override have been specified then the override from the GXHOSTS file takes precedence (i.e. the /I command line argument is effectively ignored).

The GXIO.INI file contains a subset of the settings available in the GSMWIN32.INI file.

The installation of the Global Windows Explorer (GX) creates a GXIO.INI file. It is not normally necessary to create this file. The GXIO.INI file is always initialised correctly when GX is installed and, in general, there should be no need to amend it, unless you are advised to do so in order to trouble-shoot a problem, or when adding a [loginevents] section (see section 5) when configuring GX to connect to a Unix host.

2. GXIO.INI File Sections

The following sections are supported in the GXIO.INI file:

Section	Description
[device]	The [device] section in GXIO.INI is used to specify the type of interface used to access the host system.
[miscellaneous]	Other miscellaneous settings.
[loginevents]	Only required when connecting GX to a Unix host.

3. Settings in the GXIO.INI File [device] Section

For GSMWIN32.EXE, the [device] section in GSMWIN32.INI is used to specify the type of interface used to access the host system. This equivalent section in the GXIO.INI file is

almost redundant in GXIO.INI since the only interface supported is ‘Telnet’, but other parameters regarding the interface may have to be configurable.

3.1 Interface

THIS OPTION MUST BE SET TO "Telnet".

3.2 NoDelay

This setting indicates whether the TCP/IP “SetSocket” function should be used to enable the Winsock “No Delay” option. Enabling this option will optimise the performance on some network configurations.

Although the default setting is Off (for compatibility with very early versions of GXIO), this option is set to "On" in the default GXIO.INI file provided. **FOR VERSIONS OF WINDOWS OTHER THAN ALL VARIANTS OF VISTA AND ONWARDS, YOU ARE STRONGLY ADVISED TO SET THIS OPTION TO "On".** For Windows Vista and onwards, the NoDelay setting must be disabled to prevent an error 10022 on connection.

3.3 KeepAlive

This setting indicates whether the TCP/IP “SetSocket” function should be used to enable the Winsock “Keep Alive” option.

For Windows Vista and onwards, the KeepAlive setting must be disabled to prevent an error 10022 on connection.

The default setting is Off.

3.4 Reconnection

This setting must be enabled for the “Reattach” option in the GXHOSTS.INI file to be recognized.

The default setting is On.

3.5 KeepAliveTimeout

This setting specifies the delay period, in seconds, of the optional “keep alive” message sent regularly by GXIO.EXE to the host. This option may be required on some (most) TCP/IP connections that include an inactivity timeout and where the Socket-level “KeepAlive” option (see section 3.3) is either disabled or spoofed by a software layer.

Important Note: This option may be required for some TCP/IP connections that disconnect after periods of inactivity. For example, when running GX over a VPN connection this option may have to be set to a nonzero value (e.g. 30) to prevent GX hangs and enforced reconnections.

The default value is 0, which effectively disables this option.

3.6 ConnectionStatusIcon

If this setting is enabled the GXIO.EXE connection status icon is displayed in the system tray.

The default setting is On.

The connection status icon is displayed in one of the four colours described in the following table:

GXIO.EXE Icon colour	Meaning
Red	No connection to the host.
Cyan (light blue)	Connection to host established and the link is operating normally.
Yellow	Connection to host established and either GX.EXE or GXIO.EXE are being downloaded.
Blue	Connection to host established but the link is running in single character mode. This colour appears as a diagnostic indicator to show that GXIO.EXE is running in an inappropriate mode. GXIO.EXE starts the connection in single character mode for the initial handshake of encryption and reattach information and after this is complete runs in multi-character mode (cyan or yellow).

3.7 ConnectionStatusWindowAlwaysOnTop

If this setting is enabled the connection status dialogue box remains permanently visible.

The default setting is On.

3.8 EncryptionAcknowledgementTimeout

This setting controls the maximum response time assumed for the encryption sequence.

The default value is 2000.

3.9 SaveTelnetDiagnostics

If this setting is enabled the telnet diagnostics are saved in the log-file *telnetlog.txt*.

The default setting is On.

4. Settings in the GXIO.INI File [miscellaneous] Section

As the name implies this section is used for a variety of miscellaneous settings.

4.1 ShowGXIOModule

This option is only required for problem trouble-shooting.

The default value is Off.

4.2 DisplayRDATimer

THIS OPTION MUST NOT BE MODIFIED.

The default value is 20.

4.3 Enable8BitMode

THIS OPTION MUST NOT BE MODIFIED.

The default value is On.

4.4 ExitOn$BYE

THIS OPTION MUST NOT BE MODIFIED.

The default value is On.

4.5 ExitOn$E

THIS OPTION MUST NOT BE MODIFIED.

The default value is Off.

4.6 InterpretIACSequences

THIS OPTION MUST NOT BE MODIFIED. Note that Windows/Unix option in the HostID% line in GXHOSTS.INI file (see section 2.1) effectively overrides this option.

4.7 MAPIInterface

This option is only used if the pre GSM SP-6 EMAIL$ sub-routine is being used.

4.8 NonGXDataInputWarning

This option is only required for problem trouble-shooting.

The default value is Off.

4.9 WebBrowser

This option is only used if the pre GSM SP-6 WWW$ sub-routine is being used.

4.10 TempDirectory

This option is only used if the pre GSM SP-6 WWW2$ sub-routine is being used.

4.11 LogDisplays

This option is only required for problem trouble-shooting. If this option is enabled the diagnostic file, DISPLOG.BIN, will be created in the current directory.

The default value is Off.

4.12 LogKeystrokes

This option is only required for problem trouble-shooting. If this option is enabled all keystrokes will be logged in the diagnostic file, KEYLOG.BIN, will be created in the current directory.

The default value is Off.

4.13 LogFolder

Normally, if the relevant diagnostic options are enabled, a new GXIO.EXE session will overwrite existing DISPLOG.BIN and KEYLOG.BIN log files; and append to the existing TELNETLOG.TXT and GXINT.TXT files. If the LogFolder setting is present in GXIO.INI and set to a folder name, uniquely named DISPLOG.BIN, KEYLOG.BIN, TELNETLOG.TXT and GXINT.TXT files will be saved in the specified folder. The unique file names are formed by appending the local host name to the file name (e.g. telnetlog_TISCGLOB.txt); and the time GXIO.EXE was started to the DISPLOG and KEYLOG file names (e.g. DISPLOG_TISCGLOB_09-07-07_13.45.57.bin).

4.14 DisplayLogMessages

If this option is enabled all the messages that are normally written to the LOG.TXT file are displayed in the optional display progress window which is enabled by the DisplayMessages or DisplayMessages% settings within the GXHOSTS.INI file.

The following table summarizes the combinations of the DisplayMessages and DisplayLogMessages settings:

DisplayMessages	**DisplayLogMessages**	**Comments**
Off	Off	No display messages
Off	On	No display messages
On	Off	GSM start-up dialogue appears in the Display message Window
On	On	Low-level connection dialogue between GX.EXE and GXIO.EXE appears in the Display Message Window

The default value is Off.

4.15 IgnoreGUISequences

This setting forces the GXIO.EXE module to ignore all GUI specific sequences sent from the host system.

THIS OPTION MAY BE REQUIRED WHEN RUNNING SOME 16-BIT SPEEDBASE APPLICATIONS UNDER GX.

The default setting is On.

4.16 MaximumMessageCountDifference

This setting can be used to specify the maximum number of messages sent by GXIO.EXE but not fully processed by a GX partition.

The default value is 256.

4.17 MessageCountTimeout

This setting can be used to specify the length of pause when maximum message count exceeded (ms).

The default value is 1000.

4.18 PrinterTempDirectory

This setting can be used to specify the name of the temporary directory used for some printing functions.

No default is available.

4.19 PrinterTempDirectoryStaleFileTimeout

This setting specifies the maximum amount of time (in minutes) that a print file will remain available in the PrinterTempDirectory. A value of 0 disables the timeout and ensures the directory is always cleared on receipt of a new file.

There is no upper limit on the timeout value so a 1 hour timeout can be configured with a value of 60 and a one day timeout with a value of 1440.

The default value is 0 (i.e. the directory is always cleared on receipt of a new file).

5. Settings in the GXIO.INI File [loginevents] Section

When the HostID% setting in the GXHOSTS Customisation File Group includes the Unix host option the initial window displayed by GX is used to accept the Unix user id and password (i.e. rather than the Global operator-id and password as is the case when GX is connecting to a GSM (Windows) host). The Unix user id and password are sent to the Unix host when certain events occur during the Unix login procedure. These events are triggered by GXIO detecting keywords (or phrases) sent from the host (e.g. login:).

The keyword checking is initiated in two ways:

1. After a timeout has expired (counting from the last character received);

2. When a <CR> character is received, unless an event action has already been triggered for that line.

Note that the keyword checking only takes place on the current line (hence the need to check on every <CR> character).

An event can be defined in the GXIO.INI file via settings in the [loginevents] section. Each entry consists of a event keyword (i.e. the string checked for on a line) and an event action which specifies what to do once the associated event has been detected.

5.1 LoginKeyword

There are two pre-defined events to handle the login and password prompts generated by the Unix host. For these pre-defined events only the keyword needs to be specified by the LoginKeyword and PasswordKeyword settings respectively. A typical LoginKeyword setting would be "login:". The user id string entered on the GX logon window is sent to the host with a <CR> terminator. Note that if the LoginKeyword setting is not defined in the GXIO.INI file then GXIO will display an error message and terminate the login attempt.

5.2 PasswordKeyword

There are two pre-defined events to handle the login and password prompts generated by the Unix host. For these pre-defined events only the keyword needs to be specified by the LoginKeyword and PasswordKeyword settings respectively. A typical Password Keyword setting would be "password:". The password string entered on the GX logon window is sent to the host with a <CR> terminator. Note that if the PasswordKeyword setting is not defined in the GXIO.INI file then GXIO will display an error message and terminate the login attempt.

5.3 EventKeyword% & EventAction%

Up to ten extra events can be defined by pairs of settings, EventKeyword% and EventAction% (for % in the range 1 to 10). These settings allow extra dialogue generated by the Unix system to be handled. For example, an incorrectly entered password may result in a message "Login incorrect" from Unix. This phrase could then be used as a keyword and the associated action could display a message on the GX logon window to give the user some information. The action is specified in the following way:

Command,parameter

Two commands are supported, Send and Display. The Send command transmits the parameter string as it stands, except the ':' character which signifies a <CR>. The Display command passes the parameter string to the GX logon window which displays it in a Window message box.

Note that event keywords and actions must be entered in pairs, otherwise they will be ignored.

5.4 MonitorLoginProcedure

This setting controls whether the GXIO window will be shown during the login procedure to Unix and is useful for debugging problems with the event keywords/actions set up in the INI file.Note that the GXIO window is fully active so it is possible to enter characters if the start up procedure stalls.

The default setting is Off.

5.5 CaseSensitiveKeywordComparison

This setting specifies whether the keywords entered should be compared as they are (i.e. the keyword must match the displayed prompt/message exactly) or whether a case insensitive comparison should take place (in fact all keywords/message are converted to upper case before the comparison takes place).

The default setting is Off.

5.6 An Example [loginevents] section

The following [loginevents] section has been used to login to a SCO Unix system. Note that the login and password keywords are setup to the exact string displayed (i.e. including the ':' character). There are two additional events defined. The first traps a prompt which stops the login procedure just before the user's .profile is executed. This prompt is actioned by sending a single <CR> which allows the login process to continue. The second additional event traps the condition where an invalid password has been entered and the action displays a message on the GX logon window. This is especially important if the GXIO window can't be seen to give the user information about why the login has failed.

Note also that the case sensitive comparison has been enabled because of this particular Unix server a portion of the text displayed after the password has been accepted contains the text "LOGIN:". This would trigger the login event and so a spurious user id would be returned. Making the comparison case sensitive means the all upper case "LOGIN:" display is ignored.

```
[loginevents]
MonitorLoginProcedure=On
CaseSensitiveKeywordComparison=On
LoginKeyword=login:
PasswordKeyword=Password:
EventKeyword1=Press return
EventAction1=Send,:
EventKeyword2=Login incorrect
EventAction2=Display,Login incorrect
```

5.7 GX Connections to GSM (Unix)

The initial connection between GX and a GSM (Unix) host is radically different from the connection between GX and a GSM (Windows) host. When GX connects to a GSM (Windows) host, GX forges an immediate connection to the NETWORK controller within

GLOBAL.EXE. The operator-id and password that are (normally) supplied by the GX operator are passed directly to $AUTH, $PASSWD or $AUTH32 (i.e. to whichever Global authorisation program that has been customised using $CUS).

When GX connects to a GSM (Unix) host, GX connects to the Unix telnetd daemon and interacts with the Unix login processing. The operator-id and password that are (normally) supplied by the GX operator are used to login to Unix. Once the Unix login has been completed successfully a combination of entries in the [loginevents] section of the GXIO.INI file and the Unix login script for that operator are used to start a GSM (Unix) session by (normally) running the "global" command.

This section contains a worked example of a GX to Unix, and thence GSM (Unix) login.

The GXHOSTS.INI file contains the following entries for the target Unix host:

```
HostID19=192.168.1.228,tiswyse (normal)
UnixHost19=On
GXIOINIFile19=gxiowyse.ini
DisplayMessages19=On
```

In this example "HostID19" is just the 19'th entry in the example GXHOSTS.INI file. There is no relevance to this number - any HostID number can be used for a Unix host. However, note the use of the mandatory "UnixHost" setting to specify a Unix host. Note also the use of the "DisplayMessages" option, which is **strongly recommended** when connecting to Unix hosts.

Finally, note the host-specific GXIO.INI file, gxiowyse.ini. In the example GXHOSTS.INI chosen, the use of host-specific GXIO.INI is required because HostID1 to HostID18 specify connections to GSM (Windows) hosts (which require a different GXIO.INI file as explained below).

The [loginevents] section of the gxiowyse.ini file contains the following entries for the target Unix host:

```
[loginevents]
;MonitorLoginProcedure=On
CaseSensitiveKeywordComparison=On
LoginKeyword=login:
PasswordKeyword=Password:
EventKeyword1=Press return
EventAction1=Send,:
EventKeyword2=Login incorrect
EventAction2=Display,Login incorrect
EventKeyword3=Key <CR> to continue:
EventAction3=Send,GX:
```

In this example, the generic GXIO.INI, which is used to connect to all the GSM (Windows) servers that are also configured in the GXHOSTS.INI file, does not require a [loginevents]

section. Note also, that it **may** be necessary, for aesthetic reasons, to maintain separate GX.INI files for the GSM (Unix) and GSM (Windows) servers.

The "MonitorLoginProcedure" option, which has been commented out in this example, is very useful when troubleshooting the initial hand-shaking between GX and the Unix login script. It displays the detailed dialogue and allows responses to be made to unanticipated prompts.

In the example gxiowyse.ini file the EventAction1 and EventAction2 text-strings are both required to provide responses to possible prompts from the server-specific Unix login dialogue. These example responses will almost certainly have to be changed to conform to the login dialogue for a particular target Unix server.

The EventAction3 text is included solely to provide a response to the Unix login script for the GX users. A snippet from the Unix login script (e.g. .profile) for a typical GX user is shown below.

```
unset GLTERM
echo "Key <CR> to continue:"
read FRED
if test "$FRED" = "GX"
then
      GLTERM=gx;export GLTERM
fi
global -x
```

Important Note-1: The terminal type for a GX user **MUST** be set to 911. Usually, this is achieved by setting a GLTERM variable in the start-up script to "gx" and mapping the Unix terminal name "gx" to a Global terminal code of 911 in the Systems file.

Important Note-2: The global -x **MUST** be enabled for all GX users. This is required to disable XON/XOFF flow-control characters that could otherwise interfere with the GX protocol.