

Updating GX.EXE, GXIO.EXE & GX.INI Automatically

1. Introduction

This document describes a new feature in GSM and GX that allows GX running on a thin-client PC to automatically download a new version from a central "updates" directory on the server running the Global Client.

This technique avoids the need to propagate new versions of GX.EXE (and GXIO.EXE) around **all** the thin-client PC's on a LAN or WAN.

The update-able GX.EXE (and GXIO.EXE) files are supplied by Global Business Systems in a compressed format with the following naming convention:

GXnn.EX_
GXIOnn.EX_

where *nn* is the numeric version number of a "full" release (e.g. 26), or:

GXnna.EX_
GXIOnna.EX_

where *nna* is the alpha-numeric version number of an "interim" release (e.g. 25t).

Note that the name of the compressed file includes the version number of the GX.EXE (or GXIO.EXE) contained within. It is vital that the file naming convention collates the compressed files in strict version number order.

Important Note: If a compressed file supplied by Global Business Systems is renamed then the results will be unpredictable (e.g. renaming GX25.EX_ to FRED25.EX_ will hide it from the auto-update mechanism; renaming GX25.EX_ to GX30.EX_ will artificially "promote" its version number). **DO NOT RENAME THE COMPRESSED GX.EXE OR GXIO.EXE FILES.**

2. Pre-Requisites

The following software versions are required:

GSM SP-6, or later;
GLOBAL.EXE V3.3, or later;
GX.EXE V2.5, or later

Important Note: GSM SP-6 included a bug which prevents the auto-update mechanism functioning if the case of the alphabetic character in the alpha-numeric portion of a compressed filename does not match the case of the running GX. This problem has been fixed in GSM SP-7.

3. Enabling the Automatic GX Update Option

3.1 Enabling the Automatic GX Update Option on GSM (Windows)

On the server running GLOBAL.EXE allocate a directory called "gxupdates" in the current GSM (Windows) directory. For example, if GSM (Windows) is installed in C:\GSM\ then ensure that directory C:\GSM\GXUPDATES\ is allocated.

The name of this directory can be overridden by the following registry setting:

..\Global\Client\ServicePacks\GXUpdateDirectory

It should **not** normally be necessary to use this registry setting to change the default from "gxupdates".

For GLOBAL.EXE V3.6, and later:

The following registry setting is also available:

..\Global\Client\ServicePacks\PromptForGXUpdate

By default, the operator is not given the option to download a new GX - if a new GX (or GXIO) is available, the download occurs automatically. If this option is "Off" (the default setting) the operator does not get the opportunity to cancel the automatic download of GX (or GXIO). If this option is "On" each operator is asked to confirm the download of the new GX (or GXIO) and thus gets a chance to skip the download. There should not be any need to change the default "GX update" behaviour (i.e. PromptForGXUpdate=On should not normally be required).

For GSM SP-8, the following registry setting is also available:

..\Global\Client\ServicePacks\PromptForGXReload

By default, the operator is not given the option to reload GX immediately. The GX reload occurs automatically after the operator has clicked on a single "OK" button. This default "reload always" behaviour can be changed by setting the "PromptForGXReload" to "On" in which case, each operator will be given the option to reload GX immediately or to continue the current GX session with the "old" version of GX. There should not be any need to change the default "auto re-load" behaviour (i.e. PromptForGXReload=On should not normally be required).

For GSM SP-15 (and GLOBAL.EXE V3.8) the following registry settings are also available:

..\Global\Client\ServicePacks\SilentGXReload
..\Global\Client\ServicePacks\SilentGXUpdate

As the names imply, setting these settings suppress the relevant message box completely (see sections 5 and 6).

3.2 Enabling the Automatic GX Update Option on GSM (Unix)

The automatic GX Update Option on GSM (Unix) is only supported on GSM SP-12 (and later) and BACNAT variant V3.320 (and later).

On the Unix server allocate a directory called "gxupdates" in the current GSM (Unix) directory, \$GLDIR. For example, if GSM (Unix) is installed in /u2/global then ensure that directory /u2/global/gxupdates is allocated (i.e. the new gxupdates directory is at the same level as the familiar data, bin etc.).

The name of this directory **cannot** be overridden.

The following shell variable is available:

GSM_PROMPTFORGXUPDATE

By default, the operator is not given the option to download a new GX - if a new GX (or GXIO) is available, the download occurs automatically. If the GSM_PROMPTFORGXUPDATE shell variable is set (to any string) the operator is asked to confirm the download of the new GX (or GXIO) and thus gets a chance to skip the download. See section 5.

The following shell variable is also available:

GSM_PROMPTFORGXRELOAD

By default, the operator is not given the option to reload GX immediately. The GX reload occurs automatically after the operator has clicked on a single "OK" button. This default "reload always" behaviour can be changed by setting the GSM_PROMPTFORGXRELOAD shell variable (to any string) in which case, each operator will be given the option to reload GX immediately or to continue the current GX session with the "old" version of GX. See section 6.

The following shell variable is reserved for future use:

GSM_UPDATEGXFILES

For GSM SP-15 (and BACNAT variant 3.329, and later) the following shell variables are also recognised:

GSM_SILENTGXUPDATE
GSM_SILENTGXRELOAD

As the names imply, setting these variables suppress the relevant message box completely (see sections 5 and 6).

4. What To Do With New Versions of GX.EXE and GXIO.EXE

All new versions of GX.EXE (and GXIO.EXE) provided by Global Business Systems in the compressed file format (i.e. as GXnn.EX_ or GXnna.EX_ or GXIOnn.EX_ or GXIOnna.EX_) should simply be copied to the "gxupdates" directory. The next time an operator logs in to GSM using GX the download of the new version to the GX thin-client PC will be completely automatic (see below).

Important Note: Do not attempt to use a compression tool (e.g. WinZip) to compress GX.EXE to create your own GXnn.EX_. The compressed files supplied by Global Business Systems are provided in a special file format. See Appendix A for further details.

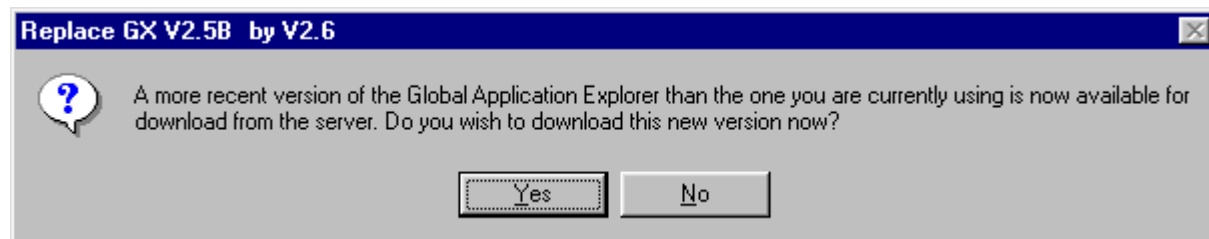
Important Note for GSM (Unix) configurations: The case of the filenames is not important (e.g. gx31a.exe_ and GX31B.EXE_ are both valid, as is GX31C.eX_ etc.). However, the results will be unpredictable if two files, with different filenames, for the same GX, or GXIO, version (e.g. GX31A.EX_ and GX31a.EX_) are present in the gxupdates directory.

Do not attempt to use a compression tool (e.g. WinZip) to compress GX.EXE to create your own GXnn.EX_. The compressed files supplied by Global Business Systems are provided in a special file format. See Appendix A for further details.

5. Download Dialogue

A special overlay in the GSM start-up process, which is only invoked on GX thin-clients, checks whether the most recent compressed file version of GX in the "gxupdates" directory is more recent than the currently running GX.EXE. The same version checking is performed for GXIO.EXE.

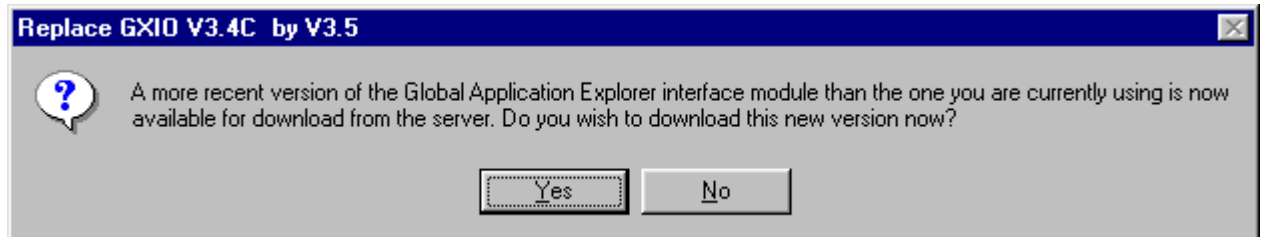
If the "gxupdates" directory contains a more recent version of GX.EXE than the one that is currently executing the following dialogue box will appear:



Note the "current" and "update" GX versions are displayed on the caption bar.

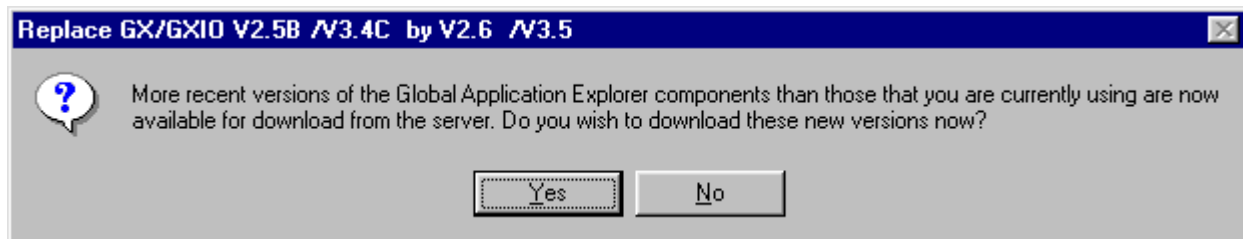
This dialogue box, with a "Yes", "No" selection will only appear if the "PromptForGXUpdate" option is set to "On" etc.

If the "gxupdates" directory contains a more recent version of GXIO.EXE than the one that is currently executing the following dialogue box will appear:



Once again, note that the "current" and "update" GXIO versions are displayed on the caption bar.

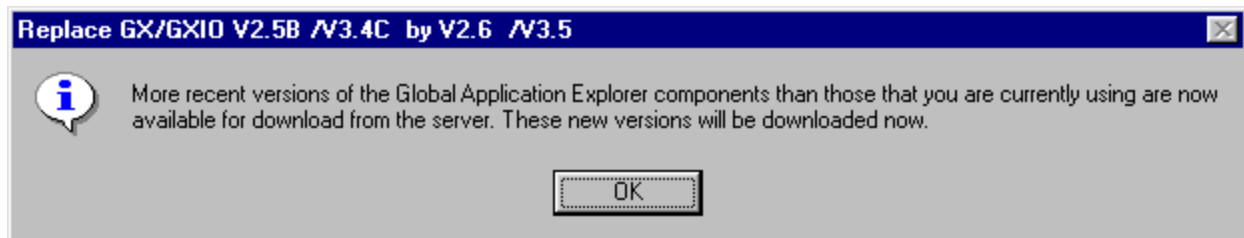
If the "gxupdates" directory contains more recent versions of both GX.EXE and GXIO.EXE than the versions that are currently executing the following dialogue box will appear:



Note that all the relevant version numbers are displayed on the caption bar.

If the "Yes" button is selected, the appropriate file(s) will be downloaded by GX itself from the "gxupdates" directory on the server running GLOBAL.EXE to the "current" directory on the GX thin-client PC. A progress bar is displayed during the file transfer.

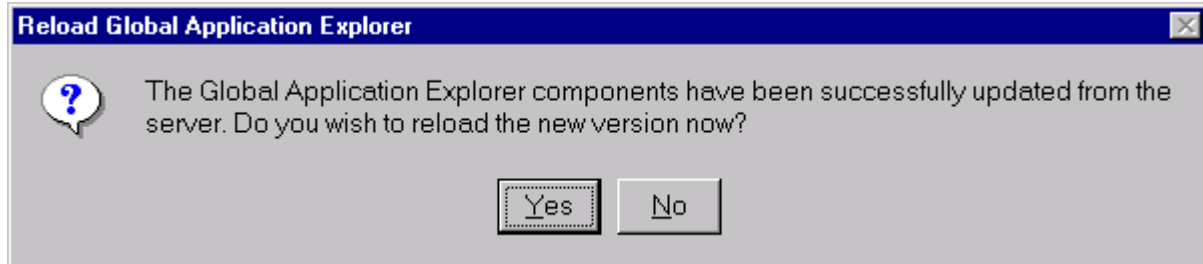
If the "PromptForGXUpdate" option is set to "Off" etc. (the default) the Yes/No dialogue box is replaced by a simple "OK" dialogue box. For example:



Once the file transfer has completed GX will expand (i.e. uncompress) the downloaded file(s) and prepare to repeat the login using the new GX.EXE (and GXIO.EXE).

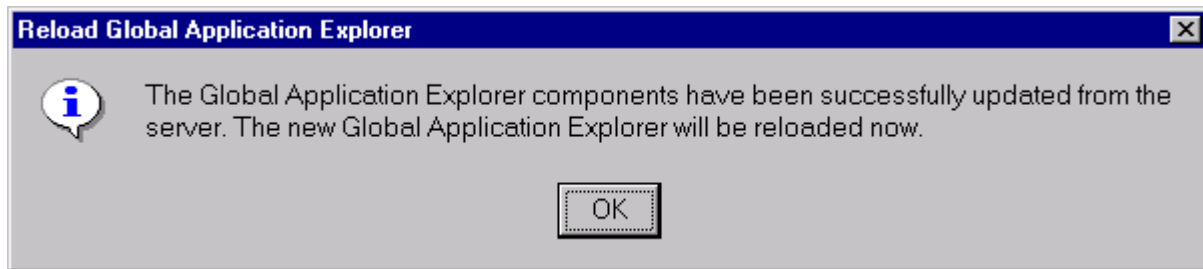
6. Reload Dialogue

The subsequent login, using the new components, is completely automatic and will use the same login name and password as the initial login. Before the GX reload commences the following dialogue box is displayed:



This dialogue box, with a "Yes", "No" selection will only appear if the "PromptForGXReload" option is set to "On" etc.

If the "PromptForGXReload" option is set to "Off" etc. (the default) the Yes/No dialogue box is replaced by a simple "OK" dialogue box. For example:



7. Recovering From Errors

If the download, file expansion or new version overwrite fails for any reason an error message with an error code will be displayed. Please quote the error code when reporting problems to the Hotline.

Before the automatic download is started GX.EXE copies the "current" version of itself (and GXIO.EXE if necessary) to a directory called "LastKnownGood" in the GX directory. Any error condition during the automatic update process, which prevents GX from reloading, can be simply recovered by copying the files from LastKnownGood back to the GX directory.

8. Updating GX.INI

The combination of GSM SP-17, GLOBAL.EXE V3.9e (or higher) and GX V3.5o (or higher) support the option to apply updates to local GX.INI file(s) from a central, template GX.INI file.

Important Note: Unlike the GX.EXE and GXIO.EXE upgrades, which both involve **copying** the relevant *.EX_ file(s), the GX.INI upgrade involves the selective **addition** of

entries present from the central, template file to the local GX.INI file. This option does **not** involve the overwriting of the local GX.INI file.

8.1 Enabling the GX.INI Upgrade Process

This GX.INI update process is simply enabled by the presence of the template GX.INI file in the gxupdates folder. No registry setting is required to enable the GX.INI upgrade process (although four settings, see section 8.3, are available to customise the process).

8.2 Details of the GX.INI Upgrade Process

The template GX.INI file (in the gxupdates folder on the server) should just contain the sections and settings that are to be **added** to the existing "live" GX.INI files on the GX PC's. For example:

```
; example of a template GX.INI file
[newsection1]
newsetting1=on
newsetting2=off
```

Only new settings will be added to a local GX.INI file. If a local GX.INI file already contains an existing setting, that setting will **NOT** be overwritten by the same setting from the template GX.INI file on the server. For example, if a local GX.INI file contains before a GX.INI upgrade:

```
; example of a local GX.INI file
[newsection1]
newsetting1=off
```

then after a GX.INI upgrade, from the template GX.INI file described above, the local GX.INI file will contain:

```
; example of a local GX.INI file
[newsection1]
newsetting1=off
newsetting2=off
```

i.e. "newsetting2", together with its associated value, has been added to the local GX.INI file. The value associated with existing "newsetting1" has not been altered.

There is currently no mechanism to force a change to an existing setting in a local GX.INI file.

There is currently no mechanism to force a "setting-less" section into a local GX.INI file. For example, if the template GX.INI file contains:

```
; example of a template GX.INI file
[emptynewsection]
```

```
[newsection1]
newsetting1=on
newsetting2=off
```

the "emptynewsection" section will not be added to the local GX.INI files. A dummy setting will be required to force the addition of a new section. For example:

```
; example of a template GX.INI file
[emptynewsection]
thisisadummysetting=on
```

```
[newsection1]
newsetting1=on
newsetting2=off
```

By default, the addition of a new setting into a local GX.INI does not force a reload of GX. However, settings can be marked as "this setting forces a GX reload" by prefixing the setting with a "*" character. For example,

```
; example of a template GX.INI file
[newsection1]
*newsettingthatforcesagxreload=on
newsettingthatdoesntrequireagxreload=off
```

The GX reload only takes place, subject to various registry settings (see section 8.3 below) if one, or more, settings that force a GX reload are actually applied to the local GX.INI file. Thus a GX reload would be considered if the local GX.INI file contains

```
; example of a local GX.INI file
[newsection1]
newsettingthatdoesntrequireagxreload=on
```

but not if the local GX.INI file already contains:

```
; example of a template GX.INI file
[newsection1]
newsettingthatforcesagxreload=off
```

Note that the "*" character in the **template** GX.INI file, that indicates "this setting forces a GX reload" is not valid, or copied to, the "live" GX.INI file on the PC running GX.

8.3 Registry Settings that control the GX.INI Upgrade Process

By default, both the GX.INI upgrade process and the possible, subsequent forced GX reload, operate silently. A number of registry settings are available to customise the

upgrade process. Change the following setting to "Off" to produce an information message box before the GX.INI upgrade process commences:

```
..\Global\Client\ServicePacks\SilentGXINIUpdate=On
```

Set the following setting to "On" to allow the operator to decide whether a GX.INI upgrade is required:

```
..\Global\Client\ServicePacks\PromptForGXINIUpdate=Off
```

Change the following setting to "Off" to produce an information message box before the GX reload commences:

```
..\Global\Client\ServicePacks\SilentGXINIReload=On
```

Set the following setting to "On" to allow the operator to decide whether a GX reload is required:

```
..\Global\Client\ServicePacks\PromptForGXINIReload=Off
```

If both the "SilentGXINIUpdate" and "PromptForGXINIUpdate" options are enabled the results will be unpredictable. If both the "SilentGXINIReload" and "PromptForGXINIReload" options are enabled the results will be unpredictable.

8.4 Errors from the GX.INI Upgrade Process

The template GX.INI file in the "gxupdates" folder is validated as each line of the file is processed. If any syntax errors are detected in the file the GX.INI upgrade process is aborted immediately and the effects on the "live" GX.INI file are unpredictable. One of the following self-explanatory messages will appear in a message box if the template GX.INI contains one, or more, errors (only the first error that is detected is actually displayed):

Syntax error on line *nnnn* of template GX.INI file: Malformed section

The section text starts with a "[" character but does not with a "]" character.

Syntax error on line *nnnn* of template GX.INI file: Setting without a section

A setting name (e.g. *xxxxx=value*) was encountered in the template GX.INI file before a valid section name (e.g. [*xxxx*]).

Syntax error on line *nnnn* of template GX.INI file: Malformed reload forcing setting

The setting line just contained a single "*" character.

Syntax error on line *nnnn* of template GX.INI file: Value without a setting

The setting must contain two strings separated by a "=" character. The erroneous line contains just a value without an associated setting string.

Syntax error on line *nnnn* of template GX.INI file: Setting without a value

The setting must contain two strings separated by a "=" character. The erroneous line contains just a setting string, followed by a "=" character but without an associated value.

Syntax error on line *nnnn* of template GX.INI file: Error updating GX.INI file

An error occurred when calling GX to update the GX.INI file.

Syntax error on line *nnnn* of template GX.INI file: Setting without a delimiter

The setting must contain two strings separated by a "=" character. The erroneous line contains just a setting string without an associated value.

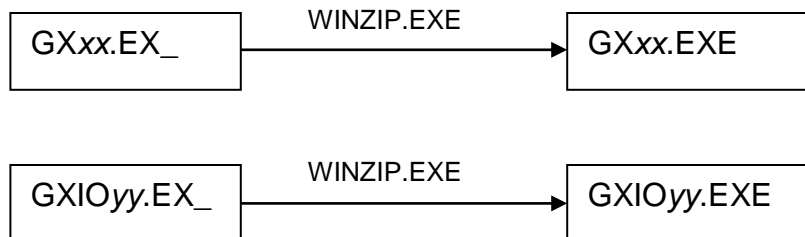
Appendix A - File Formats

The output of the compiler/linker that generates GX and GXIO are the files GX.EXE and GXIO.EXE. However, before these files are released from the Development Dept. to the QC/Support Dept. they are converted to a compressed format using a special version of the COMPRESS.EXE utility:

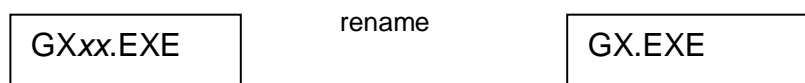


Thus, the files produced by the Development Dept. are in the format that is suitable for copying to a GXUPDATES folder.

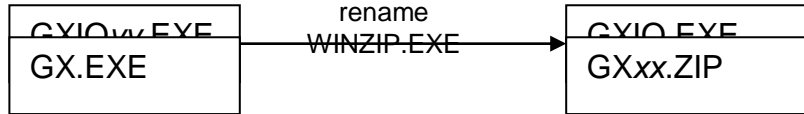
The files created by the COMPRESS.EXE utility can be uncompressed using the standard WINZIP.EXE or EXPAND.EXE utilities:



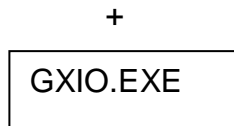
It is necessary to use WINZIP to uncompress the supplied files if GX.EXE/GXIO.EXE are to be copied directly to a thin-client GX folder instead of using the automatic update from server option described in this document. However, note that WINZIP.EXE leaves the version number identification characters (i.e. xx for GX.EXE; and yy for GXIO.EXE) in the names of the uncompressed .EXE's. To avoid an extra rename:



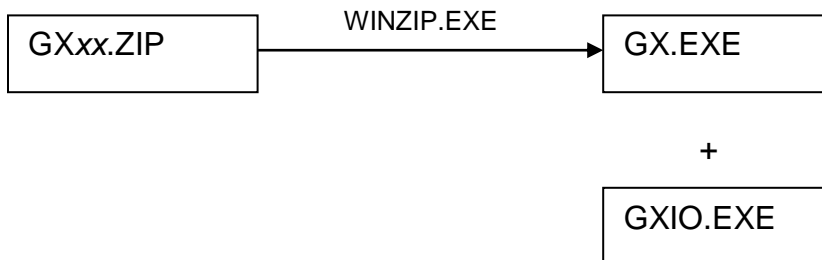
the Support Department supply the GX.EXE and GXIO.EXE files in a compressed format that does not require an addition rename stage:



Note that the name of the final .ZIP file includes the version number of GX.EXE (xx) rather than the version number of GXIO.EXE (yy) even though it includes both .EXE's.



When the GXxx.ZIP file is uncompressed "live" versions of GX.EXE and GXIO.EXE are produced:



To summarise:

- Download GXxx.EX_ and GXIOyy.EX_ if you intend to copy these files to a central GXUPDATES directory;
- Download GXxx.ZIP if you intend to copy the resultant GX.EXE and GXIO.EXE to each local GX PC.