

Building Applications with 4D 2004: Automatic Client Upgrade

By Josh Fletcher, Technical Support Engineer, 4D Inc.

Technical Note 07-22

Abstract

This Technical Note explores the automatic client upgrade feature that was added in 4th Dimension (4D) 2004. The upgrade process is explained, as well as the benefits of this feature and how to customize it.

Introduction

The ability to automatically update the client software is fundamental to 4D's client/server architecture. Changes to the database Structure, like a new method or a new form, need to be automatically distributed to all clients. Luckily 4D already handles these kinds of changes and has done so for many years.

However, what happens when a new version of 4D is released, or some external resource needs to be updated for every client?

Prior to 4D 2004 these kinds of changes had to be made manually or through 4D code. 4D 2004 now allows the developer to handle these kinds of changes automatically.

Useful Terminology for this Technical Note

Compiled Application/Database

Essentially, a .4DC file. An interpreted 4D Structure can contain compiled code but, for the sake of clarity, the term "compiled" means the Structure has been converted into a .4DC file.

Merged Application/Database

The result of "building" an application in 4D. The **compiled database** has literally been "merged" with the 4D software. This is sometimes called a "double-clickable" application.

Merged Client/Server

A merged, multi-user application. Has a client and server component.

Merged Single-User

A merged, single-user application. Also sometimes called "merged standalone".

Client Cache

The files stored on a computer that has used 4D Client (or a merged client) to connect to a 4D Server. These files are located, by default, within the user's Home directory. For example, given a database named "Test.4DB" running on a server machine with the IP address 10.96.0.79, the client cache for a user named "Joe" would be located at (on Windows XP):

```
C:\Documents and Settings\Joe\Application Data\4D\Test.4DB_10_96_0_79
```

On Mac OS it would be:

```
Mac HD:Users:Joe:Library:Application Support:4D:Test.4DB_10_96_0_79
```

Install Image

This term is used to describe all of the files that make up the installation of a merged application. For example a merged 4D Client install image contains the executable for the application, various libraries and resource files, etc. All of these files are collectively referred to as the "install image".

Install Client

The install image of a merged 4D Client application. This is the software that would be manually installed on any given computer that will be used to connect to the merged 4D Server application.

Upgrade Client

The install image that is downloaded to the client machine for the purposes of automatically upgrading. This is **not** the software that would be manually installed on a given client computer. It is used by the automatic upgrade feature only.

Automatic Client Updates in 4D

As mentioned previously, 4D has had the ability to automatically update the client software for some time. This section covers the existing upgrade functionality as well as what is new in 4D 2004.

What's Old in 2004?

This list is not all-inclusive but gives some idea of what kinds of client updates 4D already handles automatically:

- New/changed methods
- New/changed forms
- Picture library changes
- Plug-in changes
- Resource file changes
- The contents of the "Extras" folder

All of these items are automatically managed by 4D with no intervention by the developer.

What's New in 2004?

So what kinds of changes make the new automatic client upgrade feature useful?

- 4D Version change
- External resource change (something not in the "Extras" folder)
- Modifying the installation process (more on this later)
- Simply to force a refresh of the client software

The difference in terminology, "update" versus "upgrade", elucidates the difference here. 4D previously handled client updates automatically but not client upgrades.

The Solution

The automatic client upgrade feature in 4D 2004 essentially deletes the old merged client **install image** and installs an **upgrade client**.

This feature can be enabled or disabled with the Build Application XML Key <BuildCSUpgradeable>. Further customization can be accomplished with the <CurrentVers>, <RangeVersMin>, and <RangeVersMax> XML Keys.

The actual installation of the upgrade client software is handled via a script file (batch file on Windows, shell script on Mac OS).

Both the XML Keys and the install scripts are covered in this Technical Note.

For more information on the 4D Build Application XML keys refer to the "4th Dimension XML Keys BuildApplication" documentation which can be downloaded from:

<http://www.4d.com/support/documentation.html>

Automatic Client Upgrade Explained

This section of the Technical Note covers the configuration of the automatic client upgrade feature as well as how the upgrade process can be customized.

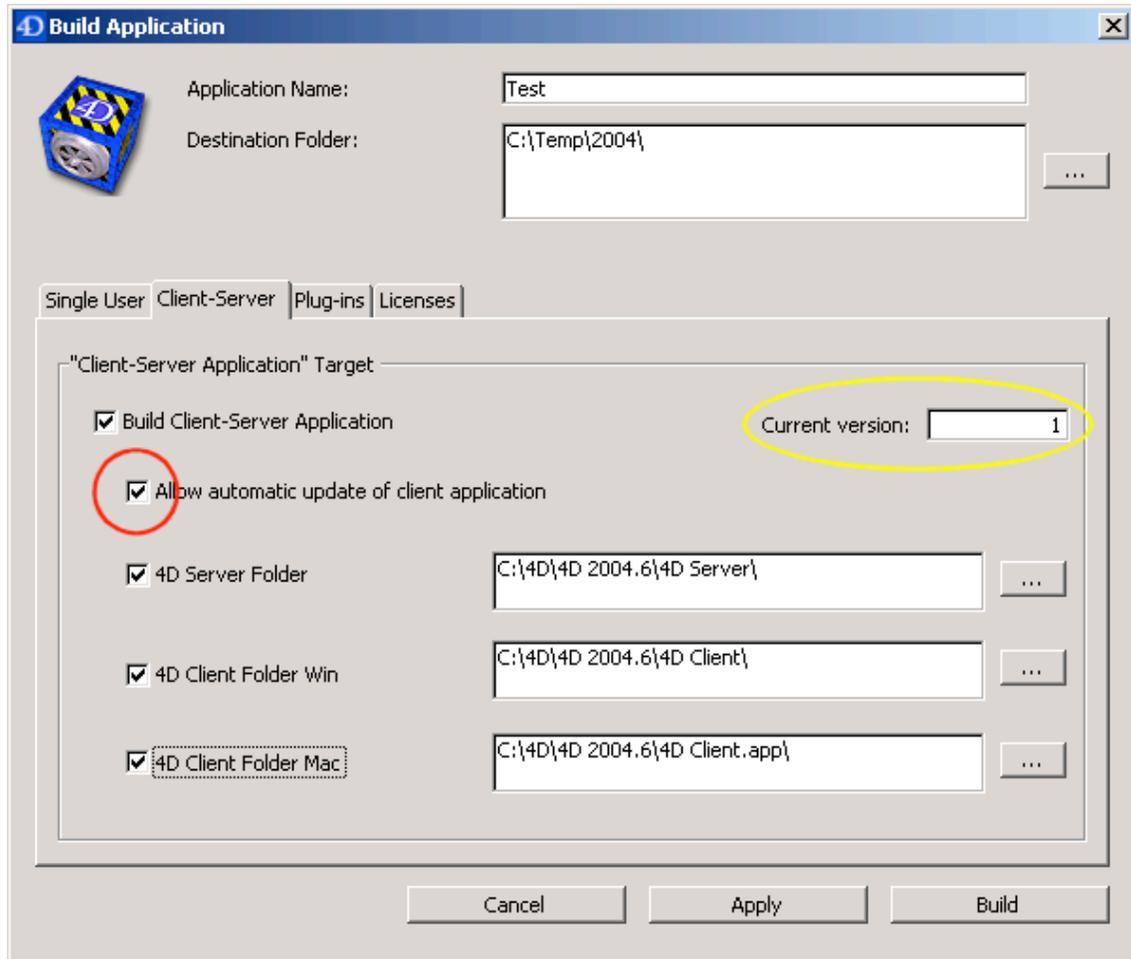
Part 1 – Configuration for Building

The automatic client upgrade feature can be enabled or disabled with the Build Application XML Key <BuildCSUpgradeable>. This XML key is a "flag" that tells 4D whether or not this merged server will allow clients to automatically upgrade. It can be set to True or False.

Note: If you currently build with the Build Application dialog you may not be familiar with the XML keys. Rest assured they are still in use. The Build Application dialog just generates the XML file for you. Refer to the "BuildApp.XML" file, which will be located in the path "Preferences/BuildApp" next to the structure. The Build Application dialog creates

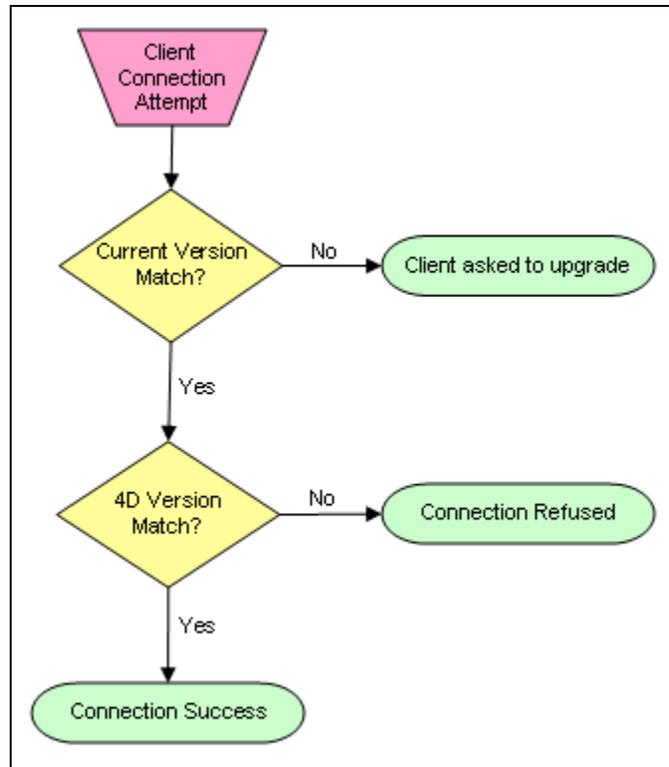
(overwrites and maintains as well) this file and 4D uses it to build the application.

The simplest way to enable the automatic client upgrade feature is via the Build Application dialog in 4D, which is available under the Design menu. Here is a screenshot:



The check box circled in red turns the feature on. Additionally a version needs to be specified (circled in yellow). This is the value that 4D uses in order to determine whether or not the client needs to upgrade.

Here is a flow chart that depicts the checks that 4D performs when a client connects with the automatic upgrade feature enabled:



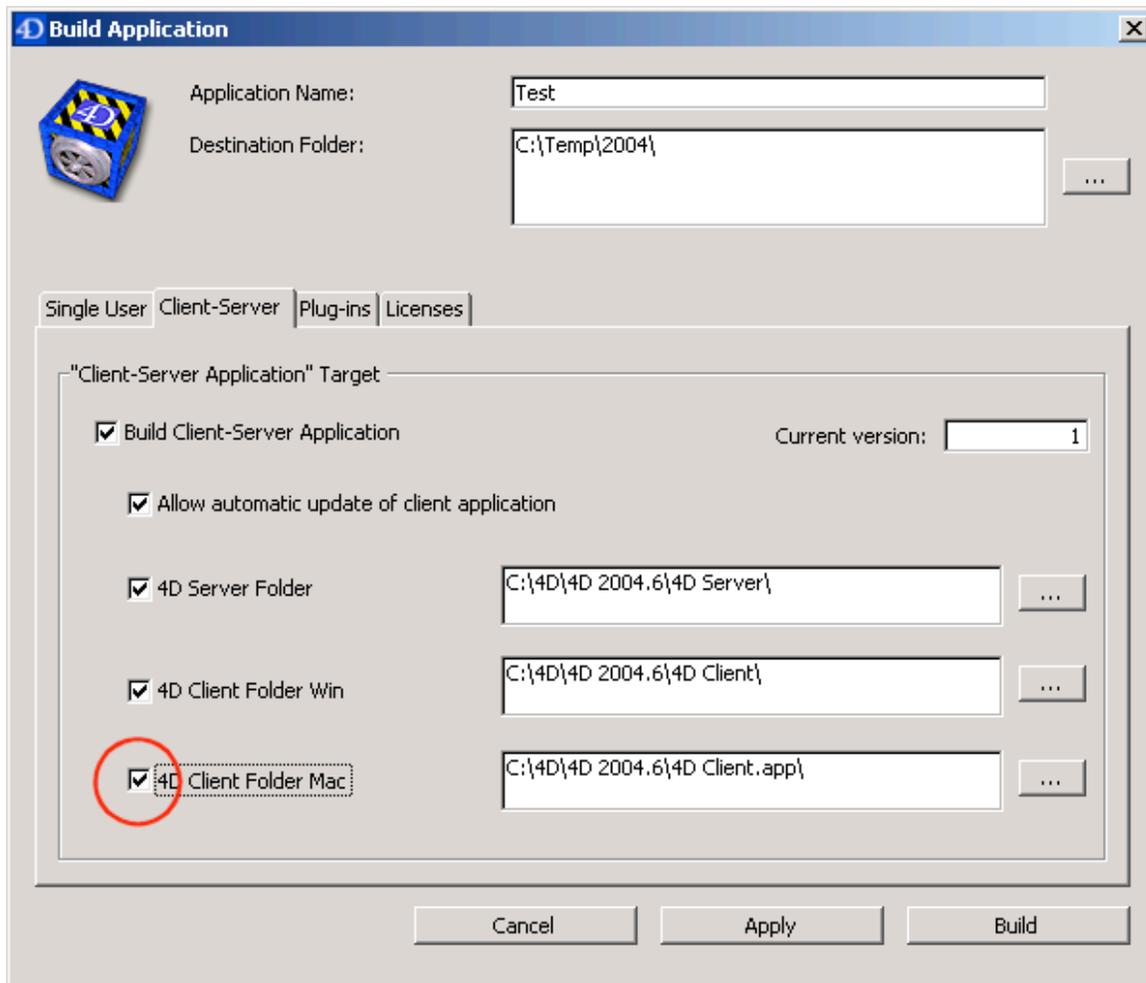
If the **current version** of the client that is attempting to connect is not the same as the current version of the server, the client is asked to upgrade. If the current version matches, the client is allowed to continue **attempting** to connect. Notice that the connection can still fail; if the current version matches but the 4D version does not, the connection will fail. For example if the current version is 5 but the client was built with 4D 2004.4 and the server is running 4D 2004.5 the connection will fail. This is why it is important to increment the current version whenever a client is built with a newer version of 4D.

Assuming the user has been prompted to upgrade, the following happens:

- The upgrade client is downloaded from the server
- The merged client quits
- A script file deletes the old merged client
- The same script file extracts the upgrade client
- Finally the upgrade client is launched

Note that the **client cache** is not affected by this process.

So where does the upgrade client come from? Here is another screenshot of the Build Application dialog:



This is on a Windows machine. Since the automatic upgrade checkbox is checked 4D will automatically build the Windows upgrade client. Additionally the option to build the upgrade client for Mac OS users has been set (circled in red). As can be seen in the screenshot the Mac OS 4D Client software (4D Client.app) has actually been copied to the Windows machine for this purpose.

In either case the Mac OS and/or Windows upgrade client will be placed with the merged server files in a folder called "Upgrade4DClient". The actual upgrade client image is called "archive.win" for Windows and "archive.mac" for Mac OS.

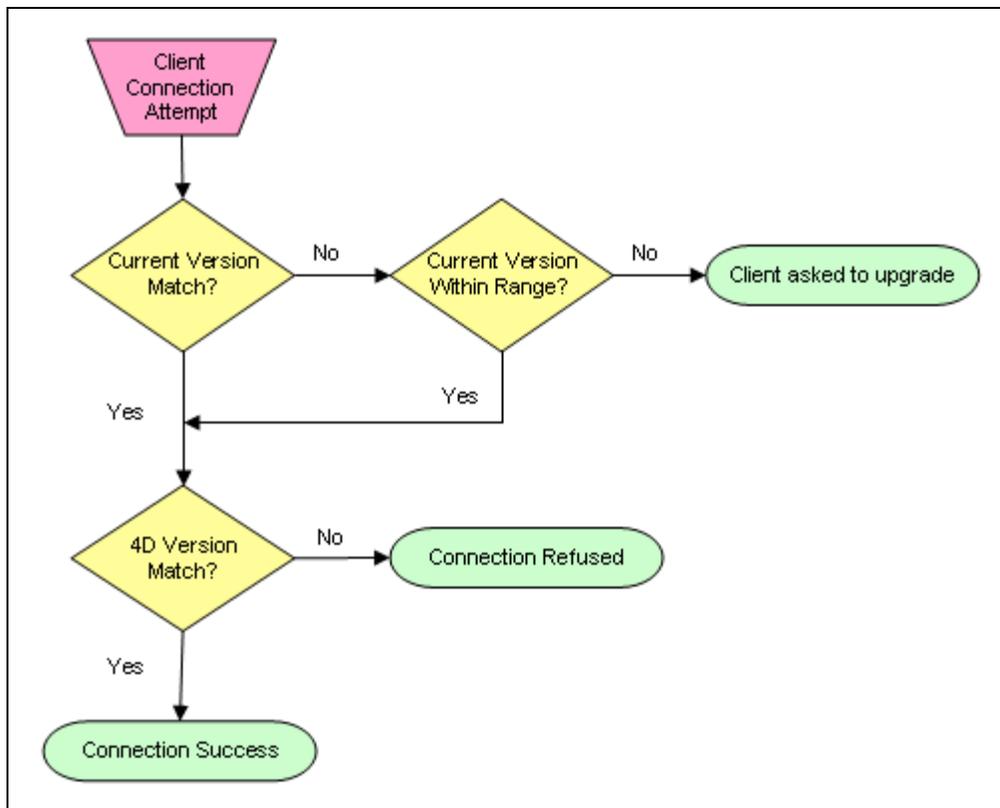
PLEASE NOTE: you cannot build the **install client** for Mac OS on a Windows machine. Similarly you cannot build the Windows install client on a Mac OS machine. The **upgrade clients** can be built on either platform for the purposes of upgrading but the install client must be built on the appropriate platform.

The paths to the files in the above screenshot can, of course, be specified using the XML Keys. Refer to the XML keys documentation for more information.

The current version of the merged application can also be set with the <CurrentVers> XML key (in fact the Build Application dialog does just that in the default "BuildApp.XML" file). This key should contain an appropriate value so that clients attempting to connect to the server will be asked to upgrade in the appropriate cases.

The current version check can be further customized by specifying a version range. This range is specified with the <RangeVersMin> and <RangeVersMax> XML Keys. Note that these XML keys cannot be set with the Build Application dialog. They must be set in a custom build project file for use with the BUILD APPLICATION command.

Here is a flow chart that depicts what happens when the version range is introduced:



In this case, as long as the current version of the client falls within the specified range it is allowed to connect. If the current version falls outside the specified range the client is asked to upgrade (including the case that the current version of the client is **higher** than the maximum version).

As a reminder, remember that in both the simple and more customized examples the connection can still be refused whether or not the current version information is accepted. Under no circumstances will a client with a mismatched 4D version be

allowed to connect. These are two independent tests and there is no way to circumvent the second test.

Also note that the logic in the above flow charts is purely observational and does not necessarily reflect the actual tests made internally in 4D. In other words treat the tests as a black box.

Part 2 – Upgrade Customization

There are two ways the installation of the client software can be customized:

- Modification of the install image
- Modification of the install script file (which is really a special case of the above item)

Modification of the Install Image

Any changes made to the 4D Client software will be included when building the merged client. For example, a new folder added to the 4D Client software will be copied into the merged client. E.g. given a 4D Client installation at:

```
C:\4D\4D 2004.6r2\4D Client 2004.6r2
```

A folder is created called "MyStuff".

A merged client that is built from this copy of 4D Client 2004.6r2 will also contain the folder "MyStuff" (and any files therein).

In other words the build process in 4D copies the contents of the 4D software folder (in this case 4D Client) to the output location of the merged application. Incidentally both 4D Server and 4D single-user software can be modified in this same way.

This technique allows the developer to add external files to the merged application as part of the build process. Of course these external files could also be copied into the merged application after the build is complete using 4D code, but this technique is quite simple (if less flexible).

Modification of the Install Scripts

PLEASE NOTE: the use and modification of the client upgrade scripts is not documented. If the developer wishes to modify these files please be prepared to take changes made in newer 4D versions into account. Backwards compatibility is not guaranteed.

The install script files are included with the 4D Client software. On Windows the file is at:

4D Client Folder\4d Extensions\upgclnt.bat

And on Mac OS:

4D Client.app/Contents/4D Extensions/Upgclnt.sh

These files can be modified to the developer's heart's content in order to customize the installation process for the upgrade client. The possibilities here are pretty much endless, i.e.:

- The install script could launch a full fledged installer instead of simply extracting the upgrade client.
- The install script can make any number of modifications to the system, including checking/altering permissions, creating folders, deleting folders, copying files, etc. (assuming the user has these permissions of course).
- The script could open a release notes file for the new software upgrade that the user could read while the upgrade takes place.

This is a very short list of what could be accomplished by customizing the install scripts. Obviously the 4D developer should be familiar with Windows batch files and Mac OS shell scripts before attempting this.

Conclusion

The automatic client upgrade feature in 4D 2004 gives the 4D developer a great way to handle the upgrade of a merged client with ease. Furthermore the process can be customized on several levels. This Technical Note aimed to provide enough insight to allow the 4D Developer to take advantage of this powerful feature.