1 Technical

This section details technical information related to the installation and use of MJFAF.

1.1 Building MJFAF in BCB5

MJFAF is installed with everything compiled and ready to go. If, for whatever reason, you need to recompile MJFAF you may come across when building the C++Builder 5 version. On a clean system, the MJFAFSystemC5 project produces the following error when compiled:

```
[C++ Error] Imagehlp.h(600, 1): E2015 Ambiguity between 'LONG64' and 'System::LONG64'
```

The simplest solution is to modify the header `ImageHlp.h` file to use `::LONG64` instead of `LONG64`.

For reference, the code in question is shown below (with the change highlighted):

```
__inline
void
Address32To64(
LPADDRESS a32,
LPADDRESS64 a64
)
{
a64->Offset = (ULONG64)( ::LONG64)(LONG)a32->Offset;
a64->Segment = a32->Segment;
a64->Mode = a32->Mode;
}
```

1.2 Updating Indy

MJFAF is regularly updated against the latest version of Indy thereby making the version provided with C++Builder out of date. In order to utilize the Indy runtimes provided with MJFAF, the Indy installation needs to be updated for each version of C++Builder being used. This section details the version of Indy required along with instructions on how to update your installation.

1.2.1 Removing Indy

This section provides instructions on removing the default installation of Indy provided with C++Builder. Before you can install a newer version of Indy, you need to perform a few basic steps as detailed below.

Initial steps

1. The first things to do is remove Indy from the list of known packages loaded by the IDE. To do this, open the IDE you will be updating Indy for and remove Indy from the list of available packages. This will prevent errors being displayed after the Indy files are deleted.
2. Check the Windows\System32 folder for any indy*.bpl files that should also be deleted. Failure to delete these files will result in incorrect behaviour in both the IDE and your own applications (when built dynamically).

The following sections detail how to remove the Indy source, headers and binaries from a default installation of C++Builder 6 through to RAD Studio XE (C++Builder 5 was not shipped with Indy). Be sure to make a full backup if you're worried about things going wrong.

C++Builder 6

1. Delete the folder "C:\Program Files\Borland\CBuilder6\Source\Indy".
2. Delete Indy.bpi and Indy.lib from the folders:
   - "C:\Program Files\Borland\CBuilder6\Lib\Debug"
   - "C:\Program Files\Borland\CBuilder6\Lib\Release"
3. Delete indy60.bpl and dclindy60.bpl from "C:\Program Files\Borland\CBuilder6\Bin"
4. Delete the following files from "C:\Program Files\Borland\CBuilder6\Lib\Obj"
   - indy.dcu
   - indy.dcp
   - dclindy.dcu
   - dclindy.dcp
   - id*.dcu (idabout.dcu through to idwinsock.dcu)

BDS 2006

1. Delete the following folders:
   - "C:\Program Files\Borland\BDS\4.0\Include\Indy9"
   - "C:\Program Files\Borland\BDS\4.0\Include\Indy10"
   - "C:\Program Files\Borland\BDS\4.0\lib\Indy9"
   - "C:\Program Files\Borland\BDS\4.0\lib\Indy10"
   - "C:\Program Files\Borland\BDS\4.0\lib\debug\Indy9"
   - "C:\Program Files\Borland\BDS\4.0\lib\debug\Indy10"
   - "C:\Program Files\Borland\BDS\4.0\Source\Win32\Indy9"
   - "C:\Program Files\Borland\BDS\4.0\Source\Indy10"
2. Delete the following files from "C:\Program Files\Borland\BDS\4.0\Bin"
   - dclindy100.bpl
   - dclindyCore100.bpl
   - dclindyProtocols100.bpl
   - indy100.bpl
   - IndyCore100.bpl
   - IndyProtocols100.bpl
   - IndySystem100.bpl

RAD 2007

1. Delete the following folders:
   - "C:\Program Files\CodeGear\RAD Studio\5.0\Include\Indy9"

- "C:\Program Files\CodeGear\RAD Studio\5.0\Include\Indy10"
- "C:\Program Files\CodeGear\RAD Studio\5.0\Lib\Indy9"
- "C:\Program Files\CodeGear\RAD Studio\5.0\Lib\Indy10"
- "C:\Program Files\CodeGear\RAD Studio\5.0\Lib\Debug\Indy9"
- "C:\Program Files\CodeGear\RAD Studio\5.0\Lib\Debug\Indy10"
- "C:\Program Files\CodeGear\RAD Studio\5.0\Source\Indy"

2. Delete the following files from "C:\Program Files\CodeGear\RAD Studio\5.0\Bin"
    - dclindy100.bpl
    - dclindyCore100.bpl
    - dclindyProtocols100.bpl
    - indy100.bpl
    - IndyCore100.bpl
    - IndyProtocols100.bpl
    - IndySystem100.bpl
3. Delete the following files from "C:\Program Files\CodeGear\RAD Studio\5.0\Lib\Debug" and "C:\Program Files\CodeGear\RAD Studio\5.0\Lib\Release"
    - indy.bpi
    - indy.lib
    - IndyCore.bpi
    - IndyCore.lib
    - IndySystem.bpi
    - IndySystem.lib
    - IndyProtocols.bpi
    - IndyProtocols.lib
    - dclIndyCore.bpi
    - dclIndyCore.lib
    - dclIndyProtocols.bpi
    - dclIndyProtocols.lib

RAD 2009

1. Delete the following folders:
    - "C:\Program Files\CodeGear\RAD Studio\6.0\Include\Indy10"
    - "C:\Program Files\CodeGear\RAD Studio\6.0\Lib\Indy10"
    - "C:\Program Files\CodeGear\RAD Studio\6.0\Lib\Debug\Indy10"
    - "C:\Program Files\CodeGear\RAD Studio\6.0\Source\Indy"
2. Delete the following files from "C:\Program Files\CodeGear\RAD Studio\6.0\Bin"
    - dclindy120.bpl
    - dclindyCore120.bpl
    - dclindyProtocols120.bpl
    - indy120.bpl
    - IndyCore120.bpl
    - IndyProtocols120.bpl
    - IndySystem120.bpl

3. Delete the following files from "C:\Program Files\CodeGear\RAD Studio\6.0\Lib\Debug" and "C:\Program Files\CodeGear\RAD Studio\6.0\Lib\Release"
    - indy.bpi
    - indy.lib
    - IndyCore.bpi
    - IndyCore.lib
    - IndySystem.bpi
    - IndySystem.lib
    - IndyProtocols.bpi
    - IndyProtocols.lib

RAD 2010

1. Delete the following folders:
    - "C:\Program Files\Embarcadero\RAD Studio\7.0\Include\Indy10"
    - "C:\Program Files\Embarcadero\RAD Studio\7.0\Lib\Indy10"
    - "C:\Program Files\Embarcadero\RAD Studio\7.0\Lib\Debug\Indy10"
    - "C:\Program Files\Embarcadero\RAD Studio\7.0\Source\Indy"
2. Delete the following files from "C:\Program Files\Embarcadero\RAD Studio\7.0\Bin"
    - dclindyCore140.bpl
    - dclindyProtocols140.bpl
    - IndyCore140.bpl
    - IndyProtocols140.bpl
    - IndySystem140.bpl
3. Delete the following files from "C:\Program Files\Embarcadero\RAD Studio\7.0\Lib\Debug" and "C:\Program Files\Embarcadero\RAD Studio\7.0\Lib\Release"
    - IndyCore.bpi
    - IndyCore.lib
    - IndySystem.bpi
    - IndySystem.lib
    - IndyProtocols.bpi
    - IndyProtocols.lib

RAD XE

1. Delete the folder "C:\Program Files\Embarcadero\RAD Studio\8.0\Source\Indy10"
2. Delete the following from "C:\Program Files\Embarcadero\RAD Studio\8.0\include\windows\rtl"
    - Id*.hpp (IdAboutVCL.hpp through to IdZLibHeaders.hpp)
3. Delete the following from "C:\Program Files\Embarcadero\RAD Studio\8.0\lib\win32\debug" and "C:\Program Files\Embarcadero\RAD Studio\8.0\lib\win32\release"
    - Id*.dcu (InAboutVCL.dcu through to IdZLibHeaders.dcu)
4. Delete the following files from "C:\Program Files\Embarcadero\RAD Studio\8.0\Bin"
    - dclindyCore150.bpl
    - dclindyProtocols150.bpl
    - IndyCore150.bpl
    - IndyProtocols150.bpl
    - IndySystem150.bpl

5. Delete the following files from "C:\Program Files\Embarcadero\RAD Studio\8.0\Lib\Win32\Debug" and "C:\Program Files\Embarcadero\RAD Studio\8.0\Lib\Win32\Release"
   - IndyCore.bpi (only present in Release folder)
   - IndyCore.lib
   - IndyCore.dcp
   - IndySystem.bpi (only present in Release folder)
   - IndySystem.lib
   - IndySystem.dcp
   - IndyProtocols.bpi (only present in Release folder)
   - IndyProtocols.lib
   - IndyProtocols.dcp

You're now ready to update your installation of Indy 9 and/or Indy 10 as detailed in the following sections.

1.2.2 Installing Indy 9

This section provides instructions on how to manually install the latest pre-built version of Indy 9 into the C++Builder IDE. Make sure you have removed all traces of the installation initially provided with C++Builder as detailed in the Removing Indy (↗ see page 1) section.

Availability

All binary files for Indy 9 is available for C++Builder 5, 6, 2006 and 2007. Instructions for updating each version of C++Builder is provided next.

Updating C++Builder 5

Copy the provided files into a **Indy9** sub-folder under the root folder of your C++Builder installation. For a default installation, this would be: **C:\Program Files\Borland\CBuilder5\Indy9**.

If you've decided to install Indy in the above recommended location, you will need to add this director5 to the PATH environment variable. If you do not perform this you will receive errors from the IDE as well as your own applications (if built dynamically).

To install Indy9 in the IDE, start C++Builder 5 and close any open projects. Open the **Project | Options** menu item and select the **Packages** tab. Click the **Add** button, browse to the **C:\Program Files\Borland\CBuilder5\Indy9\C5** folder and select the **dclIndy50.bpl** file.

Indy 9 is now ready for use in C++Builder 5.

Updating C++Builder 6

Copy the provided files into a **Indy9** sub-folder under the root folder of your C++Builder installation. For a default installation, this would be: **C:\Program Files\Borland\CBuilder6\Indy9**.

If you've decided to install Indy in the above recommended location, you will need to add this director5 to the PATH environment variable. If you do not perform this you will receive errors from the IDE as well as your own applications (if built dynamically).

To install Indy9 in the IDE, start C++Builder 6 and close any open projects. Open the **Project | Options** menu item and select the **Packages** tab. Click the **Add** button, browse to the **C:\Program Files\Borland\CBuilder6\Indy9\C6** folder and select the **dclIndy60.bpl** file.

Indy 9 is now ready for use in C++Builder 6.

Updating C++Builder 2006

Copy the provided files into a **Indy9** sub-folder under the root folder of your C++Builder installation. For a default installation, this would be: **C:\Program Files\CodeGear\BDS\4.0\Indy9**.

If you've decided to install Indy in the above recommended location, you will need to add this director5 to the PATH environment variable. If you do not perform this you will receive errors from the IDE as well as your own applications (if built dynamically).

To install Indy9 in the IDE, start C++Builder 2006 and close any open projects. Open the **Project | Default Options | C++Builder** menu item and select the **Packages** tab. Click the **Add** button, browse to the **C:\Program Files\CodeGear\BDS\4.0\Indy9\C10** folder and select the **dclIndy100.bpl** file.

Indy 9 is now ready for use in C++Builder 2006.

Updating C++Builder 2007

Copy the provided files into a **Indy9** sub-folder under the root folder of your C++Builder installation. For a default installation, this would be: **C:\Program Files\CodeGear\RAD Studio\5.0\Indy9**.

If you've decided to install Indy in the above recommended location, you will need to add this director5 to the PATH environment variable. If you do not perform this you will receive errors from the IDE as well as your own applications (if built dynamically).

To install Indy9 in the IDE, start C++Builder 2006 and close any open projects. Open the **Project | Default Options | C++Builder** menu item and select the **Packages** tab. Click the **Add** button, browse to the **C:\Program Files\CodeGear\RAD Studio\5.0\Indy9\C10** folder and select the **dclIndy110.bpl** file.

Indy 9 is now ready for use in C++Builder 2007.

1.2.3 Installing Indy 10

This section provides instructions on how to manually install the latest pre-built version of Indy 10 into the C++Builder IDE. Make sure you have removed all traces of the installation initially provided with C++Builder as detailed in the Removing Indy (↗ see page 1) section.

Availability

All binary files for Indy 10 is available for C++Builder 5, 6, 2006, 2007, 2010, 2011 and XE. Instructions for updating each version of C++Builder is provided next.

Updating C++Builder 5

Copy the provided files into a **Indy10** sub-folder under the root folder of your C++Builder installation. For a default installation, this would be: **C:\Program Files\Borland\CBuilder5\Indy10**.

To install Indy10 in the IDE, start C++Builder 5 and close any open projects. Open the **Project | Options** menu item and select the **Packages** tab. Click the **Add** button, browse to the **C:\Program Files\Borland\CBuilder5\Indy10\C5** folder and select the **dclIndyCore50.bpl** and **dclIndyProtocols50.bpl** files.

Indy 10 is now ready for use in C++Builder 5.

Updating C++Builder 6

Copy the provided files into a **Indy10** sub-folder under the root folder of your C++Builder installation. For a default installation, this would be: **C:\Program Files\Borland\CBuilder6\Indy10**.

If you've decided to install Indy in the above recommended location, you will need to add this director5 to the PATH environment variable. If you do not perform this you will receive errors from the IDE as well as your own applications (if built dynamically).

To install Indy10 in the IDE, start C++Builder 6 and close any open projects. Open the **Project | Options** menu item and select the **Packages** tab. Click the **Add** button, browse to the **C:\Program Files\Borland\CBuilder6\Indy10\C6** folder and select the **dclIndyCore60.bpl** and **dclIndyProtocols60.bpl** files.

Indy 10 is now ready for use in C++Builder 6.

Updating C++Builder 2006

Copy the provided files into a **Indy10** sub-folder under the root folder of your C++Builder installation. For a default installation, this would be: **C:\Program Files\Borland\BDS\4.0\Indy10**.

If you've decided to install Indy in the above recommended location, you will need to add this director5 to the PATH environment variable. If you do not perform this you will receive errors from the IDE as well as your own applications (if built dynamically).

To install Indy10 in the IDE, start C++Builder 2006 and close any open projects. Open the **Project | Default Options | C++Builder** menu item and select the **Packages** tab. Click the **Add** button, browse to the **C:\Program Files\Borland\BDS\4.0\Indy10\C10** folder and select the **dclIndyCore100.bpl** and **dclIndyProtocols100.bpl** files.

Indy 10 is now ready for use in C++Builder 2006.

Updating C++Builder 2007

Copy the provided files into a **Indy10** sub-folder under the root folder of your C++Builder installation. For a default installation, this would be: **C:\Program Files\CodeGear\RAD Studio\5.0\Indy10**.

If you've decided to install Indy in the above recommended location, you will need to add this director5 to the PATH environment variable. If you do not perform this you will receive errors from the IDE as well as your own applications (if built dynamically).

To install Indy10 in the IDE, start C++Builder 2007 and close any open projects. Open the **Project | Default Options | C++Builder** menu item and select the **Packages** tab. Click the **Add** button, browse to the **C:\Program Files\CodeGear\RAD Studio\5.0\Indy10\C11** folder and select the **dclIndyCore110.bpl** and **dclIndyProtocols110.bpl** files.

Indy 10 is now ready for use in C++Builder 2007.

Updating C++Builder 2009

Copy the provided files into a **Indy10** sub-folder under the root folder of your C++Builder installation. For a default installation, this would be: **C:\Program Files\CodeGear\RAD Studio\6.0\Indy10**.

If you've decided to install Indy in the above recommended location, you will need to add this director5 to the PATH environment variable. If you do not perform this you will receive errors from the IDE as well as your own applications (if built dynamically).

To install Indy10 in the IDE, start C++Builder 2009 and close any open projects. Open the **Project | Default Options | C++Builder** menu item and select the **Packages** tab. Click the **Add** button, browse to the **C:\Program Files\CodeGear\RAD Studio\6.0\Indy10\C12** folder and select the **dclIndyCore120.bpl** and **dclIndyProtocols120.bpl** files.

Indy 10 is now ready for use in C++Builder 2009.

Updating C++Builder 2010

Copy the provided files into a **Indy10** sub-folder under the root folder of your C++Builder installation. For a default installation, this would be: **C:\Program Files\CodeGear\RAD Studio\7.0\Indy10**.

If you've decided to install Indy in the above recommended location, you will need to add this director5 to the PATH environment variable. If you do not perform this you will receive errors from the IDE as well as your own applications (if built dynamically).

To install Indy10 in the IDE, start C++Builder 2010 and close any open projects. Open the **Project | Default Options | C++Builder** menu item and select the **Packages** tab. Click the **Add** button, browse to the **C:\Program**

Files\CodeGear\RAD Studio\7.0\Indy10\C14 folder and select the **dclIndyCore140.bpl** and **dclIndyProtocols140.bpl** files.

Indy 10 is now ready for use in C++Builder 2010.

Updating C++Builder XE

Copy the provided files into a **Indy10** sub-folder under the root folder of your C++Builder installation. For a default installation, this would be: **C:\Program Files\Embarcadero\RAD Studio\8.0\Indy10**.

If you've decided to install Indy in the above recommended location, you will need to add this director5 to the PATH environment variable. If you do not perform this you will receive errors from the IDE as well as your own applications (if built dynamically).

To install Indy10 in the IDE, start C++Builder XE and close any open projects. Open the **Project | Default Options | C++Builder** menu item and select the **Packages** tab. Click the **Add** button, browse to the **C:\Program Files\Embarcadero\RAD Studio\8.0\Indy10\C15** folder and select the **dclIndyCore150.bpl** and **dclIndyProtocols150.bpl** files.

Indy 10 is now ready for use in C++Builder XE.

1.2.4 Important Information

This section provides important information related to potential problems that might be encountered when updating projects that previously used the default C++Builder installed version of Indy.

Include / Library Path

The project's Include Path needs to be changed so it points to the new location, such as:

Indy 9: **$(BCB)\Indy9**

Indy 10: **$(BCB)\Indy10**

BPI/LIB filename differences

All default installed versions of C++Builder use BPI/LIB filenames that are version independent:

Indy 9

Indy.bpi and Indy.lib

Indy 10

IndyCore.bpi and IndyCore.lib

IndyProtocols.bpi and IndyProtocols.lib

IndySystem.bpi and IndySystem.lib

The updated version of Indy provided by MJ Freelancing uses filenames that reflect the version of C++Builder being used. This ensures the correct files are used in an environment where multiple IDE's are installed. All projects compiled using a previous version of Indy will need to be updated to refer to the correct BPI/LIB file.

The easiest, and safest, way to do this is to edit the project file using a text editor. Find all references to the above filenames (as applicable) and replace them with the version dependent filename. For example, a Indy 10 based project built with C++Builder 2010 would replace the filenames like so:

IndyCore.bpi and IndyCore.lib -> IndyCore140.bpi and IndyCore140.lib

IndyProtocols.bpi and IndyProtocols.lib -> IndyProtocols140.bpi and IndyProtocols140.lib

IndySystem.bpi and IndySystem.lib -> IndySystem140.bpi and IndySystem140.lib