

THINKDATA INC.

Clarion Third Party Add-Ons



qbFUSE 1.2 User Guide

CLARION THIRD PARTY ADD-ONS

qbFUSE 1.2 User Guide

© 2003 ThinkData Inc.
2508 Pacific Avenue • Suite 1
Venice Beach, California 90291
Phone 310.823.2571
info@thinkdata.com

Trademark Acknowledgements:

SoftVelocity® is a registered trademark of SoftVelocity Incorporated.

Clarion 5.5™ is a trademark of SoftVelocity Incorporated.

Microsoft, MS, Microsoft Outlook, Visual Basic, Visual C++, Win32, Windows, and Windows 2000 are registered trademarks of Microsoft Corporation.

Intuit, the Intuit logo, QuickBooks Pro 2002, QuickBooks Premiere 2002 and QuickBooks Premiere: Accountant Edition 2002 are registered trademarks and/or service marks of Intuit, Inc.

All other products and company names are trademarks of their respective owners.

Table of Contents

Introduction	1
Prerequisites	1
Installation	3
Plugware COM Overview	4
String Classes	6
Helper Classes	7
Early Binding Automation	10
Late Binding Automation	12
Multi-DLL Considerations	15
qbFUSE Overview	16
List of Interfaces	17
Example Applications	29
Links	31
Summary	32

Introduction

Overview

The qbFUSE 1.1 product provides a native COM (Component Object Model) automation interface to Intuit's QBFC (QuickBooks Foundation Classes) SDK 2.0 using Clarion 5.5 and Clarion 6. It relies on the Plugware COM classes which encapsulate the major portions of the COM Application Programming Interface (API) specification as designed by Microsoft. The Plugware COM classes usher in a new era of stability and performance when writing COM automation interfaces in Clarion. Full source code to these classes has been provided along with detailed examples demonstrating how to use the qbFUSE product. This gives the developer/end-user an opportunity to understand the underlying principles of COM while providing an open and stable platform from which to develop additional automation interfaces. The example applications included demonstrate early bound interface functionality to QuickBooks and are in no way intended to be an exhaustive demonstration of the power of the QuickBooks COM object model.

Intuit has announced that they have no plans to make any further major improvements to .iif files which have been used to create integrations with previous versions of QuickBooks. With the qbXML spec, Intuit is committed to providing much more consistent and seamless methods of integrating third-party applications with QuickBooks, for the benefit of developers and customers alike. The QBFC wraps the qbXML specification so you can write applications which share data with QuickBooks without having to build and parse qbXML in your code. Using QBFC you can manipulate the COM objects rather than raw qbXML yet still have access to all the functionality of qbXML. This makes qbFUSE the only Clarion product in the world capable of supporting the very latest technology for QuickBooks from Intuit.

Prerequisites

The QBFC (QuickBooks Foundation Classes) SDK 2.0 can be downloaded from the Intuit web site via the following link: <http://developer.intuit.com/downloads/>

You must install the QBFC SDK 2.0 before using qbFUSE and running the included demo application. According to Intuit, the following QuickBooks Products are supported by the QBFC SDK 2.0:

Supported QuickBooks Products

QuickBooks SDK v2.0 allows your applications to integrate directly with all of the following QuickBooks products:

- U.S. editions of QuickBooks 2003
 - QuickBooks Pro 2003
 - QuickBooks Premier 2003
 - QuickBooks Premier: Accountant Edition 2003
 - QuickBooks Premier: Contractor Edition 2003
- QuickBooks Enterprise Solutions 2.0
 - QuickBooks Enterprise Solutions 2.0
 - QuickBooks Enterprise Solutions: Accountant Edition 2.0
- Canadian editions of QuickBooks 2003
 - QuickBooks Pro 2003
 - QuickBooks Premier 2003
 - QuickBooks Premier: Accountant Edition 2003
- QuickBooks Online Edition

QuickBooks SDK version 2.0 also allows your applications to integrate directly with the following QuickBooks 2002 products (with the same functionality as provided by QuickBooks SDK version 1.1):

- U.S. Editions of QuickBooks 2002
 - QuickBooks Pro 2002
 - QuickBooks Premier 2002
 - QuickBooks Premier: Accountant Edition 2002
- QuickBooks Enterprise Solutions
 - QuickBooks Enterprise Solutions
 - QuickBooks Enterprise Solutions: Accountant Edition

We recommend reading the Intuit Developer Network Getting Started Guide to get a good understanding of what is provided in the QBFC SDK. You can view it at Intuit's web site: <http://developer.intuit.com/Downloads/quickbooks/sdk/doc/gsidn.pdf>. The QBFC SDK 2.0 feature list is available at: <http://developer.intuit.com/SDK-20/SDKv20Features.asp>.

This manual is targeted at intermediate Clarion 5.5 and Clarion 6 programmers with some Win32 programming experience and a basic understanding of COM. It is not intended to be a primer on COM – for that we recommend the following reading list.

Inside COM by Dale Rogerson.

This book discloses the secrets of COM programming for the advanced engineer. It includes many sample programs on CD-ROM. (Microsoft Press, ISBN 1-57231-349-8)

Essential COM by Don Box

This text covers the motivation for the design of COM and its distributed aspects. It shows how COM works and contains coverage of the core concepts of distributed COM including detailed descriptions of COM theory and remote servers. It also offers a thorough explanation of COM's basic vocabulary. (Addison Wesley ISBN 0-201-63446-5)

The Plugware COM classes and qbFUSE product are written entirely for Clarion 5.5 and Clarion 6. Earlier versions of Clarion are not currently supported because of improvements in the compiler to support interfaces.

Installation

To install qbFUSE, simply execute the supplied qbfuse.exe file and follow the instructions in the installation program. qbFUSE consists of Clarion source files, documentation and example applications. The default installation directory (\C55) is located based on your Clarion 5.5 installation. If you choose \C55 as the installation directory the installer will place files in the following order:

\C55\Libsrc – The following source files for the Plugware COM classes and the QuickBooks QBFC SDK 1.1 wrapper will default to this directory:

```
pwapi.inc
pwapifnc.inc
pwcomdef.inc
pwcom.inc
pwcom.clw
pwheap.inc
pwheap.clw
qbf.inc
qbf_2.inc
qbf.clw
qbf1.inc
qbf1.clw
qbf2.inc
qbf2.clw
qbf3.inc
qbf3.clw
qbf4.inc
qbf4.clw
qbf5.inc
qbf5.clw
qbf6.inc
qbf6.clw
qbf7.inc
qbf7.clw
```

qbf8.inc
qbf8.clw
qbf9.inc
qbf9.clw
qbf10.inc
qbf10.clw
qbf11.inc
qbf11.clw
qbf12.inc
qbf12.clw
qbf13.inc
qbf13.clw
qbfdef.inc
qbfint.inc
qbfiid.inc

\C55\Docs\qbFUSE – qbFUSE documentation including:

qbfuse.pdf

\C55\Template – qbFUSE global extension templates including:

qbfuse.gif
qfuseabc.tpl
qfuseleg.tpl

\C55\Examples\qbFUSE – Source files for the example applications including:

qbfuse.clw
qbfuse.prj
qbfuse.exe
qbfapp.app
qbfapp.dct
qbfapp.exe
thinkdata.ico

The demo executables in \C55\Examples\qbFUSE have been compiled with the QuickBooks SDK v1.1p2 wrapper under Clarion 5.5G.

Plugware COM Overview

The Plugware COM classes, written by Plugware Solutions.com Ltd., provide the Clarion developer with a set of classes to encapsulate the COM API natively in Clarion. Now any developer can write pure Clarion source code to interface with COM objects using native early or late binding without worrying about stability, performance, or flexibility. So without further ado let us explore the Plugware COM classes!

Plugware COM ships as seven Clarion source files and one export file for multi-DLL and hand-coded applications. These files are:

pwapi.inc – Contains a large number of the Win32 API data types and constants. It is included by the pwcomdef.inc file.

pwapifunc.inc – Contains a large number of the Win32 API function prototypes for Clarion. It is included by the pwcomdef.inc file.

pwcomdef.inc – Contains the common interfaces (IUnknown, IDispatch, ITypeInfo), common data types for implementing COM automation, and the function prototypes for Win32 API calls used by the COM classes. It is included by the pwcom.inc file.

pwcom.inc – Contains the class definitions for the early and late binding implementations of generic COM objects including COM string classes, variant factory object, safe array support and dispatch interface wrapper.

pwcom.clw – Contains the source implementation of the classes defined in pwcom.inc.

pwheap.inc – Contains the class definition for the PWHeap class which is used by the string classes to allocate memory properly. It is included by the pwcom.clw file.

pwheap.clw – Contains the source implementation of the class defined in pwheap.inc.

pwcom.exp – Contains the exports for the Plugware COM classes and methods used in multi-DLL and hand-coded Clarion application development.

String Classes

There are four classes provided to deal with strings in COM. Figure 1.1 shows the inheritance relationships of these classes.

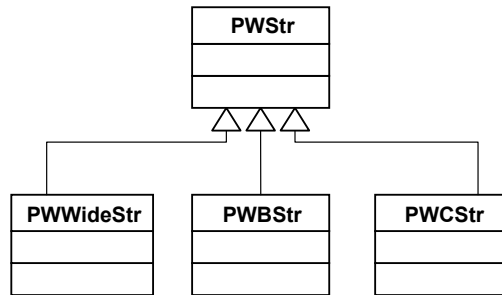


FIGURE 1.1 The relationship between the parent PWStr class and the derived classes PWWideStr, PWBStr (BSTR or binary string support in COM) and PWCStr

The PWStr is the parent class of the three other classes. It contains a virtual destructor which calls the PWStr.Release method to handle freeing the memory for the string. The PWWideStr class is used internally by the PWBStr class to allocate memory and map a character string to a wide-character (Unicode) string. The two classes of interest to the developer are the PWBStr and PWCStr classes.

The PWBStr class was written to support the COM BSTR data type. A BSTR is a pointer to a wide character string and is sometimes referred to as a Basic string or binary string. The PWBStr can take a Clarion cstring or string in its initialization method and the PWBStr.GetStr() method will be called when the developer wishes to pass a BSTR to a COM object.

The PWCStr class is designed to retrieve a Clarion CSTRING from a PWBStr object. It will be used frequently to retrieve string output parameters from calls to COM objects. It operates in the reverse of the PWWideStr by calling the WideCharToMultiByte API function to map a wide-character (Unicode) string to a character string.

A helper function called `_cstr` has been added to Plugware COM to support the automatic conversion of BSTR return values to a Clarion CSTRING. It is prototyped as follows:

```
_cstr(long bstrVal, short fFreeBStr = true),*cstring
```

The `_cstr` function is passed the BSTR parameter and returns the reference to a newly created Clarion CSTRING.

Helper Classes

There are five classes provided for making the process of interfacing to the Win32 COM API easier:

PWCOMIniter

The PWCOMIniter should be the first object instantiated on each thread using the Plugware COM classes. It calls the Win32 API function `CoInitialize()` to initialize COM for that thread. When its destructor is called the `CoUnInitialize()` API function is called. The PWCOMIniter must be instantiated before any of the other Plugware COM classes to ensure that its destructor method is called after all other classes have destructed.

PWCOMError

The PWCOMError class is responsible for translating COM errors returned as `HRESULT`s into something readable to the developer or end user. The method of interest to the developer inside the PWCOMError class is:

PWCOMError.GetError `procedure(*cstring szErrorMsg, *long dwBufferLen, HRESULT hr),long` – This method fills the passed `szErrorMsg` with the results from the Windows API error message for the passed `HRESULT`. This method uses the `FormatMessage` API function internally.

PWDateTime

The PWDateTime class converts Clarion dates and times into values compatible with COM objects. The methods of interest to the developer inside the PWDispatch class are:

PWDateTime.Init `procedure(*tVariant vtTime)` – This method takes a variant of type `VT_DATE` and initializes the PWDateTime structure with its contents.

PWDateTime.Init `procedure(*real systime)` – This method takes a Clarion real containing a `COleDateTime` date/time value and initializes the PWDateTime structure with its contents.

PWDateTime.Init `procedure(*SYSTEMTIME systime)` – This method takes a Windows API `SYSTEMTIME` structure and initializes the PWDateTime with its contents.

PWDateTime.Init `procedure(date cwDate, time cwTime)` – This method initializes the PWDateTime structure with a Clarion date and time. Use this method in conjunction with `PWDateTime.GetAsCOleDateTime` to pass dates to COM objects.

PWDateTime.GetAsCOleDateTime procedure(),real – This method returns the contents of the PWDateTime structure as a real which is equivalent to the COleDateTime class used in the Microsoft Foundation Classes (MFC). Most COM objects will accept dates in this format.

PWDateTime.GetAsClarionDate procedure(),date – This method returns the contents of the PWDateTime structure as a Clarion DATE. Use this method in conjunction with PWDateTime.GetAsClarionTime to get the complete date/time component.

PWDateTime.GetAsClarionTime procedure(),time – This method returns the contents of the PWDateTime structure as a Clarion TIME. Use this method in conjunction with PWDateTime.GetAsClarionDate to get the complete date/time component.

PWDateTime.Now procedure() – This method sets the PWDateTime structure to the current time using the GetLocalTime API function.

PWInvokeHelper

The PWInvokeHelper class is used in late binding automation to handle parameter passing to COM object methods via the PWDDispatch.Invoke method. Plugware COM provides a number of different versions of PWDDispatch.Invoke which will be discussed in the Late Binding Automation section of this documentation. These Invoke methods will cover calling methods with up to 10 parameters; therefore, one will generally not need to manipulate a PWInvokeHelper object directly. The easiest way to understand the functionality of PWInvokeHelper is to follow the code in the PWDDispatch.Invoke methods in pwcom.clw.

PWSafeArray

The PWSafeArray encapsulates the functionality of the SAFEARRAY data type. A SAFEARRAY is an array which includes boundary information. This provides the developer with the size and dimensions of the array, thus eliminating the possibility of out of bounds addressing errors. The PWSafeArray class effectively wraps the SafeArray COM API functions defined in OLEAUT32.DLL. The methods of interest to the developer inside the PWCOMError class are:

PWSafeArray.Attach procedure(procedure(*tVariant vtsa, short fSelfCleaning = true),short) – This method attaches a PWSafeArray object to a variant returned from a call to a COM object method. It returns true if the attach operation succeeds and false if it fails.

PWSafeArray.Attach procedure(* SAFEARRAY sa, short fSelfCleaning) – Attaches a PWSafeArray object to a data structure of type _SAFEARRAY. It returns true if the attach operation succeeds and false if it fails.

PWSafeArray.Create procedure(VARTYPE vt, long ulCount, long lLBound)– Creates a new array of type vt (see pwcomdef.inc for a list of valid VARTYPEs) with a size and the lower bound specification. It makes calls internally to SafeArrayCreate to create the new array descriptor and allocate and initialize the data type for the array.

PWSafeArray.Copy procedure(*HRESULT hr),*PWSafeArray - Copies the existing PWSafeArray and returns a reference to the newly created PWSafeArray.

PWSafeArray.GetType procedure,long – Returns the type of SafeArray contained in the PWSafeArray object. See pwcomdef.inc for a list of the valid VARTYPEs.

PWSafeArray.GetLowerBound procedure(long uDim = 0),long – Returns the lower bound element of the SafeArray. This return value is zero based.

PWSafeArray.GetUpperBound procedure(long uDim = 0),long – Returns the upper bound element of the SafeArray. This return value is zero based.

PWSafeArray.GetDimensions procedure(*long xElems, *long yElems, *long nDims) – Returns the number of dimensions of the array and the number of elements in the x and y dimensions.

PWSafeArray.GetDimension procedure(long nDim, *long nElems) – Returns the zero based number of elements in the dimension specified by the first parameter. Pass 0 as the first parameter to get the number of elements in the 1st dimension of the array.

PWSafeArray.AccessData procedure – Retrieves a pointer to the array data and increments the lock count of the array. You must call PWSafeArray.UnaccessData once you are finished manipulating the data after calling this method.

PWSafeArray.UnaccessData procedure – Calls SafeArrayUnaccessData internally to decrement the lock count of the array. You should call this method once you are finished manipulating the data after calling PWSafeArray.AccessData.

Early Binding Automation

The PWCOMObject class defined in pwcom.clw is responsible for handling the low level instantiation of COM objects and the proper release of those interfaces once the developer is finished using them. The PWDIsapatch class used in Late Binding Automation is actually derived from the PWCOMObject and will be covered in the next section.

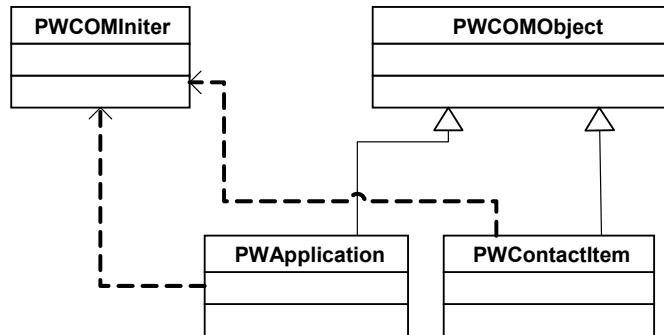


FIGURE 1.2 The relationship between the PWCOMObject and two of the Outlook COM objects, PWApplication and PWContactItem. The dotted arrows indicate that the child objects are reliant on a PWCOMInitier being instantiated for the current thread.

The methods of interest to the developer inside the PWCOMObject class are:

PWCOMObject.AttachExisting procedure(RCLSID rclsid, REFIID riid) - Attaches the object to an existing instance of the CLSID passed in, if it exists, and attaches itself to the IID passed as the second parameter.

PWCOMObject.CreateInstance procedure(REFCLSID rclsid, REFIID riid, long dwClsContext = CLSCTX_ALL) - Uses the CoCreateInstance API call to create an instance of the CLSID passed in. It attaches itself to the IID passed as the second parameter. The third parameter is the context in which the code that manages the newly created object will run in. It is taken from the CLSCTX itemized equates defined in pwcomdef.inc.

PWCOMObject.GetInterface procedure(long pInterface, REFIID riid, *long pvObject, short fRelease) - This method returns an interface from a pointer to an IDispatch or IUnknown interface. The first parameter contains the long pointer to the IDispatch interface passed in. The second parameter contains the IID of the interface we wish to get back from the internal call to QueryInterface. The third parameter will contain a pointer to the output interface. The fourth parameter is a true/false flag where a true will cause the internally referenced IUnknown to release. This method will be used most frequently in the early binding automation when a procedure call returns a pointer to an IDispatch (i.e. PWApplication.CreateItem, PWItems._Add, etc. both of which are defined in outlook.inc).

PWCOMObject.QueryInterface procedure(REFIID riid, *long pvObject) – This method calls QueryInterface on the object wrapped in the current PWCOMObject. This might be used when attaching PWCOMObject to IUnknown interfaces to determine what kind of interface is contained in the object.

PWCOMObject.AddRef procedure – This is equivalent to calling IUnknown.AddRef and is used internally by the PWCOMObject class.

PWCOMObject.Release procedure – This method releases the interface contained in the PWCOMObject and is equivalent to calling IUnknown.Release. It maintains an internal reference count to ensure that the object is released at the proper time.

PWCOMObject.Attach procedure(long pUnk, short fNoAddRef = false) – This method attaches the object to a passed pointer to an IUnknown interface. The first parameter contains the long pointer to the IUnknown interface we wish to attach to. The second parameter determines whether we should call IUnknown.AddRef on this interface.

Late Binding Automation

The PWDDispatch class defined in pwcom.clw is the derived late bound version of the PWCOMObject class. It encapsulates much of the functionality of working with IDispatch interfaces. PWDDispatch has been designed to abstract complexity away from the development of COM automation solutions by providing a simple interface with functions to perform type conversions and variant parameter passing. Figure 1.3 shows the relationship between PWDDispatch and the classes it relies upon to get its job done.

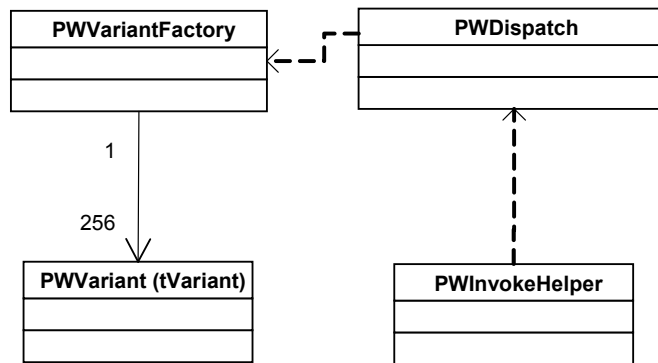


FIGURE 1.3 The PWVariantFactory object is a dependency of the PWDDispatch class because variants are cached in the global PWVariantFactory when handling the Invoke calls to the COM object inside PWDDispatch. PWVariantFactory can have a maximum of 256 cached variants at a time. PWInvokeHelper can be used independently of the PWVariantFactory to prepare parameter lists and call Invoke on them.

PWVariantFactory is a class used by a global helper function called `_vt`. The `_vt` function will convert Clarion data types into variants suitable for passing to COM objects. The PWVariantFactory is globally instantiated and it is responsible for caching the variants created when `_vt` is called. The PWVariantFactory stores a maximum of 256 cached variants to conserve memory (this can be edited in the `PWVariantFactory.Construct` method in `pwcom.clw`) so you will see references to `PWVariantFactory.Release()` in the examples to clear the cache for re-use.

The PWDDispatch class relies on internal PWInvokeHelper objects inside its Invoke methods to handle initializing the parameter lists before calling `IDispatch.Invoke`. There are four versions of the `PWDDispatch.Invoke` class which are of particular note. They take 0, 1, 5, and 10 variant parameters respectively. This allows the COM developer to pass optional parameters to procedures which take a long list. For those rare instances in which you will need to pass more than 10 parameters to a COM procedure you will need to instantiate your own PWInvokeHelper and use the `self.SetParam` method similarly to the way it is used in the `PWDDispatch.Invoke` method for 10 parameters declared in `pwcom.inc`.

The methods of interest to the developer inside the PWDDispatch class are:

PWDispatch.CreateInstance procedure(*cstring szObject) – Take a passed cstring containing the name of the object to instantiate (i.e. ‘Outlook.Application’ for Microsoft Outlook) and handles the internal calls into the COM API to call CoCreateInstance on the proper CLSID.

PWDispatch.CreateInstance procedure(*cstring szProgID, long flags) – An overloaded version of the CreateInstance above which takes a second parameter containing the context in which the code that manages the newly created object will run in. It is taken from the CLSCTX itemized equates defined in pwcomdef.inc.

PWDispatch.AttachExisting procedure(RCLSID rclsid) - Attaches the object to an existing instance of the CLSID passed in, if it exists. The PWDispatch will then be attached to the IDispatch interface of the object referenced by the CLSID.

PWDispatch.AttachExisting procedure(*cstring szProgID) – An overloaded version of the AttachExisting above which takes a cstring containing the name of the existing instance of an object to attach to (i.e. ‘Outlook.Application’ will attach to an existing instance of Microsoft Outlook).

PWDispatch.Attach procedure(long pIDisp, short fNoAddRef) – This method attaches the object to a passed pointer to an IDispatch interface. The first parameter contains the long pointer to the IDispatch interface we wish to attach to. The second parameter determines whether we should call IDispatch.AddRef on this interface.

PWDispatch.Attach procedure(*tVariant vtDisp, short fNoAddRef) – An overloaded version of the Attach above which takes a variant parameter containing an IDispatch interface instead of a long pointer to the interface. This allows us to attach a PWDispatch to an output parameter from a call to PWDispatch.Invoke.

PWDispatch.Attach procedure(*IDispatch IDisp, short fNoAddRef) – An overloaded version of the Attach above which takes an IDispatch interface parameter rather than a long pointer or a variant.

We alluded to the Invoke methods earlier and discussed the fact that there are several different versions depending on the number of parameters the COM method may take. Optional parameters are handled easily with the variant vtMissing declared globally in the Plugware COM classes. vtMissing can be used to pass empty or optional parameters to the COM methods. The next section covers the use of the PWDispatch.Invoke method (which is a wrapper around the IDispatch.Invoke COM method) and gives examples of using the vtMissing variant to pass empty or optional parameters.

PWDispatch.Invoke procedure(*cstring szMember, long wFlags, *tVariant vtRet, short fDispatchPut) – This version of PWDispatch.Invoke takes no input parameters. The first parameter is a cstring containing the name of the member method or property to

invoke. The second parameter is a flag for IDispatch::Invoke and can be one of the following constants declared in pwcomdef.inc:

DISPATCH_METHOD – This Member is a COM object method to execute
 DISPATCH_PROPERTYGET – This Member is a property we wish to retrieve
 DISPATCH_PROPERTYPUT – This Member is a property we wish to put
 DISPATCH_PROPERTYPUTREF – This Member is a property we wish to put by reference, not by value

The third parameter is a variant output parameter from the Invoke call. It will contain the return value from the call to IDispatch.Invoke once it has completed. The fourth parameter is a true/false flag and should be set to true only when the COM object expects a put-by-reference rather than a put-by-value. An example call is listed below:

```
szMember = 'Count'
hr = DispInboxItems.Invoke(szMember, DISPATCH_PROPERTYGET, vtIDisp)
```

PWDispatch.Invoke procedure(*cstring szMember, long wFlags, *tVariant vtParam1, *tVariant vtRet, short fDispatchPut = false) – This version of PWDispatch.Invoke is identical to the one above except it takes one input parameter called vtParam1. The other parameters are treated the same as the version which takes no input parameters. An example call is listed below:

```
szMember = 'GetDefaultFolder'
hr = DispNamespace.Invoke(szMember, DISPATCH_METHOD, _vt(lolFolderInbox), vtIDisp)
```

PWDispatch.Invoke procedure(*cstring szMember, long wFlags, long nParams, *tVariant vtParam1, *tVariant vtParam2, *tVariant vtParam3, *tVariant vtParam4, *tVariant vtParam5, *tVariant vtRet, short fDispatchPut) – This version of PWDispatch.Invoke is identical to the one above except it takes five input parameters called vtParam1, vtParam2, vtParam3, vtParam4, and vtParam5. This method can be used for procedures which take between 1 and 5 parameters because we can pass a vtMissing for parameters which we wish to omit. An example call listed below uses this version to call a method requiring only one parameter:

```
szMember = 'GetDefaultFolder'
hr = DispNamespace.Invoke(szMember, DISPATCH_METHOD, _vt(lolFolderInbox), _vt(vtMissing), _vt(vtMissing), _vt(vtMissing), _vt(vtMissing), vtIDisp)
```

PWDispatch.Invoke procedure(*cstring szMember, long wFlags, long nParams, *tVariant vtParam1, *tVariant vtParam2, *tVariant vtParam3, *tVariant vtParam4, *tVariant vtParam5, *tVariant vtParam6, *tVariant vtParam7, *tVariant vtParam8, *tVariant vtParam9, *tVariant vtParam10, *tVariant vtRet, short fDispatchPut) – This version of PWDispatch.Invoke is identical to the one above except it takes ten input parameters called vtParam1, ..., vtParam10. This method can be used for procedures which take between 1 and 10 parameters using the vtMissing variant.

Multi-DLL Considerations

Plugware COM provides support for compiling the classes into a multi-DLL project. All Plugware COM products ship with the pwcom.exp file installed into the Libsrc directory. This export file should be merged into the export file of the DLL containing your base class declarations and data. Once you have done this the Plugware COM objects can be referenced from any DLL or EXE which references this base DLL.

The linking of the classes is handled using project pragma settings similar to the way the ABC classes are exported in Clarion. A list of the pragmas for exporting the Plugware COM class definitions is described below for those who prefer to hand code their project in Clarion:

Applications Compiled with Data Local to the Module

```
%#pragma define(_APIDllMode_=>off)
%#pragma define(_APILinkMode=>on)
%#pragma define(_COMDllMode_=>off)
%#pragma define(_COMLinkMode_=>on)
```

Applications Compiled with Data External to the Module

```
%#pragma define(_APIDllMode_=>on)
%#pragma define(_APILinkMode=>off)
%#pragma define(_COMDllMode_=>on)
%#pragma define(_COMLinkMode_=>off)
```

qbFUSE Overview

qbFUSE consists of generated wrapper classes for the QuickBooks QBFC SDK version 1.1. The source files encapsulate the interfaces and data types for QuickBooks automation.

The seven source files which form the OutlookFUSE product and the early bound version of the Microsoft Outlook COM object model are:

qbfiid.inc – Contains the definitions for the CLSID and IID values for the entire QBFC object model. It is included by the qbfint.inc file.

qbfint.inc – Contains the QBFC object model interfaces defined with Clarion compatible prototypes. It is included by the qbf.inc.

qbfddef.inc – Contains the QBFC object model constants defined as Clarion equates. It is included by the olint.inc file.

outlook.inc – Contains the generated Plugware COM classes to implement the interfaces defined in qbfint.inc.

qbf.clw – qbf10.clw – These 11 source files contain the source implementation of the classes defined in qbf.inc – qbf10.inc.

qbf.inc – qbf10.inc – These 11 source files contain the class definitions of the classes implemented in qbf.clw – qbf10.clw.

This manual assumes that you have access to the QuickBooks SDK documentation contained in the installation of the SDK available from Intuit. It will be helpful to refer to the *QuickBooks Foundation Class (QBFC) Library Programmer Guide* and the *QuickBooks SDK Version 1.1 Reference Manual* when you are developing solutions using qbFUSE.

List of Interfaces

The following table lists the interfaces contained in the qbFUSE product along with their equivalent QuickBooks interface name and their file location. Due to the size of the wrapper and Clarion's limitations with respect to the number of class definitions in each source file, qbFUSE has been split up into many source files with each one having 40 class definitions corresponding to an interface in QuickBooks. Please refer to the QBFC SDK documentation for more details on what each interface accomplishes when automating QuickBooks.

qbFUSE Interface	QuickBooks Interface	Source File
PWIAccountAdd	IAccountAdd	qb11.inc
PWIAccountFilter	IAccountFilter	qb11.inc
PWIAccountingPreferences	IAccountingPreferences	qb11.inc
PWIAccountListFilter	IAccountListFilter	qb11.inc
PWIAccountQuery	IAccountQuery	qb11.inc
PWIAccountRet	IAccountRet	qb11.inc
PWIAccountRetList	IAccountRetList	qb11.inc
PWIAAddress	IAddress	qb11.inc
PWIAgingReportQuery	IAgingReportQuery	qb11.inc
PWIAppliedToTxnAdd	IAppliedToTxnAdd	qb11.inc
PWIAppliedToTxnAddList	IAppliedToTxnAddList	qb11.inc
PWIAppliedToTxnRet	IAppliedToTxnRet	qb11.inc
PWIAppliedToTxnRetList	IAppliedToTxnRetList	qb11.inc
PWIAtributesRqSet	IAttributesRqSet	qb11.inc
PWIAtributesRsSet	IAttributesRsSet	qb11.inc
PWIBasicPropertyType	IBasicPropertyType	qb11.inc
PWIBillAdd	IBillAdd	qb11.inc
PWIBillFilter	IBillFilter	qb11.inc
PWIBillPaymentCheckAdd	IBillPaymentCheckAdd	qb11.inc
PWIBillPaymentCheckQuery	IBillPaymentCheckQuery	qb11.inc
PWIBillPaymentCheckRet	IBillPaymentCheckRet	qb11.inc
PWIBillPaymentCheckRetList	IBillPaymentCheckRetList	qb11.inc
PWIBillPaymentCreditCardAdd	IBillPaymentCreditCardAdd	qb11.inc
PWIBillPaymentCreditCardQuery	IBillPaymentCreditCardQuery	qb11.inc
PWIBillPaymentCreditCardRet	IBillPaymentCreditCardRet	qb10.inc
PWIBillPaymentCreditCardRetList	IBillPaymentCreditCardRetList	qb10.inc
PWIBillQuery	IBillQuery	qb10.inc
PWIBillRet	IBillRet	qb10.inc
PWIBillRetList	IBillRetList	qb10.inc
PWIBillToPay	IBillToPay	qb10.inc
PWIBillToPayQuery	IBillToPayQuery	qb10.inc
PWIBillToPayRet	IBillToPayRet	qb10.inc
PWIBillToPayRetList	IBillToPayRetList	qb10.inc
PWIBSTRList	IBSTRList	qb11.inc

QBFUSE GUIDE

PWICashBackInfoAdd	ICashBackInfoAdd	qbf10.inc
PWICashBackInfoRet	ICashBackInfoRet	qbf10.inc
PWIChargeAdd	IChargeAdd	qbf10.inc
PWIChargeQuery	IChargeQuery	qbf10.inc
PWIChargeRet	IChargeRet	qbf10.inc
PWIChargeRetList	IChargeRetList	qbf10.inc
PWICheckAdd	ICheckAdd	qbf10.inc
PWICheckQuery	ICheckQuery	qbf10.inc
PWICheckRet	ICheckRet	qbf10.inc
PWICheckRetList	ICheckRetList	qbf10.inc
PWIClassAdd	IClassAdd	qbf10.inc
PWIClassQuery	IClassQuery	qbf10.inc
PWIClassRet	IClassRet	qbf10.inc
PWIClassRetList	IClassRetList	qbf10.inc
PWIClearedStatusMod	IClearedStatusMod	qbf10.inc
PWIColData	IColData	qbf10.inc
PWIColDataList	IColDataList	qbf10.inc
PWIColDesc	IColDesc	qbf10.inc
PWIColDescList	IColDescList	qbf10.inc
PWIColTitle	IColTitle	qbf10.inc
PWIColTitleList	IColTitleList	qbf10.inc
PWICompanyActivityRet	ICompanyActivityRet	qbf10.inc
PWICompanyQuery	ICompanyQuery	qbf10.inc
PWICompanyRet	ICompanyRet	qbf10.inc
PWICreditCardChargeAdd	ICreditCardChargeAdd	qbf10.inc
PWICreditCardChargeQuery	ICreditCardChargeQuery	qbf10.inc
PWICreditCardChargeRet	ICreditCardChargeRet	qbf10.inc
PWICreditCardChargeRetList	ICreditCardChargeRetList	qbf10.inc
PWICreditCardCreditAdd	ICreditCardCreditAdd	qbf10.inc
PWICreditCardCreditQuery	ICreditCardCreditQuery	qbf10.inc
PWICreditCardCreditRet	ICreditCardCreditRet	qbf10.inc
PWICreditCardCreditRetList	ICreditCardCreditRetList	qbf9.inc
PWICreditMemoAdd	ICreditMemoAdd	qbf9.inc
PWICreditMemoLineAdd	ICreditMemoLineAdd	qbf9.inc
PWICreditMemoLineGroupAdd	ICreditMemoLineGroupAdd	qbf9.inc
PWICreditMemoLineGroupRet	ICreditMemoLineGroupRet	qbf9.inc
PWICreditMemoLineRet	ICreditMemoLineRet	qbf9.inc
PWICreditMemoLineRetList	ICreditMemoLineRetList	qbf9.inc
PWICreditMemoQuery	ICreditMemoQuery	qbf9.inc
PWICreditMemoRet	ICreditMemoRet	qbf9.inc
PWICreditMemoRetList	ICreditMemoRetList	qbf9.inc
PWICreditToApply	ICreditToApply	qbf9.inc
PWICurrentAppAccessRights	ICurrentAppAccessRights	qbf9.inc
PWICustomDetailReportQuery	ICustomDetailReportQuery	qbf9.inc
PWICustomerAdd	ICustomerAdd	qbf9.inc
PWICustomerListFilter	ICustomerListFilter	qbf9.inc

QBFUSE GUIDE

PWICustomerMod	ICustomerMod	qbf9.inc
PWICustomerMsgAdd	ICustomerMsgAdd	qbf9.inc
PWICustomerMsgQuery	ICustomerMsgQuery	qbf9.inc
PWICustomerMsgRet	ICustomerMsgRet	qbf9.inc
PWICustomerMsgRetList	ICustomerMsgRetList	qbf9.inc
PWICustomerQuery	ICustomerQuery	qbf9.inc
PWICustomerRet	ICustomerRet	qbf9.inc
PWICustomerRetList	ICustomerRetList	qbf9.inc
PWICustomerTypeAdd	ICustomerTypeAdd	qbf9.inc
PWICustomerTypeQuery	ICustomerTypeQuery	qbf9.inc
PWICustomerTypeRet	ICustomerTypeRet	qbf9.inc
PWICustomerTypeRetList	ICustomerTypeRetList	qbf9.inc
PWICustomSummaryReportQuery	ICustomSummaryReportQuery	qbf9.inc
PWIDataExtAdd	IDataExtAdd	qbf9.inc
PWIDataExtDefAdd	IDataExtDefAdd	qbf9.inc
PWIDataExtDefDel	IDataExtDefDel	qbf9.inc
PWIDataExtDefDelRet	IDataExtDefDelRet	qbf9.inc
PWIDataExtDefMod	IDataExtDefMod	qbf9.inc
PWIDataExtDefQuery	IDataExtDefQuery	qbf9.inc
PWIDataExtDefRet	IDataExtDefRet	qbf9.inc
PWIDataExtDefRetList	IDataExtDefRetList	qbf9.inc
PWIDataExtDel	IDataExtDel	qbf9.inc
PWIDataExtDelRet	IDataExtDelRet	qbf9.inc
PWIDataExtMod	IDataExtMod	qbf9.inc
PWIDataExtRet	IDataExtRet	qbf9.inc
PWIDataExtRetList	IDataExtRetList	qbf8.inc
PWIDataRow	IDataRow	qbf8.inc
PWIDateDrivenTermsAdd	IDateDrivenTermsAdd	qbf8.inc
PWIDateDrivenTermsMod	IDateDrivenTermsMod	qbf8.inc
PWIDateDrivenTermsQuery	IDateDrivenTermsQuery	qbf8.inc
PWIDateDrivenTermsRet	IDateDrivenTermsRet	qbf8.inc
PWIDateDrivenTermsRetList	IDateDrivenTermsRetList	qbf8.inc
PWIDeletedDateRangeFilter	IDeletedDateRangeFilter	qbf8.inc
PWIDepositAdd	IDepositAdd	qbf8.inc
PWIDepositFilter	IDepositFilter	qbf8.inc
PWIDepositInfo	IDepositInfo	qbf8.inc
PWIDepositLineAdd	IDepositLineAdd	qbf8.inc
PWIDepositLineAddList	IDepositLineAddList	qbf8.inc
PWIDepositLineRet	IDepositLineRet	qbf8.inc
PWIDepositLineRetList	IDepositLineRetList	qbf8.inc
PWIDepositQuery	IDepositQuery	qbf8.inc
PWIDepositRet	IDepositRet	qbf8.inc
PWIDepositRetList	IDepositRetList	qbf8.inc
PWIEarnings	IEarnings	qbf8.inc
PWIEarningsList	IEarningsList	qbf8.inc
PWIEmployeeAdd	IEmployeeAdd	qbf8.inc

QBFUSE GUIDE

PWIEmployeeAddress	IEmployeeAddress	qbfs.inc
PWIEmployeeMod	IEmployeeMod	qbfs.inc
PWIEmployeePayrollInfo	IEmployeePayrollInfo	qbfs.inc
PWIEmployeePayrollInfoMod	IEmployeePayrollInfoMod	qbfs.inc
PWIEmployeeQuery	IEmployeeQuery	qbfs.inc
PWIEmployeeRet	IEmployeeRet	qbfs.inc
PWIEmployeeRetList	IEmployeeRetList	qbfs.inc
PWIENAccountTypeList	IENAccountTypeList	qbfs.inc
PWIENAssignToObjectList	IENAssignToObjectList	qbfs.inc
PWIENIncludeColumnList	IENIncludeColumnList	qbfs.inc
PWIENListDelTypeList	IENListDelTypeList	qbfs.inc
PWIENRemoveFromObjectList	IENRemoveFromObjectList	qbfs.inc
PWIEntityFilter	IEntityFilter	qbfs.inc
PWIEntityQuery	IEntityQuery	qbfs.inc
PWIENTxnDelTypeList	IENTxnDelTypeList	qbfs.inc
PWIENTxnTypeFilterList	IENTxnTypeFilterList	qbfs.inc
PWIEstimateAdd	IEstimateAdd	qbfs.inc
PWIEstimateLineAdd	IEstimateLineAdd	qbfs.inc
PWIEstimateLineGroupAdd	IEstimateLineGroupAdd	qbfs.inc
PWIEstimateLineGroupRet	IEstimateLineGroupRet	qbfs.inc
PWIEstimateLineRet	IEstimateLineRet	qbfs.inc
PWIEstimateLineRetList	IEstimateLineRetList	qbfs.inc
PWIEstimateQuery	IEstimateQuery	qbfs.inc
PWIEstimateRet	IEstimateRet	qbfs.inc
PWIEstimateRetList	IEstimateRetList	qbfs.inc
PWIExpenseLineAdd	IExpenseLineAdd	qbfs.inc
PWIExpenseLineAddList	IExpenseLineAddList	qbfs.inc
PWIExpenseLineRet	IExpenseLineRet	qbfs.inc
PWIExpenseLineRetList	IExpenseLineRetList	qbfs.inc
PWIFinanceChargePreferences	IFinanceChargePreferences	qbfs.inc
PWIGeneralDetailReportQuery	IGeneralDetailReportQuery	qbfs.inc
PWIGeneralSummaryReportQuery	IGeneralSummaryReportQuery	qbfs.inc
PWIGUIDList	IGUIDList	qbfs.inc
PWIHostRet	IHostRet	qbfs.inc
PWIIInventoryAdjustmentAdd	IInventoryAdjustmentAdd	qbfs.inc
PWIIInventoryAdjustmentLineAdd	IInventoryAdjustmentLineAdd	qbfs.inc
PWIIInventoryAdjustmentLineAddList	IInventoryAdjustmentLineAddList	qbfs.inc
PWIIInventoryAdjustmentLineRet	IInventoryAdjustmentLineRet	qbfs.inc
PWIIInventoryAdjustmentLineRetList	IInventoryAdjustmentLineRetList	qbfs.inc
PWIIInventoryAdjustmentQuery	IInventoryAdjustmentQuery	qbfs.inc
PWIIInventoryAdjustmentRet	IInventoryAdjustmentRet	qbfs.inc
PWIIInventoryAdjustmentRetList	IInventoryAdjustmentRetList	qbfs.inc
PWIIInvoiceAdd	IInvoiceAdd	qbfs.inc
PWIIInvoiceFilter	IInvoiceFilter	qbfs.inc
PWIIInvoiceLineAdd	IInvoiceLineAdd	qbfs.inc
PWIIInvoiceLineGroupAdd	IInvoiceLineGroupAdd	qbfs.inc

PWInvoiceLineGroupRet	IInvoiceLineGroupRet	qb7.inc
PWInvoiceLineRet	IInvoiceLineRet	qb7.inc
PWInvoiceLineRetList	IInvoiceLineRetList	qb7.inc
PWInvoiceQuery	IInvoiceQuery	qb7.inc
PWInvoiceRet	IInvoiceRet	qb7.inc
PWInvoiceRetList	IInvoiceRetList	qb7.inc
PWItemDiscountAdd	IItemDiscountAdd	qb7.inc
PWItemDiscountQuery	IItemDiscountQuery	qb7.inc
PWItemDiscountRet	IItemDiscountRet	qb7.inc
PWItemDiscountRetList	IItemDiscountRetList	qb7.inc
PWItemFilter	IItemFilter	qb7.inc
PWItemGroupAdd	IItemGroupAdd	qb7.inc
PWItemGroupLine	IItemGroupLine	qb7.inc
PWItemGroupLineAdd	IItemGroupLineAdd	qb7.inc
PWItemGroupLineList	IItemGroupLineList	qb6.inc
PWItemGroupLineRet	IItemGroupLineRet	qb6.inc
PWItemGroupQuery	IItemGroupQuery	qb6.inc
PWItemGroupRet	IItemGroupRet	qb6.inc
PWItemGroupRetList	IItemGroupRetList	qb6.inc
PWItemInventoryAdd	IItemInventoryAdd	qb6.inc
PWItemInventoryAssemblyAdd	IItemInventoryAssemblyAdd	qb6.inc
PWItemInventoryAssemblyLine	IItemInventoryAssemblyLine	qb6.inc
PWItemInventoryAssemblyLineList	IItemInventoryAssemblyLineList	qb6.inc
PWItemInventoryAssemblyQuery	IItemInventoryAssemblyQuery	qb6.inc
PWItemInventoryAssemblyRet	IItemInventoryAssemblyRet	qb6.inc
PWItemInventoryAssemblyRetList	IItemInventoryAssemblyRetList	qb6.inc
PWItemInventoryMod	IItemInventoryMod	qb6.inc
PWItemInventoryQuery	IItemInventoryQuery	qb6.inc
PWItemInventoryRet	IItemInventoryRet	qb6.inc
PWItemInventoryRetList	IItemInventoryRetList	qb6.inc
PWItemLineAdd	IItemLineAdd	qb6.inc
PWItemLineRet	IItemLineRet	qb6.inc
PWItemLineRetList	IItemLineRetList	qb6.inc
PWItemNonInventoryAdd	IItemNonInventoryAdd	qb6.inc
PWItemNonInventoryMod	IItemNonInventoryMod	qb6.inc
PWItemNonInventoryQuery	IItemNonInventoryQuery	qb6.inc
PWItemNonInventoryRet	IItemNonInventoryRet	qb6.inc
PWItemNonInventoryRetList	IItemNonInventoryRetList	qb6.inc
PWItemOtherChargeAdd	IItemOtherChargeAdd	qb6.inc
PWItemOtherChargeQuery	IItemOtherChargeQuery	qb6.inc
PWItemOtherChargeRet	IItemOtherChargeRet	qb6.inc
PWItemOtherChargeRetList	IItemOtherChargeRetList	qb6.inc
PWItemPaymentAdd	IItemPaymentAdd	qb6.inc
PWItemPaymentQuery	IItemPaymentQuery	qb6.inc
PWItemPaymentRet	IItemPaymentRet	qb6.inc
PWItemPaymentRetList	IItemPaymentRetList	qb6.inc

QBFUSE GUIDE

PWItemQuery	ItemQuery	qbf6.inc
PWItemSalesTaxAdd	ItemSalesTaxAdd	qbf6.inc
PWItemSalesTaxGroupAdd	ItemSalesTaxGroupAdd	qbf6.inc
PWItemSalesTaxGroupQuery	ItemSalesTaxGroupQuery	qbf6.inc
PWItemSalesTaxGroupRet	ItemSalesTaxGroupRet	qbf6.inc
PWItemSalesTaxGroupRetList	ItemSalesTaxGroupRetList	qbf6.inc
PWItemSalesTaxQuery	ItemSalesTaxQuery	qbf6.inc
PWItemSalesTaxRet	ItemSalesTaxRet	qbf6.inc
PWItemSalesTaxRetList	ItemSalesTaxRetList	qbf5.inc
PWItemServiceAdd	ItemServiceAdd	qbf5.inc
PWItemServiceMod	ItemServiceMod	qbf5.inc
PWItemServiceQuery	ItemServiceQuery	qbf5.inc
PWItemServiceRet	ItemServiceRet	qbf5.inc
PWItemServiceRetList	ItemServiceRetList	qbf5.inc
PWItemSubtotalAdd	ItemSubtotalAdd	qbf5.inc
PWItemSubtotalQuery	ItemSubtotalQuery	qbf5.inc
PWItemSubtotalRet	ItemSubtotalRet	qbf5.inc
PWItemSubtotalRetList	ItemSubtotalRetList	qbf5.inc
PWJobReportQuery	JobReportQuery	qbf5.inc
PWJobsAndEstimatesPreferences	JobsAndEstimatesPreferences	qbf5.inc
PWJobTypeAdd	JobTypeAdd	qbf5.inc
PWJobTypeQuery	JobTypeQuery	qbf5.inc
PWJobTypeRet	JobTypeRet	qbf5.inc
PWJobTypeRetList	JobTypeRetList	qbf5.inc
PWJournalCreditLine	JournalCreditLine	qbf5.inc
PWJournalDebitLine	JournalDebitLine	qbf5.inc
PWJournalEntryAdd	JournalEntryAdd	qbf5.inc
PWJournalEntryQuery	JournalEntryQuery	qbf5.inc
PWJournalEntryRet	JournalEntryRet	qbf5.inc
PWJournalEntryRetList	JournalEntryRetList	qbf5.inc
PWLinkedTxn	LinkedTxn	qbf5.inc
PWLinkedTxnList	LinkedTxnList	qbf5.inc
PWListDataExt	ListDataExt	qbf5.inc
PWListDel	ListDel	qbf5.inc
PWListDeletedQuery	ListDeletedQuery	qbf5.inc
PWListDeletedRet	ListDeletedRet	qbf5.inc
PWListDeletedRetList	ListDeletedRetList	qbf5.inc
PWListDelRet	ListDelRet	qbf5.inc
PWListFilter	ListFilter	qbf5.inc
PWModifiedDateRangeFilter	ModifiedDateRangeFilter	qbf5.inc
PWMsgSetRequest	MsgSetRequest	qbf13.inc
PWMsgSetResponse	MsgSetResponse	qbf13.inc
PWNameFilter	NameFilter	qbf5.inc
PWNameRangeFilter	NameRangeFilter	qbf5.inc
PWObjectType	ObjectType	qbf13.inc
PWIORAccountFilter	IORAccountFilter	qbf5.inc

QBFUSE GUIDE

PWIORAccountListQuery	IORAccountListQuery	qbf5.inc
PWIORApplyPayment	IORApplyPayment	qbf5.inc
PWIORBillQuery	IORBillQuery	qbf5.inc
PWIORBillToPayRet	IORBillToPayRet	qbf5.inc
PWIORCheckPrint	IORCheckPrint	qbf5.inc
PWIORCreditMemoLineAdd	IORCreditMemoLineAdd	qbf4.inc
PWIORCreditMemoLineAddList	IORCreditMemoLineAddList	qbf4.inc
PWIORCreditMemoLineRet	IORCreditMemoLineRet	qbf4.inc
PWIORCreditMemoLineRetList	IORCreditMemoLineRetList	qbf4.inc
PWIORCustomerListQuery	IORCustomerListQuery	qbf4.inc
PWIORDataExtDefQuery	IORDataExtDefQuery	qbf4.inc
PWIORDateRangeFilter	IORDateRangeFilter	qbf4.inc
PWIORDepositLineAdd	IORDepositLineAdd	qbf4.inc
PWIORDepositQuery	IORDepositQuery	qbf4.inc
PWIORDiscountRate	IORDiscountRate	qbf4.inc
PWIOREarnings	IOREarnings	qbf4.inc
PWIOREntityFilter	IOREntityFilter	qbf4.inc
PWIOREntityRet	IOREntityRet	qbf4.inc
PWIOREntityRetList	IOREntityRetList	qbf4.inc
PWIOREstimateLineAdd	IOREstimateLineAdd	qbf4.inc
PWIOREstimateLineAddList	IOREstimateLineAddList	qbf4.inc
PWIOREstimateLineRet	IOREstimateLineRet	qbf4.inc
PWIOREstimateLineRetList	IOREstimateLineRetList	qbf4.inc
PWIORInventoryAdjustmentQuery	IORInventoryAdjustmentQuery	qbf4.inc
PWIORInvoiceLineAdd	IORInvoiceLineAdd	qbf4.inc
PWIORInvoiceLineAddList	IORInvoiceLineAddList	qbf4.inc
PWIORInvoiceLineRet	IORInvoiceLineRet	qbf4.inc
PWIORInvoiceLineRetList	IORInvoiceLineRetList	qbf4.inc
PWIORInvoiceQuery	IORInvoiceQuery	qbf4.inc
PWIORItemFilter	IORItemFilter	qbf4.inc
PWIORItemLineAdd	IORItemLineAdd	qbf4.inc
PWIORItemLineAddList	IORItemLineAddList	qbf4.inc
PWIORItemLineRet	IORItemLineRet	qbf4.inc
PWIORItemLineRetList	IORItemLineRetList	qbf4.inc
PWIORItemRet	IORItemRet	qbf4.inc
PWIORItemRetList	IORItemRetList	qbf4.inc
PWIORJournalLine	IORJournalLine	qbf4.inc
PWIORJournalLineList	IORJournalLineList	qbf4.inc
PWIORListQuery	IORListQuery	qbf4.inc
PWIORListTxn	IORListTxn	qbf4.inc
PWIORMarkupRate	IORMarkupRate	qbf4.inc
PWIORNameFilter	IORNameFilter	qbf4.inc
PWIORPrice	IORPrice	qbf4.inc
PWIORPurchaseOrderLineAdd	IORPurchaseOrderLineAdd	qbf4.inc
PWIORPurchaseOrderLineAddList	IORPurchaseOrderLineAddList	qbf4.inc
PWIORPurchaseOrderLineRet	IORPurchaseOrderLineRet	qbf3.inc

QBFUSE GUIDE

PWIORPurchaseOrderLineRetList	IORPurchaseOrderLineRetList	qbf3.inc
PWIORQuantityAdjustment	IORQuantityAdjustment	qbf3.inc
PWIORRate	IORRate	qbf3.inc
PWIORRefNumberFilter	IORRefNumberFilter	qbf3.inc
PWIORReportAccountFilter	IORReportAccountFilter	qbf3.inc
PWIORReportData	IORReportData	qbf3.inc
PWIORReportDataList	IORReportDataList	qbf3.inc
PWIORReportEntityFilter	IORReportEntityFilter	qbf3.inc
PWIORReportItemFilter	IORReportItemFilter	qbf3.inc
PWIORReportPeriod	IORReportPeriod	qbf3.inc
PWIORSalesPurchase	IOSalesPurchase	qbf3.inc
PWIORSalesPurchaseMod	IOSalesPurchaseMod	qbf3.inc
PWIORSalesReceiptLineAdd	IOSalesReceiptLineAdd	qbf3.inc
PWIORSalesReceiptLineAddList	IOSalesReceiptLineAddList	qbf3.inc
PWIORSalesReceiptLineRet	IOSalesReceiptLineRet	qbf3.inc
PWIORSalesReceiptLineRetList	IOSalesReceiptLineRetList	qbf3.inc
PWIORSalesTaxPaymentCheckQuery	IOSalesTaxPaymentCheckQuery	qbf3.inc
PWIORTermsRet	IORTermsRet	qbf3.inc
PWIORTermsRetList	IORTermsRetList	qbf3.inc
PWIORTimeTrackingEntityFilter	IORTimeTrackingEntityFilter	qbf3.inc
PWIORTimeTrackingTxnQuery	IORTimeTrackingTxnQuery	qbf3.inc
PWIORToDoListQuery	IORToDoListQuery	qbf3.inc
PWIORTxnDateRangeFilter	IORTxnDateRangeFilter	qbf3.inc
PWIORTxnQuery	IORTxnQuery	qbf3.inc
PWIORTypeAdjustment	IORTypeAdjustment	qbf3.inc
PWIORVendorListQuery	IORVendorListQuery	qbf3.inc
PWIOtherNameAdd	IOtherNameAdd	qbf3.inc
PWIOtherNameMod	IOtherNameMod	qbf3.inc
PWIOtherNameQuery	IOtherNameQuery	qbf3.inc
PWIOtherNameRet	IOtherNameRet	qbf3.inc
PWIOtherNameRetList	IOtherNameRetList	qbf3.inc
PWIPaymentLine	IPaymentLine	qbf3.inc
PWIPaymentMethodAdd	IPaymentMethodAdd	qbf3.inc
PWIPaymentMethodQuery	IPaymentMethodQuery	qbf3.inc
PWIPaymentMethodRet	IPaymentMethodRet	qbf3.inc
PWIPaymentMethodRetList	IPaymentMethodRetList	qbf3.inc
PWIPayrollItemWageAdd	IPayrollItemWageAdd	qbf3.inc
PWIPayrollItemWageQuery	IPayrollItemWageQuery	qbf3.inc
PWIPayrollItemWageRet	IPayrollItemWageRet	qbf3.inc
PWIPayrollItemWageRetList	IPayrollItemWageRetList	qbf2.inc
PWIPreferencesRet	IPreferencesRet	qbf2.inc
PWIPurchaseOrderAdd	IPurchaseOrderAdd	qbf2.inc
PWIPurchaseOrderLineAdd	IPurchaseOrderLineAdd	qbf2.inc
PWIPurchaseOrderLineGroupAdd	IPurchaseOrderLineGroupAdd	qbf2.inc
PWIPurchaseOrderLineGroupRet	IPurchaseOrderLineGroupRet	qbf2.inc
PWIPurchaseOrderLineRet	IPurchaseOrderLineRet	qbf2.inc

QBFUSE GUIDE

PWIPurchaseOrderLineRetList	IPurchaseOrderLineRetList	qbf2.inc
PWIPurchaseOrderQuery	IPurchaseOrderQuery	qbf2.inc
PWIPurchaseOrderRet	IPurchaseOrderRet	qbf2.inc
PWIPurchaseOrderRetList	IPurchaseOrderRetList	qbf2.inc
PWIPurchasesAndVendorsPreferences	IPurchasesAndVendorsPreferences	qbf2.inc
PWIQBAmountType	IQBAmountType	qbf13.inc
PWIQBBase	IQBBase	qbf.inc
PWIQBBaseRef	IQBBaseRef	qbf2.inc
PWIQBBaseRefList	IQBBaseRefList	qbf2.inc
PWIQBBoolType	IQBBoolType	qbf13.inc
PWIQBDateTimeType	IQBDateTimeType	qbf13.inc
PWIQBDateType	IQBDateType	qbf13.inc
PWIQBENAccountTypeFilterType	IQBENAccountTypeFilterType	qbf13.inc
PWIQBENAccountTypeType	IQBENAccountTypeType	qbf13.inc
PWIQBENAccrualPeriodType	IQBENAccrualPeriodType	qbf13.inc
PWIQBENActiveStatusType	IQBENActiveStatusType	qbf13.inc
PWIQBENAgingReportBasisType	IQBENAgingReportBasisType	qbf13.inc
PWIQBENAgingReportTypeType	IQBENAgingReportTypeType	qbf13.inc
PWIQBENAssignToObjectType	IQBENAssignToObjectType	qbf13.inc
PWIQBENBillableStatusType	IQBENBillableStatusType	qbf13.inc
PWIQBENCalculateChargesFromType	IQBENCalculateChargesFromType	qbf13.inc
PWIQBENClearedStatusType	IQBENClearedStatusType	qbf13.inc
PWIQBENColTypeType	IQBENColTypeType	qbf13.inc
PWIQBENCustomDetailReportTypeType	IQBENCustomDetailReportTypeType	qbf13.inc
PWIQBENCustomSummaryReportTypeType	IQBENCustomSummaryReportTypeType	qbf12.inc
PWIQBENDataExtTypeType	IQBENDataExtTypeType	qbf12.inc
PWIQBENdataTypeType	IQBENdataTypeType	qbf11.inc
PWIQBENDateMacroType	IQBENDateMacroType	qbf12.inc
PWIQBENDeliveryMethodType	IQBENDeliveryMethodType	qbf12.inc
PWIQBENDetailAccountTypeType	IQBENDetailAccountTypeType	qbf12.inc
PWIQBENDoneStatusType	IQBENDoneStatusType	qbf12.inc
PWIQBENEmployeeTypeType	IQBENEmployeeTypeType	qbf12.inc
PWIQBENEntityTypeFilterType	IQBENEntityTypeFilterType	qbf12.inc
PWIQBENFirstDayOfWeekType	IQBENFirstDayOfWeekType	qbf12.inc
PWIQBENFirstMonthFiscalYearType	IQBENFirstMonthFiscalYearType	qbf12.inc
PWIQBENFirstMonthIncomeTaxYearType	IQBENFirstMonthIncomeTaxYearType	qbf12.inc
PWIQBENGenderType	IQBENGenderType	qbf12.inc
PWIQBENGeneralDetailReportTypeType	IQBENGeneralDetailReportTypeType	qbf12.inc
PWIQBENGeneralSummaryReportTypeType	IQBENGeneralSummaryReportTypeType	qbf12.inc
PWIQBENIncludeAccountsType	IQBENIncludeAccountsType	qbf12.inc
PWIQBENIncludeColumnType	IQBENIncludeColumnType	qbf12.inc
PWIQBENItemTypeFilterType	IQBENItemTypeFilterType	qbf12.inc
PWIQBENJobReportTypeType	IQBENJobReportTypeType	qbf12.inc
PWIQBENJobStatusType	IQBENJobStatusType	qbf12.inc
PWIQBENListDataExtTypeType	IQBENListDataExtTypeType	qbf12.inc
PWIQBENListDelTypeType	IQBENListDelTypeType	qbf12.inc

PWIQBENMatchCriterionType	IQBENMatchCriterionType	qbf12.inc
PWIQBENOperatorType	IQBENOperatorType	qbf12.inc
PWIQBENOtherDataExtTypeType	IQBENOtherDataExtTypeType	qbf12.inc
PWIQBENPaidStatusType	IQBENPaidStatusType	qbf12.inc
PWIQBENPayPeriodType	IQBENPayPeriodType	qbf12.inc
PWIQBENPaySalesTaxType	IQBENPaySalesTaxType	qbf12.inc
PWIQBENRemoveFromObjectType	IQBENRemoveFromObjectType	qbf12.inc
PWIQBENReportAgingAsOfType	IQBENReportAgingAsOfType	qbf12.inc
PWIQBENReportBasisType	IQBENReportBasisType	qbf12.inc
PWIQBENReportCalendarType	IQBENReportCalendarType	qbf12.inc
PWIQBENReportDateMacroType	IQBENReportDateMacroType	qbf12.inc
PWIQBENReportOpenBalanceAsOfType	IQBENReportOpenBalanceAsOfType	qbf12.inc
PWIQBENReturnColumnsType	IQBENReturnColumnsType	qbf12.inc
PWIQBENReturnRowsType	IQBENReturnRowsType	qbf12.inc
PWIQBENrowTypeType	IQBENrowTypeType	qbf11.inc
PWIQBENSummarizeColumnsByType	IQBENSummarizeColumnsByType	qbf12.inc
PWIQBENSummarizeRowsByType	IQBENSummarizeRowsByType	qbf12.inc
PWIQBENSummaryReportBasisType	IQBENSummaryReportBasisType	qbf12.inc
PWIQBENTimeReportTypeType	IQBENTimeReportTypeType	qbf12.inc
PWIQBENTxnDataExtTypeType	IQBENTxnDataExtTypeType	qbf12.inc
PWIQBENTxnDelTypeType	IQBENTxnDelTypeType	qbf11.inc
PWIQBENTxnTypeFilterType	IQBENTxnTypeFilterType	qbf11.inc
PWIQBENTxnTypeType	IQBENTxnTypeType	qbf11.inc
PWIQBENTxnVoidTypeType	IQBENTxnVoidTypeType	qbf11.inc
PWIQBENWageTypeType	IQBENWageTypeType	qbf11.inc
PWIQBGUIDType	IQBGUIDType	qbf11.inc
PWIQBIDType	IQBIDType	qbf11.inc
PWQBIntType	IQBIntType	qbf11.inc
PWQBPercentType	IQBPercentType	qbf11.inc
PWQBPriceType	IQBPriceType	qbf11.inc
PWQBQuanType	IQBQuanType	qbf11.inc
PWQBSessionManager	IQBSessionManager	qbf13.inc
PWQBStringType	IQBStringType	qbf11.inc
PWQBTimeIntervalType	IQBTimeIntervalType	qbf11.inc
PWQuantityAdjustment	IQuantityAdjustment	qbf2.inc
PWIReceivePaymentAdd	IReceivePaymentAdd	qbf2.inc
PWIReceivePaymentQuery	IReceivePaymentQuery	qbf2.inc
PWIReceivePaymentRet	IReceivePaymentRet	qbf2.inc
PWIReceivePaymentRetList	IReceivePaymentRetList	qbf2.inc
PWIReceivePaymentToDepositRet	IReceivePaymentToDepositRet	qbf2.inc
PWIReceivePaymentToDepositRetList	IReceivePaymentToDepositRetList	qbf2.inc
PWIRefNumberFilter	IRefNumberFilter	qbf2.inc
PWIRefNumberRangeFilter	IRefNumberRangeFilter	qbf2.inc
PWIReportAccountFilter	IReportAccountFilter	qbf2.inc
PWIReportData	IReportData	qbf2.inc
PWIReportEntityFilter	IReportEntityFilter	qbf2.inc

QBFUSE GUIDE

PWIReportItemFilter	IReportItemFilter	qbf2.inc
PWIReportPeriod	IReportPeriod	qbf2.inc
PWIReportRet	IReportRet	qbf2.inc
PWIReportsPreferences	IReportsPreferences	qbf2.inc
PWIReportTxnTypeFilter	IReportTxnTypeFilter	qbf2.inc
PWIResponse	IResponse	qbf13.inc
PWIResponseList	IResponseList	qbf13.inc
PWIResponseType	IResponseType	qbf13.inc
PWIRowData	IRowData	qbf2.inc
PWISalesAndCustomersPreferences	ISalesAndCustomersPreferences	qbf2.inc
PWISalesAndPurchase	ISalesAndPurchase	qbf2.inc
PWISalesAndPurchaseMod	ISalesAndPurchaseMod	qbf2.inc
PWISalesOrPurchase	ISalesOrPurchase	qbf2.inc
PWISalesOrPurchaseMod	ISalesOrPurchaseMod	qbf2.inc
PWISalesReceiptAdd	ISalesReceiptAdd	qbf2.inc
PWISalesReceiptLineAdd	ISalesReceiptLineAdd	qbf2.inc
PWISalesReceiptLineGroupAdd	ISalesReceiptLineGroupAdd	qbf2.inc
PWISalesReceiptLineGroupRet	ISalesReceiptLineGroupRet	qbf1.inc
PWISalesReceiptLineRet	ISalesReceiptLineRet	qbf1.inc
PWISalesReceiptLineRetList	ISalesReceiptLineRetList	qbf1.inc
PWISalesReceiptQuery	ISalesReceiptQuery	qbf1.inc
PWISalesReceiptRet	ISalesReceiptRet	qbf1.inc
PWISalesReceiptRetList	ISalesReceiptRetList	qbf1.inc
PWISalesRepAdd	ISalesRepAdd	qbf1.inc
PWISalesRepQuery	ISalesRepQuery	qbf1.inc
PWISalesRepRet	ISalesRepRet	qbf1.inc
PWISalesRepRetList	ISalesRepRetList	qbf1.inc
PWISalesTaxCodeAdd	ISalesTaxCodeAdd	qbf1.inc
PWISalesTaxCodeQuery	ISalesTaxCodeQuery	qbf1.inc
PWISalesTaxCodeRet	ISalesTaxCodeRet	qbf1.inc
PWISalesTaxCodeRetList	ISalesTaxCodeRetList	qbf1.inc
PWISalesTaxPaymentCheckLineRet	ISalesTaxPaymentCheckLineRet	qbf1.inc
PWISalesTaxPaymentCheckLineRetList	ISalesTaxPaymentCheckLineRetList	qbf1.inc
PWISalesTaxPaymentCheckQuery	ISalesTaxPaymentCheckQuery	qbf1.inc
PWISalesTaxPaymentCheckRet	ISalesTaxPaymentCheckRet	qbf1.inc
PWISalesTaxPaymentCheckRetList	ISalesTaxPaymentCheckRetList	qbf1.inc
PWISalesTaxPreferences	ISalesTaxPreferences	qbf1.inc
PWISetCredit	ISetCredit	qbf1.inc
PWISetCreditList	ISetCreditList	qbf1.inc
PWIShipMethodAdd	IShipMethodAdd	qbf1.inc
PWIShipMethodQuery	IShipMethodQuery	qbf1.inc
PWIShipMethodRet	IShipMethodRet	qbf1.inc
PWIShipMethodRetList	IShipMethodRetList	qbf1.inc
PWISickHours	ISickHours	qbf1.inc
PWISandardTermsAdd	IStandardTermsAdd	qbf1.inc
PWISandardTermsQuery	IStandardTermsQuery	qbf1.inc

QBFUSE GUIDE

PWISStandardTermsRet	IStandardTermsRet	qbf1.inc
PWISStandardTermsRetList	IStandardTermsRetList	qbf1.inc
PWISubtotalRow	ISubtotalRow	qbf1.inc
PWITermsQuery	ITermsQuery	qbf1.inc
PWITextRow	ITextRow	qbf1.inc
PWITimeReportQuery	ITimeReportQuery	qbf1.inc
PWITimeTrackingAdd	ITimeTrackingAdd	qbf1.inc
PWITimeTrackingEntityFilter	ITimeTrackingEntityFilter	qbf1.inc
PWITimeTrackingPreferences	ITimeTrackingPreferences	qbf1.inc
PWITimeTrackingQuery	ITimeTrackingQuery	qbf1.inc
PWITimeTrackingRet	ITimeTrackingRet	qbf1.inc
PWITimeTrackingRetList	ITimeTrackingRetList	qbf.inc
PWITimeTrackingTxnFilter	ITimeTrackingTxnFilter	qbf.inc
PWIToDoAdd	IToDoAdd	qbf.inc
PWIToDoListFilter	IToDoListFilter	qbf.inc
PWIToDoQuery	IToDoQuery	qbf.inc
PWIToDoRet	IToDoRet	qbf.inc
PWIToDoRetList	IToDoRetList	qbf.inc
PWITotalBalanceFilter	ITotalBalanceFilter	qbf.inc
PWITotalRow	ITotalRow	qbf.inc
PWITxnDataExt	ITxnDataExt	qbf.inc
PWITxnDateFilter	ITxnDateFilter	qbf.inc
PWITxnDateRangeFilter	ITxnDateRangeFilter	qbf.inc
PWITxnDel	ITxnDel	qbf.inc
PWITxnDeletedQuery	ITxnDeletedQuery	qbf.inc
PWITxnDeletedRetList	ITxnDeletedRetList	qbf.inc
PWITxnDelRet	ITxnDelRet	qbf.inc
PWITxnFilter	ITxnFilter	qbf.inc
PWITxnFilterWithItemFilter	ITxnFilterWithItemFilter	qbf.inc
PWITxnLineDetail	ITxnLineDetail	qbf.inc
PWITxnLineDetailList	ITxnLineDetailList	qbf.inc
PWITxnVoid	ITxnVoid	qbf.inc
PWITxnVoidRet	ITxnVoidRet	qbf.inc
PWIVacationHours	IVacationHours	qbf.inc
PWIValueAdjustment	IValueAdjustment	qbf.inc
PWIVendorAdd	IVendorAdd	qbf.inc
PWIVendorCreditAdd	IVendorCreditAdd	qbf.inc
PWIVendorCreditQuery	IVendorCreditQuery	qbf.inc
PWIVendorCreditRet	IVendorCreditRet	qbf.inc
PWIVendorCreditRetList	IVendorCreditRetList	qbf.inc
PWIVendorListFilter	IVendorListFilter	qbf.inc
PWIVendorMod	IVendorMod	qbf.inc
PWIVendorQuery	IVendorQuery	qbf.inc
PWIVendorRet	IVendorRet	qbf.inc
PWIVendorRetList	IVendorRetList	qbf.inc
PWIVendorTypeAdd	IVendorTypeAdd	qbf.inc

PWIVendorTypeQuery	IVendorTypeQuery	qbf.inc
PWIVendorTypeRet	IVendorTypeRet	qbf.inc
PWIVendorTypeRetList	IVendorTypeRetList	qbf.inc

Example Applications

qbFUSE ships with two demo applications, both of which can be downloaded in a single .zip file for free (including complete source code!) using the following link:

<http://www.thinkdata.com/asppwadmin/stattrack/includes/dltrack.asp?Title=QBFDemo&File=qbf.zip>

NOTE: QuickBooks must be open and running to use the demo applications and any other QBFC enabled application.

The first example application which ships with qbFUSE is qbfuse.exe. It is a Clarion source project and you can view the source in its entirety by looking at the qbfuse.clw file (most likely installed in your \C55\Examples\qbFUSE directory). It contains a conversion of all of the QBFC SDK examples included with the installation of the QuickBooks SDK. The examples from Intuit are included in both Visual Basic and Visual C++ so this example is an excellent reference to study when converting COM code from Visual Basic or Visual C++ into Clarion using the Plugware COM Classes. The documentation does not include the source code for this example due to its size. This demo implements the following QuickBooks functionality:

- Adding customers to QuickBooks
- Adding a Bill to QuickBooks for a particular Vendor
- Querying Invoices in QuickBooks using a date range
- Querying Items and setting the number of Items to return in an Item Query
- Adding a ReceivePayment transaction and deleting it if successful
- Sending multiple requests to Add Customer, Add Invoice, and Query Invoice all in the same message set

The second example application which ships with qbFUSE is called qbfapp.exe. It is a Clarion application and you can view the source in its entirety by looking at the qbfapp.app file (most likely installed in your \C55\Examples\qbFUSE directory). This example demonstrates importing and exporting data from QuickBooks into Clarion TPS data files and even directly into Microsoft Outlook using our OutlookFUSE product.

- Importing QuickBooks Accounts data into a Clarion TPS file
- Importing QuickBooks Customers data into a Clarion TPS file
- Importing QuickBooks Items data into a Clarion TPS file
- Exporting Outlook Contacts from Microsoft Outlook to QuickBooks Customers using the OutlookFUSE product from ThinkData
- Importing Outlook Contacts from QuickBooks Customers using the OutlookFUSE product from ThinkData

Links

We have provided several links which we found helpful in the development of the qbFUSE product and in its daily use. If you find helpful resources in your development process, or have questions for us related to qbFUSE, we encourage you to submit them to our support forum listed below.

ThinkData Support Forum

<http://www.thinkdata.com/forum/>

Intuit Developer Network Downloads (get the SDK here!)

<http://developer.intuit.com/downloads/>

Intuit Developer FAQ

<http://developer.intuit.com/quickbooks/faq.asp>

Summary

The examples we have shown only scratch the surface of the potential of the qbFUSE product. There are 543 interfaces defined in the qbFUSE Clarion source files and the entire wrapper contains upwards of 25,000 lines of code. qbFUSE is the only product of its kind for Clarion 5.5 and Clarion 6 providing native early binding to QuickBooks with a stable and efficient COM layer from Plugware Solutions.

qbFUSE wraps the QBFC (QuickBooks Foundation Classes) SDK which in turn wraps the qbXML specification from Intuit. Because of this, you can be assured that qbFUSE puts you on the cutting edge of software development for the QuickBooks platform.

We hope you enjoy qbFUSE and invite you to join [ThinkData's support forum](#) to ask questions and get tips on using it to its fullest potential. Thanks for your interest in qbFUSE and keep your eyes open for other COM automation products from our FUSE product line.