



QuickBooks

White Paper

Abstract

This paper describes the use of QuickBooks and QuickBooks SDK

Tometa creates custom software for you

Tometa Software designs and develops robust software solutions for virtually all industries including in-house (vertical market) and retail software, some of which is on the shelves at your local software store. We focus our unique combination of creative, technical, and problem-solving skills on meeting our client's objectives. Because of our clarity of purpose, commitment to process, and broad professional skill sets, we are able to provide our clients with world-class solutions that are functionally superior and fully aligned with our client's strategic focus.

Balancing development speed, quality and cost is what we are all about. Tometa combines agile development practices with fixed pricing, so you know what the cost, end product, and delivery time table look like—up front. If we underestimate the effort, we complete the overrun on our dime. Simple as that. That's why large enterprise firms like Alcoa and NASDAQ choose Tometa.

Tometa's agile development expertise and low-overhead US location keep our prices comparable to offshore vendors – without offshore challenges. Using a fixed pricing model, we provide upfront visibility into a project's ultimate costs, end product and delivery schedule. Our clients like knowing that we have “skin in the game” – a fixed price that aligns our goals with yours, incenting us to get the job done right and fast.

Lastly, as a Microsoft Certified Gold Partner, Tometa Software, can customize its products or create custom web, client/server, and traditional applications. With programming experience in C#, C++, Visual Basic, CGI, HTML, RPG, Delphi, Java and many others; Tometa Software is uniquely positioned to meet your needs as a development firm.

[Check us out today](#)

QuickBooks by Intuit is a common-place tool used by small businesses to help conduct business transactions and accounting. It can also be used to conduct point-of-sale transactions and keep track of inventory and customers. It can automate invoices and provide administrative functions in running a business. It has sold millions of copies so a large user-base exists for it. With this many people relying on a program for such a large part of their business, it is certainly a solution that will be around for a long time.

It's a pretty lofty goal, however, to provide one program that will fulfill the accounting and administrative needs of all the small businesses that exist. Every small business operates differently and many small businesses won't benefit from a program that isn't capable of embracing that diversity. The part of business that QuickBooks tries to automate is a part of business many small businesses have spent a lot of time already trying to make as easy as possible. If QuickBooks can't work with the solutions they already have, then it can't offer its full benefit to the business.

Intuit has provided an SDK for QuickBooks for this reason and several others. With the SDK QuickBooks offers, developers can create customized solutions to match the needs of the business owner while keeping development costs down and utilizing the power of QuickBooks. Basic programming skills and familiarity with Microsoft's COM design method is all a developer needs to create powerful solutions tailored to specific needs. The SDK can be used with any programming language that supports OLE.

The SDK comes with an affordable price; it's free. While it has no initial price, it still provides great flexibility and ease of development making the market for QuickBooks solutions one that is hard to ignore. It's quite easy to envision a paperless office with many administrative functions automated by a system that has QuickBooks as its keystone.

Why use QuickBooks and an SDK?

Small businesses have tried to find a solution to accounting and administrative problems for as long as businesses have been going active. Many businesses have already put into place systems to make accounting easier and the time spent in development shouldn't just be ignored. Switching to a QuickBooks based system may seem like an extra headache that no one wants to deal with. A common concern may be that between the cost of installation and configuration, then the cost of training the employees to learn the new system makes switching to QuickBooks seemingly more trouble than its worth.

Simply put, the initial cost of a QuickBooks solution will save money in the long run. QuickBooks allows data to be entered one time and then be replicated from that one entry everywhere. If a business changes its address, you change the address once and it is changed everywhere within the system. If a price of a product changes, it's reflected on all open invoices. Once the information is updated, no detail (no matter how small) will slip through the cracks, which means no more need for strenuous double-checking. Situations like these are where QuickBooks makes up its initial development and installation costs.

The SDK provides even more flexibility to the hesitant yet prospective user. The SDK allows for a developer to quickly create a custom solution that matches the way business is already being done. The developer can create a solution that is seamless for the employees of a business who are not involved in accounting or administration. The developer can also create custom solutions that a stock installation of QuickBooks might not be capable of. This creates a scenario where QuickBooks can be implemented with a minimal amount of adaptation from the business. This is the advantage the SDK provides. It allows QuickBooks to be integrated into any solutions that are already in place and allows for innovation from the business using it. If you can dream a solution, the SDK allows it to be created.

Programming, Development and Deployment

The QuickBooks SDK can use any development language that supports OLE. Most modern development languages in use for Windows do support OLE. Even without an OLE compliant system, plug-ins written using the QuickBooks SDK can synchronize the data contained in QuickBooks to a database, allowing any system and development environment to work with QuickBooks.

The QuickBooks SDK uses COM to communicate between the solutions being created and QuickBooks. This is typical of most SDKs. They contain a group of functions and libraries containing a lot of the functionality that will prospectively be used in bridging two systems.

The information can also be synchronized with most databases for ODBC and ADO models of data access, allowing for further flexibility in solution development. This requires functionality the SDK does not natively provide. The synchronization is created by a plug-in that is readily available to developers. If a database isn't necessary, but ODBC functionality is required, a third-party API allows QuickBooks to be accessed as an ODBC database.

QuickBooks uses a server/client model for communications. The server/client model provides orthogonal persistence for QuickBooks while allowing communications with the product developed with the SDK. QuickBooks behaves as a server, sending and receiving information with the client program developed using the SDK.

The QuickBooks SDK communicates between the Server and the Client using messages. These messages can be sent using the qbXML request processor or communicated with the QBFC API that provides automatic parsing of qbXML using COM objects. The XML messages allow the QuickBooks's SDK to be used with web applications using technologies like XMLHTTP. The two methods of sending messages use two different API's, both of which are included in the SDK.

There are two choices of API to use. The first choice, the qbXML interface, provides a development method that can be used across platforms. The second choice, the QBFC, uses Windows COM and MFC to pass the messages and therefore can only be used on the windows platform.

The messages form the building blocks for requests and responses which are sent between the Server and the Client. The request object contains operation and identification codes for an object or contains the object itself. The response object contains data and a return status code to communicate the results of the requested operation.

This builds the foundation for the transfer of data that the QuickBooks's SDK facilitates. A skilled developer can quickly learn this simple model and then have no problem developing applications for use with QuickBooks.

Examples of applications developed with QuickBooks SDK

Applications developed using the QuickBooks SDK can have any purpose imaginable, but most of them are going to be fairly similar. QuickBooks, in its most basic use, keeps track of how much money is received and spent. Although it is more complex than just money tracking, it typically stays within that scope. The SDK allows that scope to be expanded to purposes relative to administrative functions of business. These are usually the functions that will have the most value from being integrated with QuickBooks.

Probably the most common integration that makes sense to be used with QuickBooks is a utility that allows viewing, analysis and manipulation of data in a way that is customized to specific business needs. This can be done by writing an application, which would provide the most flexibility and be the most elegant solution. It can also be done through synchronization with a database. These solutions allow for automation of financial statements, more effective research and the visualization of the strengths and weaknesses of a business.

Another function that QuickBooks applications are commonly used for is taxes. Several options already exist to help automate filling out tax forms and providing information for various other tax functions. Often this is included into the functionality of other programs. Point-of-Sale programs will include sales tax functionality. Payroll programs can automate W2 information. There are several programs that exist that can export an entire business's tax forms from the information in QuickBooks with minimal user input.

QuickBooks programs can also offer expanded functionality for keeping track of inventories or employees. A solution can be designed specifically for the way your business works, I could keep track of how many hours employees work, what they do while their working, what materials they use, and how much revenue they generate. Another solution can be designed independently to keep track of your inventory and automatically order things when supply runs low or show you statistics related to any item in your inventory.

Many businesses have specialized needs that are common from one company to the next. There are generalized solutions that are available on the commercial market already existing for several kinds of businesses. Most major niches of business have an extension meant for them. There are extensions designed specifically for law firms, construction management companies, freelance consultants and even specialty niches like print houses and coffee shops.

Finally, since the QuickBooks SDK is easy to learn, new solutions are being developed all the time. It isn't out-of-the-ordinary for a business to contract out design of a new solution for integration with QuickBooks that will meet their needs.

Conclusion

The QuickBooks SDK is an essential part of Intuit's product. This SDK is the keystone in providing flexibility to a product that can provide an amazing amount of organization and automation right out of the box. With the proper extensions and specialized development, QuickBooks is a gigantic step towards a paperless office and will provide automation that will return whatever costs are involved in the development of the solution used.

This document is for informational purposes only. Tometa Software, Inc. MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.

© 2004 Tometa Software, Inc. All rights reserved.