Intuit Inc.

Introducing qbXML™

Enabling Software Developers to Harness the Power of Data

Introducing qbXML™

Enabling Software Developers to Harness the Power of Data

This document introduces Intuit's QuickBooks eXtensible Markup Language, or qbXML™. It is the language at the core of a new framework that allows electronic exchange, creation and management of accounting and other business data.

Please note that the current version, qbXML v0.6, is in draft form and cannot be used for any kind of implementation. Also note that qbXML will be supported for the first time by the next version of QuickBooks®.

An Enabling Technology

Unleashing the Power of Data

QuickBooks is the financial management system of choice for small and medium-sized businesses, who depend on it to manage business-critical data such as accounts receivable, accounts payable, customer, vendor and employee databases, and expense and time tracking.

With qbXML, software developers will now be empowered to build specialized vertical applications and horizontal productivity applications that mine, enrich and share this data. Businesses stand to benefit by gaining maximum leverage of their most critical data.

Never Enter Data Twice (NED2™) – qbXML maximizes productivity and efficiency for businesses by making data universally available to all applications.

Elevating Small Businesses

Small businesses today trail behind in the electronic revolution. Seamless connectivity, the Web and a myriad of powerful software solutions are taken for granted by bigger, more economically able companies. Small businesses demand simple and cost-effective solutions. Unhampered innovation is required to elevate the millions of small businesses to the ranks of their enterprise counterparts. Innovation in the large community of independent software developers is encouraged and enabled by qbXML, helping them build solutions that meet the needs of the small business community.

Seamless Communications – With qbXML, independent developers will now be able to employ a standards-based framework to give small businesses the benefits of streamlined communication with employees, customers, partners, vendors and accountants.

World-Class Technology Framework

Intuit adhered to design principles for qbXML to provide developers and businesses with a powerful framework that is:

- **Standard.** The framework is based on standard, open technologies and languages such as XML and HTTP.
- **Extensible**. The framework can easily be extended over time as new technologies and standards emerge.
- **Secure.** Security is built into the framework to protect business transactions.
- **Robust.** The framework provides a strong error-recovery mechanism specially designed to handle a business's financial transactions.
- **Desktop- and Web-Ready.** The qbXML framework is designed to support Web-based, remote data integration as well as local data integration on the desktop. This means that any kind of application can use qbXML: Web, wireless and desktop.
- **OFX-Like.** Intuit co-authored and supports the Open Financial Exchange (OFX) industry standard. OFX is a unified specification for the electronic exchange of financial data between financial institutions, businesses and consumers via the Internet. For consistency, the qbXML specification has a style very similar to that of OFX. For more information about OFX, please visit http://www.ofx.net.

Business Data

qbXML v0.6 is defined by two Document Type Definition (DTD) files. These text files specify the syntax used to build qbXML requests. The following data objects and the operations supported by them are described in detail by qbXML v0.6. Additional objects and functionality will be added in future versions.

- Account
- Customer
- Employee
- Vendor
- Job
- Terms
- Service Item
- Inventory Item
- Non-Inventory Item
- Other Charge Item
- Discount
- Payment
- Sales Tax
- Invoice
- Receive Payment
- Estimate
- Sales Receipt
- Credit Memo
- Purchase Order
- Check
- Credit Card
- Vendor Credit
- Bill
- Journal Entry

Application Types

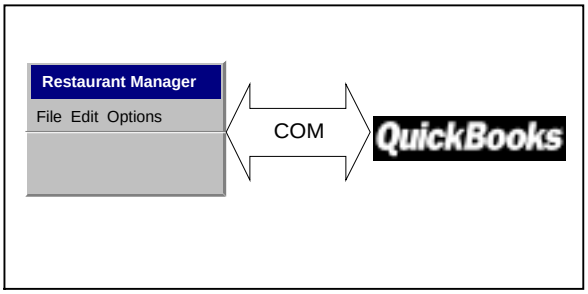
Virtually any application can use qbXML. Desktop applications can use it to interact directly with QuickBooks. Web applications can use framework services to interact with QuickBooks.

qbXML allows multiple requests to be batched together in one qbXML transaction. This option enhances the efficiency and performance of applications and suits certain batch-oriented applications well. The qbXML developer sets an option flag to determine how processing errors in batch requests should be handled. With one option, the batch processing is halted and the appropriate data and status information is returned in the response. With the other option, processing of the remaining requests continues and the appropriate data and status information is returned.

Desktop Applications

A desktop application can interact with QuickBooks directly using qbXML. It must first build a qbXML request, or batch of requests, adhering to the syntax defined in the qbXML DTDs. Next, it must pass the qbXML request to QuickBooks by means of a Microsoft Component Object Model, or COM, interface supplied by the framework. This COM interface is a very simple and light interface that only requires three parameters: business designator, qbXML request (input) and qbXML response (output).

1) Business designator is necessary to identify the correct QuickBooks company database.
2) The qbXML request may be a memory buffer containing the qbXML-formatted string or the name of a file that contains the string.
3) The response from QuickBooks is returned in the third parameter, which also may be a buffer or the name of a file.



Desktop-to-Desktop Integration

Many kinds of desktop applications can make use of qbXML to integrate their data with QuickBooks. For the purpose of illustration, let's see what a restaurant management package could do. This vertical solution can use QuickBooks as its accounting engine. If the restaurant manager already uses QuickBooks to manage business finances, initial set-up of the new restaurant management solution is greatly simplified and expedited because it can seamlessly pull the data it needs from QuickBooks. This data can include the database of employees, vendors, company information and so forth.

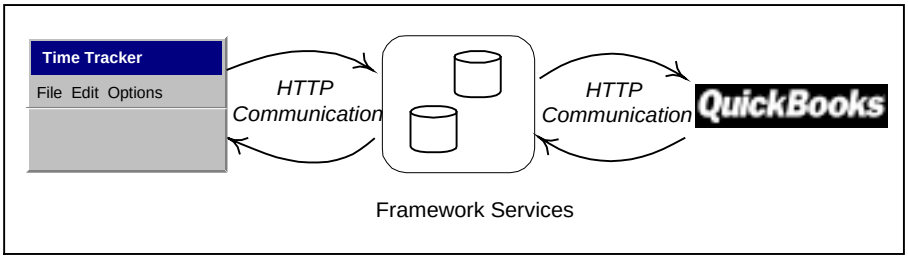
Furthermore, the following activities will generate data that needs to be managed and tracked by QuickBooks:

- Receive patron payment by credit card, check or cash
- Make payment to meat supplier
- Place order with vegetable supplier
- Purchase new oven
- Hire new waiter
- Change flower vendor
- Record Chef check-in and check-out times

Clearly, this is a very short selection from a long list of daily business management activities. One can easily imagine a sophisticated software package tailored specifically to the management of restaurants. With QuickBooks as the financial management package for such a business, one can further imagine a complete, seamlessly integrated solution for the restaurant manager.

Web Applications

Web applications build qbXML requests the same way desktop applications do. These applications can then take advantage of the framework's Web-based intermediary services and its support of HTTP. Requests are processed immediately on the server and the response returned to the application. Subsequently, the QuickBooks user synchronizes QuickBooks records with data on the server.



Web-to-Desktop Integration

An example of this kind of application is one that allows remote entry of timesheet information. One can imagine a business whose employees often work remotely. The Time Tracker application would enable them to use a browser or PDA to enter their timesheet data. The business owner or payroll administrator/bookkeeper could then download the timesheet data into QuickBooks.

Another example is that of a Web store. Data related to Web store transactions such as received payments, inventory items and customer information can now be easily synchronized with QuickBooks.

Give Us Your Feedback

We encourage you to give us your feedback on qbXML. The DTDs and a feedback form are available at http://developer.intuit.com. Your comments and suggestions will be invaluable as we strive to meet your needs and as we continue to evolve the qbXML specification toward a production version 1.0.