

Networking & Forums

[→ Log In](#)

Overview

[Developer Forums](#)

[Development Tools](#)

Members Write

A ProAdvisor Sheds Light on the QuickBooks® Software Development Kit (SDK)

By Nancy Smyth , QuickBooks ProAdvisor and Intuit® Developer Network member

Introduction

I first learned about the QuickBooks Software Development Kit (SDK) through an Intuit press release announcing both the SDK and a training seminar on how it worked, which my husband and I attended. Our company, Sunburst Software Solutions, had already built our Certified Payroll Solution product, which solved one of my biggest double-entry nightmares as a ProAdvisor/Bookkeeper. At that time, our product interfaced with QuickBooks using another means of data extraction. We could tell that this would be a much less stressful, as well as faster, reliable, and "approved" means of getting the information we needed out of QuickBooks to have for our program to function.

I wrote this article initially to add to the whitepapers about our own products to alleviate our customers' fears and misconceptions about third-party products and what they could/would do to existing QuickBooks data. After hearing these same key concerns from other ProAdvisors in phone calls, surveys, and at face-to-face meetings, it became obvious to me that many ProAdvisors wanted to learn how the SDK worked, and wanted reassurance that third party applications would not corrupt their clients' existing data files, without having to learn all the ins and outs of writing program code.

Why a QuickBooks SDK?

The QuickBooks SDK is a collection of tools, reference materials, and sample program code that Intuit distributes to application developers, system integrators, software product managers, consultants, and other technical professionals interested in developing applications that share data with QuickBooks.

Intuit developed the QuickBooks SDK because of customer demand. QuickBooks customers were asking how they could avoid entering the same data multiple times into multiple places: QuickBooks, and other, "third-party," business applications they were using. They wanted to be able to transfer data in to, and out of, QuickBooks to save time and avoid errors.

More About Customers Need

Often times, QuickBooks data—such as accounts receivable, accounts payable, customer lists, vendor lists, employee lists, and expense and time tracking information—must be managed in a way that is unique to a specific industry, therefore requiring the use of additional applications to supplement QuickBooks. For example, a retail store uses a point-of-sale application in the store, and QuickBooks in the back office. In other cases, the third-party application enters and stores the QuickBooks data differently, for example, on a per-client or per-location basis. In both of

Benefits of Membership

- Download the SDK
- Forums
- Marketing
- Support

What's Hot



Developer Spotlight
IDN member Ed Hofer works with QuickBooks Online Edition



Get the latest and greatest!
Download here

Other Opportunities



QuickBase Developer Opportunities

QuickBase offers developers an easy, standards-based Web application development environment. [Learn more.](#)



Master Builder Developer Opportunities

Master Builder software offers developers a construction management platform for building solutions that put contractor in control of all aspect of their business. [Learn more.](#)

these examples, large quantities of important data need to be entered twice, resulting in extra work and potential errors, both of which cost time and money.

Third-party applications created with the SDK can share QuickBooks data. By using these “QuickBooks-integrated” applications, small-business owners no longer need to enter the same data twice, and in fact now can have the best of both worlds—the power, ease, and comprehensiveness of QuickBooks, combined with the benefits of an application tailored to the unique needs of a particular small-business concern.

More About the SDK

The SDK provides a common methodology for integrating an application with QuickBooks: once a developer creates an application for one QuickBooks product, modifying that application to support any (or all) of the other SDK-supported QuickBooks products is a straightforward task. This shared SDK approach is based on qbXML, which is a version of XML (eXtensible Markup Language, a computer code) that has been designed specifically for QuickBooks.

Intuit designed the QuickBooks SDK with the following principles in mind:

- **Keep the small-business owner in control.**
The business owner or the administrator for the business **must authorize** the connection between the third-party application and QuickBooks. They also set up **permissions** for authorized users and access rights for each third-party application through the QuickBooks, Edit Menu, Preferences, Integrated Applications, and Company Preferences section. The SDK supports the user privilege structure as established by QuickBooks, wherein, for example, certain users can access only certain types of information.
- **Provide robust mechanisms to protect QuickBooks data.**
The SDK provides strong error recovery, data logging, and synchronization facilities to ensure that data is not lost or destroyed and that third-party application data remains synchronized with QuickBooks data. Additionally, the Inuit SDK enforces all the QuickBooks business rules no matter what the third-party application does.

The use of the **qbXML Validator utility** allows a developer to verify that a given qbXML document conforms to the qbXML specifications before it is sent to QuickBooks.

The **SDKTest utility** is used to test the request/response cycle, which accepts a qbXML request, sends it to QuickBooks, and returns the response to the 3rd party application.

The qbXML specifications are built into QuickBooks, and the SDK provides methods for querying which version of the qbXML specification is supported by the version of QuickBooks a small business owner is using. I.e., if a developer writes “smart” code that checks the version and responds accordingly, a third-party application can run with multiple versions of QuickBooks.

- **Use existing programming standards.**
The QuickBooks SDK uses standard interfaces and XML formats, and as a result is compatible with most software programming languages.
- **Adhere to a single specification for the entire SDK.**
The QuickBooks products, except for QuickBooks Online Edition, all use the same qbXML specification. Therefore, the content and behavior of the request and response messages between QuickBooks and the third-party application is consistent across all products.

The SDK allows applications to:

- **“Query,”** or read, most (but not all) information contained in the QuickBooks file
- **Add** some but not all types of information—for example, a Customer or a Job can be added to QuickBooks by a third-party application, but an Item or a Non-wage payroll item cannot
- **Modify** some (but not all) types of information—for example, an existing customer can be modified, but an Item cannot.
- **Delete** some but not all types of information—for example, a request for an Account to be deleted can be made, but an Item may not be deleted
- **Void** some but not all types of information—for example, a Bill may be voided, but an Estimate may not.

Conclusion

As a ProAdvisor, I understand how important it is to be sure that data remains intact and undamaged when using any third-party product. I hope that this article explaining the principals behind the SDK and its built-in mechanisms to protect that data has been useful.

Next: [Back to the Networking and ForumsOverview page](#)

[Privacy](#)

[Legal](#)

[Calendar](#)

[Membership Required:](#)

