

QXpress SDK

Developer's Guide

Version 1.0

Alocet Inc.
55 Eglinton Ave. East, Suite 804
Toronto, ON M4P 1G8
Canada

www.alocet.com
Toll Free: 1.877.529.6659

CHAPTER 1

Introduction

The purpose of this document is to offer you an introduction into the QXSDK, so that you can get started on your way to building powerful applications that interact with QXpress.

The topics discussed in this section include:

- Why an SDK for QXpress?
- What are the runtime requirements?
- What is included in the SDK?
- What should you know before you begin?
- How do you register your application with Alocet Inc.?
- Who do you contact for support or questions?

Why an SDK for QXpress?

Avoid double-entry

- QXpress users often use a variety of software programs to automate their company. They use QuickBooks for their accounting, and QXpress for scheduling, routing, job-costing and batch invoicing. They may also use some type of design / CAD software, estimating software or PDA software. If these software programs do not directly communicate with QXpress, the user may run into double-entry which can lead to increased data entry time, and the possibilities of costly mistakes.

Extend the power of QXpress

- Alocet Inc., the makers of QXpress, simply cannot create all the potential software programs QXpress users need run their businesses. With the SDK, however, QXpress users can get the best of all worlds, as each software vendor can concentrate on the area they specialize in, and a QXpress user benefits from the absence of double-entry. For example, imagine a QXpress user who has a web designer create a web site for them. The QXpress user has very specific demands on the functions of the web site. For instance, the QXpress user wants their customers to be able to log

into the web site, see their schedule and history, as well as schedule their own service calls. Using this SDK, the web designer can program the web site to interact with QXpress precisely to the specifications of the QXpress user.

- Some of the possibilities include: integrating QXpress data with web sites; converting estimates from Microsoft Excel into QXpress services, advanced analysis of QXpress data to be used in marketing modules; integration with existing estimating software; integration with GPS devices, etc.

What are the runtime requirements?

QX_SDK.dll

- Your application will be interacting directly with the file QX_SDK.dll, which is the QXMLRequestProcessor. This dll is installed automatically with QXpress version 5.0 or higher.

QXpress installation

- QXpress is not *required* to be installed on the computer, however, all of the dependency files required for the QX_SDK.dll to operate are installed with QXpress. Therefore, it is *recommended* that QXpress is installed on the computer that you are using to interact with QX_SDK.dll.

QXpress database

- QXpress does not need to be open on the computer while you are interacting with QX_SDK.dll.
- If QXpress is open, you can specify to interact with a different QXpress database than the one open.

QXpress version

- Only QXpress version 5.0 and higher is compatible with the QX_SDK.dll.

What is included in the SDK?

Developers Guide (this document)

- This guide is your starting point to get you familiar with the SDK. You should read this guide from start to finish before moving onto other resources.
- To access this guide online, please visit:
<http://www.alocet.net/qxsdk/developersguide.pdf>

Onscreen Reference Guide

- The Onscreen Reference Guide is an excellent resource. It details every request you can make to the QX_SDK.dll, and explains each field, field type and attribute.
- To access the Onscreen Reference Guide, please visit:
<http://www.alocet.net/qxsdk/onscreenref/index.html>
- To extract all files of the Onscreen Reference Guide locally onto your computer, unzip the following self-extracting zip file:
<http://www.alocet.net/qxsdk/onscreenref/onscreenref.exe>

Sample Applications in Visual Basic

- Various sample applications exist, written in Visual Basic 6.
- To download these applications, please visit:
<http://www.alocet.com/qxsdk.cfm>

Developer Forum

- The developers forum is a great place to ask questions, and learn from answers given to other developers.
- To visit the developers forum, please follow the link on the site:
<http://www.alocet.com/qxsdk.cfm>

What should you know before you begin?

XML

- QX_SDK.dll only receives requests, and provides responses, in XML. It will, therefore, be an advantage to have at least an intermediate understanding of XML. If you are only a beginner in XML coding, however, you can likely survive by simply learning from the code examples in the visual basic sample applications.

QuickBooks

- Since QXpress is a QuickBooks add-on, many of the requests made to QX_SDK.dll will interact directly with QuickBooks. It is assumed that you have at least a basic understanding of QuickBooks.

QXpress

- It is assumed that you at least have an intermediate understanding of QXpress. This will enable you to understand what the requests you make to QX_SDK.dll are attempting to accomplish.
- If you do not have an intermediate level of understanding, it recommended that you visit <http://www.qxpress.com>, and watch the movie tutorials specific to the industries you are interested in creating your application for. You can also download a free trial version of QXpress from this site.

Programming Languages

- Although the sample applications are written in Visual Basic, you may choose any programming language capable of referencing dll files. This SDK assumes that you are an experienced programmer in your preferred programming language.

How do you register your application with Alocet Inc.?

ApplicationName and ApplicationAccessPassword attributes

- **ApplicationName:** In every request you make to QX_SDK.dll, you must specify the ApplicationName attribute (*for more information on required attributes, please visit the Onscreen Reference guide*). The ApplicationName attribute is a descriptive name of your application. QXpress users will see this name when you try to connect to their QXpress database, and will either accept or deny access to your application.
 - **ApplicationAccessPassword:** Another attribute you must pass to QX_SDK.dll is your ApplicationAccessPassword. Based on your ApplicationName, you will have a unique ApplicationAccessPassword assigned to you. You must apply to Alocet Inc. for your ApplicationAccessPassword by emailing developer@alocet.com for further instructions. The process is fast and painless, it just serves as an extra level of protection to QXpress users to prevent applications with misleading ApplicationNames from accessing their data.
-

Who do you contact for support or questions?

QXpress related questions

- All questions related directly to QXpress must go through regular support channels. To access these channels, please follow the Support link on the <http://www.qxpress.com> web site.
- Free, non-self serve support is not available to developers on questions related to QXpress. If you require assisted support you must either call on behalf of your client (with their permission), or you must be a registered QXpress user.

SDK related questions

- If your questions relate directly to the SDK, and your answers cannot be found in the Onscreen Reference Guide, sample applications, or this Developers Guide document, please direct your questions to the Developers Forum at <http://www.alocet.com/qxsdk.cfm>. Only questions pertaining directly to the SDK will be answered on the Developers Forum.
- Please do not ask questions pertaining to the SDK in regular support channels (e-Support, phone support, users forums). QXpress Support staff will direct you to ask your questions on the Developers Forum.

CHAPTER 2

Communicating with QXpress

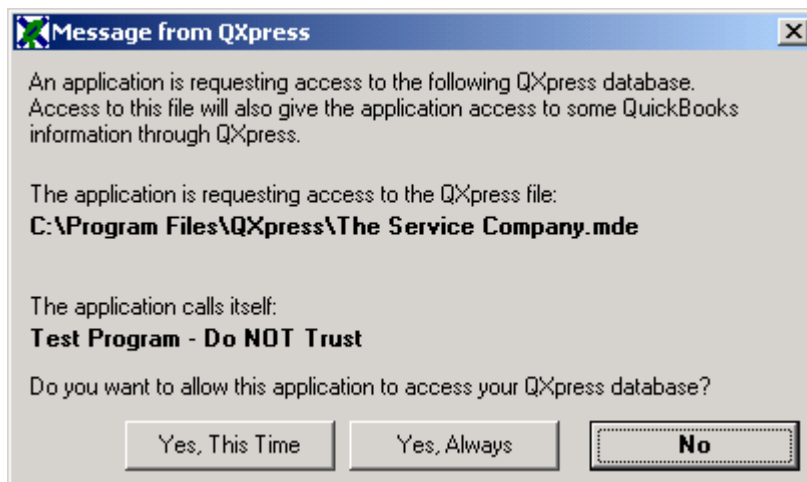
This chapter discusses the basics of what information your application needs to pass to QX_SDK.dll, and what information will be passed back to your application. The topics discussed in this section include:

- What is the QXMLRequestProcessor?
- What is inside a request?
- What is inside a response?

What is the QXMLRequestProcessor?

Gatekeeper

- The QXMLRequestProcessor talks directly to a QXpress database. Therefore, it alone has the responsibility of analyzing and validating your request, executing it, and finally providing your application with a response detailing the request's success.
- It also has the responsibility of prompting a QXpress user that your application is trying to access their QXpress database to see if they want to grant you access.



XML

- Interacting with QX_SDK.dll is straightforward. Your application must reference the QXXMLRequestProcessor in the QX_SDK.dll, and send it an XML request. In return it will provide an XML response. The following Visual Basic example shows a request and associated response to get a list of crews from QXpress:

Request:

```
Private Sub cmdSendRequest_Click()
On Error Resume Next
Dim Request As New QXXMLRequestProcessor
Dim Response As String
Dim sRequestXML As String

'Build request
sRequestXML = "<?xml version=""1.0"" ?>" _
& "<QXXML>" _
& "<QXXMLMsgsRq onError=""stopOnError"" loginuser=""Admin"" loginpassword=""AdminPassword"" " _
& "database=""C:\Program Files\QXpress\The Service Company.mde"" " _
& "ApplicationName=""Test Program - Do NOT Trust"" ApplicationAccessPassword=""430806"">" _
& "<CrewQueryRq RequestID=""124"">" _
& "</CrewQueryRq>" _
& "</QXXMLMsgsRq>" _
& "</QXXML>"

'Send request
Response = Request.ProcessRequest(sRequestXML)

'Close connection
Set MyRequest = Nothing

End Sub
```

Response

```

<QXXML>
  <QXXMLMsgsRs>
    <CrewQueryDefinitionRs...
    ...
  </CrewQueryDefinitionRs>
  <CrewQueryRs statuscode="0" requestID="124" statusSeverity="success" statusMessage="">
    <CrewRet>
      <CrewLeader>Andrew Jones</CrewLeader>
      <CrewName></CrewName>
      <CrewNumber>001</CrewNumber>
      <CrewRate>23.5</CrewRate>
    </CrewRet>
    <CrewRet>
      <CrewLeader>Amy Hazel</CrewLeader>
      <CrewName></CrewName>
      <CrewNumber>AMY</CrewNumber>
      <CrewRate>21</CrewRate>
    </CrewRet>
    <CrewRet>
      <CrewName>Not Applicable</CrewName>
      <CrewNumber>n/a</CrewNumber>
      <CrewRate>0</CrewRate>
    </CrewRet>
  </CrewQueryRs>
</QXXMLMsgsRs>
</QXXML>

```

What is inside a request?

Header

- Each request begins with the header, `<?xml version="1.0" ?>`, that specifies to the QXXMLRequestProcessor the version of XML used.

QXXML element

- The request will then start and end with the QXXML element. Everything found inside of `<QXXML>` and `</QXXML>` is the value of the QXXML element.
- No attributes need to be set for the QXXML element.

QXXMLMsgsRq element

- The request will continue with the QXXMLMsgsRq element. Within this element all requests will be listed.
- There are several attributes that must be set in this element, such as the 'database' attribute that specifies which QXpress database you wish your application to have access to, your ApplicationName, and the login information for the QXpress user. For more information on the required attributes, click on the Attributes link in the Onscreen Reference Guide.

Request Object

- The type of request you would like to make. In the previous example the Request Object to list the crews of the QXpress database is: `<CrewQueryRq RequestID="124">...</CrewQueryRq>`. For a full list of all the available Request Objects, select from the "QXML Message" in the Onscreen Reference Guide.

Special Characters

- Avoid using special characters in the value of your XML. For instance, if you must use an "&" symbol in your XML request, you must instead substitute "&". The most common special characters are:

Table 1: Common XML Special Characters

Special Character	XML accepted format
&	&
<	<
>	>
'	'
"	"

- The following example shows an incorrect and correct format of adding a customer named "Bob & Wendy's Restaurant":

Incorrect:

```
...<CustomerAddRq><CustomerAdd><CustomerName>Bob &
Wendy's Restaurant</CustomerAddRq></CustomerAdd></
CustomerName>...
```

Correct:

```
...<CustomerAddRq><CustomerAdd><CustomerName>Bob &amp;
Wendy&apos;s Restaurant</CustomerAddRq></CustomerAdd></
CustomerName>...
```

What is inside a response?

QXXML element

- The response will start and end with the QXXML element. Everything found inside of <QXXML> and </QXXML> is the value of the QXXML element.
- No attributes will be passed in the QXXML element.

QXXMLMsgsRs element

- The response will continue with the QXXMLMsgsRs element. Within this element a response for each request will be listed.
- In Version 1.0 of the QXpress SDK, no attributes are returned with this element.

Object Query Definition

- For your convenience, a definition of type and size of each field that will be returned is first listed in the Object Query Definition. You may choose to ignore this portion of the response, or you may choose to use the definitions to dynamically build a structure (such as a database or collection) to parse the following Object response into.

Response Object

- In the previous CrewQuery example, the Response Object is <CrewQueryRs statuscode="0" requestID="124" statusSeverity="Success" statusMessage="">. Then for each crew record, a <CrewQueryRet> element lists the fields for each crew and the associated values.
- The Response Object will contain meaningful information about the success or failure of your request, in the form of the statuscode, statusSeverity and statusMessage attributes. For more information on these response attributes, click on the Attributes link in the Onscreen Reference Guide.

