Using 4D to manage eBay

By Frank Chang, 4D Evangelist
Technical Note 04-51

Introduction

More and more of today's small businesses are trying to gain web exposure. They want to make available, their products and services to the world without investing heavily in an internet infrastructure. It's a dog eat dog world in the business world, and you need every advantage you can get. Enter eBay, a company that has re-defined the way small businesses do business. With their innovative ideas, a new medium was born for linking businesses to their consumers. 4D developers can take advantage of eBay's medium and architecture through the use of the eBay Developers Program and 4D's Web Services capabilities.

You've got great ideas for new products and services. The eBay Platform provides what you need to take your business to the next level. Do you have a new way of using eBay that no one has thought of yet? Have you built the next killer app? With 4D and the eBay Developers Program, you can build a killer application that connects to eBay or simply add eBay functionalities right into your current 4D application. In this Tech Note, we will discover how to integrate eBay into a 4D application with the help of the 4D - eBay Object Library.

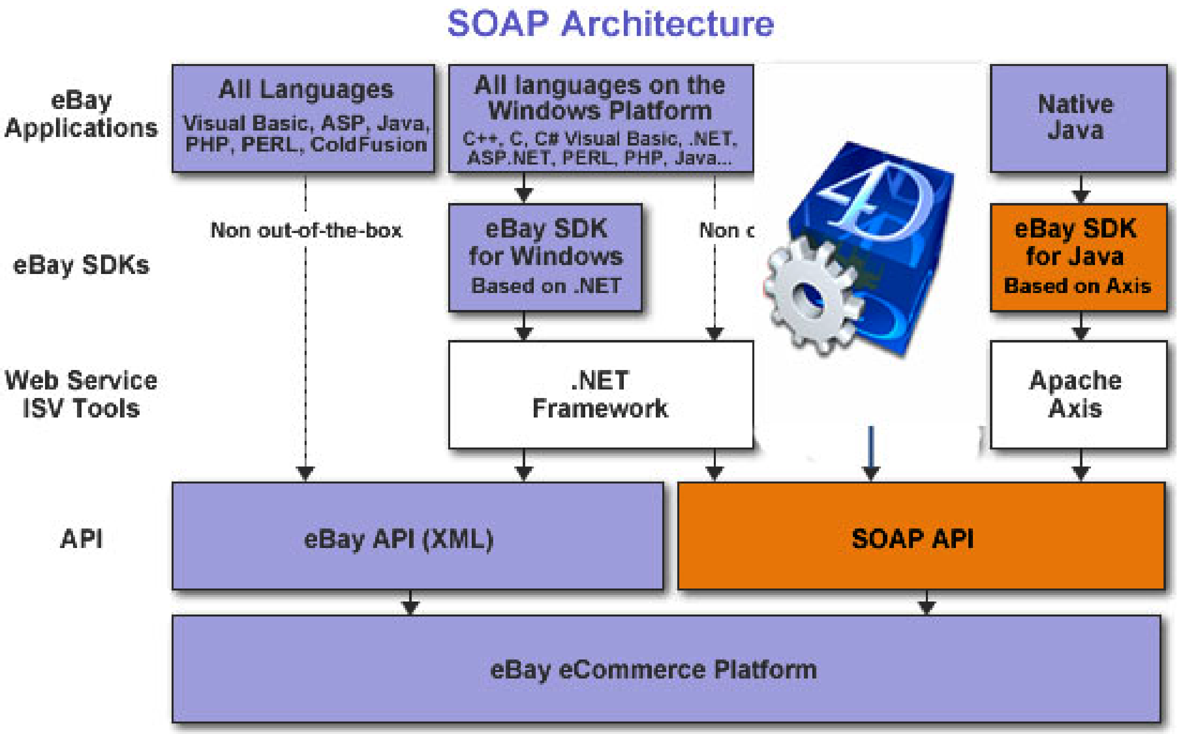
About eBay and the eBay developers program

By now, almost everyone has heard of eBay. Most of us reading this Tech Note probably have been on eBay during the last few months or so. EBay is the world's largest online marketplace for the sale of goods and services by a diverse community of individuals and businesses. The eBay community consists of millions of members from all over the world. It is one of the most popular shopping destinations on the internet. People come to the eBay marketplace to buy and sell items in thousands of categories. There are also a variety of features and services that enable members to buy and sell their products quickly and conveniently. Buyers have the option to purchase items in auction-style format or items can be purchased at fixed price through a feature called Buy-It-Now.

The eBay Developers Program provides technology and resources that enable you to fully leverage eBay's platform, strategies and community. The program provides you, the developer, with unparalleled innovation, competitive advantage and stability. Included in the eBay Developers Program is a complete Web Services API (the eBay SOAP API). This API allows 4D Developers the ability to communicate seamlessly with eBay application servers. You can create customized interfaces to integrate your customer's product line on eBay's web site.

The eBay SOAP API consists of database calls that return data in a standard format for easy use within your application. Using the SOAP API, you can build highly reliable tools and bring them to market faster than is possible with other means. 4D can directly access this API through the use of the Web Services Wizard.

Figure 1: 4D interacting with eBay's SOAP API



The uses of eBay are many and varied. Some users are individuals who occasionally sell personal item(s) from home. Other users are small businesses that do not wish to have the overhead of a physical store, and offer their products and services on eBay. And still other users are major manufacturers, services, or distributors for which eBay is another outlet for marketing their products. All of these cases can be supported and enabled by applications created using the API.

New Opportunities

New businesses are being created on eBay everyday. Established manufacturers and retailers are adding eBay as one of their standard sales channels. How is this possible, you ask? Well, it is because eBay is a leveled playing field, where an individual or small business seller operating a home-based business has the same capabilities (and the same regulations) as a large corporation. Anyone with access to the Internet can participate in the eBay marketplace.

EBay sellers range from individuals to small businesses, retailers and large manufacturers. Because of this, the breadth of available products in the eBay Marketplace continues to grow. Even though eBay's history lies in collectibles and antique sold at auction via individuals, the number of new and in-season goods from businesses and retailers that are available in a fixed-format is large (approximately 30% of all eBay listings) and growing. Thus, when evaluating the opportunities that eBay may provide, keep in mind that eBay is much more than auction-format listings and used items. This fixed-format enables a buyer to easily purchase an item without competing with other buyers or waiting for the end of an auction.

Here are a couple of software application development scenarios that eBay's Web Services API can benefit 4D Developers.

Individual Applications - You can build using 4D, single apps intended to accelerate the activity of a single eBay user or business. Such applications can be highly optimized for a particular type of business sector or product type. For example, managing a catalog of certain types of items for sale, such as collectible, cars, clothing, or anything tailored to the preferences of a specific user or class of users.

Bridge Applications - Another type of application that you can build with 4D is an integration applications designed to create a bridge between the inventory and fulfillment systems of an established business. For example, if you or your client has an existing product database that you would like to list as products on eBay's marketplace. This can be accomplished quite easily by adding the functionalities found in the 4D - eBay Object Library.

Even if your business does not deal with physical goods, lots of opportunities exist for eBay Developers. Many technology consulting firms provide eBay API integration to customers as part of their consulting practice. In addition, independent software contractors can utilize 4D and the eBay API to create custom applications with unique interfaces for the eBay platform that are designed to accelerate the user experience for specialized types of buyers and sellers. Through 4D, a powerful eBay API application can be created in a short amount of time.

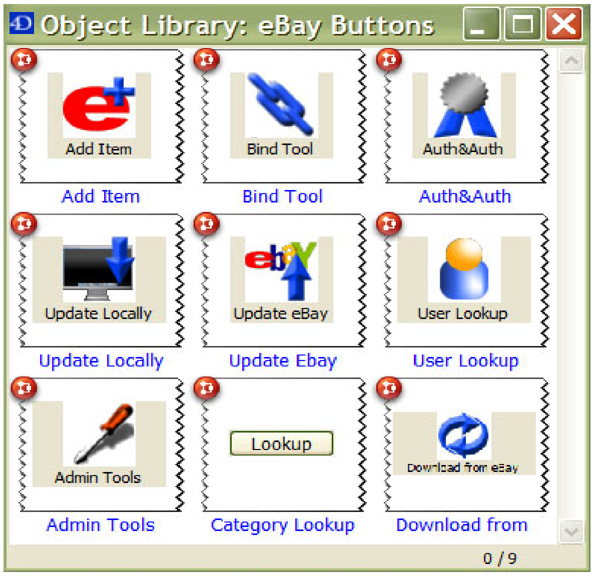
Creating an eBay application

Creating an eBay application is simple. With 4D 2004, we have taken advantage of one of the many new features, the Object Library. Based on the eBay API, We have created an Object Library and template combo that allows 4D developers to develop their eBay application from the ground up. In this Tech Note we will focus on the eBay toolkit that we’ve created as well as on an application created with this toolkit.

The 4D – eBay Object Library

The 4D - eBay Object Library consists of nine (9) objects which you can drag and drop onto your database. These objects gives you the ability to communicate with the eBay SOAP server to create items on eBay, download items from eBay, update items… etc.

Figure 2: the 4D – eBay Object Library



Here are a few of the object descriptions in detail:

The Bind Tool

This object is used to bind the current table fields to the corresponding fields in eBay. Currently, there are (14) fields that are required by the eBay SOAP API. This object will call the project method Binding_SetBindings and display a form for you to configure the binding properties. Once the fields have been mapped to eBay's fields, we can proceed to upload items to eBay or download them from eBay.

Add Item(s) to eBay

This object allows for the insertion of items from your table to eBay's online marketplace. This object will call the project method eBay_PostAdd. This method will cycle through the selected records and build a SOAP request for each item and call the eBay AddItem. All the code involved in the building of the SOAP request is done in the 4D method proxy_AddItem.

Download from eBay

This object will download the existing product(s) that I have on eBay's online marketplace. This object calls the project method eBay_PostGetSellerList, which then downloads and parses the XML for items listed on eBay

Update eBay

This object allows you to directly update the auction id on eBay corresponding to the record edited in 4D. This object is quite useful when you want to display the changes you've made locally in your 4D app, on eBay. The records must be already on eBay's site, otherwise you won't be able to update anything, and an XML error is returned.

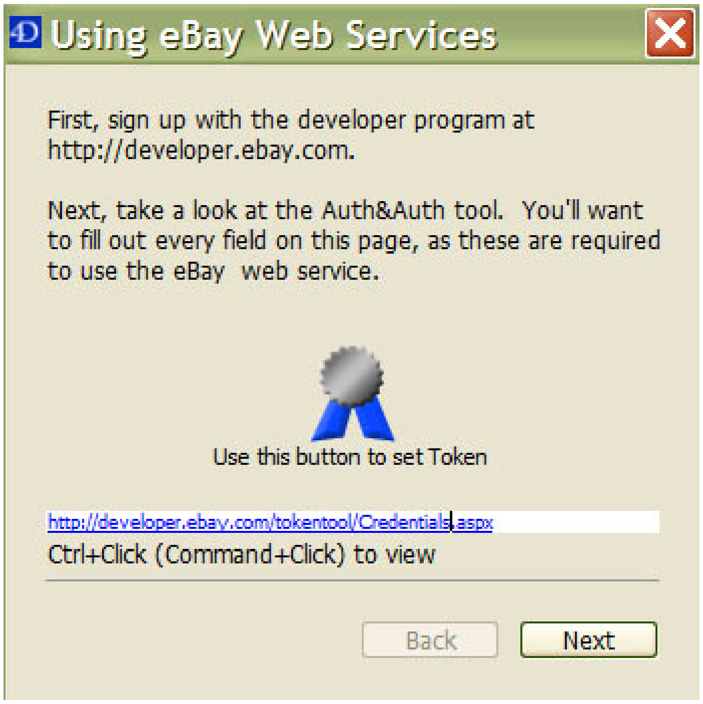
Update Locally

This object will update your 4D application with the latest data on eBay. The matching auction ID on your local system will be overwritten with the information from eBay.

Building an eBay app

Now that we have an understanding of how the tools work. Let's take a look at how to actually set this up. When you first start up the demo database, you will be instructed to follow a few pages of instructions (the tutorial). The first page deals with obtaining an eBay token. Follow the instructions to navigate eBay's developer site to get your token.

Figure 3: the Auth&Auth tool



First, sign up with the developer program at http://developer.ebay.com.

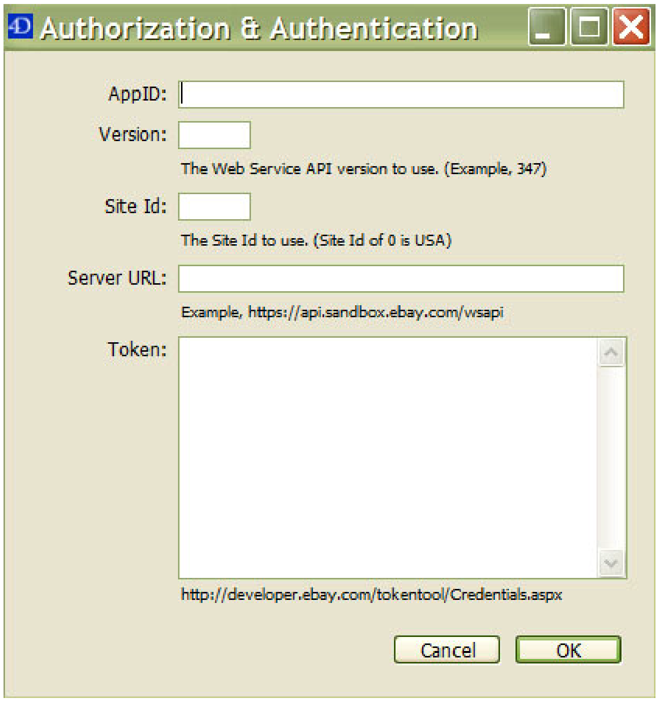
Next, take a look at the Auth&Auth tool. You'll want to fill out every field on this page, as these are required to use the eBay web service.

http://developer.ebay.com/tokentool/Credentials.aspx

Ctrl+Click (Command+Click) to view

Visit http://developer.ebay.com for more information on how to obtain a developer's token. Clicking on the Auth&Auth tool button, you will be prompted for the information provided when you received your developer's token.

Figure 4: Authorization & Authentication information



The Web Service API version to use. (Example, 347)

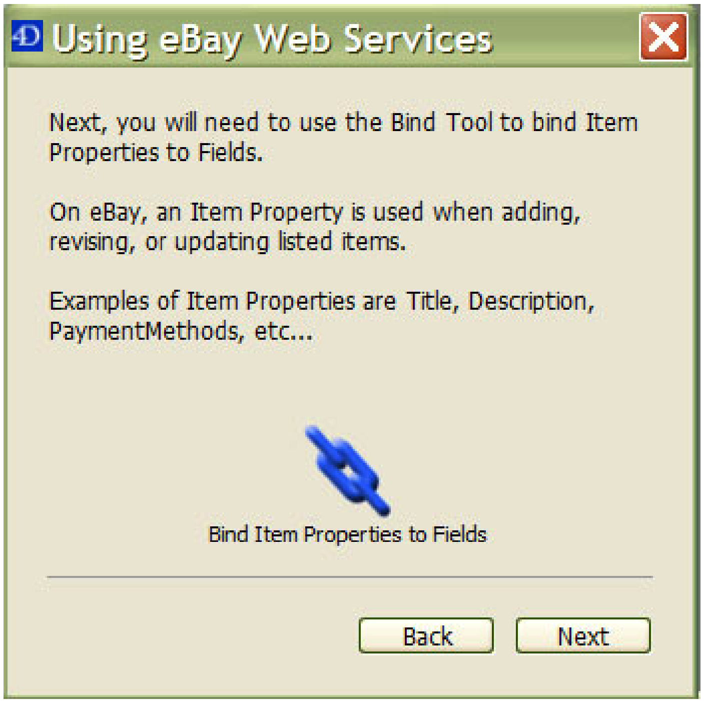
The Site Id to use. (Site Id of 0 is USA)

Example, https://api.sandbox.ebay.com/wsapi

http://developer.ebay.com/tokentool/Credentials.aspx

Page two of the tutorial, explains the details of the bind tool. The bind tool is used for binding data from 4D to eBay. With each Web Services call to eBay, data from our database needs to be placed into the correct fields on eBay's side. Data fields in 4D will be "bound" to their respective fields on eBay's end.

Figure 5: The Bind Tool

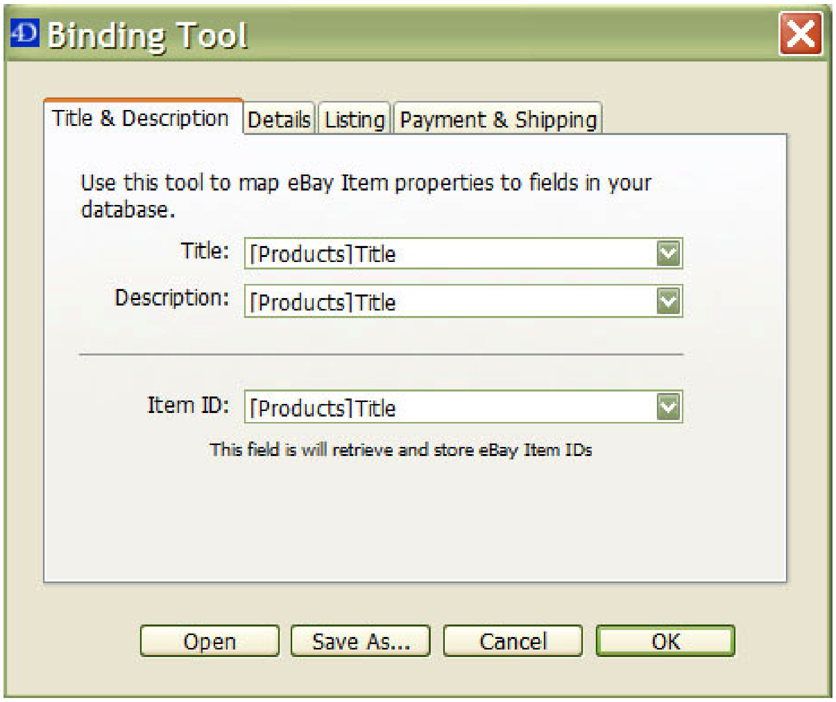


Next, you will need to use the Bind Tool to bind Item Properties to Fields.

On eBay, an Item Property is used when adding, revising, or updating listed items.

Examples of Item Properties are Title, Description, PaymentMethods, etc...

By clicking on the bind tool icon button, the properties of the bind tool are displayed. From this property dialog, simply select the 4D table/field from the drop down lists on the right to bind to the eBay fields on the left.



Use this tool to map eBay Item properties to fields in your database.

Title: [Products]Title

Description: [Products]Title

Item ID: [Products]Title

This field is will retrieve and store eBay Item IDs

Note: The properties of the bind tool only needs to be set once for the entire database

Building the Interface

Once the bind is complete, we are ready to build the look and feel of our eBay application. As mentioned earlier, the Object Library that is included with this database contains many 4D objects that will perform certain functions. Open the object library and add the objects to your output form. These objects are generic enough so that you may use them with any table that you've already bound with the bind tool.

The project method AbstractAction_Startup is one of the main methods behind the scenes of many of the objects found in the 4D-eBay Object Library. This method is quite generic, which gives it the ability to be called from many objects with different parameters. The key in this method is the third parameter, which contains the method name to call in another process. From here, the actual function that the object was meant to do is called.

Method Name: AbstractAction_Startup

```
C_POINTER($1;$pTable)
C_TEXT($2;$currentSelect)
C_TEXT($3;$methodName)
C_BOOLEAN($4;$xmlDump)

  ` use the default binding for adding items
If (Test path name(eBay_GetBindingPath )#Is a document )
       Binding_SetBindings
End if

If (Test path name(eBay_GetBindingPath )=Is a document )

       $pTable:=$1
       $currentSelect:=$2
       $methodName:=$3

       If (Count parameters>3)
         $xmlDump:=$4
       Else
         $xmlDump:=False
       End if

       C_LONGINT($processCount)
       C_TEXT($processName)

       Repeat
         $processCount:=$processCount+1
         $processName:=$methodName+String($processCount)
       Until (Process number($processName)=0)

       $proc:=New process($methodName;1024*32;$processName;$pTable;$currentSelect;$xmlDump)
```

```
End if
```

For the Add Item object, the variable $methodName above is eBay_PostAdd. This method will build the XML blob based on the current selected record. The order of procedure is as follows:

1) Construct the item encapsulation that we pass to the proxy method when we want to add the item.
2) Initializes the necessary elements that are required in a soap request. Such as, the Token, Application ID, version, server, and the site id.
3) Once the encapsulation and initialization are completed, it is appended with the data that we want to add to eBay (all the item information).
4) Execute the proxy call to eBay. This sends all the information we have gathered to eBay.
5) Finally, the response is parsed and the item id for this item is retrieved and saved with the current selected record.

Note: initially the item id is not available when the item was first created. This is because the item id has to be generated by eBay. The item id is the auction id that can only be generated by eBay. Once we have obtained an item id, we can be assured that the Add Item process was successful. Below is the section of the project method eBay_PostAdd that is explained in the order of procedure above.

```
Method Name: eBay_PostAdd

$item:=eBay_ConstructItem

 ` Initialize and ready eBay, then make the call using the node references to each
 `  item encapsulation
If (eBay_Initialize (EBaySoapVer;EBaySoapServer))

        $addItemResponse:=proxy_AddItem ($item)

        If (eBay_isOKResponse ($addItemResponse))
             ` Set the ID field to the id we retrieve from the soap request
            $id:=eBay_GetItemID ($addItemResponse)

            If ($id#"")
                $ptrID->:=$id
                SAVE RECORD(CurrentTable->)
                $addCount:=$addCount+1
            End if
        Else
            $errorMessage:=eBay_GetError ($addItemResponse)
            $i:=Records in selection(CurrentTable->)
        End if

End if
```

Most other objects in the 4D-eBay Object Library follow the same sequence. That is, the objects themselves call a series of methods to obtain the selected record or records and then call the AbstractAction_Startup method. From this method, it is where the intended function of the object is executed.

To use the included database, start with the tutorial. Once you’ve set the Authorization and Authentication, you’ll be ready to create items on eBay and download your existing auction items. In the Custom environment, selecting Tutorial from the File menu will bring up the tutorial to get you started while selecting Store Example, will bring up the My Store table interface with the eBay functionalities already built in.

Summary

4D's ability to utilize Web Services has dramatically increased the opportunities for developers to develop new business applications. Applications that can communicate with and facilitate transactions on eBay can prove to be very exciting to say the least. With many companies providing services via Web Services, 4D has once again paved the way. With the 4D - eBay Object Library, developers can add functionalities that can mainstream an existing product line to a whole new market exposure, namely the eBay marketplace.

The eBay SOAP API allows a developer to create an application that performs many of the same important auction-related operations as the eBay site. It used to be very time consuming to put a product line on eBay. Each item would have to be created manually on eBay's web site. With 4D and the eBay SOAP API, your existing product line can be viewed by the eBay community in a matter of seconds.