Customizing MarketBlast

By Sati Hillyer -- 4D Evangelist, 4D Inc.
Technical Note 05-29

Overview

MarketBlast was created with 4D 2004. Using some advanced programming logic, we have implemented an automatic updating mechanism. Therefore as new versions of MarketBlast are created, our customers can easily upgrade and install the changes at the touch of a button. MarketBlast will contact remote servers and if a newer version is found, automatically download and reinstall the new version. It'll even re-launch MarketBlast for you so you never miss a beat.

MarketBlast is sold open-source allowing users to add more functionality to the application. To make this process even easier, MarketBlast provides another interface to add customization. MarketBlast enables the user to automate many if not all tasks using Automation. Within Automation, a user can create custom events and actions that can propel MarketBlast's already powerful engine. So without even having to go to the Design environment, you will be able to modify the structure of MarketBlast.

And the best part, new custom events and actions that you create will remain in the application forever. Even when you upgrade to another revision, MarketBlast will keep track of your custom changes and incorporate them into the new version. Have your cake and eat it to!

Introduction

The following technical note will guide a MarketBlast user/developer on how to use Automation and add custom events and actions for even more custom-made power. It is important to understand the use of custom events and actions, because only these will maintain after upgrades. Other changes made to the underlying structure (form, methods, triggers, resources, etc.) will be lost once upgraded.

Custom actions are actually 4D methods; however the reason they are not affected is because a custom macro is used to pull the content of the method and store it externally. Therefore, when an upgrade is made, the structure will change and the external methods will be added after.

Combining the power of custom events and actions with automation, MarketBlast will pave the way for a new level of power and finesse with 4D applications.

Understanding Automation

Automation provides MarketBlast users the power to set it and forget it. Build your custom scripts to perform various functions, filter specific data with a variety of events and have this all happen when you want it with the integrated scheduler. In addition,

you can add custom events and actions for even greater control. To bring up the **Automation** window, select **Automation** from the Automation menu (*Figure 1)*.

Figure 1: MarketBlast Automation Window

Automation is comprised of 3 main components:

- **Events**
- **Actions**
- **Scripts**

An Event is a rule or condition that can be used to filter data retrieved from eBay when running synchronization. An Action is how MarketBlast will respond when triggered by an Event or Script. A Script is a series of Actions. These comprise the building blocks of MarketBlast's Automation. By default, MarketBlast has some predefined values to begin the automation process. These defaults cannot be deleted. There are a number of events that you can begin filtering against. There are set of predefined actions that can be triggered instantly, through a menu item or at a scheduled date and time.

Events and actions can be even more powerful through customization. Think of it this way, if you could create your own 4D form event, how much more power and customization could you achieve? Well with MarketBlast, you now can! Create custom

events that can include a number of conditions and use these at a global or individual level. Global events affect all records (listings, questions, contacts, etc.) that fit under the specified conditions. Or for more control custom events can be placed at an individual level where only that item will be affected. This allows you to control every nook and cranny of MarketBlast.

For more information about the predefined events and actions, please refer to MarketBlast documentation.

Creating a Custom Event

Custom Events provide the user with the option of personalization when working. MarketBlast is created to grow with its user and Custom Events are one component of this dynamic. To create a Custom Event, select the Events button on the left side of the Automation screen. On the screen, press the "+" button on the top of the screen (*Figure 2)*.

Click the Add button to create a custom event

Provide a unique name

Figure 2: Creating a Custom Event

When creating a new event, you have 4 main categories of properties to build the event. Each category is tied to an underlying table in MarketBlast:

- Item – Listing Table
- Feedback – Transaction Table
- Bidder – Offer Table
- Question – MemberMessageExchange

The respective properties are mapped to the respective fields of that table. Now, you simply build a rule as you would the Query editor. Pick a property (field), a condition (comparison) and a value. Click Add to generate the new Rule. An event can be made up of multiple rules. You can add advanced logic to determine what will finally trigger the event using AND or OR operators. Once done, click Save to finish creating your new custom event.

So what have we just done? We have just created a new filter when retrieving data from eBay. With custom events, you can build practically any filter to make MarketBlast more effective and productive. Once the event has been created, you will see the new custom event in your list of events. Find your new event and select the corresponding actions you'd like MarketBlast to take.

Most events are triggered by the Synchronization process. Once we sync with eBay, MarketBlast will look for the active events and filter the data accordingly. You can manually sync MarketBlast anytime. However, for more productivity you can use the scripts and build routines for MarketBlast to execute. These scripts can later be triggered on a reoccurring schedule. Let's now discuss how to create a script.

Creating a Script

Scripts are intended to run an instance of actions at one time. A user would build a script of actions, run them instantly, turn the script into a menu item or choose to have the script execute at a scheduled date and time.

To create a script, select the Script button on the left hand side of the Automation screen.

Click the Add button to create a custom event

Provide a unique name

Figure 3: Creating a Script

For this example, we are going to create a script that will bulk list all our new items to eBay. We named our script, "Bulk Publish" and selected the predefined action, List New Items. Now we have three ways to execute this script. You can select the Play button, top right of *Figure 3*. This will execute the script immediately. Your second option is to select the checkbox, "Available from menu bar" which will take this script and append it as a menu item in the **Automation** menu. It will use the same name you provided when creating the script. The third and final way to execute this script is using the scheduler (*Figure 4*). Click the Scheduler button on the left pane. Select your script from the drop-down menu at the top of the screen. Pick whether you want the script to run hourly, daily, weekly, monthly or at a certain day and time.

A number of actions are predefined when you start using MarketBlast. For more power, MarketBlast enables the user to create personalized custom actions that can pretty much do anything you desire. Let's now discuss how to create a custom action.

Figure 4: Schedule a Script

Creating a Custom Action

A custom action is your gateway to the core of MarketBlast. It is actually a 4D method that you can write and seamlessly integrate back into MarketBlast. Since you have access to the method editor, the possibilities are endless as to what you can accomplish. A user can create as many custom actions as they want so that they can have MarketBlast perform just as they wish. What's more, we store the custom actions externally, so when you run an update to a new version of MarketBlast, you can rest assure that all of your custom actions will still be there.

To create a custom action, be in either the Events or Scripts screen. Click the blue gear button on the top right hand side of the window (*Figure 5*).

Click to create a Custom Action

Provide a unique name

Figure 5: Creating a Custom Action

After providing a unique name for the action, you will be taken to the 4D's Method Editor. Here you can begin writing a method utilizing any and all of the 4D commands. Once complete, run the Macro, "Create Custom Action" shown in *Figure 5*. You will then be taken back to the Automation screen where you can now incorporate your new custom action. Whether you build an event or script, you can now use one of your custom actions for greater control and functionality. If at any time you do not want your Custom Action showing up in the Automation window, go back to the original method tied to the action and run the macro, "Remove Custom Action." This will remove the action from the Automation window, but it does not delete the method.

Behind the scenes, when the macro is called, MarketBlast will copy the content of the method and create a file externally. Therefore, when a new version is installed, all of

the external methods will be dynamically recreated and added back into your version. Note: Only custom actions will maintain from version to version. Since the structure file is replaced, others changes made such as forms will be lost.

Now let's take a look at some sample scenarios.

Scenario 1: Updating an External Data Source

As a large seller, you may maintain a separate inventory in another database engine. With that, every time an item has sold, it would be great to update MarketBlast inventory as well as your external database. Well, with custom actions, we can do this!

Let's begin by creating a custom action; we'll call it *Update External*. The custom action will begin by connecting to the ODBC data source. Once connected, we start with our first sold item and load the related inventory. We build our SQL statement and then execute it to update the external ODBC data source. If there are more than one sold item, we'll loop through them all and repeat the same steps. Once complete, we log out and unload our records. Note: You will have to change the ODBC data source and SQL statement to reflect your system.

```
  `Log into your ODBC Data Source – You will have to change these values to fit your system
ODBC LOGIN("odbcdatasource";"username";"password")

If (OK=1)

  C_TEXT($sql)

    `Start with the first Sold item
  FIRST RECORD([Listing])
  Repeat
      `Find the related Inventory item
    RELATE ONE([Listing]ProductId)

      `Build the sql statement to update the ODBC data source – You will have to changes this statement
    $sql:="UPDATE MyInv SET Stock ="
    $sql:=$sql+String([Inventory]Quantity_InStock)
    $sql:=$sql+" WHERE Title="
    $sql:=$sql+"'"+[Inventory]Title+"'"

      `Execute the statement
    ODBC EXECUTE($sql)

      `Go to the next sold item
    NEXT RECORD([Listing])
  Until (End selection([Listing]))

    `Log out of the ODBC data source
  ODBC LOGOUT
End if
```

```
  `Release used records
UNLOAD RECORD([Listing])
```

The SQL statement is using the UPDATE command which will modify values on the ODBC data source. The SQL statement is written as follows:

```
UPDATE tableName
        SET QtyField = MarketBlastQty
        WHERE TitleField = MarketBlastTitle
```

The statement updates the tableName you provide by using the SET command to update the QtyField you have in your ODBC data source. In our code, we set it to the value stored in the [Inventory]Quantity_InStock field in MarketBlast. We then use the WHERE command to make the comparison. For this example we compare based on the Title of the item. You may want to modify this to compare on an ID or SKU number.

Once the method has been written, execute the macro: "Create Custom Action" shown in *Figure 5*. You can now use this custom action with the event "When Items are Sold."

Scenario 2: Alert Seller if Negative Feedback Received

Now there may come a time where you received negative feedback. Heaven forbid, but the possibility is there. Since this is not in your control, your best option is to be on top of it if and when it does arrive. Negative feedback is bad for business. Therefore, let's create a custom action that alerts via email when we receive a negative feedback. That way we can eliminate any delay in handling this matter. We'll call our custom action: *Notify Seller*

In the custom action, we load our SMTP settings that we set in the Email Preferences. We want to alert the seller, so we do a quick search for the seller contact info. Now we'll start with the first response if there are multiple, write our subject and body and send the message using ***SMTP_QuickSend*** command. If there is more than one, we'll repeat this process over.

```
  `Load the SMTP settings so you can send the email
C_LONGINT($smtpauthmode_l)
C_TEXT($hostname_t;$smtpuser_t;$smtppswd_t;$emailaddres_t;$Subject_t)
eBay_eLoadSMTPSettings (->$hostname_t;->$smtpuser_t;->$smtppswd_t;->$smtpauthmode_l;-
>$emailaddres_t)

  `Find the Seller Contact info
QUERY([Contact];[Contact]IsSeller=True)

  `The "When Negative Feedback is Received " event will create a current selection of feedback response
  `Start with the first Negative Feedback response
  `Hopefully you don't get more than one :-)
FIRST RECORD([Transaction])
```

```
  `Write our subject
$Subject_t:="Negative Feedback Received!"

  `Loop through each response
Repeat
     `Write our body
   $Message_t:="Oh NO! You have received negative feedback from "+[Transaction]UserID+". "
   $Message_t:=$Message_t+"The feedback stated: "+[Transaction]Feedback.CommentText+". "
   $Message_t:=$Message_t+Char(13)+Char(13)+[Contact]Buyer.Ship.Name

     `Send the email
   $Error:=SMTP_QuickSend ($hostname_t;[Contact]Email;[Contact]Email;$Subject_t;$Message_t)
   If ($Error#0)
    ALERT(IT_ErrorText ($Error)+" An error occurred when sending email. Please check your Outgoing Email
settings")
   End if

     `Go to the next feedback response
   NEXT RECORD([Transaction])
Until (End selection([Transaction]))
```

Once the method is complete, run the Create Custom Action macro and set the action in the "When Negative Feedback is Received" event. This is a great action if you want to quickly send an email. It can be reused with a different subject, body and recipient and combined with any event.

Scenario 3: Auto Respond to certain Emails

MarketBlast 1.10 introduces a new event that will auto respond to incoming emails. Simply select "When Receiving an Email" and choose the action "Send an Automatic Email Reply to Sender" and you can then have MarketBlast act as you personal Out-of-the-Office assistant.

Combined with Custom Actions we can make this new event even more powerful. In this scenario, let's create a custom action that acts as a filter. We can then pass a certain keyword that will be used to extract certain messages you desire. For instance, if you know there has been a lot of attention on a certain listing. Pass in the Listing Item ID into $searchVar and all new emails received will be filter accordingly. This way you can tailor your automatic email response. Let's call our custom action: *Filter Email Message*

```
C_TEXT($searchVar)

  `This is what you will be filtering for – Type in the keyword you desire
$searchVar:=""

  `Create a set to store the filtered emails
CREATE EMPTY SET([Emails];"EmailSet")

  `The "When Receiving an Email" event will create a current selection of all new email
  `Start with the first email
```

```
FIRST RECORD([Emails])
Repeat
    ` Search for the variable we defined earlier
  If (Position($searchVar;[Emails]Body)>0) | (Position($searchVar;[Emails]Subject)>0)
      `If we find a match, add it to our set
    ADD TO SET([Emails];"EmailSet")
  End if

    `Move to the next Email
  NEXT RECORD([Emails])
Until (End selection([Emails]))

  `Loads the set we just created into the current selection. Now we can focus on the filter messages
USE SET("EmailSet")

  `Always clear your sets!
CLEAR SET("EmailSet")
```

Once the method is complete, run the macro Create Custom Action. Remember, the filter will rely on the value you define for $searchVar.

Now we can set the next action, the automatic email response. Select the action; Send an Automatic Email Reply to Sender. For this example, let's create a new Email template to respond. Select an Email Template and click the Edit Template button (*Figure 6*). In the Edit Email Template screen, select New from Template drop-down

Edit an existing Email Template

Figure 6: Automatic Email Response

menu. Write a new email template and press Save. For our example we use some custom formulas, such as eBay Item ID and Listing Name, since our keyword from earlier will help us filter the appropriate messages. Once complete, select Choose to set the new email template as your automatic response. Now every time new email messages are received, we can filter certain messages based off a keyword using our custom action. Then we can respond automatically with a tailored response using our email templates.

Scenario 4: Creating a Custom Event for Offer Second Chance

As a seller, you may have certain items that you want automated tasks triggered. In our previous scenarios, we have discussed events that affect the global aspect of MarketBlast. In other words, any listing that falls into that event will be affected. With Custom Events, a seller can automate tasks for individual listings thus providing superior control.

In this scenario, we are going to create two custom events. One that filters if an item has been sold; the other event is if the Second Chance offer has been accepted. Now you may be wondering why we want to create an event for an item sold since we already have that in our event list. Remember, we want this event to be triggered at the listing level, not globally. So we need to use Custom Events to accomplish this.

The scenario is for those sellers who when they list an item, they want to automatically offer Second Chance to the next highest bidder. Then if the bidder decides to accept the Second Chance offer, we want to automatically send the winner a thank you email.
Custom Event: *When Item is Sold*

Following the steps from *Figure 2*, create a custom action with the following conditions:

- Properties = Item
- SellingStatus is equal to 0
- QuantitySold is greater than 0
- TimeLeft is equal to PT0S

Figure 7: When Item is Sold

Note: To turn this event into *When Item is Unsold*, set the QuantitySold condition to equal 0. The other 2 rules remain the same.

Custom Event: *Second Chance Offer is Accepted*

Following the steps from *Figure 2*, create a custom action with the following conditions:

- Properties = Item
- SecondChanceOriginalItemID is not equal to "No Value" (leave it blank)
- SellingStatus is equal to 0
- QuantitySold is greater than 0
- TimeLeft is equal to PT0S

Figure 8: Second Chance Offer is Accepted

Save the custom events. Now let's go to a specific listing and activate the desired actions. In the Listing form, navigate to the last tab – Automation. A list of your

custom events will be displayed (*Figure 9*). So the first this we want to do is provide a Second Chance Offer to the next highest bidder. Check the custom event, "When Item is Sold" and scroll through the list of actions until you find Offer Second Chance. Press the green arrow to add it to the My Action list. This list will be the action(s) that will be executed when the event is triggered. Continue to our second custom event, "Second Chance offer is Accepted" and select the action, Send an Email to Buyer. Once the buyer has agreed to accept the offer, next time sync is performed MarketBlast will automatically send an email to the buyer. Select the action in the My Action list and you will see options appear below. Let's choose an Email Template and select the Edit Template button to bring up the MarketBlast Email Template window (*Figure 9*). For our example, we created a new template by selecting the Template drop-down menu and choosing New. Write your template and press Save, and then Choose.

Figure 9: Automation Listing Specific

Using the Database

This technical note has been written for MarketBlast users. In order to try these examples, you must own a valid MarketBlast license. To purchase MarketBlast, visit our store online. **http://store.marketblast.com/**

MarketBlast is completely customizable and delivered open-source. For more information about MarketBlast, be sure to check out the website. **http://www.marketblast.com**

Summary

To summarize, we have explored much of the customization you can achieve with MarketBlast. We've outlined using automation at both the Global and Item specific level. Through our scenarios, we've discussed real-life examples incorporating Custom Events, Custom Scripts and Custom Actions. Custom actions allow the MarketBlast user/developer to add functionality by writing 4D methods. Once more, these methods will remain even as you upgrade to the next revision. With the use of these features, MarketBlast can be manipulated to function and operate how you do. Inefficient will be a thing of the past!