Use Scripting to Change a Warehouse in Sage 100 ERP

Posted on **March 3, 2015** by **Alnoor Cassim**

Scripting Tips for Sage 100 ERP

Blogger: Alnoor Cassim, xkzero Technical Services

When creating Sales Order Entries in Sage 100 ERP, have you ever keyed in many lines of shipping information on multiple orders only to learn that the product needs to ship from a different warehouse than the location originally entered?

If so, you know that manually correcting the warehouse code on each order line is quite time consuming (and not exactly fun). You may have even thought, "If only there were a way to change the warehouse information *just once"* to apply the update to all the line warehouses… Good news! This wish can come true by creating a fairly straightforward script.

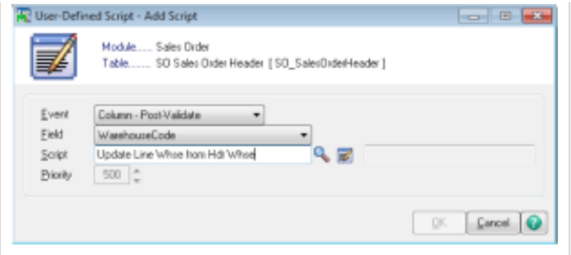
What kind of script do you need to create?

Create an *event script* that runs when the Warehouse Code on the Header is changed.

How do you create this script?

Follow these steps:

1. Go to the Custom Office/Main Menu, and make these selections:
   "User-Defined Field"
   "Table Maintenance"
2. Under the "Sales Order" heading, choose "SO Sales Order Header."
3. Right-click and choose "User Defined Scripts."
4. In the "User Defined Script" window, click the "Add" button. This will open the "Add Script" window.
5. In the "User-Defined Script – Add Script" window, as shown below, make these choices:
   Event: "Column Post-Validate"
   Field: "WarehouseCode"
6. Type in a name for your script, such as this:
   "Update Line Whse from Hdr Whse"



User-Defined Script - Add Script

Module..... Sales Order
Table...... SO Sales Order Header [SO_SalesOrderHeader]

Event: Column - Post-Validate
Field: WarehouseCode
Script: Update Line Whse from Hdr Whse
Priority: 500

7. You will see a message indicating that the script file does not exist, followed by a question, “Do you want to create it? Yes/No?”
   Yes, you do want to create the script! Click “Yes.”
8. Now the “Edit Script” window will appear. Copy the following code and paste it into your script editor window. **Note: It is important to use the code *exactly* as shown below, including line breaks and spaces.**

```
‘Init VARs
newWhse = value : sItemCode = “” : sItemType = “”‘If an existing sales order and not during update
If oBusObj.EditState = 1 and oSession.Updating = 0 Then

    Set oLines = oBusObj.AsObject(oBusObj.Lines)retVal = oLines.MoveFirst()sMsg = “Would you like
    to update the Whse Lines to Whse Code ” & newWhse & “?”
    retMsg = oSession.AsObject(oSession.UI).MessageBox(“”,sMsg,”Style=YesNo”)If retMsg = “NO” Then

            Exit Sub ‘
    Else

            retVal = oLines.MoveFirst()
            ‘Loop through all the lines and choose regular inventory item lines only
            Do Until CBool(oLines.EOF)

                        retVal = oLines.GetValue(“ItemCode$”, sItemCode)
                        retVal = oLines.GetValue(“ItemType$”, sItemType)If sItemType = “1” Then

                                        retVal = oLines.SetValue(“WarehouseCode$”, newWhse)
                                        retVal = oLines.Write()
                        End If

                        retVal = oLines.MoveNext()

            Loop

    End If

End If
```

How does this work?

To better understand the logic behind the script’s functionality, read this:

“Init VARs”
In the first section, we “initialize” (init) the variables to prepare for use later in the script.

“IF/THEN/ELSE”
Through these stated conditions, we run our main logic. In this particular script, we want one of the conditions to be “existing order.”

How do we find out if the order exists?

We do this by checking the “edit state” as indicated by the following values:

2 = new order
1 = existing order
0 = no order on the screen

Next, to find out if you have landed on a regular inventory item, follow these steps:

1. Go to the Lines object. Move to the first line and start a loop.
2. Check the value of the “Item Type” field.
3. If it is a regular inventory item, set it with the new value and then write the line. Continue until all the lines are completed.

You’re almost done!

Now you can accept the script. This will return you to the “Add Script” window.

1. To close this screen and return to the “User Defined Scripts” window, click “OK.”
2. Next, to go back to the main “User Defined Field” and “Table Maintenance” window, click “Close.” Here, you must also click “Close” **again** to see the “Script Compile” window.
3. IMPORTANT: When the “Script Compile” window appears, ***click the “Compile” button***.

Note that you can further edit the script, if necessary. To do so, from the “Custom Office/Main” menu, choose the “User Defined Script Maintenance” task.

This is the first of many scripting examples that we will share through the xkzero blog. If you need help with scripting or other technical issues with your Sage ERP, no matter how complex, please feel free to contact us:

xkzero Technical Services
Email: info@xkzero.com
Call: 847-416-2009

This entry was posted in **Sage 100 ERP**, **xkzero Technical Services** and tagged **Alnoor**, **Alnoor Cassim**, **Sage 100 ERP**, **scripting**, **warehouse line automation** by **Alnoor Cassim**. Bookmark the **permalink [http://erpappsblog.com/?p=1359]** .

About Alnoor Cassim

Director, xkzero Technical Services for Sage 100 ERP. Providing companies solutions to utilize and enhance their ERP software to grow their business, identify and create solutions that streamline business processes, improve productivity, increase profitability, and provide win-win scenarios for all.
View all posts by Alnoor Cassim →