**Table of Contents**

**OPEN2\_TEST.vbs 2**

Option Explicit 2

Sub Main() | End Sub 2

“Oracle” Connection 2

“Business Rule” Connection 3

“Debug” Connection 3

CleanUp 3

OPEN2\_TEST.vbs File 4

**NEW\_PACKAGE2\_TEST.vbs 5**

Option Explicit 5

Sub Main() | End Sub 5

ShowWindow 5 - 11

Window Scripting Host 11

Subroutines with Arguments 11

**START\_SHIP2\_TEST.vbs 12**

Subroutines with Arguments 0

**OPEN2\_TEST.vbs**

“OPEN2\_TEST.vbs” is the first script that is executed once the “ConnectShip” application is launched. The function of this script is to open the database connection to Oracle, the “Business Rules” database and the “Debug” database. For tracking each script easily, the first line will identify the name of the file, second line shows the last time the script was modified followed by the third line briefly displaying the purpose. The following is what it currently looks like:

**1 | ' Document Name: OPEN2\_TEST.vbs**

**2 | ' Last Modified: 03/17/2017**

**3 | ' Purpose: Open Database connection to Oracle and Business Rules**

In every script, the statement “Option Explicit” is the first line of code. It prevents the accidental reuse of the name of a previously declared variable. If a declared variable is mistype or used while being undeclared, an error message is generated. This is essential for debugging purposes.

**5 | Option Explicit**

For the script to run successfully, there has to be at least one “Main” procedure. To show when the process should finish, the “End” procedure is placed at the end of all code. This is shown below:

**7 | Sub Main()**

**... |**

**36 | End Sub**

It is of best practice to include comments on major functions for quick and efficient modification of the script. The following section written inside the “Sub Main()” and “End Sub” procedures. “Dim” is used for declaring the variables: “objDB”, “objDD”, “conString”, “dd\_ConnStr”. “objDB” is set to create an open connection to a data source, allowing to access and manipulate the database.

**8 | ' Connection to Oracle**

**9 | Dim objDB, objDD, conString, dd\_ConnStr**

**10 | Set objDB = CreateObject( "ADODB.Connection" )**

To access an individual database, the “credentials” should be stated in the “conString” variable. Once stated, the connection is opened with the connection string and stored in the “ScriptDataManager.StoredData(“PBD\_CONNECTION”)”. This is stored in the “objDB” variable. Please note the following:

Provider = MSDASQL;

DSN = IBMDA720train;

UID = UPSCSTST;

PWD = CSTS031808

**12 | conString = "Provider=MSDASQL;DSN=IBMDA720train;uid=UPSCSTST;pwd=CSTS031808"**

**13 | objDB.Open conString**

**14 | ScriptDataManager.StoredData("PBD\_CONNECTION") = objDB**

The process repeats for the connection to the “Business Rules” database. The connection is opened with the connection string and stored in the “ScriptDataManager.StoredData(“SQL\_CONNECTION”)”. This is stored in the “sqlobjDB” variable. Please note the following:

Provider = SQLOLEDB;

Data Source = qts-sql2008;

Initial Catalog = CSW\_Ship\_Rules;

User Id = ‘connectship’;

Password = ‘c0nn3ct5h1p#’

**16 | ' Connect to Business Rules database**

**17 | Dim sqlobjDB, sqlconstring**

**18 | Set sqlobjDB = CreateObject("ADODB.Connection")**

**20 | sqlconstring = "Provider=SQLOLEDB.1;Data Source=qts-sql2008;Initial Catalog=CSW\_SHIP\_Rules;user id='connectship'; password='c0nn3ct5h1p#'"**

**21 | sqlobjDB.Open sqlconstring**

**22 | ScriptDataManager.StoredData("SQL\_CONNECTION") = sqlobjDB**

The process repeats for the connection to the “Debug” database. The connection is opened with the connection string and stored in the “ScriptDataManager.StoredData(“DEBUG\_CONNECTION”)”. This is stored in the “debugobjDB” variable. Please note the following:

Provider = SQLOLEDB.1;

Data Source = qts-csdev;

Initial Catalog = CSW\_Logs;

User Id = ‘connectship\_test’;

Password = ‘connectship’

**24 | ' Connect to Debug database**

**25 | Dim debugobjDB, debugconstring**

**26 | Set debugobjDB = CreateObject("ADODB.Connection")**

**28 | debugconstring = "Provider=SQLOLEDB.1;Data Source=qts-csdev;Initial Catalog=CSW\_Logs;user id='connectship\_test'; password='connectship'"**

**29 | debugobjDB.Open debugconstring**

**30 | ScriptDataManager.StoredData("DEBUG\_CONNECTION") = debugobjDB**

Finally, CleanUp is used to reset all variables and release any used memory:

**32 | ' Cleanup**

**33 | Set objDB = Nothing**

**34 | Set sqlobjDB = Nothing**

**35 | Set debugobjDB = Nothing**

The whole script document should look like on the next page:

1. **' Document Name: OPEN2\_TEST.vbs**
2. **' Last Modified: 03/17/2017**
3. **' Purpose: Open Database connection to Oracle and Business Rules**
4. **Option Explicit**
5. **Sub Main()**
6. **' Connection to Oracle**
7. **Dim objDB, objDD, conString, dd\_ConnStr**
8. **Set objDB = CreateObject( "ADODB.Connection" )**
9. **conString = "Provider=MSDASQL;DSN=IBMDA720train;uid=UPSCSTST;pwd=CSTS031808"**
10. **objDB.Open conString**
11. **ScriptDataManager.StoredData("PBD\_CONNECTION") = objDB**
12. **' Connect to Business Rules database**
13. **Dim sqlobjDB, sqlconstring**
14. **Set sqlobjDB = CreateObject("ADODB.Connection")**
15. **sqlconstring = "Provider=SQLOLEDB.1;Data Source=qts-sql2008;Initial Catalog=CSW\_SHIP\_Rules;user id='connectship'; password='c0nn3ct5h1p#'"**
16. **sqlobjDB.Open sqlconstring**
17. **ScriptDataManager.StoredData("SQL\_CONNECTION") = sqlobjDB**
18. **' Connect to Debug database**
19. **Dim debugobjDB, debugconstring**
20. **Set debugobjDB = CreateObject("ADODB.Connection")**
21. **debugconstring = "Provider=SQLOLEDB.1;Data Source=qts-csdev;Initial Catalog=CSW\_Logs;user id='connectship\_test'; password='connectship'"**
22. **debugobjDB.Open debugconstring**
23. **ScriptDataManager.StoredData("DEBUG\_CONNECTION") = debugobjDB**
24. **' Cleanup**
25. **Set objDB = Nothing**
26. **Set sqlobjDB = Nothing**
27. **Set debugobjDB = Nothing**
28. **End Sub**

**NEW\_PACKAGE2\_TEST.vbs**

“NEW\_PACKAGE2\_TEST.vbs” is the second script that is executed after shipping an order. The function of this script is show the appropriate windows in the application and clear any data from previous shipment. For tracking each script easily, the first line will identify the name of the file, second line shows the last time the script was modified followed by the third line briefly displaying the purpose. The following is what it currently looks like:

**1 | ' Document Name: NEW\_PACKAGE2\_TEST.vbs**

**2 | ' Last Modified: 03/17/2017**

**3 | ' Purpose: Show appropriate windows in the application and clear any data from previous shipment.**

In every script, the statement “Option Explicit” is the first line of code. It prevents the accidental reuse of the name of a previously declared variable. If a declared variable is mistype or used while being undeclared, an error message is generated. This is essential for debugging purposes.

**5 | Option Explicit**

For the script to run successfully, there has to be at least one “Main” procedure. To show when the process should finish, the “End” procedure is placed at the end of all code. This is shown below:

**7 | Sub Main()**

**... |**

**40 | End Sub**

The “ConnectShip” application shows multiple windows with specific details of each package. Due to one being accidentally closed, this script shows/reopens the dialog boxes as shown below:

**8 | showWindow "CONSIGNEE\_WINDOW", "", True**

**9 | showWindow "LTL\_WINDOW", "", True**

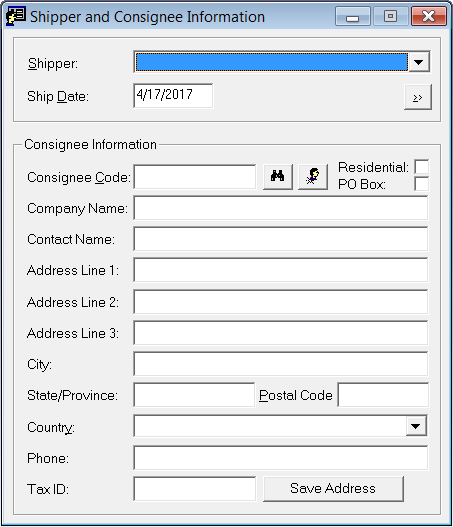
**10 | showWindow "PRIMARY\_SUBCATEGORY\_WINDOW", "", True**

**11 | showWindow "RATE\_RESULTS\_WINDOW", "", True**

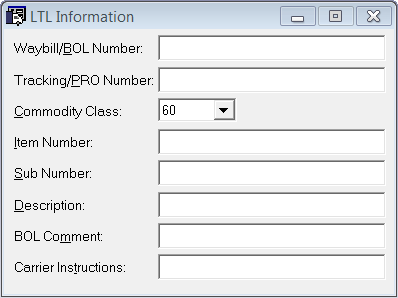
**12 | showWindow "PACKAGE\_WINDOW", "", True**

Pictures of the windows are shown in the next pages:

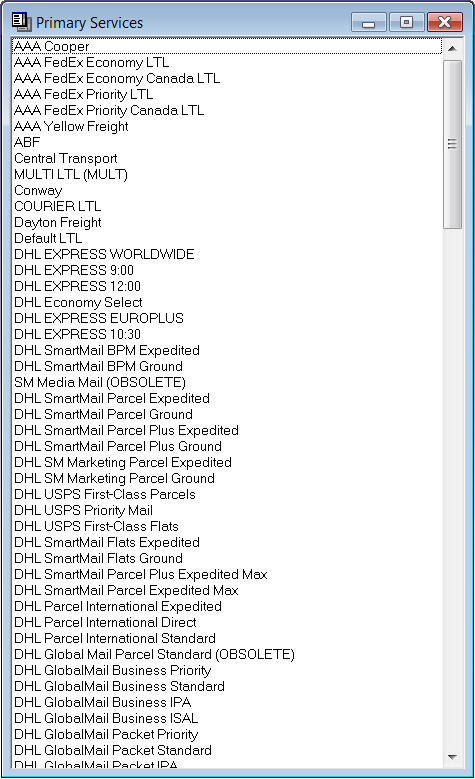
**8 | showWindow "CONSIGNEE\_WINDOW", "", True**



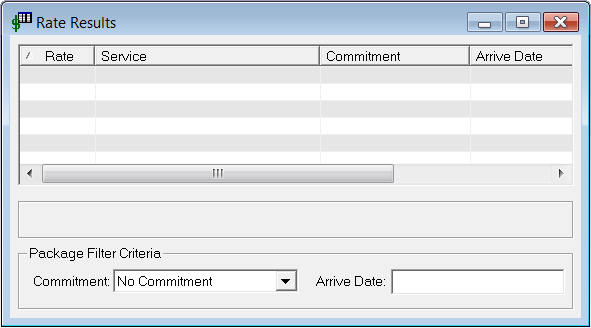
**9 | showWindow "LTL\_WINDOW", "", True**



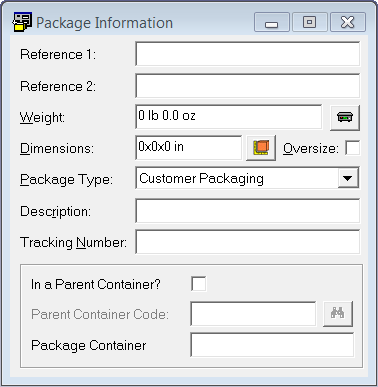
**10 | showWindow "PRIMARY\_SUBCATEGORY\_WINDOW", "", True**



**11 | showWindow "RATE\_RESULTS\_WINDOW", "", True**



**12 | showWindow "PACKAGE\_WINDOW", "", True**



The script continues with declarations of “WshShell” and “WshSysEnv” which allow the reference of user or account on the computer. The “CreateObject()” allows the creation of a Windows Scripting Host (WSH) automation object. “Wsh.Environment()” gains access to the environment variables, or details of the computer in which the script is executing. In this instance, it is used to store the “USERNAME” stored in the declared “windowUserName” variable shown below:

**14 | ‘ Set the Windows Username**

**15 | Dim WshShell**

**16 | Dim WshSysEnv**

**17 |**

**18 | Set WshShell = CreateObject(“WScript.Shell”)**

**19 | Set WshSysEnv = WshShell.Environment(“PROCESS”)**

**20 |**

**21 | Dim windowUserName**

**22 | windowUserName = WshSysEnv(“USERNAME”)**

Once the previous code has been executed, the following can be used to fill in the textbox with the user details for our records. “REFERENCE\_10” refers to the user logged in the computer/manifest station.

**24 | setField "REFERENCE\_10", windowsUserName, True, 0**

TODO: Look into why the following code is there:

**26 | ScriptDataManager.StoredData("Consignee") = False**

**27 | ScriptDataManager.StoredData("TPB") = False**

As mentioned before, the declared variables are then destroyed to release used memory.

**29 | Set WshShell = Nothing**

**30 | Set WshSysEnv = Nothing**

TODO: Look into why the following code is there:

**32 | ScriptDataManager.StoredData("MYVALUE") = 0**

TODO: Look into why the following code is there:

**34 | ' Weigh Package**

**35 | Dim objWMacro**

**36 | Set objWMacro = CreateObject("Progistics.Dictionary")**

**37 | objWMacro.Value("NAME") = "MACRO\_WEIGH"**

**38 | ScriptDataManager.AddMacro objWMacro**

**39 | Set objWMacro = Nothing**

The previous code concludes the Main procedure. In most of the scripts, there exists subroutine and functions that allow the code to be read efficiently and cuts down on the number of lines of code as well. A “Sub” procedure is a series of statements inside the “Sub” and “End Sub” statements that execute but do not return a value. The “Function” procedure is exactly the same as the “Sub” procedure, but it can return a value. Both can take arguments such as constants, variables, or expressions that are passed by a calling procedure. In the case below, we have the “Sub” procedure called “showWindow” with arguments listed as: “window”, “field”, and “suppresScript”.

**42 | Sub showWindow(window, field, suppresScript)**

**43 | Dim macro**

**44 | Set macro = CreateObject("Progistics.Dictionary")**

**45 | macro.Value("NAME") = "MACRO\_VIEW\_WINDOW"**

**46 | macro.Value("WINDOW\_NAME") = window**

**47 | macro.Value("FIELD\_NAME") = field**

**48 | macro.Value("SUPPRESS\_SCRIPT") = suppresScript**

**49 | ScriptDataManager.AddMacro macro**

**50 | Set macro = Nothing**

**51 | End Sub**

Lastly, we finish the “NEW\_PACKAGE2\_TEST.vbs” script with the “Sub” procedure called “setfield” with arguments listed as: “fieldName”, “fieldValue”, “suppressScript”, and “fieldIndex”.

**53 | Sub setField(fieldName, fieldValue, suppressScript, fieldIndex)**

**54 | Dim macro**

**55 | Set macro = CreateObject("Progistics.Dictionary")**

**56 | macro.Value("NAME") = "MACRO\_SET\_FIELD"**

**57 | macro.Value("WINDOW\_NAME") = fieldName**

**58 | macro.Value("FIELD\_NAME") = fieldValue**

**59 |**

**60 | If fieldName > 0 Then**

**61 | macro.Value(“FIELD\_INDEX”) = fieldIndex**

**62 | End If**

**63 |**

**64 | macro.Value("SUPPRESS\_SCRIPT") = suppressScript**

**65 | ScriptDataManager.AddMacro macro**

**66 | Set macro = Nothing**

**67 | End Sub**

The whole script document should look like on the next two pages:

**1 | ' Document Name: NEW\_PACKAGE2\_TEST.vbs**

**2 | ' Last Modified: 03/17/2017**

**3 | ' Purpose: Show appropriate windows in the application and clear any data from previous shipment.**

**4 |**

**5 | Option Explicit**

**6 |**

**7 | Sub Main()**

**8 | showWindow "CONSIGNEE\_WINDOW", "", True**

**9 | showWindow "LTL\_WINDOW", "", True**

**10 | showWindow "PRIMARY\_SUBCATEGORY\_WINDOW", "", True**

**11 | showWindow "RATE\_RESULTS\_WINDOW", "", True**

**12 | showWindow "PACKAGE\_WINDOW", "", True**

**13 |**

**14 | ‘ Set the Windows Username**

**15 | Dim WshShell**

**16 | Dim WshSysEnv**

**17 |**

**18 | Set WshShell = CreateObject(“WScript.Shell”)**

**19 | Set WshSysEnv = WshShell.Environment(“PROCESS”)**

**20 |**

**21 | Dim windowUserName**

**22 | windowUserName = WshSysEnv(“USERNAME”)**

**23 |**

**24 | setField "REFERENCE\_10", windowsUserName, True, 0**

**25 |**

**26 | ScriptDataManager.StoredData("Consignee") = False**

**27 | ScriptDataManager.StoredData("TPB") = False**

**28 |**

**29 | Set WshShell = Nothing**

**30 | Set WshSysEnv = Nothing**

**31 |**

**32 | ScriptDataManager.StoredData("MYVALUE") = 0**

**33 |**

**34 | ' Weigh Package**

**35 | Dim objWMacro**

**36 | Set objWMacro = CreateObject("Progistics.Dictionary")**

**37 | objWMacro.Value("NAME") = "MACRO\_WEIGH"**

**38 | ScriptDataManager.AddMacro objWMacro**

**39 | Set objWMacro = Nothing**

**40 | End Sub**

**41 |**

**42 | Sub showWindow(window, field, suppresScript)**

**43 | Dim macro**

**44 | Set macro = CreateObject("Progistics.Dictionary")**

**45 | macro.Value("NAME") = "MACRO\_VIEW\_WINDOW"**

**46 | macro.Value("WINDOW\_NAME") = window**

**47 | macro.Value("FIELD\_NAME") = field**

**48 | macro.Value("SUPPRESS\_SCRIPT") = suppresScript**

**49 | ScriptDataManager.AddMacro macro**

**50 | Set macro = Nothing**

**51 | End Sub**

**52 |**

**53 | Sub setField(fieldName, fieldValue, suppressScript, fieldIndex)**

**54 | Dim macro**

**55 | Set macro = CreateObject("Progistics.Dictionary")**

**56 | macro.Value("NAME") = "MACRO\_SET\_FIELD"**

**57 | macro.Value("WINDOW\_NAME") = fieldName**

**58 | macro.Value("FIELD\_NAME") = fieldValue**

**59 |**

**60 | If fieldName > 0 Then**

**61 | macro.Value(“FIELD\_INDEX”) = fieldIndex**

**62 | End If**

**63 |**

**64 | macro.Value("SUPPRESS\_SCRIPT") = suppressScript**

**65 | ScriptDataManager.AddMacro macro**

**66 | Set macro = Nothing**

**67 | End Sub**

**START\_SHIP2\_TEST.vbs**

“START\_SHIP2\_TEST.vbs” is the first script that is executed when shipping an order. The function of this script is prompt the user to scan the Carton Id. Here would also go all the code that would need to be executed before the “PRE\_SHIP” script. For tracking each script easily, the first line will identify the name of the file, second line shows the last time the script was modified followed by the third line briefly displaying the purpose. The following is what it currently looks like:

**1 | ' Document Name: START\_SHIP2\_TEST.vbs**

**2 | ' Last Modified: 03/17/2017**

**3 | ' Purpose: Business Rules that are triggered before "PreShip"**

In every script, the statement “Option Explicit” is the first line of code. It prevents the accidental reuse of the name of a previously declared variable. If a declared variable is mistype or used while being undeclared, an error message is generated. This is essential for debugging purposes.

**5 | Option Explicit**

This is the first script that declares the variables outside if the “Sub” procedure. This allows the variables to be global, or used and multiple procedures.

**7 | Dim context, currentPackage, PRIMARY\_SUBCATEGORY**

For the script to run successfully, there has to be at least one “Main” procedure. To show when the process should finish, the “End” procedure is placed at the end of all code. This is shown below:

**9 | Sub Main()**

**... |**

**18 | End Sub**

This script inside the “Sub” procedure known as “Main” starts setting “context” and “currentPackage” with values as shown below. The “PRIMARY\_SUBCATEGORY” refers to the way of shipping while the “CURRENT\_PACKAGE” points to the order that was scanned ready for shipping.

**10 | Set context = ScriptDataManager.ClientContext**

**11 | Set currentPackage = context(“CURRENT\_PACKAGE”)**

**12 | PRIMARY\_SUBCATEGORY = currentPackage(“PRIMARY\_SUBCATEGORY”)**

When modifications are done in any code, it is very important to document validating that it will not affect other components. Below is live example of such, Project #7377 was created and urged to alter this script prompting the user to scan the Carton Id with stored data “AMAZON” and being shipped via “LTL”.

**14 | ‘ Project 7377 prompt user to scan Carton Id**

**15 | If (ScriptDataManager.StoredData(“AMAZON”) = “Y” and InStr(PRIMARY\_CATEGORY, “LTL”)) Then**

**16 | cartonIDs**

**17 | End If**

“CartonIDs” is calling the subroutine from the “Main” procedure. You can just specify the name of the procedure when calling it, or include “Call” before the name. If “Call” is included before the name of the procedure, then the name has to include after it parenthesis where the arguments would be inserted. For exmaple:

**16 | Call cartonIDs()**

The previous code concludes the Main procedure. In most of the scripts, there exists subroutine and functions that allow the code to be read efficiently and cuts down on the number of lines of code as well. A “Sub” procedure is a series of statements inside the “Sub” and “End Sub” statements that execute but do not return a value. The “Function” procedure is exactly the same as the “Sub” procedure, but it can return a value. Both can take arguments such as constants, variables, or expressions that are passed by a calling procedure.

**20 | Sub cartonIDs()**

**21 | Dim Dim dictFormData, dictUserInput, cartonID, csutils, strTitle**

**22 |**

**23 | Set csutils = CreateObject(“CSWUtils.Utils”)**

**24 | Set dictFormData = CreateObject(“Progistics.Dictionary”)**

**25 |**

**26 | dictFormData(“FIELD1\_CAPTION”) = “Carton ID”**

**27 | dictFormData(“FIELD1\_DEFAULT”) = “”**

**28 | strTitle = “Carton ID”**

**29 |**

**30 | Set dictUserInput = csutils.UserInput(strTitle, dictFormData)**

**31 | Set dictFormData = Nothing**

**32 | cartonID = dictUserInput(“FIELD1\_DEFAULT”)**

**33 | setField “REFERENCE\_16” cartonID, True, 0**

**34 | Set csutils = Nothing**

**35 | End Sub**

The whole script document should look like on the next page:

**1 | ' Document Name: START\_SHIP2\_TEST.vbs**

**2 | ' Last Modified: 03/17/2017**

**3 | ' Purpose: Business Rules that are triggered before "PreShip"**

**4 |**

**5 | Option Explicit**

**6 |**

**7 | Dim context, currentPackage, PRIMARY\_SUBCATEGORY**

**8 |**

**9 | Sub Main()**

**10 | Set context = ScriptDataManager.ClientContext**

**11 | Set currentPackage = context(“CURRENT\_PACKAGE”)**

**12 | PRIMARY\_SUBCATEGORY = currentPackage(“PRIMARY\_SUBCATEGORY”)**

**13 |**

**14 | ‘ Project 7377 prompt user to scan Carton Id**

**15 | If (ScriptDataManager.StoredData(“AMAZON”) = “Y” and InStr(PRIMARY\_CATEGORY, “LTL”)) Then**

**16 | cartonIDs**

**17 | End If**

**18 | End Sub**

**19 |**

**20 | Sub cartonIDs()**

**21 | Dim Dim dictFormData, dictUserInput, cartonID, csutils, strTitle**

**22 |**

**23 | Set csutils = CreateObject(“CSWUtils.Utils”)**

**24 | Set dictFormData = CreateObject(“Progistics.Dictionary”)**

**25 |**

**26 | dictFormData(“FIELD1\_CAPTION”) = “Carton ID”**

**27 | dictFormData(“FIELD1\_DEFAULT”) = “”**

**28 | strTitle = “Carton ID”**

**29 |**

**30 | Set dictUserInput = csutils.UserInput(strTitle, dictFormData)**

**31 | Set dictFormData = Nothing**

**32 | cartonID = dictUserInput(“FIELD1\_DEFAULT”)**

**33 | setField “REFERENCE\_16” cartonID, True, 0**

**34 | Set csutils = Nothing**

**35 | End Sub**

**PRE\_SHIP2\_TEST.vbs**

“PRE\_SHIP2\_TEST.vbs” is the second script that is executed when shipping an order. The function of this script applies business rules immediately before a package is shipped. For tracking each script easily, the first line will identify the name of the file, second line shows the last time the script was modified followed by the third line briefly displaying the purpose. The following is what it currently looks like:

**1 | ' Document Name: PRE\_SHIP2\_TEST.vbs**

**2 | ' Last Modified: 03/17/2017**

**3 | ' Purpose: This script applies business rules immediately before a package is shipped.**

In every script, the statement “Option Explicit” is the first line of code. It prevents the accidental reuse of the name of a previously declared variable. If a declared variable is mistype or used while being undeclared, an error message is generated. This is essential for debugging purposes.

**5 | Option Explicit**

This is the first script that declares the variables outside if the “Sub” procedure. This allows the variables to be global, or used and multiple procedures.

**7 | ' Used to Populate variables from ConnectShip UI**

**8 | Dim REFERENCE\_1, REFERENCE\_2, REFERENCE\_3, REFERENCE\_4, REFERENCE\_5, REFERENCE\_6, REFERENCE\_7, REFERENCE\_8, REFERENCE\_9, REFERENCE\_10**

**9 | Dim REFERENCE\_15, REFERENCE\_17, REFERENCE\_19**

**10 | Dim user, WORLDEASEPRINT, shipDate, WEIGHT, PRIMARY\_SUBCATEGORY, PROOF, COMMODITY\_CLASS, CONSIGNEE, customsValue**

**11 | Dim NOFN\_SEQUENCE, needByDate, TERMS, CONSIGNEE\_ACCOUNT, TPB\_ACCOUNT, TPB\_ADDRESS, TPB\_TF, ORDER\_TAKEN\_BY, SHIPPER\_ID**

**13 | Dim context, currentPackage, sc, package, contents, content, NumofPkgs, NEW\_TRACKING\_NUMBER, SHIPPER, macro**

**14 | Dim MAIL\_SERVICEOPTION, HARMONIZED\_CODE, AES\_TRANSACTION\_NUMBER, SED\_EXEMPTION\_NUMBER**

**15 | Dim shipDateToUse, currPkgWeight, dblMyWeight, errMsg**

For the script to run successfully, there has to be at least one “Main” procedure. To show when the process should finish, the “End” procedure is placed at the end of all code. This is shown below:

**17 | Sub Main()**

**... |**

**266 | End Sub**

This script inside the “Sub” procedure known as “Main” starts setting “context” and “currentPackage” with values as shown below. “sc” has now stored the “shipped via” variable and the “NumofPkgs” has the total number of packages for this one order:

**18 | Set context = ScriptDataManager.ClientContext**

**19 | Set currentPackage = context(“CURRENT\_PACKAGE”)**

**20 | sc = currentPackage(“PRIMARY\_CATEGORY”)**

**21 | NumofPkgs = currentPackage(“NOFN\_TOTAL”)**

Now comes the long list of variables that are used to populate the ConnectShip User Interface:

**24 | user = LCase(currentPackage(“USER”)) ' Login Name**

**25 | REFERENCE\_1 = currentPackage(“REFERENCE\_1”) ' Order/Pick List #**

**26 | REFERENCE\_2 = currentPackage(“REFERENCE\_2”) ' Account/Business #**

**27 | REFERENCE\_3 = currentPackage(“REFERENCE\_3”) '**

**28 | REFERENCE\_3 = Trim(REFRENCE\_3) '**

**29 | REFERENCE\_4 = currentPackage(“REFERENCE\_4”) '**

**30 | REFERENCE\_5 = currentPackage(“REFERENCE\_5”) ' PBD**

**31 | REFERENCE\_6 = currentPackage(“REFERENCE\_6”) ' “REFERENCE\_1”**

**32 | REFERENCE\_7 = currentPackage(“REFERENCE\_7”) ' “REFERENCE\_2”**

**33 | REFERENCE\_8 = currentPackage(“REFERENCE\_8”) '**

**34 | REFERENCE\_9 = currentPackage(“REFERENCE\_9”) ' “REFERENCE\_5”**

**35 | REFERENCE\_10 = currentPackage(“REFERENCE\_10”) ' USER**

**36 | REFERENCE\_15 = currentPackage(“REFERENCE\_15”) '**

**37 | REFERENCE\_17 = currentPackage(“REFERENCE\_17”) ' International?**

**38 | REFERENCE\_17 = Trim(REFRENCE\_17) '**

**39 | REFERENCE\_19 = currentPackage(“REFERENCE\_19”) '**

**40 | WORLDEASEPRINT = currentPackage(“REFERENCE\_18”) '**

**41 | WORLDEASEPRINT = Trim(WORLDEASEPRINT) '**

**42 | shipDate = currentPackage(“SHIPDATE”) ' Date of Shipment**

**43 | WEIGHT = currentPackage(“N”) ' WEIGHT**

**44 | PRIMARY\_SUBCATEGORY = currentPackage(“N”) ' Form of Shipment**

**45 | PROOF = currentPackage(“N”) ' Signature required**

**46 | COMMODITY\_CLASS = currentPackage(“N”) ' COMMODITY\_CLASS**

**47 | Set CONSIGNEE = currentPackage(“N”) '**

**48 | customsValue = currentPackage(“N”) ' Custom Value**

**49 | NOFN\_SEQUENCE = currentPackage(“N”) ' Current Package #**

**50 | needByDate = currentPackage(“N”) '**

**51 | TERMS = currentPackage(“N”) ' Terms**

**52 | CONSIGNEE\_ACCOUNT = currentPackage(“N”) ' Consignee Account #**

**53 | TPB\_ACCOUNT = currentPackage(“N”) ' 3rd Party Billing Acc.**

**54 | Set TPB\_ADDRESS = currentPackage(“N”) ' 3rd Party Billing Addr.**

**55 | TPB\_TF = currentPackage(“N”) ' 3rd Party Billing**

**56 | ORDER\_TAKEN\_BY = currentPackage(“N”) ' Who took order**

**57 | SHIPPER\_ID = currentPackage(“N”) ' Who will ship order**

The next section verifies if FedEx order is billable, which if any were false, the script would hault:

**59 | ' Verify Billable FedEx Fields**

**60 | If Not IsFedExBillable(currentPackage(“SHIPPER”), REFERENCE\_2, PRIMARY\_SUBCATEGORY, TERMS, CONSIGNEE\_ACCOUNT, TPB\_ADDRESS, TPB\_ACCOUNT, TPB\_TF) Then**

**61 | Exit Sub**

**62 | End If**

After the last If Then Statement, there is a check to si if the weight is valid. The “IsNumeric” verifies the value of “WEIGHT” if it only contains numeric characters:

**64 | ' Check that weight is valid**

**65 | If (IsNumeric(WEIGHT) and (WEIGHT > 0 and WEIGHT <= 50000)) Then**

**66 | dblMyWeight = CDbl(WEIGHT)**

**67 | Else**

**68 | errMsg = "Invalid Weight: Weight cannot be negative or greater than 50000 pounds."**

**69 | setError(errMsg)**

**70 | End If**

Proceeding, a check is made to see if the order is set to ship to any restricted countries:

**72 | ' Do not ship to the following restricted countries:**

**73 | If (CONSIGNEE.CountrySymbol = "CUBA" Or CONSIGNEE.CountrySymbol = "IRAN" Or CONSIGNEE.CountrySymbol = "KOREA\_NORTH" Or CONSIGNEE.CountrySymbol = "BURMA" Or CONSIGNEE.CountrySymbol = "MYANMAR" Or CONSIGNEE.CountrySymbol = "SUDAN" Or CONSIGNEE.CountrySymbol = "SYRIA") Then**

**74 | seterror("Shipment denied to a restricted country: " & CONSIGNEE.CountrySymbol & "." & vbCr & vbLf & vbCr & vbLf & "Please return pick ticket to client rep." & vbCr & vbLf & vbCr & vbLf )**

**75 | End If**

Below, in reference to the Project #7121, If a package being shipped via “USPS First Class Mail” is over “13 oz”, then it will be switched over the “USPS Priority”:

**77 | ' Per Project 7121: Switch USPS First Class Mail shipments over 13 oz to USPS Priority**

**78 | If (PRIMARY\_SUBCATEGORY = "TANDATA\_USPS.USPS.FIRST") and (dblMyWeight > .8125) Then**

**79 | setfield "PRIMARY\_SUBCATEGORY", "TANDATA\_USPS.USPS.PRIORITY", False, 0**

**80 | End If**

For all “UPS Mail Innovations (UPSMI BPM)” shipments greater than 1 pound will be switched over to ship via “UPSMI Expedited Parcel Lightweight”:

**82 | ' Set all UPSMI BPM shipments < 1 pound to ship UPSMI Expedited Parcel Lightweight**

**83 | If ((PRIMARY\_SUBCATEGORY = "CONNECTSHIP\_UPSMAILINNOVATIONS.UPS.BPM") And (dblMyWeight < 1.00)) Then**

**84 | setfield "PRIMARY\_SUBCATEGORY", "CONNECTSHIP\_UPSMAILINNOVATIONS.UPS.EPD", False, 0**

**85 | End If**

For all “UPSMI Expedited” shipments greater than 1 pound will be switched over to ship via “UPSMI Media Mail”:

**87 | ' For shipments over 1 lb, switch from UPSMI Expedited / Parcel Select to UPSMI Media Mail**

**88 | If ((PRIMARY\_SUBCATEGORY = "CONNECTSHIP\_UPSMAILINNOVATIONS.UPS.EPD" or PRIMARY\_SUBCATEGORY = "CONNECTSHIP\_UPSMAILINNOVATIONS.UPS.PS") and dblMyWeight >= 1.00) Then**

**89 | setfield "PRIMARY\_SUBCATEGORY", "CONNECTSHIP\_UPSMAILINNOVATIONS.UPS.MEDIA", False, 0**

**90 | setfield "PROOF", True, True, 0**

**91 | setfield "TRACKING\_NUMBER", "", True, 0**

**92 | End If**

For “FedEx Ground” shipments under “71 pounds” going to residential, switch to “FedEx Home Delivery” per request on Ticket #43004”

**94 | ' For shipments over 1 lb, switch from UPSMI Expedited / Parcel Select to UPSMI Media Mail**

**95 | If (PRIMARY\_SUBCATEGORY = "TANDATA\_FEDEXFSMS.FEDEX.GND" And CONSIGNEE.residential = True And dblMyWeight < 71) Then**

**96 | setfield "PRIMARY\_SUBCATEGORY", "TANDATA\_FEDEXFSMS.FEDEX.FHD", False, 0**

**97 | End If**

The next line calls the subroutine called “clientBusinessRules()”:

**102 | clientBusinessRules()**

If no service were to be selected and attempted to ship, an error would be generated and prompt to select a shipping service:

**106 | ' For shipments over 1 lb, switch from UPSMI Expedited / Parcel Select to UPSMI Media Mail**

**107 | If (PRIMARY\_SUBCATEGORY = “”) Then**

**108 | setError("No service was selected. Please select a service.")**

**109 | Exit Sub**

**110 | End If**

The next line calls the subroutine called “weightCheck()” only if validateWeight was processed and accepted pre request on Project #7469:

**112 | ' Weight check (Project #7469)**

**113 | If (ScriptDataManager.StoredData("validateWeight") = True) Then**

**114 | weightCheck()**

**115 | End If**

Proceeding this is the preparation of documents and labels to print for “WorldEase” or other shipments:

**117 | ' WorldEase documents**

**118 | If (WORLDEASEPRINT = "WORLDEASE") Then**

**119 | setDocumentsWE PRIMARY\_SUBCATEGORY, CONSIGNEE**

**120 | Else**

**121 | setDocuments PRIMARY\_SUBCATEGORY, CONSIGNEE**

**122 | setPaperless PRIMARY\_SUBCATEGORY, CONSIGNEE**

**123 | End If**

The following section functions to carry the LTL Pro Number over from previous package and populate LTL Bol:

**125 | ' Carry LTL pro number over from previous package and populate LTL bol**

**126 | If (InStr(currentPackage("PRIMARY\_SUBCATEGORY"), "TANDATA\_LTL")) Then**

**127 | Dim pro, bol, lastPackage**

**128 | pro = currentPackage(“TRACKING\_NUMBER”)**

**129 | pro = Trim(pro)**

**130 | bol= currentPackage(“WAYBILL\_BOL\_NUMBER”)**

**131 | bol = Trim(bol)**

**132 |**

**133 | If (currentPackage("NOFN\_SEQUENCE") > 1) Then**

**134 | Set lastPackage = context("LAST\_PACKAGE")**

**135 | If (pro = "") Then**

**136 | pro = lastPackage("TRACKING\_NUMBER")**

**137 | Set macro = CreateObject("Progistics.Dictionary")**

**138 | macro.Value("NAME") = "MACRO\_SET\_FIELD"**

**139 | macro.Value("FIELD\_NAME") = "TRACKING\_NUMBER"**

**140 | macro.Value("FIELD\_VALUE") = pro**

**141 | macro.Value("SUPPRESS\_SCRIPT") = True**

**142 | ScriptDataManager.AddMacro macro**

**143 | Set macro = Nothing**

**144 | End If**

**145 | End If**

**146 |**

**147 | If (bol = "") Then**

**148 | Set macro = CreateObject("Progistics.Dictionary")**

**149 | macro.Value("NAME") = "MACRO\_SET\_FIELD"**

**150 | macro.Value("FIELD\_NAME") = "WAYBILL\_BOL\_NUMBER"**

**151 | macro.Value("FIELD\_VALUE") = pro**

**152 | macro.Value("SUPPRESS\_SCRIPT") = True**

**153 | ScriptDataManager.AddMacro macro**

**154 | Set macro = Nothing**

**155 | End If**

**156 | End If**

Following is to generate tracking numbers for “UMI shipments” (14-15 character string: POSTAL\_CODE(5) + REFERENCE\_2(2) + REFERENCE\_1(7-8))”

**158 | ' Generate Tracking Numbers for UMI shipments - 14-15 character string: POSTAL\_CODE (5) + REFERENCE\_2 (2) + REFERENCE\_1 (7-8)**

**159 | If (PRIMARY\_SUBCATEGORY = "TANDATA\_MISC.OMS.OMS") Then**

**160 | Dim currentTrackingNumber, newTrackingNumber, postalCodeVal, ref1Val, ref2Val, consigneePack**

**161 | currentTrackingNumber = currentPackage("TRACKING\_NUMBER")**

**162 | currentTrackingNumber = Trim(currentTrackingNumber)**

**163 |**

**164 | ref1Val = currentPackage("REFERENCE\_1")**

**165 | ref2Val = currentPackage("REFERENCE\_2")**

**166 | postalCodeVal = CONSIGNEE.POSTALCODE**

**167 |**

**168 | newTrackingNumber = Left(postalCodeVal, 5) & ref2Val & ref1Val**

**169 | If (currentTrackingNumber <> newTrackingNumber) Then**

**170 | setfield "TRACKING\_NUMBER", newTrackingNumber, True, 0**

**171 | End If**

**172 | End If**

The If Then Statement checks to see if a “TRACKING\_NUMBER” exists for the order scanned. If non exists, then an error is generated:

**174 | ' Tracking number should not be blank as it is normally scanned in by the user from a book of PRO numbers**

**175 | If (PRIMARY\_SUBCATEGORY = "TANDATA\_LTL.FXNL.FXNL" Or PRIMARY\_SUBCATEGORY = "TANDATA\_LTL.FXFE.FXFE") Then**

**176 | currentTrackingNumber = currentPackage("TRACKING\_NUMBER")**

**177 | If (currentTrackingNumber = "") Then**

**178 | setError ("Error: Tracking Number is required for this LTL shipment.")**

**179 | End If**

**180 | End If**

Per Project #6737, it requires Washington to use Endicia component:

**182 | ' Per Project 6737: Require Washington to use Endicia component**

**183 | If (InStr(PRIMARY\_SUBCATEGORY, "TANDATA\_USPS.USPS")) Then**

**184 | If (user = "philly") or (user= "wdc") Then**

**185 | seterror("\*Error\* Regular USPS services not allowed." & vbCr & vbLf & "Please use Endicia USPS services." & vbCr & vbLf & "" )**

**186 | End If**

**187 | End If**

For FedEx Smart Post, we are setting the tracking number to blank:

**189 | ' FedEx Smart Post - set tracking # to blank**

**190 | If ((PRIMARY\_SUBCATEGORY = "TANDATA\_MISC.POST.SMPT") and (currentPackage("TRACKING\_NUMBER") <> "")) Then**

**191 | setfield "TRACKING\_NUMBER", "", False, 0**

**192 | End If**

The following section sets “PROOF” to false for “Generic Label” shipments:

**194 | ' Set proof to false for "Generic Label" shipments**

**195 | If (InStr(PRIMARY\_SUBCATEGORY, "GSLB")) Then**

**196 | setfield "PROOF", False, False, 0**

**197 | End If**

For all “UPSMI Expedited” shipments, set the “NONMACHINABLE” flag to “False” and “PROOF” to “True”:

**199 | ' For all UPSMI Expedited shipments, set the NonMachinable flag to false and proof to true**

**200 | If (PRIMARY\_SUBCATEGORY = "CONNECTSHIP\_UPSMAILINNOVATIONS.UPS.EPD") Then**

**201 | setField "NONMACHINABLE\_MAIL", False, True, 0**

**202 | setfield "PROOF", True, False, 0**

**203 | End If**

Prevents domestic “UPS” packages from being shipped to PO Box

**205 | ' Prevent domestic UPS packages from being shipped to PO Box**

**206 | Dim tempAddress, Utils, retval, retval2**

**207 | tempAddress = UCase(CONSIGNEE.ADDRESS1)**

**208 | tempAddress = Replace (tempAddress, " ","")**

**209 | If (InStr(tempAddress, "POBOX") and (InStr(PRIMARY\_SUBCATEGORY, "MISC") or InStr(PRIMARY\_SUBCATEGORY, "USPS") or InStr(PRIMARY\_SUBCATEGORY, "EPD") or InStr(PRIMARY\_SUBCATEGORY, "MEDIA") or InStr(PRIMARY\_SUBCATEGORY, "UPSMAILINNOVATIONS"))) Then**

**210 | End Sub**

**211 | End If**

**212 |**

**213 | If (InStr(tempAddress, "POBOX") and InStr(PRIMARY\_SUBCATEGORY, "UPS") and CONSIGNEE.CountrySymbol = "UNITED\_STATES") Then**

**214 | Set Utils = CreateObject(“CSWUtils.Utils”)**

**215 | retval = Utils.Biginform(“Is this shipping to a PO BOX?", "PO BOX Verification”, 4, 0)**

**216 | If (retval = 6) Then**

**217 | setError("UPS does not ship domestic packages to a PO BOX")**

**218 | Set Utils = Nothing**

**219 | End Sub**

**220 | Else**

**221 | Set Utils = Nothing**

**222 | End Sub**

**223 | End If**

**224 | End If**

Remind user to add consignee phone # to Invoice/Waybill for international PO BOX

**226 | ' Remind user to add consignee phone # to invoice/waybill for international PO BOX**

**227 | If (InStr(tempAddress, "POBOX") and InStr(PRIMARY\_SUBCATEGORY, "UPS") and (CONSIGNEE.CountrySymbol = "UNITED\_ARAB\_EMIRATES" or CONSIGNEE.CountrySymbol = "AFGHANISTAN" or CONSIGNEE.CountrySymbol = "BAHRAIN" or CONSIGNEE.CountrySymbol = "BURUNDI" or CONSIGNEE.CountrySymbol = "IRAN" or CONSIGNEE.CountrySymbol = "JORDAN" or CONSIGNEE.CountrySymbol = "KUWAIT" or CONSIGNEE.CountrySymbol = "LEBANON" or CONSIGNEE.CountrySymbol = "OMAN" or CONSIGNEE.CountrySymbol = "QATAR" or CONSIGNEE.CountrySymbol = "SAUDI\_ARABIA" or CONSIGNEE.CountrySymbol = "SYRIA" or CONSIGNEE.CountrySymbol = "YEMEN" or CONSIGNEE.CountrySymbol = "ZAMBIA")) Then**

**228 | Set Utils = CreateObject("CSWUtils.Utils")**

**229 | retval2 = Utils.Biginform("Is this shipping to a International PO BOX?", "International PO BOX Verification", 4, 0)**

**230 | If (retval2 = 6) Then**

**231 | MsgBox("You are shipping to a International PO Box! Please verify that consignee phone number is on the order or Invoice or Waybill")**

**232 | Set Utils = Nothing**

**233 | Else**

**234 | Set Utils = Nothing**

**235 | Exit Sub**

**236 | End If**

**237 | End If**

Set harmonized Code

**239 | ' Set harmonized Code**

**240 | If ((InStr(PRIMARY\_SUBCATEGORY, "USPS.I\_PRIORITY") or InStr(PRIMARY\_SUBCATEGORY, "USPS.I\_FIRST") or InStr(PRIMARY\_SUBCATEGORY, "UPSMAILINNOVATIONS.UPS.ECO")) and trim(HARMONIZED\_CODE) = "N/A") Then**

**241 | HARMONIZED\_CODE = "4901990050"**

**242 | End If**

As per request on Ticket #44322 accounting for user error, the following locks down certain carriers to only ship collect.

**244 | ' \*\*\*Accounting for user error\*\*\***

**245 | ' Ticket #44322: Lock down certain carriers to only ship collect**

**246 | If (PRIMARY\_SUBCATEGORY = "TANDATA\_LTL.SAIA.SAIA" or PRIMARY\_SUBCATEGORY = "TANDATA\_LTL.RLC.RLC" or PRIMARY\_SUBCATEGORY = "TANDATA\_LTL.ESTS.ESTS" or PRIMARY\_SUBCATEGORY = "TANDATA\_LTL.AAAC.AAAC" or PRIMARY\_SUBCATEGORY = "TANDATA\_LTL.SEF.SEF" or PRIMARY\_SUBCATEGORY = "TANDATA\_LTL.USFH.USFH" or PRIMARY\_SUBCATEGORY = "TANDATA\_LTL.OLDD.OLDD" or PRIMARY\_SUBCATEGORY = "TANDATA\_LTL.CNWY.CNWY" or PRIMARY\_SUBCATEGORY = "TANDATA\_LTL.CT.CT" or PRIMARY\_SUBCATEGORY = "TANDATA\_LTL.DAYF.DAYF" or PRIMARY\_SUBCATEGORY = "TANDATA\_LTL.NEMF.NEMF" or PRIMARY\_SUBCATEGORY = "TANDATA\_LTL.NEWP.NEWP") Then**

**247 | setfield "TERMS", "CONSIGNEE", False, 0**

**248 | End If**

If Third Party Billing data was passed from Oracle and user unchecks TPB box before shipping, show error and do not allow user to ship per Ticket #42426:

**250 | ' If Third Party Billing data was passed from Oracle and user unchecks TPB box before shipping, show error and do not allow user to ship (Ticket #42426)**

**251 | If (ScriptDataManager.StoredData("TPB") = True and TPB\_TF = False) Then**

**252 | setError("Third Party Information was provided. Please check the 3rd Party Billing Box")**

**253 | ScriptDataManager.StoredData("TPB") = False**

**254 | Exit Sub**

**255 | End If**

If consignee account is passed from Oracle, check that the account number is filled in as per Ticket #42426:

**257 | ' If consignee account is passed from Oracle, check that the account number is filled in (Ticket #42426)**

**258 | If (ScriptDataManager.StoredData("Consignee") = True and CONSIGNEE\_ACCOUNT = "") Then**

**259 | setError("Consignee account was provided. Please make sure the account data has been entered.")**

**260 | ScriptDataManager.StoredData("Consignee") = False**

**261 | Exit Sub**

**262 | End If**

The “Main” procedure ense with a call to the “WritePreShipDebug()” function:

**264 | WritePreShipDebug()**

The following subroutine called “setDocuments()” sets all labels to not print unless conditions is true. This allows the correct label to be printed out:

**268 | Sub setDocuments(PRIMARY\_SUBCATEGORY, CONSIGNEE)**

**269 | Dim category, REFERENCE\_2**

**270 | Dim comInv\_document\_print, domestic\_document\_print, intl\_document\_print**

**271 | Dim comInv\_document\_format, domestic\_document\_format, intl\_document\_format**

**272 |**

**273 | If (CONSIGNEE.CountrySymbol = "UNITED\_STATES") Then**

**274 | comInv\_document\_print = False**

**275 | domestic\_document\_print = True**

**276 | intl\_document\_print = False**

**277 | Else**

**278 | comInv\_document\_print = True**

**279 | domestic\_document\_print = False**

**280 | intl\_document\_print = True**

**281 | End If**

**282 |**

**283 | comInv\_document\_format = “”**

**284 | domestic\_document\_format = “”**

**285 | intl\_document\_format = “”**

**286 |**

**287 | category = getCategory(PRIMARY\_SUBCATEGORY)**

**288 | REFERENCE\_2 = currentPackage("REFERENCE\_2")**

**290 | If (REFERENCE\_2 = "27" and comInv\_document\_print = True and intl\_document\_print = True) Then**

**291 | Dim EXpna**

**292 | Set EXpna = CreateObject("Progistics.NameAddress")**

**293 | EXpna.Company = “Customer Service - Kimberly Gates”**

**294 | EXpna.Contact = “Distribution Center”**

**295 | EXpna.Address1 = “3280 Summit Ridge Pkwy”**

**296 | EXpna.City = “Duluth”**

**297 | EXpna.StateProvince = “GA”**

**298 | EXpna.PostalCode = “30096-0000”**

**299 | EXpna.Phone = “16785391152”**

**300 | EXpna.CountrySymbol = “UNITED\_STATES”**

**301 | currentPackage(“EXPORTER”) = EXPna**

**302 | End If**

**304 | Set macro = CreateObject(“Progistics.Dictionary”)**

**305 | macro.Value(“NAME”) = “MACRO\_MODIFY\_DOCUMENT\_ATTRIBUTES”**

**306 | macro.Value(“CATEGORY”) = category**

**307 | macro.Value(“DOCUMENT\_FORMAT”) = “CONNECTSHIP\_UPS\_MAXICODE\_PACKAGE\_LABEL.STANDARD”**

**308 | macro.Value(“PRINT”) = False**

**309 | macro.Value(“VIEW\_DOCUMENT\_MANAGER”) = False**

**310 | ScriptDataManager.AddMacro macro**

**311 | Set marco = Nothing**

**313 | Set macro = CreateObject(“Progistics.Dictionary”)**

**314 | macro.Value(“NAME”) = “MACRO\_MODIFY\_DOCUMENT\_ATTRIBUTES”**

**315 | macro.Value(“CATEGORY”) = category**

**316 | macro.Value(“DOCUMENT\_FORMAT”) = “CUSTOM\_TANDATA\_USPS\_LABEL.MMS”**

**317 | macro.Value(“PRINT”) = False**

**318 | macro.Value(“VIEW\_DOCUMENT\_MANAGER”) = False**

**319 | ScriptDataManager.AddMacro macro**

**320 | Set marco = Nothing**

**322 | Set macro = CreateObject(“Progistics.Dictionary”)**

**323 | macro.Value(“NAME”) = “MACRO\_MODIFY\_DOCUMENT\_ATTRIBUTES”**

**324 | macro.Value(“CATEGORY”) = category**

**325 | macro.Value(“DOCUMENT\_FORMAT”) = “CUSTOM\_TANDATA\_USPS\_EMM\_LABEL.STANDARD”**

**326 | macro.Value(“PRINT”) = False**

**327 | macro.Value(“VIEW\_DOCUMENT\_MANAGER”) = False**

**328 | ScriptDataManager.AddMacro macro**

**329 | Set marco = Nothing**

**331 | Set macro = CreateObject(“Progistics.Dictionary”)**

**332 | macro.Value(“NAME”) = “MACRO\_MODIFY\_DOCUMENT\_ATTRIBUTES”**

**333 | macro.Value(“CATEGORY”) = category**

**334 | macro.Value(“DOCUMENT\_FORMAT”) = “USPS\_2976.STANDARD”**

**335 | macro.Value(“PRINT”) = False**

**336 | macro.Value(“VIEW\_DOCUMENT\_MANAGER”) = False**

**337 | ScriptDataManager.AddMacro macro**

**338 | Set marco = Nothing**

**340 | Set macro = CreateObject(“Progistics.Dictionary”)**

**341 | macro.Value(“NAME”) = “MACRO\_MODIFY\_DOCUMENT\_ATTRIBUTES”**

**342 | macro.Value(“CATEGORY”) = category**

**343 | macro.Value(“DOCUMENT\_FORMAT”) = “USPS\_2976A.STANDARD”**

**344 | macro.Value(“PRINT”) = False**

**345 | macro.Value(“VIEW\_DOCUMENT\_MANAGER”) = False**

**346 | ScriptDataManager.AddMacro macro**

**347 | Set marco = Nothing**

**349 | Set macro = CreateObject(“Progistics.Dictionary”)**

**350 | macro.Value(“NAME”) = “MACRO\_MODIFY\_DOCUMENT\_ATTRIBUTES”**

**351 | macro.Value(“CATEGORY”) = category**

**352 | macro.Value(“DOCUMENT\_FORMAT”) = “TANDATA\_USPS\_FORM2976.STANDARD”**

**353 | macro.Value(“PRINT”) = False**

**354 | macro.Value(“VIEW\_DOCUMENT\_MANAGER”) = False**

**355 | ScriptDataManager.AddMacro macro**

**356 | Set marco = Nothing**

**358 | Set macro = CreateObject(“Progistics.Dictionary”)**

**359 | macro.Value(“NAME”) = “MACRO\_MODIFY\_DOCUMENT\_ATTRIBUTES”**

**360 | macro.Value(“CATEGORY”) = category**

**361 | macro.Value(“DOCUMENT\_FORMAT”) = “TANDATA\_USPS\_FORM2976A.STANDARD”**

**362 | macro.Value(“PRINT”) = False**

**363 | macro.Value(“VIEW\_DOCUMENT\_MANAGER”) = False**

**364 | ScriptDataManager.AddMacro macro**

**365 | Set marco = Nothing**

**367 | Set macro = CreateObject(“Progistics.Dictionary”)**

**368 | macro.Value(“NAME”) = “MACRO\_MODIFY\_DOCUMENT\_ATTRIBUTES”**

**369 | macro.Value(“CATEGORY”) = category**

**370 | macro.Value(“DOCUMENT\_FORMAT”) = “CONNECTSHIP\_UPSMAILINNOVATIONS\_USPS\_FORM2976.STANDARD”**

**371 | macro.Value(“PRINT”) = False**

**372 | macro.Value(“VIEW\_DOCUMENT\_MANAGER”) = False**

**373 | ScriptDataManager.AddMacro macro**

**374 | Set marco = Nothing**

**376 | Set macro = CreateObject(“Progistics.Dictionary”)**

**377 | macro.Value(“NAME”) = “MACRO\_MODIFY\_DOCUMENT\_ATTRIBUTES”**

**378 | macro.Value(“CATEGORY”) = category**

**379 | macro.Value(“DOCUMENT\_FORMAT”) = “CONNECTSHIP\_UPSMAILINNOVATIONS\_USPS\_FORM2976A.STANDARD”**

**380 | macro.Value(“PRINT”) = False**

**381 | macro.Value(“VIEW\_DOCUMENT\_MANAGER”) = False**

**382 | ScriptDataManager.AddMacro macro**

**383 | Set marco = Nothing**

**385 | Set macro = CreateObject(“Progistics.Dictionary”)**

**386 | macro.Value(“NAME”) = “MACRO\_MODIFY\_DOCUMENT\_ATTRIBUTES”**

**387 | macro.Value(“CATEGORY”) = category**

**388 | macro.Value(“DOCUMENT\_FORMAT”) = “TANDATA\_COMMERCIAL\_INVOICE.STANDARD”**

**389 | macro.Value(“PRINT”) = False**

**390 | macro.Value(“VIEW\_DOCUMENT\_MANAGER”) = False**

**391 | ScriptDataManager.AddMacro macro**

**392 | Set marco = Nothing**

**394 | Set macro = CreateObject(“Progistics.Dictionary”)**

**395 | macro.Value(“NAME”) = “MACRO\_MODIFY\_DOCUMENT\_ATTRIBUTES”**

**396 | macro.Value(“CATEGORY”) = category**

**397 | macro.Value(“DOCUMENT\_FORMAT”) = “CUSTOM\_TANDATA\_COMMERCIAL\_INVOICE\_PACKAGES.STANDARD”**

**398 | macro.Value(“PRINT”) = False**

**399 | macro.Value(“VIEW\_DOCUMENT\_MANAGER”) = False**

**400 | ScriptDataManager.AddMacro macro**

**401 | Set marco = Nothing**

**403 | Set macro = CreateObject(“Progistics.Dictionary”)**

**404 | macro.Value(“NAME”) = “MACRO\_MODIFY\_DOCUMENT\_ATTRIBUTES”**

**405 | macro.Value(“CATEGORY”) = category**

**406 | macro.Value(“DOCUMENT\_FORMAT”) = “CUSTOM\_TANDATA\_USPS\_LABEL\_RTRN\_ADDR.STANDARD”**

**407 | macro.Value(“PRINT”) = False**

**408 | macro.Value(“VIEW\_DOCUMENT\_MANAGER”) = False**

**409 | ScriptDataManager.AddMacro macro**

**410 | Set marco = Nothing**

**412 | Set macro = CreateObject(“Progistics.Dictionary”)**

**413 | macro.Value(“NAME”) = “MACRO\_MODIFY\_DOCUMENT\_ATTRIBUTES”**

**414 | macro.Value(“CATEGORY”) = category**

**415 | macro.Value(“DOCUMENT\_FORMAT”) = “EVENT\_SUMMARY C.STANDARD”**

**416 | macro.Value(“PRINT”) = False**

**417 | macro.Value(“VIEW\_DOCUMENT\_MANAGER”) = False**

**418 | ScriptDataManager.AddMacro macro**

**419 | Set marco = Nothing**

**421 | Set macro = CreateObject(“Progistics.Dictionary”)**

**422 | macro.Value(“NAME”) = “MACRO\_MODIFY\_DOCUMENT\_ATTRIBUTES”**

**423 | macro.Value(“CATEGORY”) = category**

**424 | macro.Value(“DOCUMENT\_FORMAT”) = “EVENT\_SUMMARY R.STANDARD”**

**425 | macro.Value(“PRINT”) = False**

**426 | macro.Value(“VIEW\_DOCUMENT\_MANAGER”) = False**

**427 | ScriptDataManager.AddMacro macro**

**428 | Set marco = Nothing**

**430 | Set macro = CreateObject(“Progistics.Dictionary”)**

**431 | macro.Value(“NAME”) = “MACRO\_MODIFY\_DOCUMENT\_ATTRIBUTES”**

**432 | macro.Value(“CATEGORY”) = category**

**433 | macro.Value(“DOCUMENT\_FORMAT”) = “CONNECTSHIP\_UPSMAILINNOVATIONS\_LABEL.STANDARD”**

**434 | macro.Value(“PRINT”) = False**

**435 | macro.Value(“VIEW\_DOCUMENT\_MANAGER”) = False**

**436 | ScriptDataManager.AddMacro macro**

**437 | Set marco = Nothing**

**439 | Set macro = CreateObject(“Progistics.Dictionary”)**

**440 | macro.Value(“NAME”) = “MACRO\_MODIFY\_DOCUMENT\_ATTRIBUTES”**

**441 | macro.Value(“CATEGORY”) = category**

**442 | macro.Value(“DOCUMENT\_FORMAT”) = “TANDATA\_LTL\_BOL.STANDARD”**

**443 | macro.Value(“PRINT”) = False**

**444 | macro.Value(“VIEW\_DOCUMENT\_MANAGER”) = False**

**445 | ScriptDataManager.AddMacro macro**

**446 | Set marco = Nothing**

**448 | Set macro = CreateObject(“Progistics.Dictionary”)**

**449 | macro.Value(“NAME”) = “MACRO\_MODIFY\_DOCUMENT\_ATTRIBUTES”**

**450 | macro.Value(“CATEGORY”) = category**

**451 | macro.Value(“DOCUMENT\_FORMAT”) = “WF\_BOL.STANDARD”**

**452 | macro.Value(“PRINT”) = False**

**453 | macro.Value(“VIEW\_DOCUMENT\_MANAGER”) = False**

**454 | ScriptDataManager.AddMacro macro**

**455 | Set marco = Nothing**

**457 | Set macro = CreateObject(“Progistics.Dictionary”)**

**458 | macro.Value(“NAME”) = “MACRO\_MODIFY\_DOCUMENT\_ATTRIBUTES”**

**459 | macro.Value(“CATEGORY”) = category**

**460 | macro.Value(“DOCUMENT\_FORMAT”) = “DHL\_HOLSTED\_FRAGILE\_LABEL.STANDARD”**

**461 | macro.Value(“PRINT”) = False**

**462 | macro.Value(“VIEW\_DOCUMENT\_MANAGER”) = False**

**463 | ScriptDataManager.AddMacro macro**

**464 | Set marco = Nothing**

**466 | Set macro = CreateObject(“Progistics.Dictionary”)**

**467 | macro.Value(“NAME”) = “MACRO\_MODIFY\_DOCUMENT\_ATTRIBUTES”**

**468 | macro.Value(“CATEGORY”) = category**

**469 | macro.Value(“DOCUMENT\_FORMAT”) = “DHL\_HOLSTED\_NON\_FRAGILE\_LABEL.STANDARD”**

**470 | macro.Value(“PRINT”) = False**

**471 | macro.Value(“VIEW\_DOCUMENT\_MANAGER”) = False**

**472 | ScriptDataManager.AddMacro macro**

**473 | Set marco = Nothing**

**475 | Set macro = CreateObject(“Progistics.Dictionary”)**

**476 | macro.Value(“NAME”) = “MACRO\_MODIFY\_DOCUMENT\_ATTRIBUTES”**

**477 | macro.Value(“CATEGORY”) = category**

**478 | macro.Value(“DOCUMENT\_FORMAT”) = “CONNECTSHIP\_DHLGLOBALMAIL\_DOMESTIC\_LABEL.STANDARD”**

**479 | macro.Value(“PRINT”) = False**

**480 | macro.Value(“VIEW\_DOCUMENT\_MANAGER”) = False**

**481 | ScriptDataManager.AddMacro macro**

**482 | Set marco = Nothing**

**484 | Set macro = CreateObject(“Progistics.Dictionary”)**

**485 | macro.Value(“NAME”) = “MACRO\_MODIFY\_DOCUMENT\_ATTRIBUTES”**

**486 | macro.Value(“CATEGORY”) = category**

**487 | macro.Value(“DOCUMENT\_FORMAT”) = “TANDATA\_COMMERCIAL\_INVOICE\_PACKAGES.STANDARD”**

**488 | macro.Value(“PRINT”) = False**

**489 | macro.Value(“VIEW\_DOCUMENT\_MANAGER”) = False**

**490 | ScriptDataManager.AddMacro macro**

**491 | Set marco = Nothing**

**493 | ' SUREPOST LABEL ASSIGNMENT**

**494 | If (category = "CONNECTSHIP\_UPS.UPS") Then**

**495 | comInv\_document\_format = "CUSTOM\_TANDATA\_COMMERCIAL\_INVOICE\_PACKAGES.STANDARD"**

**496 | domestic\_document\_format = "PBD\_CUSTOM\_UPS\_DOM\_LABEL.STANDARD"**

**497 | intl\_document\_format = "PBD\_CUSTOM\_UPS\_INTL\_LABEL.STANDARD"**

**498 |**

**499 | If (PRIMARY\_SUBCATEGORY = "CONNECTSHIP\_UPS.UPS.SPPS" and (REFERENCE\_2 = "65" or REFERENCE\_2 = "1A")) Then**

**500 | Set macro = CreateObject("Progistics.Dictionary")**

**501 | macro.Value("NAME") = "MACRO\_MODIFY\_DOCUMENT\_ATTRIBUTES"**

**502 | macro.Value("CATEGORY") = category**

**503 | macro.Value("DOCUMENT\_FORMAT") = "PBD\_CUSTOM\_UPS\_DOM\_LABEL.STANDARD"**

**504 | macro.Value("PRINT") = True**

**505 | macro.Value("VIEW\_DOCUMENT\_MANAGER") = False**

**506 | ScriptDataManager.AddMacro macro**

**507 | Set macro = Nothing**

**508 | End If**

**509 | End If**

**511 | If (category = "TANDATA\_DHL.DHL") Then**

**512 | comInv\_document\_format = "CUSTOM\_TANDATA\_COMMERCIAL\_INVOICE\_PACKAGES.STANDARD"**

**513 | domestic\_document\_format = ""**

**514 | intl\_document\_format = ""**

**515 | End If**

**517 | If (category = "TANDATA\_USPS.USPS" and InStr(PRIMARY\_SUBCATEGORY, "USPS.I\_PRIORITY") > 0) Then**

**518 | Set macro = CreateObject("Progistics.Dictionary")**

**519 | macro.Value("NAME") = "MACRO\_MODIFY\_DOCUMENT\_ATTRIBUTES"**

**520 | macro.Value("CATEGORY") = category**

**521 | macro.Value("DOCUMENT\_FORMAT") = "USPS\_2976A.STANDARD"**

**522 | macro.Value("PRINT") = True**

**523 | macro.Value("VIEW\_DOCUMENT\_MANAGER") = False**

**524 | ScriptDataManager.AddMacro macro**

**525 | Set macro = Nothing**

**526 | End If**

**528 | If (category = "TANDATA\_USPS.USPS" and InStr(PRIMARY\_SUBCATEGORY, "USPS.I\_FIRST") > 0) Then**

**529 | Set macro = CreateObject("Progistics.Dictionary")**

**530 | macro.Value("NAME") = "MACRO\_MODIFY\_DOCUMENT\_ATTRIBUTES"**

**531 | macro.Value("CATEGORY") = category**

**532 | macro.Value("DOCUMENT\_FORMAT") = "USPS\_2976.STANDARD"**

**533 | macro.Value("PRINT") = True**

**534 | macro.Value("VIEW\_DOCUMENT\_MANAGER") = False**

**535 | ScriptDataManager.AddMacro macro**

**536 | Set macro = Nothing**

**537 | End If**

**539 | If (category = "TANDATA\_DHL.DHL") Then**

**540 | comInv\_document\_format = "CUSTOM\_TANDATA\_COMMERCIAL\_INVOICE\_PACKAGES.STANDARD"**

**541 | domestic\_document\_format = ""**

**542 | intl\_document\_format = ""**

**453 | End If**

**545 | ' International commercial invoice**

**546 | If (comInv\_document\_format <> "") Then**

**547 | Set macro = CreateObject("Progistics.Dictionary")**

**548 | macro.Value("NAME") = "MACRO\_MODIFY\_DOCUMENT\_ATTRIBUTES"**

**549 | macro.Value("CATEGORY") = category**

**550 | macro.Value("DOCUMENT\_FORMAT") = comInv\_document\_format**

**551 | macro.Value("PRINT") = comInv\_document\_print**

**552 | macro.Value("VIEW\_DOCUMENT\_MANAGER") = False**

**553 | ScriptDataManager.AddMacro macro**

**554 | Set macro = Nothing**

**555 | End If**

**557 | ' International Package Document**

**558 | If (intl\_document\_format <> "") Then**

**559 | Set macro = CreateObject("Progistics.Dictionary")**

**560 | macro.Value("NAME") = "MACRO\_MODIFY\_DOCUMENT\_ATTRIBUTES"**

**561 | macro.Value("CATEGORY") = category**

**562 | macro.Value("DOCUMENT\_FORMAT") = intl\_document\_format**

**563 | macro.Value("PRINT") = intl\_document\_print**

**564 | macro.Value("VIEW\_DOCUMENT\_MANAGER") = False**

**565 | ScriptDataManager.AddMacro macro**

**566 | Set macro = Nothing**

**567 | End If**

**569 | ' Domestic Package Document**

**570 | If (domestic\_document\_format <> "") Then**

**571 | Set macro = CreateObject("Progistics.Dictionary")**

**572 | macro.Value("NAME") = "MACRO\_MODIFY\_DOCUMENT\_ATTRIBUTES"**

**573 | macro.Value("CATEGORY") = category**

**574 | macro.Value("DOCUMENT\_FORMAT") = domestic\_document\_format**

**575 | macro.Value("PRINT") = domestic\_document\_print**

**576 | macro.Value("VIEW\_DOCUMENT\_MANAGER") = False**

**577 | ScriptDataManager.AddMacro macro**

**578 | Set macro = Nothing**

**579 | End If**

**581 | ' OVERRIDE THE DOCUMENT ASSIGNMENTS SO THE CUSTOM LABEL FOR AMA LABEL PRINTS FOR AMA ORDERS**

**582 | If (PRIMARY\_SUBCATEGORY = "TANDATA\_USPS.USPS.SPCL" And REFERENCE\_2 = "44") Then**

**583 | Set macro = CreateObject("Progistics.Dictionary")**

**584 | macro.Value("NAME")= "MACRO\_MODIFY\_DOCUMENT\_ATTRIBUTES"**

**585 | macro.Value("CATEGORY") = category**

**586 | macro.Value("DOCUMENT\_FORMAT") = "CUSTOM\_TANDATA\_USPS\_LABEL.MMS"**

**587 | macro.Value("PRINT") = True**

**588 | macro.Value("VIEW\_DOCUMENT\_MANAGER") = False**

**589 | ScriptDataManager.AddMacro macro**

**590 | Set macro = Nothing**

**591 | End If**

**593 | ' PRINT PRYOR EVENT RETURN ENVELOPE LABEL**

**594 | If (ScriptDataManager.StoredData("EVENTLABEL") = True) Then**

**595 | Set macro = CreateObject("Progistics.Dictionary")**

**596 | macro.Value("NAME") = "MACRO\_MODIFY\_DOCUMENT\_ATTRIBUTES"**

**597 | macro.Value("CATEGORY") = category**

**598 | macro.Value("DOCUMENT\_FORMAT") = "EVENT\_SUMMARY R.STANDARD"**

**599 | macro.Value("COPIES") = 1**

**600 | macro.Value("PRINT") = True**

**601 | macro.Value("VIEW\_DOCUMENT\_MANAGER") = False**

**602 | ScriptDataManager.AddMacro macro**

**603 | Set macro = Nothing**

**604 | End If**

**606 | ' PRINT PRYOR EVENT CARTON LABEL**

**607 | If (ScriptDataManager.StoredData("EVENTLABEL") = True) Then**

**608 | Set macro = CreateObject("Progistics.Dictionary")**

**609 | macro.Value("NAME") = "MACRO\_MODIFY\_DOCUMENT\_ATTRIBUTES"**

**610 | macro.Value("CATEGORY") = category**

**611 | macro.Value("DOCUMENT\_FORMAT") = "EVENT\_SUMMARY C.STANDARD"**

**612 | macro.Value("COPIES") = currentPackage("NOFN\_TOTAL")**

**613 | macro.Value("PRINT") = True**

**614 | macro.Value("VIEW\_DOCUMENT\_MANAGER") = False**

**615 | ScriptDataManager.AddMacro macro**

**616 | Set macro = Nothing**

**617 | End If**

**619 | If (Instr(category, "LTL") and REFERENCE\_2 = "2Y") Then**

**620 | Set macro = CreateObject("Progistics.Dictionary")**

**621 | macro.Value("NAME") = "MACRO\_MODIFY\_DOCUMENT\_ATTRIBUTES"**

**622 | macro.Value("CATEGORY") = sc**

**623 | macro.Value("DOCUMENT\_FORMAT") = "WF\_BOL.STANDARD"**

**624 | macro.Value("COPIES") = 1**

**625 | macro.Value("PRINT") = True**

**626 | ScriptDataManager.AddMacro macro**

**627 | Set macro = Nothing**

**628 | End If**

**630 | If (REFERENCE\_2 = "3F") Then**

**631 | If (PRIMARY\_SUBCATEGORY = "CONNECTSHIP\_DHLGLOBALMAIL.DHL.PS\_EXP") Then**

**632 | Set macro = CreateObject("Progistics.Dictionary")**

**633 | macro.Value("NAME") = "MACRO\_MODIFY\_DOCUMENT\_ATTRIBUTES"**

**634 | macro.Value("CATEGORY") = category**

**635 | macro.Value("DOCUMENT\_FORMAT") = "DHL\_HOLSTED\_FRAGILE\_LABEL.STANDARD"**

**636 | macro.Value("PRINT") = True**

**637 | macro.Value("VIEW\_DOCUMENT\_MANAGER") = False**

**638 | ScriptDataManager.AddMacro macro**

**639 | Set macro = Nothing**

**640 | Else**

**641 | Set macro = CreateObject("Progistics.Dictionary")**

**642 | macro.Value("NAME") = "MACRO\_MODIFY\_DOCUMENT\_ATTRIBUTES"**

**643 | macro.Value("CATEGORY") = category**

**644 | macro.Value("DOCUMENT\_FORMAT") = "DHL\_HOLSTED\_NON\_FRAGILE\_LABEL.STANDARD"**

**645 | macro.Value("PRINT") = True**

**646 | macro.Value("VIEW\_DOCUMENT\_MANAGER") = False**

**647 | ScriptDataManager.AddMacro macro**

**648 | Set macro = Nothing**

**649 | End If**

**650 | End If**

**651 | End Sub**

Below is the subroutine called “setPaperless” with both “PRIMARY\_SUBCATEGORY” and “CONSIGNEE” arguments:

**653 | Sub setPaperless(PRIMARY\_SUBCATEGORY, CONSIGNEE)**

**654 | Set context = ScriptDatamanager.ClientContext**

**655 | Set currentPackage = context("CURRENT\_PACKAGE")**

**656 | sc = currentPackage("PRIMARY\_CATEGORY")**

**657 |**

**658 | Dim category, user2, REFERENCE\_17**

**659 | category = getCategory(PRIMARY\_SUBCATEGORY)**

**660 |**

**661 | ' Used to turn off Commercial Invoice printing for Alpha Users**

**662 | user2 = currentPackage("USER")**

**663 | user2 = LCase(user2)**

**664 |**

**665 | REFERENCE\_17 = currentPackage("REFERENCE\_17")**

**666 | REFERENCE\_17 = Trim(REFERENCE\_17)**

**667 |**

**668 | If (CONSIGNEE.CountrySymbol <> "UNITED\_STATES" And REFERENCE\_17 = "YES" And InStr(PRIMARY\_SUBCATEGORY, "CONNECTSHIP\_UPS")) Then**

**669 | ' COMMERCIAL\_INVOICE\_METHOD**

**670 | paperless\_on**

**671 | End If**

**672 |**

**673 | If (CONSIGNEE.CountrySymbol <> "UNITED\_STATES" And REFERENCE\_17 = "NO" And InStr(PRIMARY\_SUBCATEGORY, "CONNECTSHIP\_UPS")) Then**

**674 | Set macro = CreateObject("Progistics.Dictionary")**

**675 | macro.Value("NAME") = "MACRO\_MODIFY\_DOCUMENT\_ATTRIBUTES"**

**676 | macro.Value("CATEGORY") = category**

**677 | macro.Value("DOCUMENT\_FORMAT") = "CUSTOM\_TANDATA\_COMMERCIAL\_INVOICE\_PACKAGES.STANDARD"**

**678 | macro.Value("PRINT") = True**

**679 | macro.Value("VIEW\_DOCUMENT\_MANAGER") = False**

**680 | ScriptDataManager.AddMacro macro**

**681 | Set macro = Nothing**

**682 |**

**683 | Set macro = CreateObject("Progistics.Dictionary")**

**684 | macro.Value("NAME") = "MACRO\_MODIFY\_DOCUMENT\_ATTRIBUTES"**

**685 | macro.Value("CATEGORY") = category**

**686 | macro.Value("DOCUMENT\_FORMAT") = "TANDATA\_COMMERCIAL\_INVOICE\_PACKAGES.STANDARD"**

**687 | macro.Value("PRINT") = True**

**688 | macro.Value("VIEW\_DOCUMENT\_MANAGER") = False**

**689 | ScriptDataManager.AddMacro macro**

**690 | Set macro = Nothing**

**691 |**

**692 | Set macro = CreateObject("Progistics.Dictionary")**

**693 | macro.Value("NAME") = "MACRO\_MODIFY\_DOCUMENT\_ATTRIBUTES"**

**694 | macro.Value("CATEGORY") = category**

**695 | macro.Value("DOCUMENT\_FORMAT") = "TANDATA\_COMMERCIAL\_INVOICE.STANDARD\_1"**

**696 | macro.Value("PRINT") = True**

**697 | macro.Value("VIEW\_DOCUMENT\_MANAGER") = False**

**698 | ScriptDataManager.AddMacro macro**

**699 | Set macro = Nothing**

**700 | End If**

**701 | End Sub**

Next comes the subroutine called “setDocumentsWE” with the same arguments as before:

**703 | ' Used to set the documents for WorldEase Shipments**

**704 | Sub setDocumentsWE(PRIMARY\_SUBCATEGORY, CONSIGNEE)**

**705 | Set context = ScriptDatamanager.ClientContext**

**706 | Set currentPackage = context("CURRENT\_PACKAGE")**

**707 | sc = currentPackage("PRIMARY\_CATEGORY")**

**708 |**

**709 | Dim WorldeaseLabel, category, REFERENCE\_18, ref18**

**710 | WorldeaseLabel = "CONNECTSHIP\_UPS\_WORLD\_EASE\_MAXICODE\_LABEL.STANDARD"**

**711 |**

**712 | category = getCategory(PRIMARY\_SUBCATEGORY)**

**713 | REFERENCE\_18 = currentPackage("REFERENCE\_18")**

**714 | ref18 = Trim(REFERENCE\_18)**

**715 |**

**716 | If (REFERENCE\_18 = "WORLDEASE") Then**

**717 | Set macro = CreateObject("Progistics.Dictionary")**

**718 | macro.Value("NAME")= "MACRO\_MODIFY\_DOCUMENT\_ATTRIBUTES"**

**719 | macro.Value("CATEGORY") = category**

**720 | macro.Value("DOCUMENT\_FORMAT") = "CONNECTSHIP\_UPS\_WORLD\_EASE\_MAXICODE\_LABEL.STANDARD"**

**721 | macro.Value("PRINT") = True**

**722 | macro.Value("VIEW\_DOCUMENT\_MANAGER") = False**

**723 | ScriptDataManager.AddMacro macro**

**724 | Set macro = Nothing**

**725 | End If**

**726 | End Sub**

Following is the “setField” subroutine with arguments: “fieldName”, “fieldValue”, “scppressScript”, and “fieldIndex”. Not sure what it does...

**728 | Sub setField(fieldName, fieldValue, suppressScript, fieldIndex)**

**729 | Dim macro**

**730 | Set macro = CreateObject("Progistics.Dictionary")**

**731 | macro.Value("NAME") = "MACRO\_SET\_FIELD"**

**732 | macro.Value("WINDOW\_NAME") = fieldName**

**733 | macro.Value("FIELD\_NAME") = fieldValue**

**734 |**

**735 | If fieldName > 0 Then**

**736 | macro.Value(“FIELD\_INDEX”) = fieldIndex**

**737 | End If**

**738 |**

**739 | macro.Value("SUPPRESS\_SCRIPT") = suppressScript**

**740 | ScriptDataManager.AddMacro macro**

**741 | Set macro = Nothing**

**742 | End Sub**

Below

**744 | Sub setError( buf )**

**745 | Dim macro**

**746 | Set macro = CreateObject("Progistics.Dictionary")**

**747 | macro.Value("NAME") = "MACRO\_SET\_PACKAGE\_ERROR"**

**748 | macro.Value("ERROR\_MESSAGE") = buf**

**749 | ScriptDataManager.AddMacro macro**

**750 | Set macro = Nothing**

**751 | End Sub**

Below

**753 | Sub setCustomField( fieldName, fieldValue )**

**754 | Dim macro**

**755 | Set macro = CreateObject("Progistics.Dictionary")**

**756 | macro.Value("NAME") = "MACRO\_SET\_CUSTOM\_FIELD"**

**757 | macro.Value("FIELD\_NAME") = fieldName**

**758 | macro.Value("FIELD\_VALUE") = fieldValue**

**759 | ScriptDataManager.AddMacro macro**

**760 | Set macro = Nothing**

**761 | End Sub**

Below

**763 | Sub DebugLog( buf )**

**764 | setField "REFERENCE\_20", buf, True, 0**

**765 | End Sub**

Below

**767 | Function getCategory( primary\_subcategory )**

**768 | Dim retString, index**

**769 | index = InStrRev(primary\_subcategory,".")**

**770 | retString = Left(primary\_subcategory, index-1)**

**771 | getCategory = retString**

**772 | End Function**

Below

**774 | Function paperless\_on ()**

**775 | Dim context, package**

**776 | Set context = ScriptDataManager.ClientContext**

**777 | Set package = context("CURRENT\_PACKAGE")**

**778 |**

**779 | If (ScriptDataManager.StoredData("MYVALUE") = 0) Then**

**780 | Dim i**

**781 | For i = 1 To package("NOFN\_TOTAL")**

**782 | Dim macro**

**783 | Set macro = CreateObject("Progistics.Dictionary")**

**784 | macro.Value("NAME") = "MACRO\_MOVE\_TO\_PACKAGE"**

**785 | macro.Value("PACKAGE\_NUMBER") = i**

**786 | ScriptDataManager.AddMacro macro**

**787 | Set macro = Nothing**

**788 | Set macro = CreateObject("Progistics.Dictionary")**

**789 | macro.Value("NAME") = "MACRO\_SET\_CUSTOM\_FIELD"**

**790 | macro.Value("FIELD\_NAME") = "COMMERCIAL\_INVOICE\_METHOD"**

**791 | macro.Value("FIELD\_VALUE") = 1**

**792 | ScriptDataManager.AddMacro macro**

**793 | Set macro = Nothing**

**794 | Next**

**795 | ScriptDataManager.StoredData("MYVALUE") = 1**

**796 | End If**

**797 | End Function**

Below

**799 | Function IsFedExBillable(shipper, companyNumber, service, paymentTerms, accountNumber, thirdPartyAddress, thirdPartyAccountNumber, thirdPartyBilling)**

**800 | If IsFedExService(service) Then**

**801 | If IsADM(companyNumber) Then**

**802 | setError("Cannot ship FedEx under company ADM.")**

**803 | IsFedExBillable = False**

**804 | Else**

**805 | If VerificationNeeded(shipper) Then**

**806 | If IsFreightCollect(paymentTerms) Then**

**807 | If IsValidAccountNumber(accountNumber) Then**

**808 | IsFedExBillable = True**

**809 | Else**

**810 | setError("Invalid consignee account number.")**

**811 | IsFedExBillable = False**

**812 | End If**

**813 | ElseIf thirdPartyBilling Then**

**814 | If IsValidAccountNumber(thirdPartyAccountNumber) Then**

**815 | IsFedExBillable = True**

**816 | Else**

**817 | setError("Invalid third party account number.")**

**818 | IsFedExBillable = False**

**819 | End If**

**820 | Else**

**821 | setError("FedEx orders must ship Freight Collect or Third Party for Company " & companyNumber & ".")**

**822 | IsFedExBillable = False**

**823 | End If**

**824 | Else**

**825 | IsFedExBillable = True**

**826 | End If**

**827 | End If**

**828 | Else**

**829 | IsFedExBillable = True**

**830 | End If**

**831 | End Function**

Below

**833 | ' Verifies the company is a company billing verification is needed for.**

**834 | Function VerificationNeeded(shipper)**

**835 | If shipper = "JONESBLET" or shipper = "ACS" or shipper = "AUGSBURG" or shipper = "ALPHA" Then**

**836 | VerificationNeeded = False**

**837 | Else**

**838 | VerificationNeeded = True**

**839 | End If**

**840 | End Function**

Below

**842 | ' Verifies the service is a FedEx service**

**843 | Function IsFedExService(service)**

**844 | If InStr(service, "TANDATA\_FEDEXFSMS") Or service = "TANDATA\_LTL.FXNJ" Or service = "TANDATA\_LTL.FXFJ" Then**

**845 | IsFedExService = True**

**846 | Else**

**847 | IsFedExService = False**

**848 | End If**

**849 | End Function**

Below

**851 | ' Determines if the package is shipping Freight Collect**

**852 | Function IsFreightCollect(paymentTerms)**

**853 | If paymentTerms = "CONSIGNEE" Then**

**854 | IsFreightCollect = True**

**855 | Else**

**856 | IsFreightCollect = False**

**857 | End If**

**858 | End Function**

Below

**860 | ' Determines if an account number is valid**

**861 | Function IsValidAccountNumber(accountNumber)**

**862 | Dim regex**

**863 | Set regex = New RegExp**

**864 | regex.Global = False**

**865 | regex.IgnoreCase = True**

**866 | regex.Pattern = "^[0-9]{9}$"**

**867 | IsValidAccountNumber = regex.Test(accountNumber)**

**868 | End Function**

Below

**870 | ' Determines if an address object is valid**

**871 | Function IsValidAddress(address)**

**872 | If (Len(address.Address1) > 0 or Len(address.Address2) > 0 or Len(address.Address3) > 0) And Len(address.City) > 0 And Len(address.StateProvince) > 0 And Len(address.PostalCode) > 0 Then**

**873 | IsValidAddress = True**

**874 | Else**

**875 | IsValidAddress = False**

**876 | End If**

**877 | End Function**

Below

**879 | ' Determines if the company number is ADM**

**880 | Function IsADM(companyNumber)**

**881 | If UCase(Trim(companyNumber)) = "ADM" Then**

**882 | IsADM = True**

**883 | Else**

**884 | IsADM = False**

**885 | End If**

**886 | End Function**

Below

**888 | ' Documents orders that fail weight check**

**889 | Function writeFailed(msg)**

**890 | Dim debugfilepath, debugFso, debugOut, strHeader, context, currentPackage**

**891 |**

**892 | debugfilepath = "\\main.pbd.com\data\Client Updates\WeightChkLogs\FailedWeightCheck.txt"**

**893 | Set debugFso = CreateObject("Scripting.FileSystemObject")**

**894 | Set debugOut = debugFso.OpenTextFile(debugFilePath, 8, True)**

**895 |**

**896 | debugOut.WriteLine Now()**

**897 | debugOut.WriteLine msg**

**898 | debugOut.WriteLine**

**899 |**

**900 | ' Cleanup objects**

**901 | Set debugOut = Nothing**

**902 | Set debugFso = Nothing**

**903 | End Function**

Below

**913 | Sub clientBusinessRules()**

**914 | ' \*\*\* AFP (2A) Business Rules \*\*\***

**915 | If (REFERENCE\_2 = "2A") Then**

**916 | If (InStr(PRIMARY\_SUBCATEGORY, "FEDEXFSMS.FEDEX.SP")) Then**

**917 | ' Proof should be false per Augsburg's FedEx contract (produces a tracking barcode)**

**918 | setField "PROOF\_REQUIRE\_SIGNATURE\_ADULT", False, True, 0**

**919 | setField "PROOF\_REQUIRE\_SIGNATURE\_CONSIGNEE", False , True, 0**

**920 | setField "PROOF\_REQUIRE\_SIGNATURE", False, True, 0**

**921 | End If**

**922 | ' AFP (Augsburg) SmartPost Logic**

**923 | If (InStr(PRIMARY\_SUBCATEGORY, "FEDEXFSMS.FEDEX.SP") Or InStr(PRIMARY\_SUBCATEGORY, "MISC.POST.SMPT")) Then**

**924 | If (dblMyWeight < 1.0) Then**

**925 | setField "PRIMARY\_SUBCATEGORY", "TANDATA\_FEDEXFSMS.FEDEX.SP\_STD", False, 0**

**926 | Else**

**927 | setField "PRIMARY\_SUBCATEGORY", "TANDATA\_FEDEXFSMS.FEDEX.SP\_PS", False, 0**

**928 | End If**

**929 | End If**

**930 | End If**

**931 | ' \*\*\* TPM (07) Business Rules \*\*\***

**932 | If (REFERENCE\_2 = "07" and PRIMARY\_SUBCATEGORY = "CONNECTSHIP\_UPSMAILINNOVATIONS.UPS.EPD" and CONSIGNEE.CountrySymbol = "UNITED\_STATES") Then**

**933 | If (dblMyWeight > .81 and dblMyWeight <= 10) Then**

**934 | setfield "PRIMARY\_SUBCATEGORY", "TANDATA\_USPS.USPS.SPCL", False, 0**

**935 | setfield "TRACKING\_NUMBER", "", False, 0**

**936 | ElseIf (dblMyWeight > 10) Then**

**937 | setfield "PRIMARY\_SUBCATEGORY", "CONNECTSHIP\_UPS.UPS.GND", False, 0**

**938 | setfield "TRACKING\_NUMBER", "", False, 0**

**939 | End If**

**940 | End If**

**941 | ' \*\*\* CFA (11) business rules \*\*\***

**942 | If (REFERENCE\_2 = "11") Then**

**943 | If Not InStr(PRIMARY\_SUBCATEGORY, "TANDATA\_USPS") Then**

**944 | Select Case user**

**945 | Case "duluth"**

**946 | setField "SHIPPER", "CFA\_DULUTH" , True, 0**

**947 | Case "alpha"**

**948 | setField "SHIPPER", "CFA\_ALPHA" , True, 0**

**949 | End Select**

**950 | End If**

**951 | End If**

**952 | ' \*\*\* ANA (17) Business Rules \*\*\***

**953 | If (PRIMARY\_SUBCATEGORY = "TANDATA\_USPS.USPS.BPM" and REFERENCE\_2 = "17") Then**

**954 | If (dblMyWeight < .6) Then**

**955 | setfield "PRIMARY\_SUBCATEGORY", "TANDATA\_USPS.USPS.FIRST", False, 0**

**956 | End If**

**957 | If (dblMyWeight > 9.99) Then**

**958 | setfield "PRIMARY\_SUBCATEGORY", "CONNECTSHIP\_UPS.UPS.GND", False, 0**

**959 | End If**

**960 | End If**

**961 | ' \*\*\* UMCOMM (35) Business Rules \*\*\***

**962 | ' Residential shipments under 9 lbs should switch from UPS Ground to UPS Surepost (Ticket #34967)**

**963 | If (CONSIGNEE.Residential = True And PRIMARY\_SUBCATEGORY = "CONNECTSHIP\_UPS.UPS.GND" And REFERENCE\_2 = "35") Then**

**964 | If (dblMyWeight < 1) Then**

**965 | setField "PRIMARY\_SUBCATEGORY", "CONNECTSHIP\_UPS.UPS.SPSTD", False, 0**

**966 | ElseIf (dblMyWeight >= 1 And dblMyWeight < 9) Then**

**967 | setField "PRIMARY\_SUBCATEGORY", "CONNECTSHIP\_UPS.UPS.SPPS", False, 0**

**968 | setfield "PROOF", False, False, 0**

**969 | ElseIf (dblMyWeight >= 9) Then**

**970 | setField "PRIMARY\_SUBCATEGORY", "CONNECTSHIP\_UPS.UPS.GND", False, 0**

**971 | End If**

**972 | End If**

**973 | ' \*\*\* JOM (54) business rules \*\*\***

**974 | ' Set shipments to address service requested (Project 7346)**

**975 | If (REFERENCE\_2 = "54") Then**

**976 | If (ScriptDataManager.StoredData("FLAG") = 0) then**

**977 | Dim context1, objMacro**

**978 | ' Load data from the client**

**979 | Set context1 = ScriptDataManager.ClientContext**

**980 | Set package = context1( "CURRENT\_PACKAGE" )**

**981 |**

**982 | ' SET DISPOSITION METHOD**

**983 | Set objMacro = CreateObject("Progistics.Dictionary")**

**984 | objMacro.Value("NAME") = "MACRO\_SET\_CUSTOM\_FIELD"**

**985 | objMacro.Value("FIELD\_NAME") = "DISPOSITION\_METHOD"**

**986 | objMacro.Value("FIELD\_VALUE") = 3**

**987 | ScriptDataManager.AddMacro objMacro**

**988 | Set objMacro = Nothing**

**989 |**

**990 | ' Set address change notification**

**991 | Set objMacro = CreateObject("Progistics.Dictionary")**

**992 | objMacro.Value("NAME") = "MACRO\_SET\_CUSTOM\_FIELD"**

**993 | objMacro.Value("FIELD\_NAME") = "ADDRESS\_CHANGE\_NOTIFICATION"**

**994 | objMacro.Value("FIELD\_VALUE") = True**

**995 | ScriptDataManager.AddMacro objMacro**

**996 | Set objMacro = Nothing**

**997 |**

**998 | ' Cleanup remaining objects**

**999 | Set context1 = Nothing**

**1000 | Set package = Nothing**

**1001 |**

**1002 | ' Update StoredData flag**

**1003 | ScriptDataManager.StoredData("FLAG") = 1**

**1004 | End if**

**1005 | End If**

**1006 | ' \*\*\* ICMA (58) Business Rules \*\*\***

**1007 | If (PRIMARY\_SUBCATEGORY = "CONNECTSHIP\_UPS.UPS.GND" and REFERENCE\_2 = "58" and REFERENCE\_3 <> "AMAZON ORDER") Then**

**1008 | If (dblMyWeight <= 0.81) Then**

**1009 | setfield "PRIMARY\_SUBCATEGORY", "TANDATA\_USPS.USPS.FIRST", False, 0**

**1010 | ElseIf (dblMyWeight > .81 and dblMyWeight <= 0.99) Then**

**1011 | setfield "PRIMARY\_SUBCATEGORY", "TANDATA\_USPS.USPS.PRIORITY", False, 0**

**1012 | End If**

**1013 | End If**

**1014 | ' \*\*\* UMW (68) Business Rules \*\*\***

**1015 | ' Shipments < 1 lb: UPSMI, Shipments 1-9 lb: UPS Surepost, 9+ lb: UPS Ground**

**1016 | If (REFERENCE\_2 = "68") Then**

**1017 | If (PRIMARY\_SUBCATEGORY = "CONNECTSHIP\_UPS.UPS.GND" or PRIMARY\_SUBCATEGORY = "TANDATA\_USPS.USPS.FIRST") Then**

**1018 | If (dblMyWeight < 1) Then**

**1019 | setField "PRIMARY\_SUBCATEGORY", "CONNECTSHIP\_UPSMAILINNOVATIONS.UPS.EPD", False, 0**

**1020 | Else**

**1021 | If (dblMyWeight >= 1 and dblMyWeight <= 9) Then**

**1022 | Dim tempStreet**

**1023 | tempStreet = CONSIGNEE.ADDRESS1 & CONSIGNEE.ADDRESS2**

**1024 | If (CONSIGNEE.Residential = True or InStr(tempStreet, "POBOX")) Then**

**1025 | setField "PROOF", False, False, 0**

**1026 | setField "PRIMARY\_SUBCATEGORY", "CONNECTSHIP\_UPS.UPS.SPPS", False, 0**

**1027 | End If**

**1028 | Else**

**1029 | setField "PRIMARY\_SUBCATEGORY", "CONNECTSHIP\_UPS.UPS.GND", False, 0**

**1030 | End If**

**1031 | End If**

**1032 | End If**

**1033 | End If**

**1034 | ' \*\*\* Heifer (69) Business Rules \*\*\***

**1035 | If (PRIMARY\_SUBCATEGORY = "TANDATA\_USPS.USPS.FIRST" and REFERENCE\_2 = "69"**

**and dblMyWeight > 1.75) Then**

**1036 | setfield "PRIMARY\_SUBCATEGORY", "CONNECTSHIP\_UPS.UPS.GND", False, 0**

**1037 | setField "SHIPPER", "HEI" , True, 0**

**1038 | End If**

**1039 | ' If service is UPSMI Expedited, change service based on weight**

**1040 | If (REFERENCE\_2 = "69" and PRIMARY\_SUBCATEGORY = "CONNECTSHIP\_UPSMAILINNOVATIONS.UPS.EPD") Then**

**1041 | If (dblMyWeight > .0625 and dblMyWeight <= 1) Then**

**1042 | setfield "PRIMARY\_SUBCATEGORY", "TANDATA\_USPS.USPS.FIRST", False, 0**

**1043 | setfield "TRACKING\_NUMBER", "", True, 0**

**1044 | ElseIf (dblMyWeight > 1 and dblMyWeight <= 7 And CONSIGNEE.Residential = True) Then**

**1045 | setfield "PRIMARY\_SUBCATEGORY", "CONNECTSHIP\_UPS.UPS.SPPS", False, 0**

**1046 | setfield "TRACKING\_NUMBER", "", True, 0**

**1047 | setfield "PROOF", False, False, 0**

**1048 | ElseIf (dblMyWeight > 7) Then**

**1049 | setfield "PRIMARY\_SUBCATEGORY", "CONNECTSHIP\_UPS.UPS.GND", False, 0**

**1050 | setfield "TRACKING\_NUMBER", "", True, 0**

**1051 | setField "SHIPPER", "HEI" , True, 0**

**1052 | End If**

**1053 | End If**

**1054 | ' \*\*\* Arthritis Foundation (77) Business Rules \*\*\***

**1055 | ' Residential shipments under 9 lbs switch from UPS Ground to UPS SurePost**

**1056 | ' Anything with taken\_by of "AFYRP": < 1 lb, ship UPSMI. > 1 lb & residential, ship SurePost (Project 6661)**

**1057 | If (REFERENCE\_2 = "77") Then**

**1058 | If (ORDER\_TAKEN\_BY = "AFYRP" ) Then**

**1059 | If (dblMyWeight < 1.0) Then**

**1060 | setField "PRIMARY\_SUBCATEGORY", "CONNECTSHIP\_UPSMAILINNOVATIONS.UPS.EPD", False, 0**

**1061 | Else**

**1062 | If (CONSIGNEE.Residential = True And dblMyWeight <= 9.0 ) Then**

**1063 | setfield "PROOF", False, False, 0**

**1064 | setField "PRIMARY\_SUBCATEGORY", "CONNECTSHIP\_UPS.UPS.SPPS", False, 0**

**1065 | Else**

**1066 | setField "PRIMARY\_SUBCATEGORY", "CONNECTSHIP\_UPS.UPS.GND", False, 0**

**1067 | End If**

**1068 | End If**

**1069 | Else**

**1070 | If (CONSIGNEE.Residential = True And PRIMARY\_SUBCATEGORY = "CONNECTSHIP\_UPS.UPS.GND") Then**

**1071 | If (dblMyWeight <= 9) Then**

**1072 | setField "PRIMARY\_SUBCATEGORY", "CONNECTSHIP\_UPS.UPS.SPPS", False, 0**

**1073 | setfield "PROOF", False, False, 0**

**1074 | End If**

**1075 | End If**

**1076 | End If**

**1077 | End If**

**1078 | ' \*\*\* WF (2Y) Business Rules \*\*\***

**1079 | ' Handle BES FOB code (which passes down UPSMI Expedited as the service) for WF (Project #7642 - Task #727)**

**1080 | If (REFERENCE\_2 = "2Y" and PRIMARY\_SUBCATEGORY = "CONNECTSHIP\_UPSMAILINNOVATIONS.UPS.EPD") Then**

**1081 | If (CONSIGNEE.Residential = True and dblMyWeight >= 1 and dblMyWeight < 10) Then**

**1082 | setfield "PRIMARY\_SUBCATEGORY", "CONNECTSHIP\_UPS.UPS.SPPS", False, 0**

**1083 | setfield "PROOF", False, False, 0**

**1084 | ElseIf (CONSIGNEE.Residential = True and dblMyWeight >= 10) Then**

**1085 | setfield "PRIMARY\_SUBCATEGORY", "CONNECTSHIP\_UPS.UPS.GND", False,**

**1086 | ElseIf (CONSIGNEE.Residential = False and dblMyWeight >= 1) Then**

**1087 | setfield "PRIMARY\_SUBCATEGORY", "CONNECTSHIP\_UPS.UPS.GND", False, 0**

**1088 | Else**

**1089 | setfield "PRIMARY\_SUBCATEGORY", "CONNECTSHIP\_UPSMAILINNOVATIONS.UPS.EPD", False, 0**

**1090 | End If**

**1091 | End If**

**1092 | ' Ticket #44174 Bill Wahoo shipments to 3rd party**

**1093 | If (REFERENCE\_2 = "2Y" And CONSIGNEE.CountrySymbol = "CANADA" And PRIMARY\_SUBCATEGORY = "CONNECTSHIP\_UPS.UPS.STD" And ScriptDataManager.StoredData("LOOP\_FLAG") = 0) Then**

**1094 | setField "TERMS", "DDU", False, 0**

**1095 | setField "THIRD\_PARTY\_BILLING", True, True, 0**

**1096 | setField "THIRD\_PARTY\_BILLING\_ACCOUNT", "309A77", True, 0**

**1097 | setField "THIRD\_PARTY\_BILLING\_COUNTRY", "UNITED\_STATES", True, 0 ' Country**

**1098 | setField "THIRD\_PARTY\_BILLING\_COMPANY", "WAHOO FITNESS" , True, 0**

**1099 | setField "THIRD\_PARTY\_BILLING\_CONTACT", "Steve Hochradel", True, 0**

**1100 | setField "THIRD\_PARTY\_BILLING\_ADDRESS\_LINE\_1", "1650 Bluegrass Lakes Pkwy", True, 0**

**1101 | setField "THIRD\_PARTY\_BILLING\_ADDRESS\_LINE\_2", "", True, 0**

**1102 | setField "THIRD\_PARTY\_BILLING\_ADDRESS\_LINE\_3", "", True, 0**

**1103 | setField "THIRD\_PARTY\_BILLING\_CITY", "Alpharetta", True, 0**

**1104 | setField "THIRD\_PARTY\_BILLING\_STATEPROVINCE", "GA", True, 0**

**1105 | setField "THIRD\_PARTY\_BILLING\_POSTALCODE", "30004", True, 0**

**1106 | setField "THIRD\_PARTY\_BILLING\_PHONE", "770-280-4097", True, 0 ' Telephone**

**1107 | Dim objCTPBna**

**1108 | Set objCTPBna = CreateObject("Progistics.NameAddress")**

**1109 | Set objMacro = CreateObject("Progistics.Dictionary")**

**1110 |**

**1111 | objCTPBna.CountrySymbol = "CANADA"**

**1112 | objCTPBna.Company = "Wahoo Fitness"**

**1113 | objCTPBna.Contact = "Steve Hochradel"**

**1114 | objCTPBna.Address1 = "790 Belgrave Way"**

**1115 | objCTPBna.Phone = "770-280-4097"**

**1116 | objCTPBna.City = "Delta"**

**1117 | objCTPBna.StateProvince = "BC"**

**1118 | objCTPBna.PostalCode = "V3M5R9"**

**1119 |**

**1120 | Set objMacro = CreateObject("Progistics.Dictionary")  
1121 | objMacro.Value("NAME") = "MACRO\_SET\_CUSTOM\_FIELD"**

**1121 | objMacro.Value("FIELD\_NAME") = "CONSIGNEE\_THIRD\_PARTY\_BILLING"**

**1122 | objMacro.Value("FIELD\_VALUE") = True**

**1123 | objMacro.Value("SUPPRESS\_SCRIPT") = True**

**1124 | ScriptDataManager.AddMacro objMacro**

**1125 | Set objMacro = Nothing**

**1126 |**

**1127 | Set objMacro = CreateObject("Progistics.Dictionary")**

**1128 | objMacro.Value("NAME") = "MACRO\_SET\_CUSTOM\_FIELD"**

**1129 | objMacro.Value("FIELD\_NAME") = "CONSIGNEE\_THIRD\_PARTY\_BILLING\_ACCOUNT"**

**1130 | objMacro.Value("FIELD\_VALUE") = "37X46V"**

**1131 | objMacro.Value("SUPPRESS\_SCRIPT") = True**

**1132 | ScriptDataManager.AddMacro objMacro**

**1133 | Set objMacro = Nothing**

**1134 |**

**1135 | Set objMacro = CreateObject("Progistics.Dictionary")**

**1136 | objMacro.Value("NAME") = "MACRO\_SET\_CUSTOM\_FIELD"**

**1137 | objMacro.Value("FIELD\_NAME") = "CONSIGNEE\_THIRD\_PARTY\_BILLING\_ADDRESS"**

**1138 | objMacro.Value("FIELD\_VALUE") = objCTPBna**

**1139 | objMacro.Value("SUPPRESS\_SCRIPT") = True**

**1140 | ScriptDataManager.AddMacro objMacro**

**1141 | Set objMacro = Nothing**

**1142 |**

**1143 | Set objCTPBna = Nothing**

**1144 | ScriptDataManager.StoredData("LOOP\_FLAG") = 1**

**1145 | End If**

**1146 |**

**1147 | ' \*\*\* Pryor (3C) business rules \*\*\***

**1148 | ' Handle BES FOB code for Pryor - Project #7765**

**1149 | If (REFERENCE\_2 = "3C" And PRIMARY\_SUBCATEGORY = "CONNECTSHIP\_UPSMAILINNOVATIONS.UPS.EPD") Then**

**1150 | If (dblMyWeight => 15) Then**

**1151 | If (CONSIGNEE.Residential = False) Then**

**1152 | setfield "PRIMARY\_SUBCATEGORY", "CONNECTSHIP\_UPS.UPS.GND", False, 0**

**1153 | End If**

**1154 | If (CONSIGNEE.Residential = True) Then**

**1155 | setfield "PRIMARY\_SUBCATEGORY", "CONNECTSHIP\_UPS.UPS.SPPS", False, 0**

**1156 | setfield "PROOF", False, False, 0**

**1157 | End If**

**1158 | End If**

**1159 | If (dblMyWeight > 1 And dblMyWeight < 15 ) Then**

**1160 | setfield "PRIMARY\_SUBCATEGORY", "CONNECTSHIP\_UPSMAILINNOVATIONS.UPS.MEDIA", False, 0**

**1161 | End If**

**1162 | End If**

**1163 | ' \*\*\* CPI (2W) Business Rules \*\*\***

**1164 | ' Ticket #42228**

**1165 | If (REFERENCE\_2 = "2W" And PRIMARY\_SUBCATEGORY = "CONNECTSHIP\_UPSMAILINNOVATIONS.UPS.EPD") Then**

**1166 | If (dblMyWeight < 1) Then**

**1167 | setfield "PRIMARY\_SUBCATEGORY", "CONNECTSHIP\_UPSMAILINNOVATIONS.UPS.EPD", False, 0-9**

**1168 | ElseIf (dblMyWeight > 1) Then**

**1169 | If (CONSIGNEE.Residential = True) Then**

**1170 | setfield "PROOF", False, False, 0**

**1171 | setfield "PRIMARY\_SUBCATEGORY", "CONNECTSHIP\_UPS.UPS.SPPS", False, 0-9**

**1172 | Else**

**1173 | setfield "PRIMARY\_SUBCATEGORY", "CONNECTSHIP\_UPS.UPS.GND", False, 0**

**1174 | End If**

**1175 | End If**

**1176 | End If**

**1177 | ' \*\*\* Holsted (3F) Business Rules \*\*\***

**1178 | ' Set Holsted DHL orders to return service requested**

**1179 | If (REFERENCE\_2 = "3F") Then**

**1180 | If (ScriptDataManager.StoredData("LOOPFLAG2") = 0) Then**

**1181 | Dim context2**

**1182 | ' Load data from the client**

**1183 | Set context2 = ScriptDataManager.ClientContext**

**1184 | Set package = context2( "CURRENT\_PACKAGE" )**

**1185 |**

**1186 | ' SET DISPOSITION METHOD**

**1187 | Set objMacro = CreateObject("Progistics.Dictionary")**

**1188 | objMacro.Value("NAME") = "MACRO\_SET\_CUSTOM\_FIELD"**

**1189 | objMacro.Value("FIELD\_NAME") = "DISPOSITION\_METHOD"**

**1190 | objMacro.Value("FIELD\_VALUE") = 1**

**1191 | ScriptDataManager.AddMacro objMacro**

**1192 | Set objMacro = Nothing**

**1193 |**

**1194 | ' Cleanup remaining objects**

**1195 | Set context2 = Nothing**

**1196 | Set package = Nothing**

**1197 |**

**1198 | ' Update StoredData flag**

**1199 | ScriptDataManager.StoredData("LOOPFLAG2") = 1**

**1200 | End If**

**1201 | ' Round up weight of <= 1oz to 1.01oz per Tonya, Jonathan, & Steve H.**

**1202 | ' This should allow users to ship flats and parcels with a total weight of less than an ounce.**

**1203 | If (Instr(PRIMARY\_SUBCATEGORY, "CONNECTSHIP\_DHLGLOBALMAIL.DHL") and dblMyWeight <= .0625) Then**

**1204 | setfield "WEIGHT", 0.063125, False, 0**

**1205 | dblMyWeight = 0.063125**

**1206 | End If**

**1207 | End If**

**1208 | ' Set Holsted orders going to Puerto Rico to DDP per Ticket #46487**

**1209 | If (REFERENCE\_2 = "3F" And CONSIGNEE.CountrySymbol = "PUERTO\_RICO") Then**

**1210 | setField "TERMS", "SHIPPER", False, 0**

**1211 | End If**

**1212 | ' \*\*\* ASQ (28) Business Rules \*\*\***

**1213 | If (REFERENCE\_2 = "28" And InStr(PRIMARY\_SUBCATEGORY, "DHL")) Then**

**1214 | setField "COMMERCIAL\_INVOICE\_METHOD", "0", True, 0**

**1215 | If (REFERENCE\_5 <> "BLANK") Then**

**1216 | setField "REFERENCE\_6", REFERENCE\_5, True, 0**

**1217 | setField "REFERENCE\_1", REFERENCE\_5, True, 0**

**1218 | End If**

**1219 | End If**

**1220 | End Sub**

Below

**1222 | Sub weightCheck()**

**1223 | Dim perVar, tolerance, avgBox, fillerObject, actualWeight, expectedWeight, weightWithBox, variance**

**1224 | Set fillerObject = CreateObject("Progistics.Dictionary")**

**1225 |**

**1226 | If (IsNull(ScriptDataManager.StoredData("ORDER\_WEIGHT")) Or ScriptDataManager.StoredData("ORDER\_WEIGHT") = "") Then**

**1227 | expectedWeight = 0**

**1228 | Else**

**1229 | expectedWeight = ABS(ScriptDataManager.StoredData("ORDER\_WEIGHT"))**

**1230 | End If**

**1231 |**

**1232 | If (REFERENCE\_2 = "1Q") Then**

**1233 | If (expectedWeight <= .5) Then**

**1234 | avgBox = .5**

**1235 | tolerance = .75**

**1236 | ElseIf (expectedWeight > .5 And expectedWeight <= 3) Then**

**1237 | avgBox = .8**

**1238 | tolerance = 1**

**1239 | ElseIf (expectedWeight > 3 And expectedWeight <= 10) Then**

**1240 | avgBox = .8**

**1241 | tolerance = 1.5**

**1242 | ElseIf (expectedWeight > 10 And expectedWeight <= 20) Then**

**1243 | avgBox = 2.1**

**1244 | tolerance = 1.5**

**1245 | ElseIf (expectedWeight > 20 And expectedWeight <= 40) Then**

**1246 | avgBox = 2.1**

**1247 | tolerance = 1.75**

**1248 | ElseIf (expectedWeight > 40 And expectedWeight <= 60) Then**

**1249 | avgBox = 2.1**

**1250 | tolerance = 2**

**1251 | ElseIf (expectedWeight > 60 And expectedWeight <= 100) Then**

**1252 | avgBox = 2.1**

**1253 | tolerance = 3**

**1254 | ElseIf (expectedWeight > 100 And expectedWeight <= 150) Then**

**1255 | avgBox = 2.1**

**1256 | tolerance = 6**

**1257 | ElseIf (expectedWeight > 150) Then**

**1258 | avgBox = 2.1**

**1259 | tolerance = 10**

**1260 | End If**

**1261 | End If**

**1262 |**

**1263 | If (REFERENCE\_2 = "84") Then**

**1264 | If (expectedWeight <= .5) Then**

**1265 | avgBox = .3**

**1266 | tolerance = .75**

**1267 | ElseIf (expectedWeight > .5 And expectedWeight <= 1) Then**

**1268 | avgBox = .4**

**1269 | tolerance = 1**

**1270 | ElseIf (expectedWeight > 1 And expectedWeight <= 5) Then**

**1271 | avgBox = 1.5**

**1272 | tolerance = 2.25**

**1273 | ElseIf (expectedWeight > 5 And expectedWeight <= 10) Then**

**1274 | avgBox = 1.5**

**1275 | tolerance = 2.25**

**1276 | ElseIf (expectedWeight > 10 And expectedWeight <= 15) Then**

**1277 | avgBox = 2**

**1278 | tolerance = 2.5**

**1279 | ElseIf (expectedWeight > 15 And expectedWeight <= 30) Then**

**1280 | avgBox = 2.3**

**1281 | tolerance = 3**

**1282 | ElseIf (expectedWeight > 30 And expectedWeight <= 50) Then**

**1283 | avgBox = 2**

**1284 | tolerance = 3.5**

**1285 | ElseIf (expectedWeight > 50 And expectedWeight <= 100) Then**

**1286 | avgBox = 2.3**

**1287 | tolerance = 4**

**1288 | ElseIf (expectedWeight > 100) Then**

**1289 | avgBox = 2.3**

**1290 | tolerance = 6**

**1291 | End If**

**1292 | End If**

**1293 |**

**1294 | If (REFERENCE\_2 = "004") Then**

**1295 | If (expectedWeight <= 1) Then**

**1296 | avgBox = .3**

**1297 | tolerance = .15**

**1298 | ElseIf (expectedWeight > 1 And expectedWeight <= 6) Then**

**1299 | avgBox = .3**

**1300 | tolerance = .25**

**1301 | ElseIf (expectedWeight > 6 And expectedWeight <= 10) Then**

**1302 | avgBox = .6**

**1303 | tolerance = .5**

**1304 | ElseIf (expectedWeight > 10 And expectedWeight <= 20) Then**

**1305 | avgBox = .65**

**1306 | tolerance = .75**

**1307 | ElseIf (expectedWeight > 20) Then**

**1308 | avgBox = .65**

**1309 | tolerance = 1**

**1310 | End If**

**1311 | End If**

**1312 |**

**1313 | If (REFERENCE\_2 <> "1Q" And REFERENCE\_2 <> "84" And REFERENCE\_2 <> "65" And REFERENCE\_2 <> "004") Then**

**1314 | If (expectedWeight <= .5) Then**

**1315 | avgBox = .1**

**1316 | tolerance = .75**

**1317 | ElseIf (expectedWeight > .5 And expectedWeight <= 1) Then**

**1318 | avgBox = .1**

**1319 | tolerance = 1**

**1320 | ElseIf (expectedWeight > 1 And expectedWeight <= 5) Then**

**1321 | avgBox = .55**

**1322 | tolerance = 1**

**1323 | ElseIf (expectedWeight > 5 And expectedWeight <= 10) Then**

**1324 | avgBox = .65**

**1325 | tolerance = 1.5**

**1326 | ElseIf (expectedWeight > 10 And expectedWeight <= 15) Then**

**1327 | avgBox = .85**

**1328 | tolerance = 2**

**1329 | ElseIf (expectedWeight > 15 And expectedWeight <= 30) Then**

**1330 | avgBox = 1.15**

**1331 | tolerance = 3**

**1332 | ElseIf (expectedWeight > 30 And expectedWeight <= 50) Then**

**1333 | avgBox = 1.25**

**1334 | tolerance = 3.5**

**1335 | ElseIf (expectedWeight > 50 And expectedWeight <= 100) Then**

**1336 | avgBox = 1.25  
1337 | tolerance = 4**

**1338 | ElseIf (expectedWeight > 100) Then**

**1339 | avgBox = 1.25**

**1340 | tolerance = 5**

**1341 | End If**

**1342 | End If**

**1343 |**

**1344 | If (REFERENCE\_2 = "65") Then**

**1345 | If (expectedWeight <= .5) Then**

**1346 | avgBox = .1**

**1347 | tolerance = 2**

**1348 | ElseIf (expectedWeight > .5 And expectedWeight <= 1) Then**

**1349 | avgBox = .1**

**1350 | tolerance = 3.5**

**1351 | ElseIf (expectedWeight > 1 And expectedWeight <= 5) Then**

**1352 | avgBox = .55**

**1353 | tolerance = 3**

**1354 | ElseIf (expectedWeight > 5 And expectedWeight <= 10) Then**

**1355 | avgBox = .65**

**1356 | tolerance = 3**

**1357 | ElseIf (expectedWeight > 10 And expectedWeight <= 15) Then**

**1358 | avgBox = .85**

**1359 | tolerance = 3.5**

**1360 | ElseIf (expectedWeight > 15 And expectedWeight <= 30) Then**

**1361 | avgBox = 1.15**

**1362 | tolerance = 4**

**1363 | ElseIf (expectedWeight > 30 And expectedWeight <= 50) Then**

**1364 | avgBox = 1.25**

**1365 | tolerance = 4.5**

**1366 | ElseIf (expectedWeight > 50 And expectedWeight <= 100) Then**

**1367 | avgBox = 1.25**

**1368 | tolerance = 5**

**1369 | ElseIf (expectedWeight > 100) Then**

**1370 | avgBox = 1.25**

**1371 | tolerance = 6**

**1372 | End If**

**1373 | End If**

**1374 |**

**| If (isobject(context("CURRENT\_PACKAGE\_LIST")) And context("PACKAGE\_LIST\_SHIPPING\_MODE") = 2) Then**

**| Dim packagelist, pkgListWeight, numPackages**

**| Set packagelist = context("CURRENT\_PACKAGE\_LIST")**

**| pkgListWeight = 0**

**| weightWithBox = 0**

**| numPackages = 0**

**|**

**| For Each fillerObject In packagelist**

**| pkgListWeight = pkgListWeight + fillerObject("WEIGHT")**

**| weightWithBox = weightWithBox + fillerObject("WEIGHT")**

**| ' Subtract Average Box Weights**

**| pkgListWeight = pkgListWeight - avgBox**

**| numPackages = numPackages + 1**

**| Next**

**| actualWeight = pkgListWeight**

**| Else**

**| 'Use existing weight - avg carton weight**

**| actualWeight = dblMyWeight - avgBox**

**| weightWithBox = dblMyWeight**

**| End If**

**|**

**| variance = Abs(expectedWeight - actualWeight)**

**|**

**| If (variance < .2 And actualWeight < expectedWeight And REFERENCE\_2 = "65") Then**

**| variance = Abs(weightWithBox - (avgBox\*numPackages) - expectedWeight)**

**| End If**

**|**

**| ' If variance is outside of specified percentage, notify user and log error**

**| If (variance > tolerance) or (weightWithBox < (expectedWeight - .2)) Then**

**| Dim retPkgVal**

**| Dim cswUtil**

**| Dim failedWC**

**|**

**| Set cswUtil = CreateObject("CSWUtils.Utils")**

**| retPkgVal = cswUtil.Inform("Weight check failed. Expected Weight: " & Round(expectedWeight, 2) & " lbs. " & vbCrLf & "Calculated Weight: " & Round(actualWeight, 2) & "lbs." & vbCrLf & "Continue Shipment?", "Process Shipment", 4, 256)**

**|**

**| If (retPkgVal = 6) Then**

**| failedWC = ",Invoice#:," & REFERENCE\_1 & ",Expected weight:," & expectedWeight & ",Calculated weight:," & actualWeight & ",UserID:," & REFERENCE\_10 & ",Facility:," & user**

**| writeFailed(failedWC)**

**| ElseIf retPkgVal = 7 or retPkgVal <> 6 Then**

**| setError("Weight outside approved variance. Shipping cancelled")**

**| End If**

**|**

**| Set cswUtil = Nothing**

**| End If**

**| End Sub**

Below

**|**

Below

**|**

Below

**|**

Below

**|**

Below

**|**