Pueblo Computer Systems, Inc.

The Anatomy of Drag & Drop

This process requires a calculated container field in the table you want to drag from and the table you want to drag to.

In the table you want to drag from…in this case Order Line Items

zDND_Container – you setup a list of values you will need to pass to the script.

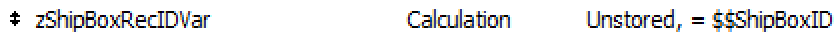
```
Let ( _trigger = zDnD_Container ;

"Item" & ¶ &
StyleNbr & ¶ &
Description & ¶ &
OrderLIrecID & ¶ &
OrderID & ¶ &
ShippingBoxID
)
```

In the receiving table – in this case ship boxes - this will create a variable with the primary key of the receiving record

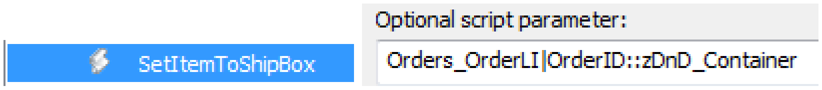
```
zDND_Handle
Let (
[
_trigger = zDnD_Handle ;
$$ShipBoxID = ShipBoxRecID
] ; "" )
```

In a separation model system, because we can't access the variable from the UI file, you will also need another field in the receiving table. It's just a calculation of the variable you create.



zShipBoxRecIDVar	Calculation	Unstored, = $$ShipBoxID

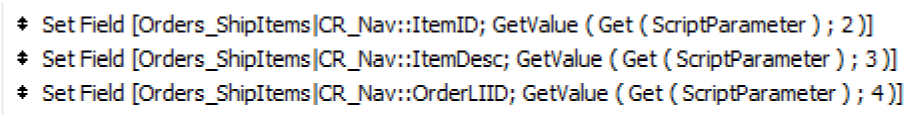
In the UI file you set a script trigger zDND_Container – OnObjectEnter with the primary script. You also set the field as the script parameter.



Optional script parameter:

Orders_OrderLI|OrderID::zDnD_Container

The script parameter will pass the values you need to the script… here is that section



```
Set Field [Orders_ShipItems|CR_Nav::ItemID; GetValue ( Get ( ScriptParameter ) ; 2 )]
Set Field [Orders_ShipItems|CR_Nav::ItemDesc; GetValue ( Get ( ScriptParameter ) ; 3 )]
Set Field [Orders_ShipItems|CR_Nav::OrderLIID; GetValue ( Get ( ScriptParameter ) ; 4 )]
```

You are dragging from zDnD_Container to zDnD_Handle

Or from Order Line Items to Ship Boxes

The Script

The comment at the top is a reference to the field order you set in zDnD_Container

```
#Item • StyleNr • Description • OrderLIrecID • OrderID • ShippingBoxRecID
```

When you drag into zDnD_Handle it created the Variable which in turn updates the matching field.
This first set field sets a global field to the record ID of the box you are dragging the item to.

```
Set Field [Orders::zgCR_Nav; Orders_ShipBoxes|OrderID::zShipBoxRecIDVar]
Set Variable [$QtyOrdered; Value:Orders_OrderLI|OrderID::QtyOrdered]
Commit Records/Requests
Set Variable [$BoxNbr; Value:Right ( Orders_ShipBoxes|CR_Nav::BoxID ; 1 )
```

If you don’t drag to zDnD_Handle this field will not have a value and the script will exit

```
If [IsEmpty ( Orders::zgCR_Nav )]
Exit Script [ ]
End If
```

This next section tests if the item is already in the box

```
If [
        Let (
        [
        ~Item = GetField ( "Orders_ShipItems|Box" & $BoxNbr & "::ItemID" ) ;
        ~Values = List ( ~Item ) ;
        ~String = GetValue ( Get ( ScriptParameter ) ; 2 ) ];
        PatternCount ( ~Values ; ~String ))

        Beep Show Custom Dialog [Title:"Add To Box"; Message: "Item: " & GetValue ( Get ( ScriptParameter ) ; 3 ) & " is
        Already In that Box!"; Default Button:“OK”, Commit:“Yes”]
        Perform Script [“ClearDnDApptID_Var” from file: “Data_ASI”] Set Field [Orders::zgCR_Nav; ""]
        Commit Records/Requests
        Exit Script [ ]
End If
Commit Records/Requests
```

Set the Ship Box Record ID to a variable to pass it to the Ship Box line item record.

```
Set Variable [$ShipBoxID; Value:Orders::zgCR_Nav]
```

This creates the line item record and refreshes the portal

```
Set Field [Orders::zgCR_Nav; ""]
Set Field [Orders_ShipItems|CR_Nav::OrderID; GetValue ( Get ( ScriptParameter ) ; 5 )]
Commit Records/Requests Set Field [Orders_ShipItems|CR_Nav::ShipBoxID; $ShipBoxID]
Set Field [Orders_ShipItems|CR_Nav::ItemID; GetValue ( Get ( ScriptParameter ) ; 2 )]
Set Field [Orders_ShipItems|CR_Nav::ItemDesc; GetValue ( Get ( ScriptParameter ) ; 3 )]
Set Field [Orders_ShipItems|CR_Nav::OrderLIID; GetValue ( Get ( ScriptParameter ) ; 4 )]
Commit Records/Requests
Refresh Object [Object Name:"Box" & $BoxNbr]
```