



The Anatomy of Drag & Drop

The standard drag & drop method is to drag from one table directly to another. This example is a bit different. In this example will be moving a record from Order Line Items to Ship Box Items. However we need the Ship Box record ID in order to create the Ship Box Items record.

Tables

- Order Line Items
- Ship Boxes
- Ship Box Items

DnD occurs between Order Line Items and Ship Boxes but just to capture Ship Box ID which is used to create the Ship Box Items.

Overview

Drag and Drop requires 2 calculated container fields.

In the table you want to drag from, in this case Order Line Items, the field creates a list of values you need to send to the receiving table. This field will have an OnObjectEnter trigger which calls your script and send this field as the script parameter making all the values available to GetValue (Get (ScriptParameter) ; x)

```
zDND_Container
```

```
Let ( _trigger = zDnD_Container ;
```

```
"Item" & ¶ &
```

```
StyleNbr & ¶ &
```

```
Description & ¶ &
```

```
OrderLIrecID & ¶ &
```

```
OrderID
```

```
)
```

Specify a script to be run for an event	
Event	Script
<input checked="" type="checkbox"/> OnObjectEnter	" SetItemToShipBox"; 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::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 )]
```



PUEBLO COMPUTER SYSTEMS, INC.

We will be dragging zDND_Container to Ship Boxes table but we are not creating a record there. We are just capturing the ShipBoxRecID to use when we create the record in Ship Box Items.

This field this field will create a variable with the Ship Boxes primary key.

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

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

✦ zShipBoxRecIDVar Calculation Unstored, = \$\$ShipBoxID

You are dragging from zDnD_Container to zDnD_Handle from Order Line Items to Ship Boxes



The Script

The comment at the top of the script is a reference to the field order you set in zDnD_Container
Item • StyleNbr • Description • OrderLIrecID • OrderID

When you drag into zDnD_Handle it creates 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. I left that field on the
layout so you can see it update. Normally it wouldn't be on the layout.

```
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
```



The Script Continued...

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]
```

Additional notes:

Other fields in Order Line Items help to track the item moved. ShippingBoxItemCT – it's a calculation which returns the Total ordered from Ship Box Items.

There is a section in the script which checks to see when Qty Shipped = Box CT. I have these script steps disabled because set the field zDND_Container to Hide Object when Qty Shipped = Box CT.