

# Drag and Drop

Implementing a complete drag and drop solution.

Drag and drop in the SRP Tree Control is actually quite simple to implement, but there are a lot of options available to achieve your drag and drop needs. It can be intimidating to sift through the many properties, methods, and events needed to make drag and drop a reality. This article aims to provide you with clarity on the subject.

## Sources and Targets

To help explain the drag and drop system, it helps to understand some terminology. In every drag and drop operation, there is a Source and a Target. In the case of the SRP Tree Control, the Source Tree is the tree from which the user is dragging items and the Target Tree, sometimes called the Drop Target, is the tree over which the user released the dragged items. There are some important steps necessary to make a tree a source, a target, or both. But first, there is the number one, most important step.

## CtrlEntId

For drag and drop to work effectively between SRP Tree Controls, there must be a way to identify another tree control. OpenInsight can already do this: all controls have a unique control entity identifier, the CtrlEntId. However, OLE Controls do not know what their CtrlEntId is, nor can they discover it on their own. You must tell the SRP Tree Controls, especially those to be involved in drag and drop transactions, what their CtrlEntIds are using the [CtrlEntId](#) property. Setting this property allows any other SRP Tree Control to identify it, making it much easier to transfer items between controls and to identify drag and drop sources and targets.

**IMPORTANT:** Always set the [CtrlEntId](#) during your CREATE event just to be safe. If you use promoted events, you can do this automatically for all controls. The code looks like this:

```
Set_Property(Ctrl, "OLE.CtrlEntId", Ctrl)
```

## Enabling a Tree as a Drag and Drop Source

By default, the user cannot drag any items from a tree. In order to allow the user that privilege, you must set the [DragEnabled](#) property to 1. Doing so will allow the user to drag any items he/she chooses. You have two choices on how to limit the items the user can drag. The [ItemDraggable](#) property allows you to choose which items are draggable and which ones are not, though this can be tedious. An easier way is to use the [DragCondition](#) property. Using [Conditions](#), you can specify Boolean expression that, when true, permits the user to drag an item.

This is all that is needed to set up a drag source. At this point, the user can drag items to other SRP Tree Controls, but releasing the items over these controls will have no effect. There is still some work to do.

***At this point, however, you can drag items to any drag and drop enabled text control. Doing so will cause a textual representation of the tree to appear, with tabs used to show hierarchy.***

## Enabling a Tree as a Drag and Drop Target

By default, a tree control will accept drag and drop items from any SRP Tree Control. This is usually not desirable, so the [DropSources](#) property has been provided to limit valid drop sources. Simply set this property to a list of CtrlEntIds from which items are allowed to be dragged and dropped. This list can include the tree itself if you wish to be able to drag and drop within the same control.

Like source trees, all items are valid drop targets by default. This means that the user can drop items onto or in between any item. Again, you have two choices on how to limit the drop targets. The [ItemDroppable](#) property lets you be very specific about which items can be drop targets whereas the [DropCondition](#) uses [Conditions](#) to automatically decide the drop targets.

Another property you may want to modify is the [DropBehavior](#) property. This property defines the kind of feedback the user will see as items are dragged over its surface. There are three drop behaviors:

Behavior	Description	Use
Add	The user sees items highlight, indicating that dropping the items will add them as children to the target item	Use when you want to allow users to move or copy items to new parents (regrouping)
Insert	The user sees an insertion marker between items, indicating that dropping the items will insert them between the items	Use when you want to allow users to move or copy items to new locations (reordering)
Control	The user sees the entire control light up, indicating the dropping the items anywhere on the control will cause them to be added to predetermined locations within the tree	Use when you want to allow users to only move from one tree to the other, but you plan to place the items yourself

While we've managed to completely establish the control as a drop target, the end user will still get nothing more than visual feedback. We have a couple choices on how to actually move the items from the Source to the Target.

## Manual Drag and Drop

When the user drops items onto a Target, it fires the [OnDrop](#) event. This event lets you know which items were being dropped as well as the drop source. You can use this information to implement that actual copy or move of items. To copy items, use either the [SendItems](#) or [TransferItems](#) methods. The [SendItems](#) method sends copies of the items from one tree to another. The [TransferItem](#) does the same thing with the addition that it also copies over the items' parents.

```
// Perform a copy
Convert ',' to @FM in Items
SendMessage(SourceCtrl, "OLE.SendItems":@FM:1, Items:@FM:CtrlEntId:@FM:DestItem:@FM:Action:@FM:0:@FM:1)
```

If you want to perform a move, then follow the above method with a call to [RemoveItems](#).

```
// Perform a move
Convert ',' to @FM in Items
SendMessage(SourceCtrl, "OLE.SendItems":@FM:1, Items:@FM:CtrlEntId:@FM:DestItem:@FM:Action:@FM:0:@FM:1)
SendMessage(SourceCtrl, "OLE.RemoveItems", Items)
```

## Automatic Drag and Drop

The SRP Tree Control can do this work for you as long as you only need simple drag and drop. The [AutoDrop](#) property can be set to 1 to enable automatic copy of items from the Source to the Target. The Target tree uses the [DropBehavior](#) property to determine how the items are copied:

DropBehavior	Action	Equivalent Method
Add	Copies of the items are add to the target item	<a href="#">SendItems</a> using "Add" action
Insert	Copies of the items are inserted between the target items	<a href="#">SendItems</a> using "Before" or "After" action
Control	Copies of the items are added to the Target at the same location they appeared in the Source	<a href="#">TransferItems</a>

Note that the items are copied, not moved. If you wish to simulate a move, simply capture the [OnDrop](#) event and call the Source tree's [RemoveItems](#) method.

## See Also

[DragEnabled](#), [DropSources](#), [DragCondition](#), [DropCondition](#), [OnDragStart](#), [OnDragEnter](#), [OnDragExit](#), [OnDrop](#), [ItemDraggable](#), [ItemDroppable](#)