Writing a Custom JS Function

1. In Innovator/Client/Solutions/3DV/Scripts/Controls/TGV/DynamicView.ts add the following function:

```
this.getSelectedItemsHandler = function () {
var ccItem;
const tgvGrid = mainPageExtension.getTgvGrid();
const treeGrid = mainPageExtension.getTreeGrid();
const selectedRowsIds = treeGrid.settings.selectedRows;
const count = selectedRowsIds ? selectedRowsIds.length : 0;

// Add a dummy item to start the result collection (removed later)
var resultItem = aras.newIOMItem("DeleteMe", "DeleteMe");
for (let i = 0; i < count; i++) {
const rowId = selectedRowsIds[i];
const selectedRow = tgvGrid.getRow(rowId);
const dataStr = selectedRow.cells[0].data;
if (dataStr) {
const data = JSON.parse(dataStr);
var thisType = data.type; // this.getType();
var thisId = data.id;
var relshipItemType;
```

```
var sourceItemTypeName;
var isRelationship = false;

// Check whether the method is being called from a relationship grid, the main grid or on an
// individual item and build an array of item ids
var relTypeId = aras.getRelationshipTypeId(thisType);
if (relTypeId) {
isRelationship = true;

relshipItemType = aras.getRelationshipType(relTypeId);
var sourceItemType = aras.getItemTypeDictionary(relshipItemType.getProperty("source_id"), "id");
sourceItemTypeName = sourceItemType.getProperty("name");
}


// When starting from a relationship, get the parent item from the cache and then retrieve the related item
if (isRelationship) {
var parentId = relshipItemType.getProperty("source_id", "");
```

```
var parentItem = aras.getItemById(sourceItemTypeName, parentId, 0);
if (parentItem) {
ccItem = parentItem.selectSingleNode("Relationships/Item[@id='" + thisId + "']");

//If item isn't cached, get
if (!ccItem) {
ccItem = aras.getItemById(thisType, thisId, 0);
}

//get related Part
if (ccItem) {
ccItem = aras.getRelatedItem(ccItem);
}
}
}
else {
ccItem = aras.getItemById(thisType, thisId, 0);
}
```

```
// Add the item to the collection to be returned
if (ccItem) {
if (aras.isNew(ccItem)) {
aras.AlertError(aras.getResource("PLM",
"affecteditem.add_not_saved_item", aras.getKeyedNameEx(ccItem)));
return;
}
var tempItem = aras.newIOMItem("", "");
tempItem.loadAML(ccItem.xml);
resultItem.appendItem(tempItem);
}
}
}

if (resultItem.getItemCount() < 2) { return; }

// Remove the dummy item
resultItem.removeItem(resultItem.getItemByIndex(0));

// Set the ChangeItem attribute for debugging purposes and return the
results
```

```
resultItem.getItemByIndex(0).setAttribute("ChangeItem", "Pending");


var methodArgs = {};
methodArgs.results = resultItem;
results = aras.evalItemMethod(
'PE_ChooseCMItem',
ccItem,
methodArgs
);
return resultItem;
};
```

This function calls a new function which is an adapted version of the standard **PE_GetSelectedItems** Server Method.

2. Create the **dpn_GetSelectedItems** Method of the **JavaScript** Method Type with its execution allowed to the **World** Identity:

```
// MethodTemplateName=CUI;
const dynamicView = options.dpnContext.getDynamicView();
dynamicView.getSelectedItemsHandler ();
```

This Method will be a menu item click handler.