2. Hookflow Script

Hookflow script is used for inserting custom logic at a certain point in the flow

There are *input* and *output* parameters defined in the Hookflow class.
The value from the parameter can be loaded by the *Get()* method.
Set value in the parameter can be done by the Set() methd: BackRequested.Set(true);
It is not possible to define additional input or output parameters.

Customization Articles for WMS scripting:
How to make custom bin location list in Put Away
How to make custom bin location list in AdHoc move

There are two classes in every HookFlow script that are pre defined in the *Execute()* method.
It is the **Session** and the **ISboProviderService** classes.

References:

```
using Produmex.Foundation.SlimScreen;
using Produmex.Foundation.Wwf.Sbo.LocalServices;
```

Remove the comment before the variables in case you would like to use them!

```
Session session = GetScopeParameter("Session") as Session;
ISboProviderService sboProviderService = GetScopeParameter("
<WwfService>ISboService") as ISboProviderService;
```

2.1. Database Connection

Necessary classes:

Class	Reference
PmxDbConnection	`Produmex.Foundation.Data.Sbo;`

You can get the connection from **sboProviderService**.

Example - creating a Picklist provider with connection

```
sboProviderService.InvokeMethodWithDbConnection<object>(false, false,
null, null, delegate (PmxDbConnection conn, object[] parameters)
```

```
{

PmxPickListProvider plProv = new PmxPickListProvider(conn);
PmxPickList pickList = plProv.GetBO( WaveKey.Get() );
return null;
});
```

2.2. Screens

Customization Articles for WMS scripting:
How to display product image after selecting the item in Picking and Ad-Hoc picking

Screen can be generated by the *ShowScreen* method of the *session* object.
There are different types that we can use.

Necessary classes:

Class	Message

Reference	`using Produmex.Foundation.Messages;`
	`using Produmex.Foundation.Wwf.Sbo.LocalServices;`
	`using Produmex.Foundation.SlimScreen;`

2.2.1. Message Screen type

Parameters

Name	Type	Description
MessageKey	String	Set your message
ShowButton	Bool	-

Example:

```
Message msg = null;
session.ShowScreen(typeof(Produmex.Foundation.SlimScreen.Interfaces.ISho
wMessageScreen),
this.DefaultCultureInfo, BuildParamCollection(
     "MessageKey", "YOUR MESSAGE" + DLoc,
     "ShowButton", true
));
msg = WaitForMessage();
```

Result:

2.2.2. Image screen type

Parameters collection

Name	Type	Description
TitleKey	String	Title of the screen
ImagePath	String	Full path of the picture
MessageKey	String	Message under the screen
ShowButton	Bool	-

Example:

```
Message msg = null;
session.ShowScreen(typeof(Produmex.Foundation.SlimScreen.Interfaces.ISho
wImageScreen),
            this.DefaultCultureInfo, BuildParamCollection(
                "TitleKey", "Picture of the product",
                "MessageKey", "message under the picture",
                "ImagePath", "<PATH OF THE IMAGE>",
                "ShowButton", true
            ));
            msg = WaitForMessage();
```

Result:

2.2.3. Enter String Value type

You can capture additional text information on this screen. The captured data can be get from the message object.
The value can be used in the Hookflow for further processing, or if the Hookflow script has an output parameter, then we can put the captured value into the output parameter.

Parameters

Name	Type	Description
InitialErrorKey	String	n.a. in custom usage
TitleKey	String	Title of the screen
Information	String	Additional information on the screen
Parameters	Object of sting	n.a. in custom usage
AllowToGoBack	Bool	-

Name	Type	Description
ForceDataEntry	Bool	-
AllowMultiLine	Bool	-
MinimumNumberOfCharacters	Int	Minimum number of characters that must be typed

Example:
The entered text will be used on a message screen.

```
string initialErrorKey = null;
        string FreeText = "";

        Message msg = null;
session.ShowCustomizedScreen(typeof(Produmex.Foundation.SlimScreen.Inter
faces.IEnterStringValueScreen),
                DefaultCultureInfo.Get(), BuildParamCollection(
                        "InitialErrorKey", initialErrorKey,
                        "TitleKey", "Title of the screen",
                        "Information", "Information text",
                        "Parameters", new object[] { "" },
                        "AllowToGoBack", true,
                        "ForceDataEntry", true,
                        "AllowMultiLine", true,
                        "MinimumNumberOfCharacters", 5
                        ),
                    WorkflowId,
nameof(PickingScript_Screens.EnterStringValueScreen1));
        msg = WaitForMessage();
        if (msg.Name.EndsWith(".StringEntered"))
        {
            FreeText = ExtractParameter<string>(msg.Parameters,
"stringValue");
        }

        msg = null;
session.ShowScreen(typeof(Produmex.Foundation.SlimScreen.Interfaces.ISho
wMessageScreen),
        this.DefaultCultureInfo, BuildParamCollection(
            "MessageKey", "Entered text: " + FreeText,
            "ShowButton", true
        ));
        msg = WaitForMessage();
```

Result:

2.2.4. Select Product Screen type

You can create an item list in a DataSet object to select an item from a list. The captured data can be get from the message object.
The value can be used in the Hookflow for further processing, or if the Hookflow script has an output parameter, then we can put the captured value into the output parameter.

Parameters

Name	Type	Description
InitialErrorKey	String	n.a. in custom usage
TitleKey	String	Title of the screen
Information	String	Additional information on the screen
Parameters	Object of sting	n.a. in custom usage
AllowToGoBack	Bool	-
ForceDataEntry	Bool	-
AllowMultiLine	Bool	-
MinimumNumberOfCharacters	Int	Minimum number of characters that must be typed

Example:
The selected item will be used on a message screen.

```
string initialErrorKey = null;
        string FreeText = "";
        Message msg = null;
        DataSet dsItems = null;

        string query = "SELECT DISTINCT PMX_OITMANAGED_BY_PMX.ItemCode
AS ProductCode, PMX_OITMANAGED_BY_PMX.U_PMX_CUDE AS ProductDescription,
PMX_OITMANAGED_BY_PMX.CodeBars AS GTIN,
PMX_OITMANAGED_BY_PMX.U_PMX_HBBD, PMX_OITMANAGED_BY_PMX.U_PMX_PILR,
PMX_OITMANAGED_BY_PMX.ManBtchNum, PMX_OITMANAGED_BY_PMX.U_PMX_LOUN,
PMX_OITMANAGED_BY_PMX.NumInBuy, PMX_OITMANAGED_BY_PMX.BuyUnitMsr,
PMX_OITMANAGED_BY_PMX.InvntryUom, PMX_OITMANAGED_BY_PMX.CodeBars AS
CodeBars, PMX_OITMANAGED_BY_PMX.ItemName FROM PMX_OITMANAGED_BY_PMX WITH
(NOLOCK)  WHERE PMX_OITMANAGED_BY_PMX.InvntItem = 'Y' AND
PMX_OITMANAGED_BY_PMX.InvntItem = 'Y'  AND  NOT (
PMX_OITMANAGED_BY_PMX.frozenFor = 'Y' AND ( (
PMX_OITMANAGED_BY_PMX.frozenFrom IS NULL OR CURRENT_TIMESTAMP >=
PMX_OITMANAGED_BY_PMX.frozenFrom ) AND ( PMX_OITMANAGED_BY_PMX.frozenTo
IS NULL OR CURRENT_TIMESTAMP < DATEADD( day, 1,
PMX_OITMANAGED_BY_PMX.frozenTo ) ) ) ) ORDER BY ProductDescription";

        dsItems = sboProviderService.RunView(false, null, null, query);
session.ShowCustomizedScreen(typeof(Produmex.Foundation.SlimScreen.Inter
faces.ISelectProductScreen),
            DefaultCultureInfo.Get(), BuildParamCollection(
            "InitialErrorKey", initialErrorKey,
            "TitleKey", "Title of the screen",
```

```
                "ProductDS", dsItems
                ),
    WorkflowId,
        nameof(ChecksScript_Screens.SelectProductScreen1));
    msg = WaitForMessage();

        if (msg.Name.EndsWith(".ProductSelected"))
        {
            FreeText = ExtractParameter<string>( msg.Parameters,
"itemCode" );
        }

        msg = null;
session.ShowScreen(typeof(Produmex.Foundation.SlimScreen.Interfaces.ISho
wMessageScreen),
        this.DefaultCultureInfo, BuildParamCollection(
            "MessageKey", "Entered text: " + FreeText,
            "ShowButton", true
        ));
        msg = WaitForMessage();
```

Result:

2.2.5. Yes/No question Screen type

Parameters

Name,Type,Description TitleKey,String,Name of the screen MessageKey,String,question string

Example:

```
Message msg = null;
session.ShowCustomizedScreen(typeof(Produmex.Foundation.SlimScreen.Interface
s.IDecisionScreen),
DefaultCultureInfo.Get(), BuildParamCollection(
"TitleKey", "Title of the screen",
"MessageKey", "Do you want to continue?"),
WorkflowId,
nameof(ReceptionScript_Screens.DecisionScreen22));
msg = WaitForMessage();

if (msg.Name.EndsWith(".Yes"))
{
// goto Step_ClearDataBeforeNextItem;
```

```
}
if (msg.Name.EndsWith(".No"))
{
BackRequested.Set(true);
}
```

From:
https://wiki.produmex.name/ - **Produmex**

Permanent link:
https://wiki.produmex.name/doku.php?id=implementation:wms:wms_scripting_site:hookflow

Last update: **2024/11/18 09:18**