

Mobile Client Screens

4.2.1. Difference between scanner and touchscreen mode

The application can run either on a scanner or on a touch screen. The scanner mode is used in mobile scanner devices. The touchscreen mode can be used in addition to the scanner, on immobile devices. The minimum screen resolution is 240 x 320 for the scanner mode and 1024 x 768 for the touchscreen mode.

The menu structure is the same on both mode, but the available functions are different. Because the touchscreen is used on immobile devices, flows connected to fixed stations are available in touch screen mode.



4.2.2. Sections of the screen



Title

The title is located on the top of the screen. It can display system commands or the name of the current step. While navigating among the menus, the title shows the menu structure. Next to the menu name, the system also displays the menu page number.



Menu screens have a blue background. Screens in a normal flow have a grey background.

Toolbar



The toolbar is located on the bottom of the screen. When a button is inactive, it is displayed in gray.



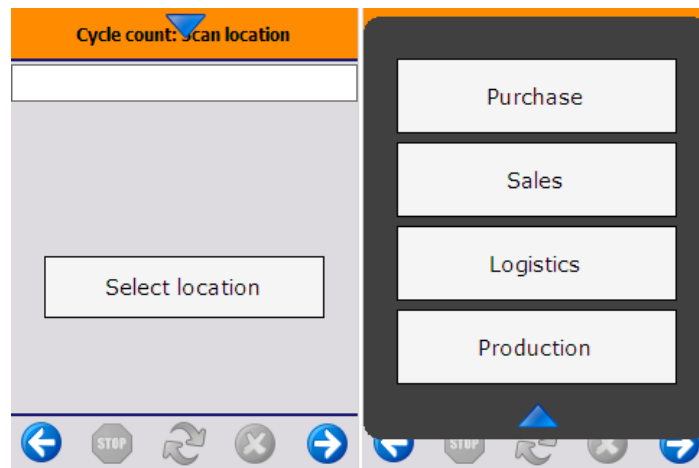
- (1) Left arrow: Displays the previous screen.
- (2) Stop: Exits from the current flow.
- (3) Refresh: Updates the document list.
- (4) Delete: Removes the selected item.
- (5) Right arrow: Displays the next screen.

 Additional toolbar buttons on the touchscreen:

- (1) Left arrow: Displays the previous screen.
- (2) Stop: Exits from the current flow.
- (3) Keyboard: Displays the touchscreen keyboard.
- (4) Clock: Used in the production flows to start the time registration.
- (5) Delete: Removes the selected item.
- (6) Settings (*This button is not currently in use but is reserved for future use.*)
- (7) Refresh: Updates the document list.
- (8) Right arrow: Displays the next screen.

4.2.3. Quick access menu

When hovering over or pressing on the title, a downward arrow appears. Press the arrow to open the Quick Access menu. To close the menu, press the upward arrow on the bottom.



In the standard product the Quick Access menu shows the main menu buttons. The menu can be reached from any flow. When pressing a button on the menu, the user will leave the current flow and all parent flows without any warning and the system will open the selected flow.

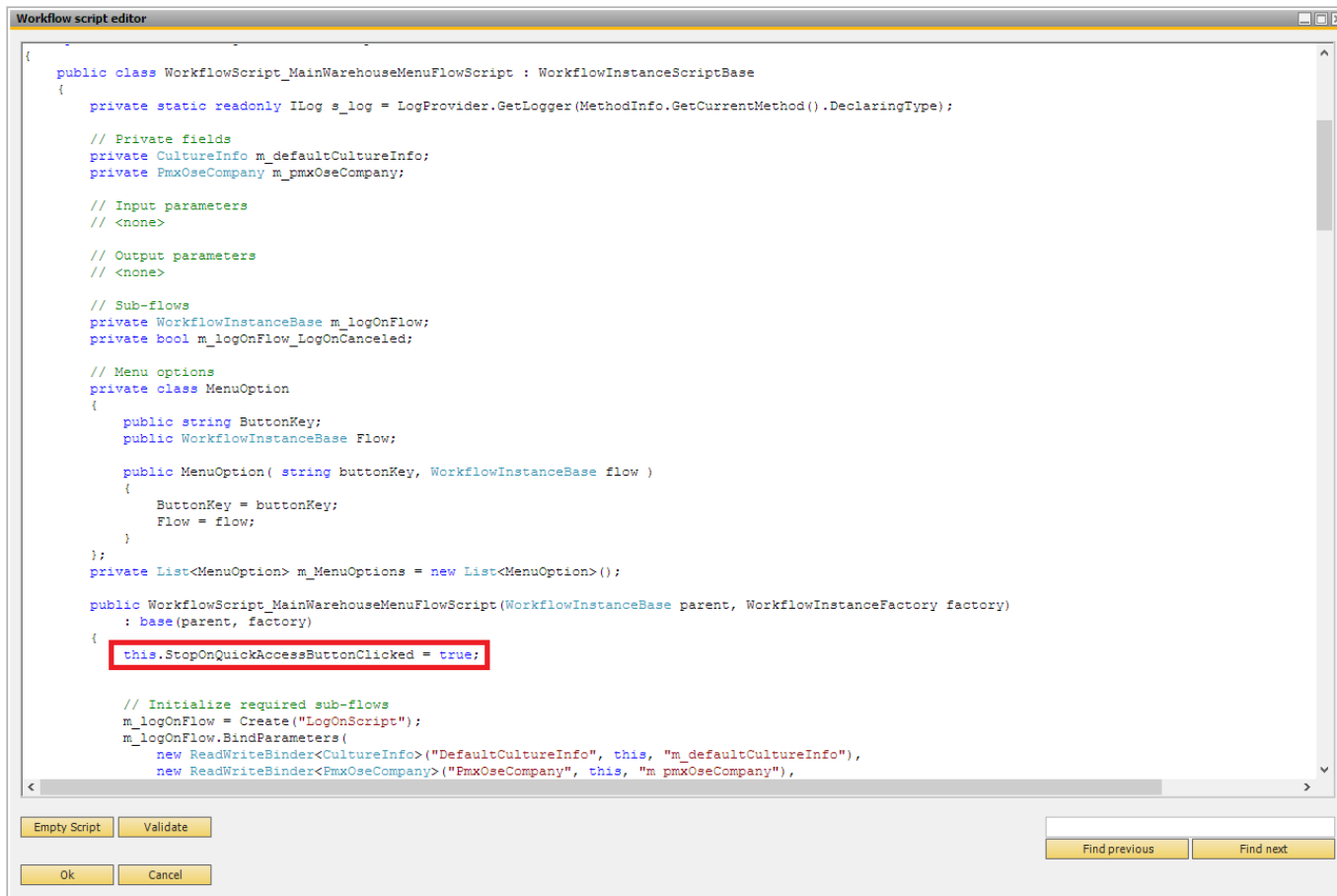
The data registered in the current flow will be lost when clicking on a Quick Access button.

Configuration

When a quick access button is pressed, the current flow will be left, and all parent flows also until there is a flow whose constructor contains the following:

```
this.StopOnQuickAccessButtonClicked = true
```

In the standard product the stop is added in the MainWarehouseMenuFlowScript.

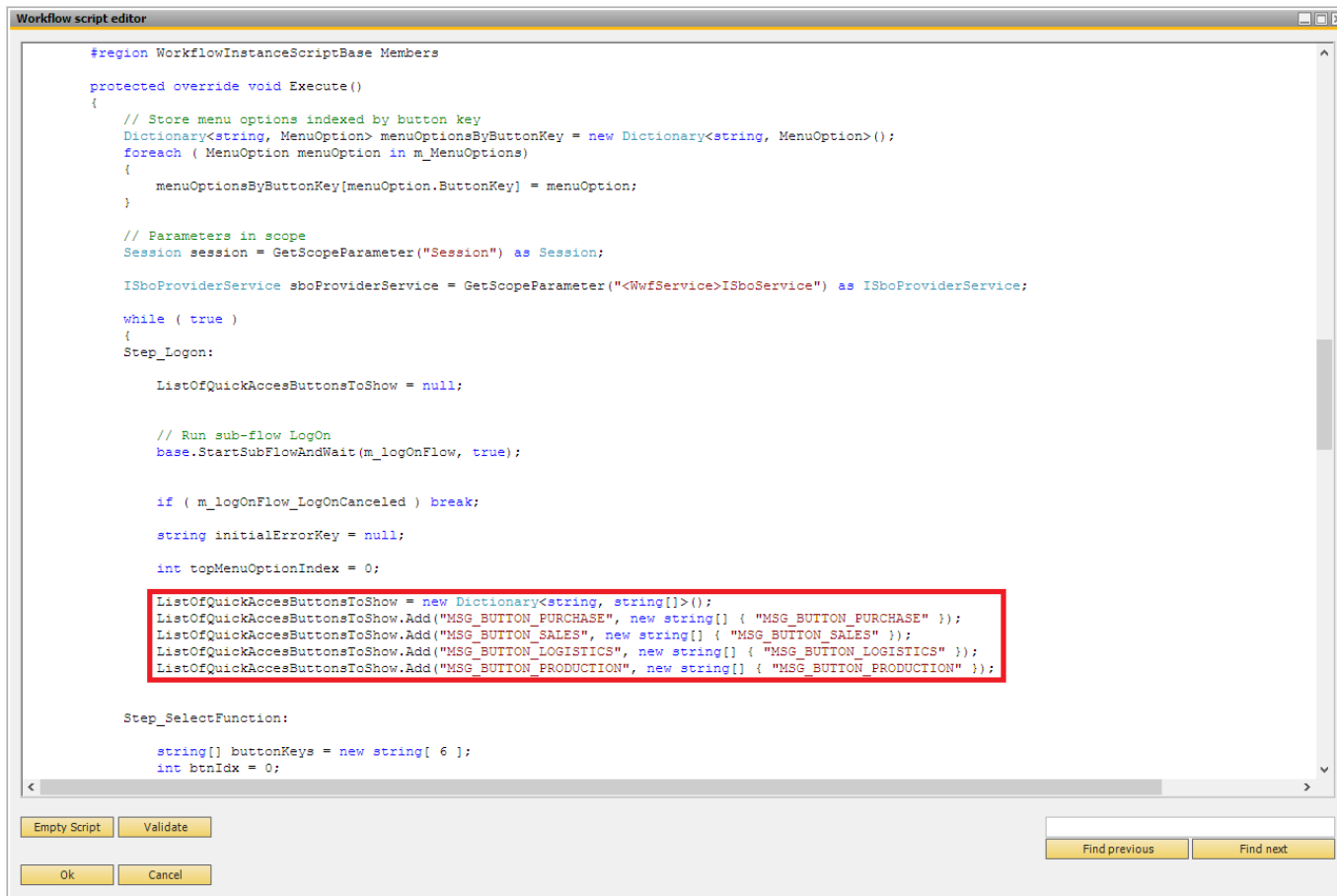


To be able to show the quick menu, a configuration needs to be done on the main flow to add a list of buttons to show on the quick access menu.

The buttons shown in the quick access menu are the ones defined in the ListOfQuickAccesButtonsToShow dictionary. In the standard product the MainWarehouseMenuFlowScript has this configuration after the user logged on:

```
ListOfQuickAccesButtonsToShow = new Dictionary<string, string[]>();
ListOfQuickAccesButtonsToShow.Add("MSG_BUTTON_PURCHASE", new string[] {
"MSG_BUTTON_PURCHASE" });
ListOfQuickAccesButtonsToShow.Add("MSG_BUTTON_SALES", new string[] {
"MSG_BUTTON_SALES" });
ListOfQuickAccesButtonsToShow.Add("MSG_BUTTON_LOGISTICS", new string[] {
"MSG_BUTTON_LOGISTICS" });
ListOfQuickAccesButtonsToShow.Add("MSG_BUTTON_PRODUCTION", new string[] {
"MSG_BUTTON_PRODUCTION" });
```

This builds the list of buttons to show, and the 'path' to get to that flow.



Customization

It is possible to customize the Quick Access menu and show buttons one level deeper or disable quick buttons.

It is recommended to customize the Quick Access menu in a custom flow that is created based on the standard Produmex main flows. For more information about how to customize main flows please see: [5.1.13. Workflows](#)

In the example we will add the Picking flow to the Quick access menu and disable the Purchase, Logistic and Production buttons. To add the Picking flow, insert the following after the other quick access buttons in the workflow script:

```
ListOfQuickAccesButtonsToShow.Add("MSG_BUTTON_PICKING", new string[] {
"MSG_BUTTON_SALES", "MSG_BUTTON_PICKING" });
```

The [translation key](#) of the new button is 'MSG_BUTTON_PICKING'.

The path to the flow is "MSG_BUTTON_SALES", "MSG_BUTTON_PICKING".

After the Picking button is pressed, the system leaves the current flow and all its parent flows until there is a flow with the option

```
'StopOnQuickAccessButtonClicked' = true
```

(With default settings it is the main menu flow). Then the main menu flow will start the flow behind

the "MSG_BUTTON_SALES" button. This flow contains a list of buttons. The system will start the flow behind the "MSG_BUTTON_PICKING" button that was defined in the path.

Please note: If the path does not exist, an error message will be shown.

Quick Access buttons can be disabled in the same way as other menu buttons. Please see: [5.1.13. Workflows.](#)

```

Workflow script editor
#region WorkflowInstanceScriptBase Members
protected override void Execute()
{
    // Store menu options indexed by button key
    Dictionary<string, MenuOption> menuOptionsByButtonKey = new Dictionary<string, MenuOption>();
    foreach ( MenuOption menuOption in m_MenuOptions)
    {
        menuOptionsByButtonKey[menuOption.ButtonKey] = menuOption;
    }

    // Parameters in scope
    Session session = GetScopeParameter("Session") as Session;

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

    while ( true )
    {
        Step_Logon:

        ListOfQuickAccesButtonsToShow = null;

        // Run sub-flow LogOn
        base.StartSubFlowAndWait(m_logOnFlow, true);

        if ( m_logOnFlow_LogOnCanceled ) break;

        string initialErrorKey = null;

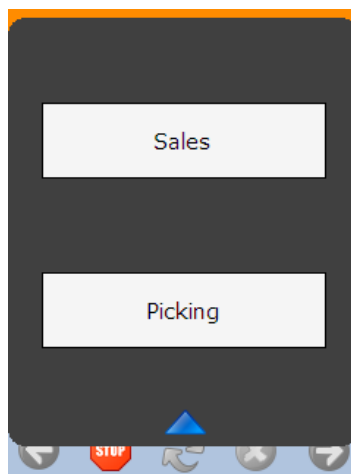
        int topMenuOptionIndex = 0;

        ListOfQuickAccesButtonsToShow = new Dictionary<string, string[]>();
        //ListOfQuickAccesButtonsToShow.Add("MSG_BUTTON_PURCHASE", new string[] { "MSG_BUTTON_PURCHASE" });
        ListOfQuickAccesButtonsToShow.Add("MSG_BUTTON_SALES", new string[] { "MSG_BUTTON_SALES" });
        //ListOfQuickAccesButtonsToShow.Add("MSG_BUTTON_LOGISTICS", new string[] { "MSG_BUTTON_LOGISTICS" });
        //ListOfQuickAccesButtonsToShow.Add("MSG_BUTTON_PRODUCTION", new string[] { "MSG_BUTTON_PRODUCTION" });
        ListOfQuickAccesButtonsToShow.Add("MSG_BUTTON_PICKING", new string[] { "MSG_BUTTON_SALES", "MSG_BUTTON_PICKING" });

        Step_SelectFunction:

        string[] buttonKeys = new string[ 6 ];
    }
}

```



If there are existing customized main flows, this functionality will not be enabled by default, because the customized flows do not contain the configuration, nor does it have the functionality to automatically proceed to the correct flow using the patch defined on the Quick Access menu.

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

Permanent link:
<https://wiki.produmex.name/doku.php?id=implementation:wms:fatclientscreen>

Last update: **2021/10/21 11:38**

