

The requirement is simple: Create a mobile application, that will allow you to view/edit SAP Equipment Masters using a Windows Based mobile device.

In this example, I am using a Motorola Handheld device, with a built in Barcode Scanner, Camera, 3G and WiFi capable.

To put the scenario into perspective: Let's assume you have to manage and create new assets in your organization, and you want to create those assets in a register, stored as SAP Equipment Masters.

What you need:

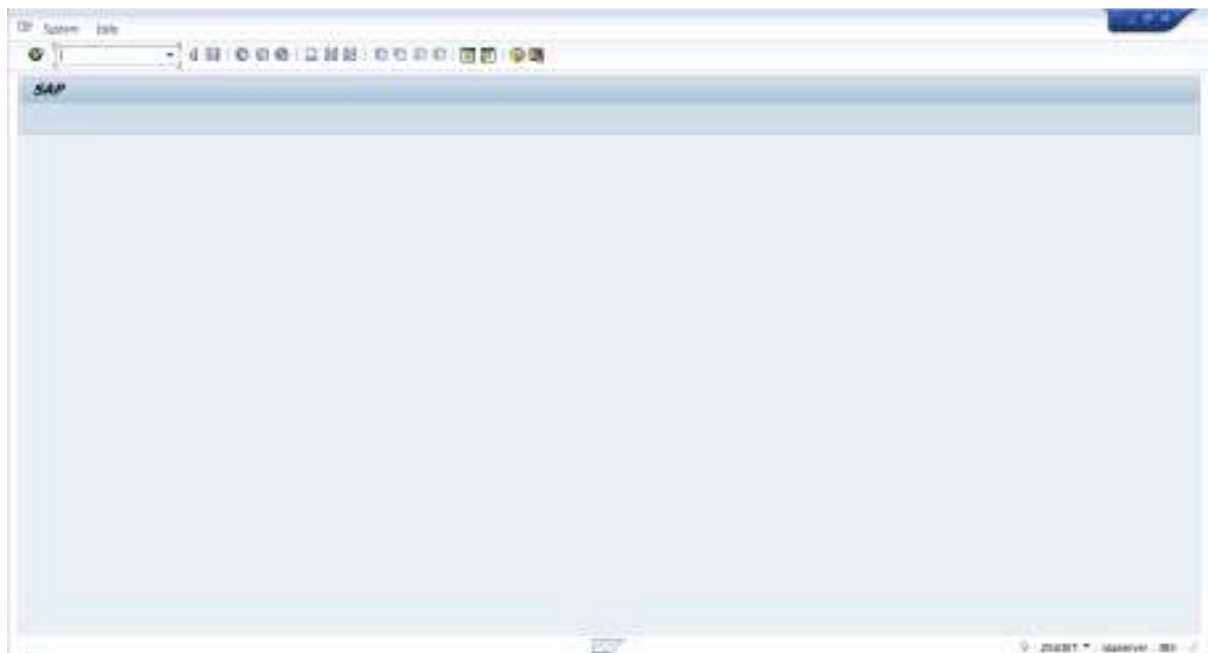
- One hand held Windows Based Barcode Scanner
- GuiXT Mobile (Loaded onto the device)
- GuiXT Designer
- GuiXT Server (Interpretation Software for GuiXT Simplified Screens)
- A GuiXT Programmer, like myself

Very little code is actually written by myself, as the Designer generates most of the code for me.

What you need to do:

I prefer always starting with a SAP ZTransaction as my base, to create a Launchpad. From here, I can create navigational steps, and provide the user with an easy to use navigation.

I create a Program, ZGUIXT, in ABAP, and give it a screen, 100. Activate the program. Now navigate to SE93 and create a transaction to call the program, I used ZGUIXT. When you execute the transaction, this is what you will see



Open a text editor, and paste the following code into it:

```

// Defaults Paths - This will tell GuiXT where our scripts are stored,
images are stored, etc.
set V[z_img_path] "C:\GuiXT\scripts\mobileassetmanagement\images"
set V[z_script_path] "C:\GuiXT\scripts\mobileassetmanagement"

// Solution Overview - To display/create an equipment master in SAP, we
would need a Launch Screen (1) that would allow us to navigate back and
forth. We would need a Create Screen (2) that allows us to capture
information, a Search Screen (3) that allows us to scan a barcode which we
will then submit to retrieve the details on a Detail Screen (4)

I am using a single transaction and screen, but by using display levels, I
can conditionalise which information I am processing, and thus emulate as
if I am using multiple screens.

By assigning a value to a variable, and testing that variable, I can then
determine which screen elements I want to show/hide

// Initialise Launchpad
if not V[z_disp_level]
  set V[z_disp_level] "1"          // display level = main menu
endif

// Determine what screen should be displayed
if V[z_disp_level=1]
  goto DisplayMain
endif

if V[z_disp_level=2]
  goto CreateEquipment
endif

if V[z_disp_level=3]
  goto ScanEquipment
endif

if V[z_disp_level=4]
  goto DisplayEquipment
endif

goto EndDisplay

// On my main screen, i.e. the launch pad, I want two pushbuttons, one to
create, and one to display an equipment. I also want to be able to log off
of the system at this point in time.

I can do some additional things here, like providing icons for my buttons,
changing my screen title, create group boxes etc.

// ***** Display main screen *****

label DisplayMain

// display screen title
title "Asset Management"

box (0,1) (11,31) "Select A Function"

```

```
// For my pushbuttons, I need to tell them that they need to show/hide
certain elements when pressed. To do this, I pass a value to a text file,
and I interrogate the text file, to see which level I want to display.
```

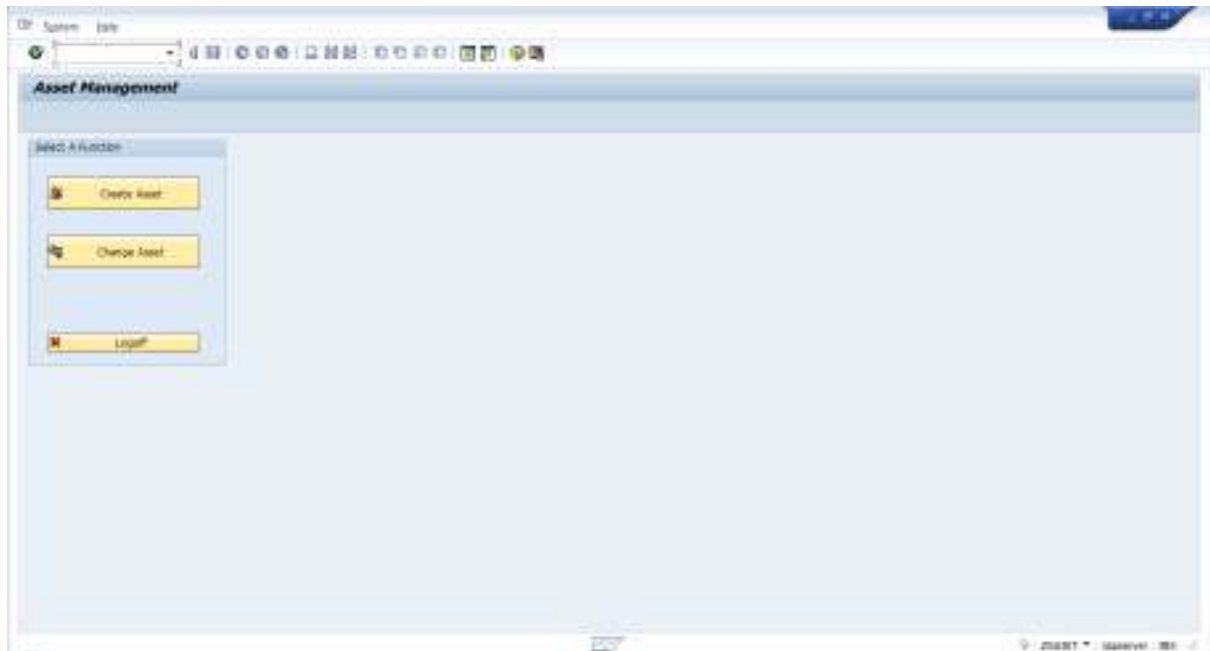
```
pushbutton (2,4) "@9X@Create Asset "
process="&V[z_script_path]\selectdisplay.txt" size="(2,24)"
    Using ZDISPLAY = "2"

pushbutton (5,4) "@AJ@Change Asset "
process="&V[z_script_path]\selectdisplay.txt" size="(2,24)"
    Using ZDISPLAY = "3"

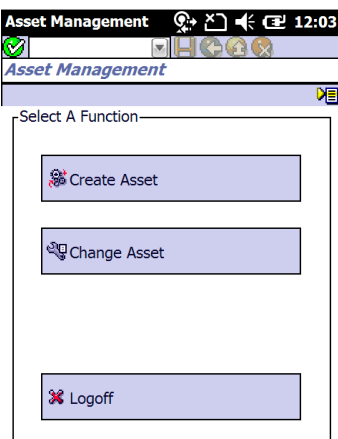
pushbutton (10,4) "@0W@Logoff " "/nex" size="(1,24)"

goto EndDisplay
```

The end result is a screen that looks like this:



On the device, it looks like this:



The process is as follows:

- User presses “Create Asset”
- Variable is passed to text file with a value
- Value is interpreted and you return to the start of the script
- Test the value and based on that, perform a part of the code

To Create an Equipment, we need to tell GuiXT to show the elements of level 2.

When we return to the start of the script, it will process the following code and match on the second condition and “goto” CreateEquipment as shown below:

```
// Determine what screen should be displayed
```

```
if V[z_disp_level=1]
```

```
    goto DisplayMain
```

```
endif
```

```
if V[z_disp_level=2]
```

```
    goto CreateEquipment
```

```
endif
```

```
if V[z_disp_level=3]
```

```
    goto ScanEquipment
```

```
endif
```

```
if V[z_disp_level=4]
```

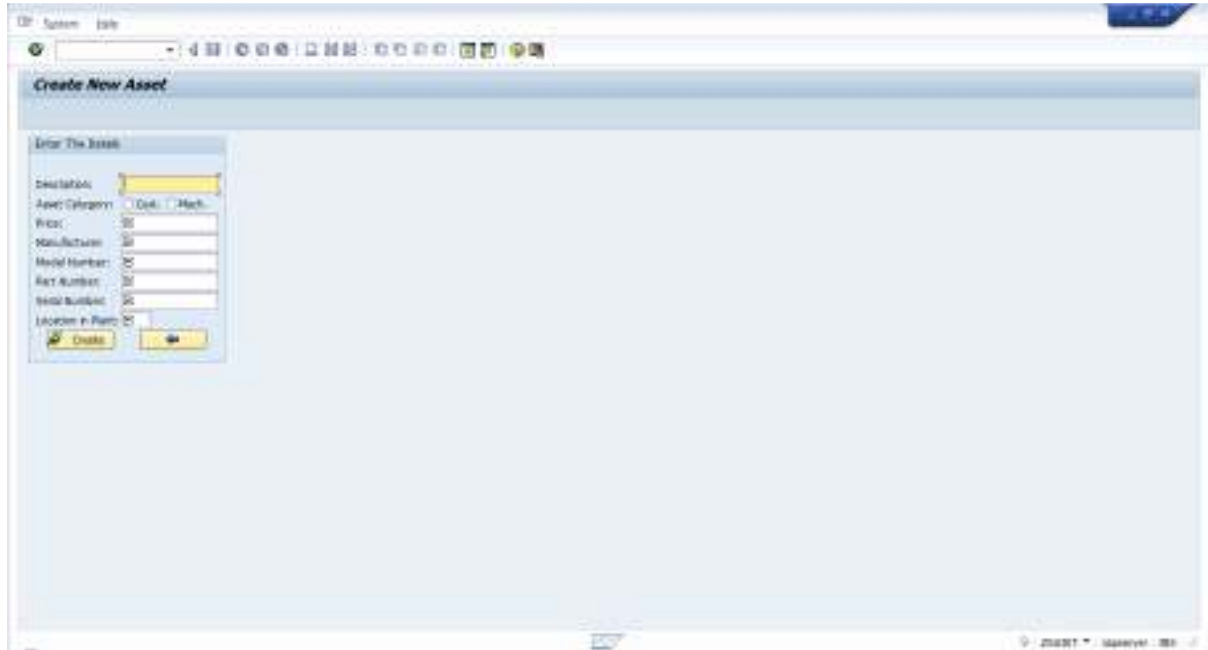
```
    goto DisplayEquipment
```

```
endif
```

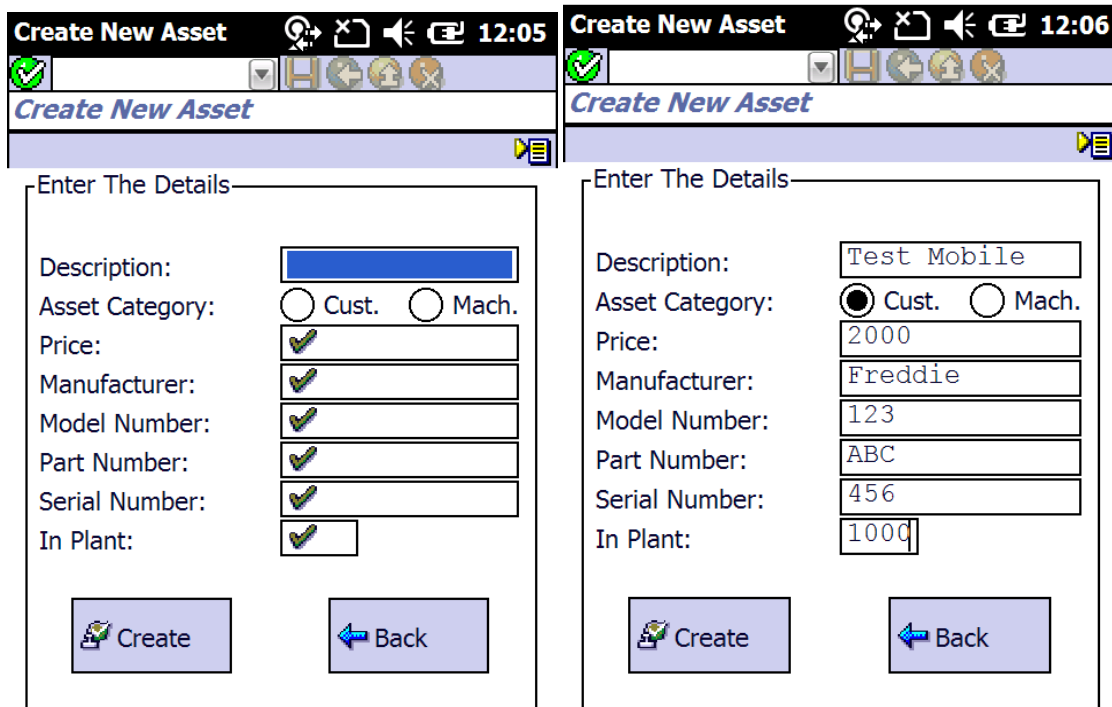
```
goto EndDisplay
```

// The result that we want to achieve is shown below. The analysis is as follows, to create an equipment, we need to execute transaction IE01. We need to then populate certain information, and save, which will create the equipment master in SAP.

We will need a pushbutton to "Create" the equipment master in SAP. No need to code the action that needs to be performed, as we will simply record the process to create an equipment, and then replace the recorded values with variable values that we create.



On the device, it looks like this:



```

// ***** Create A New Asset Master *****
Label CreateEquipment
if not V[z_disp_level=2]
    goto EndDisplay
endif

title "Create New Asset"
box (0,1) (11,31) "Enter The Details"

// Asset Information

set V[z_disp_line] "2" // display at line number

// Asset Description
text (&V[z_disp_line],2) "Description:" size="12"
inputfield (&V[z_disp_line],16) name="z_equi_descr" size="15" -required -
nolabel
set V[z_disp_line] "&V[z_disp_line]" + 1 // next display line

// Asset Type
text (&V[z_disp_line],2) "Asset Category:" size="14"
RadioButton (&V[z_disp_line],16) "Cust." name="z_equi_category" value="S"
// to define an InputAssistant variable
RadioButton (&V[z_disp_line],23) "Mach." name="z_equi_category" value="M"
// to define an InputAssistant variable
set V[z_disp_line] "&V[z_disp_line]" + 1 // next display line

// Asset Price
text (&V[z_disp_line],2) "Price:" size="12"
inputfield (&V[z_disp_line],16) name="z_equi_price" size="15" -required -
nolabel
set V[z_disp_line] "&V[z_disp_line]" + 1 // next display line

// Asset Manufacturer
text (&V[z_disp_line],2) "Manufacturer:" size="12"
inputfield (&V[z_disp_line],16) name="z_equi_man" size="15" -required -
nolabel
set V[z_disp_line] "&V[z_disp_line]" + 1 // next display line

// Asset Model Number
text (&V[z_disp_line],2) "Model Number:" size="12"
inputfield (&V[z_disp_line],16) name="z_equi_model" size="15" -required -
nolabel
set V[z_disp_line] "&V[z_disp_line]" + 1 // next display line

// Asset Part Number
text (&V[z_disp_line],2) "Part Number:" size="12"
inputfield (&V[z_disp_line],16) name="z_equi_part" size="15" -required -
nolabel
set V[z_disp_line] "&V[z_disp_line]" + 1 // next display line

// Asset Serial Number
text (&V[z_disp_line],2) "Serial Number:" size="12"
inputfield (&V[z_disp_line],16) name="z_equi_serial" size="15" -required -
nolabel
set V[z_disp_line] "&V[z_disp_line]" + 1 // next display line

// Asset Location
text (&V[z_disp_line],2) "Location in Plant:" size="12"

```

```

inputfield (&V[z_disp_line],16) name="z_equi_location" size="4" -required -
nolabel
set V[z_disp_line] "&V[z_disp_line]" + 1 // next display line

pushbutton (10,4) "@8X@Create " "/nIE01"
process="&V[z_script_path]\createequipment.txt" size="(1,10)"
pushbutton (10,19) "@9S@ " "/nzguixt"
process="&V[z_script_path]\backbutton.txt" size="(1,10)"

```

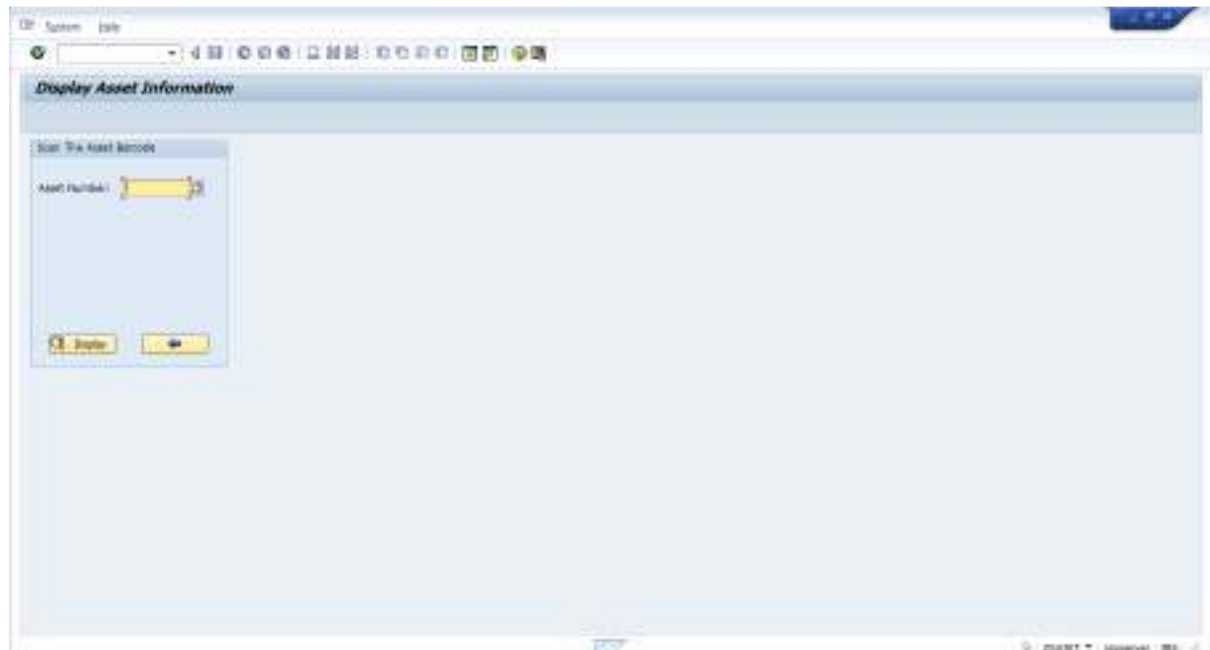
// Each field on the screen will need an associated variable, which will hold the value that the user enters into it. These values will then replace the values in the recorded transaction, thus making it completely dynamic.

When the user presses the "Create Button" we will execute an input script, that performs the creation function.

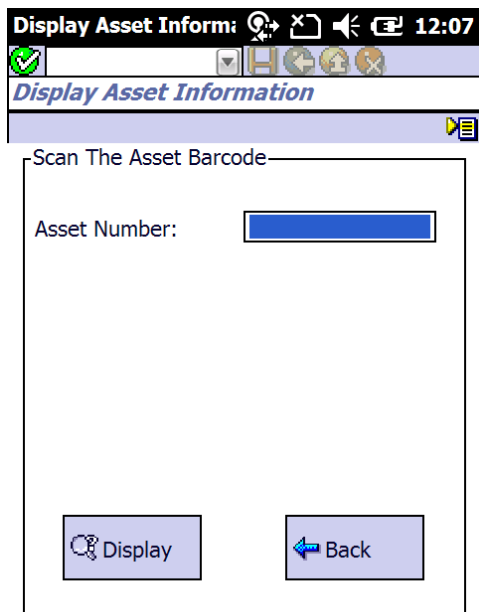
If the user presses the "Back Button" we will change the value of the display level to the current value - 1 so that the user is returned to the level that they came from.

Next in the solution, we want the ability to "Scan" a barcode, and then, based on the retrieved value, execute transaction IE03, and display information about the equipment to the user on the screen.

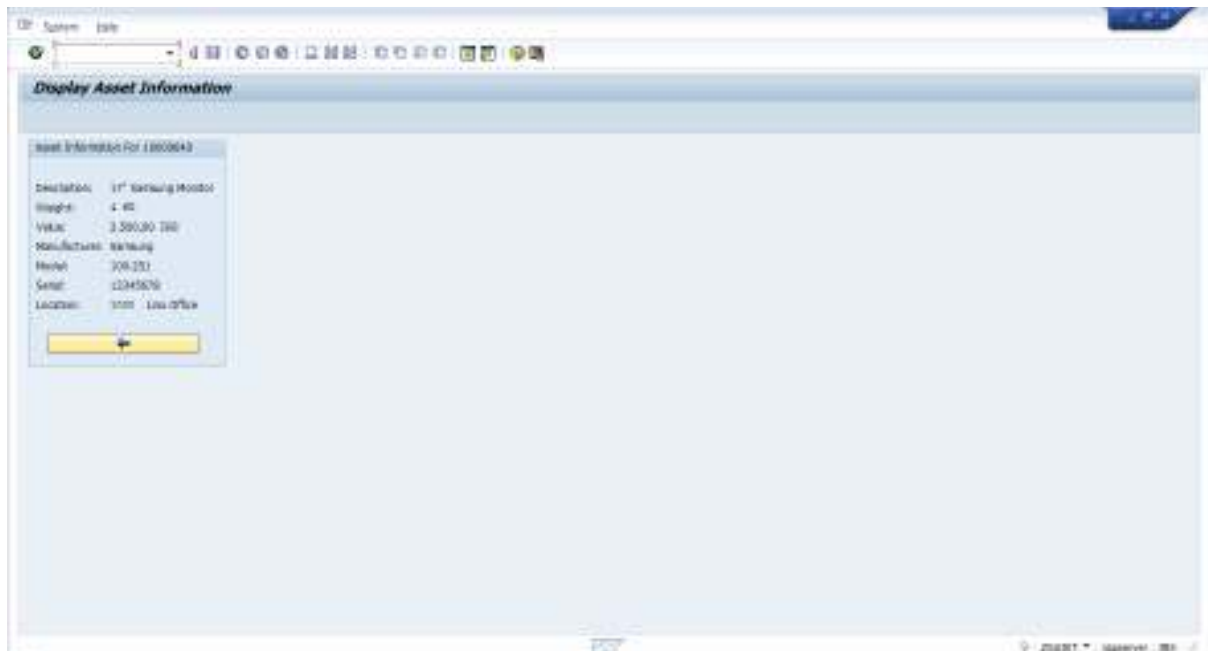
Initially, this is the screen that we want to see:



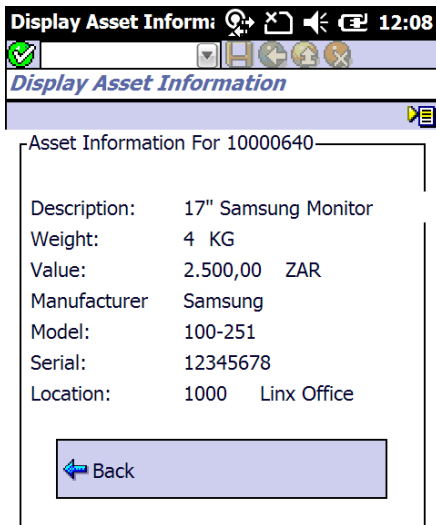
On the device, it looks like this:



And then when the user presses "Display" we want to see this:



On the device, it looks like this:



Again, the reading of the information from the equipment master is done using GuiXT's InputAssistant technology, which simply runs a recording of the transaction.

```
// ***** Scan A Current Asset Barcode *****
Label ScanEquipment
if not V[z_disp_level=3]
    goto EndDisplay
endif

title "Display Asset Information"
box (0,1) (11,31) "Scan The Asset Barcode" //10000640 Can be used
as an example

set V[z_disp_line] "2" // display at line number

text (&V[z_disp_line],2) "Asset Number:" size="14"
inputfield (&V[z_disp_line],16) name="z_get_equi" size="10" -required -
nolabel searchhelp="EQUI"
set V[z_disp_line] "&V[z_disp_line]" + 1 // next display line

pushbutton (10,4) "@GS@Display " "/nIE03"
process="&V[z_script_path]\showequipment.txt" size="(1,10)"
pushbutton (10,19) "@9S@ " "/nzquixt"
process="&V[z_script_path]\backbutton.txt" size="(1,10)"
```

After the user presses the Display button, this is the code that will be executed:

```
// ***** Display A Current Asset*****
Label DisplayEquipment
if not V[z_disp_level=4]
    goto EndDisplay
endif

title "Display Asset Information"
box (0,1) (11,31) "Asset Information For &V[z_get_equi]"
//10000640 Can be used as an example

set V[z_disp_line] "2" // display at line number
```

```

//text (&V[z_disp_line],2) "Asset Number:" size="14"
//inputfield (&V[z_disp_line],14) name="z_get_equi" size="18" -required -
nolabel searchhelp="EQUI"
//set V[z_disp_line] "&V[z_disp_line]" + 1 // next display line

// Basic Equipment Data

text (&V[z_disp_line],2) "Description:" size="10"
text (&V[z_disp_line],14) "&V[z_sap_equidesc]"
set V[z_disp_line] "&V[z_disp_line]" + 1 // next display line

// Weight
text (&V[z_disp_line],2) "Weight:" size="10"
text (&V[z_disp_line],14) "&V[z_sap_equiweight]"
text (&V[z_disp_line],16) "&V[z_sap_equiweightdesc]"
set V[z_disp_line] "&V[z_disp_line]" + 1 // next display line

// Value
text (&V[z_disp_line],2) "Value:" size="10"
text (&V[z_disp_line],14) "&V[z_sap_equiprice]"
text (&V[z_disp_line],22) "&V[z_sap_equicurrency]"
set V[z_disp_line] "&V[z_disp_line]" + 1 // next display line

// Manufacturer Details
text (&V[z_disp_line],2) "Manufacturer:" size="10"
text (&V[z_disp_line],14) "&V[z_sap_equimanname]"
set V[z_disp_line] "&V[z_disp_line]" + 1 // next display line

text (&V[z_disp_line],2) "Model:" size="10"
text (&V[z_disp_line],14) "&V[z_sap_equimanmodel]"
set V[z_disp_line] "&V[z_disp_line]" + 1 // next display line

text (&V[z_disp_line],2) "Serial:" size="10"
text (&V[z_disp_line],14) "&V[z_sap_equimanserial]"
set V[z_disp_line] "&V[z_disp_line]" + 1 // next display line

// Location Details
text (&V[z_disp_line],2) "Location:" size="10"
text (&V[z_disp_line],14) "&V[z_sap_equiplant]"
text (&V[z_disp_line],20) "&V[z_sap_equiplantdescription]"
set V[z_disp_line] "&V[z_disp_line]" + 1 // next display line

pushbutton (10,4) "@9S@ " "/nzquixt"
process="&V[z_script_path]\backbutton.txt" size="(1,24)"

Label EndDisplay

//***** EOF *****

```

I hope you enjoyed that quick walk through.

If you need any additional information, please do not hesitate to contact me.

Freddie Botha

+27 72 733 3034

freddieb@documation.co.za