Offline Implementations

5.01.01 Standard Implementation

The standard implementation for Offline is one in which a user would install Offline and then may also need to manually configure the parameters via the config.js files previously discussed. It is not always necessary to manually configure the config.js file; often the default parameters plus the user name and connection information input via the Connection Screen are sufficient. Once the installation is properly configured, the user would then need to build the following:

- Create transaction objects for any SAP transactions that will be performed
- Create a database object that will in turn create the database for the given transaction.
- Create the necessary screens for the given transaction by using screen objects. It is necessary to create as many screens as the transaction in question requires
- Screen elements in the given screens are then mapped to the columns in the database for that transaction

Creating Transaction Object

As an example, we will first create a transaction object. This particular transaction object refers to a work order and takes the work order's transaction code as an argument.

```
objSelWkdr = new SR3TransactionObject(strWorkOrderTCode);
```

Building a Database

We then build a database for this new transaction object. In our example, we are building a database that will hold a range of work order transaction codes:

```
objSelWkdr.addDatabaseTable([
{fieldname:"g_wo_no",columntitle:"Work-OrderNumber",columnwidth:8,fiel
dtype:DBF_STRING,fieldlength:10,keytype:DBT_NON_PRIMARY_KEY,columntype
:CTRL_TABLE_EDIT,columnstate:CTRL_CS_ENABLE },
{fieldname:"g_wo_activity",columntitle:"Activity",columnwidth:8,fieldt
ype:DBF_STRING,fieldlength:10,keytype:DBT_NON_PRIMARY_KEY,columntype:C
TRL_TABLE_EDIT,columnstate:CTRL_CS_ENABLE},
{fieldname:"g_wo_location",columntitle:"FuncLoc",columnwidth:10,fieldt
ype:DBF_STRING,fieldlength:30,keytype:DBT_NON_PRIMARY_KEY,columntype:C
TRL_TABLE_EDIT,columnstate:CTRL_CS_ENABLE},
{fieldname:"g_sys_status_info",columntitle:"StatusInfo",columnwidth:10
,fieldtype:DBF_STRING,fieldlength:28,keytype:DBT_NON_PRIMARY_KEY,colum
ntype:CTRL_TABLE_EDIT,columnstate:CTRL_CS_ENABLE },
{fieldname:"g_order_info_second",columntitle:"ord-infosec",columnwidth
:8,fieldtype:DBF_STRING,fieldlength:8,keytype:DBT_NON_PRIMARY_KEY,colu
mntype:CTRL_TABLE_EDIT,columnstate:CTRL_CS_ENABLE} ]);
```


(c) 2026 Liquid UI | Synactive | GuiXT <dev@guixt.com> | 2026-11-02 14:01
URL: https://www.liquid-ui.com/knowledge_base/content/136/1069/en/50101-standard-implementation.html

Creating Screens

After creating the database, it is necessary to create the screens for a given transaction. These screens, as defined previously, are built using screen objects. The screen objects are defined as follows:

```
objSelWkdr.addScreen([
{type:SCR_CAPTION,label:"Work Order List"},
{type:SC
R_TABLE,caption
:"View and Select Work Orders"
,position:{
top:0,left:0,bottom:10,right:54},
linkedtableinfo:strW
orkOrderTCode,linkedshowinfo:[
"g_wo_no
","g_wo_activity
","g_order_info_third","g_wo_location"],fieldname:UNIQUE_TABLE_NAME},
{type:SCR_ONAPPKEY,keyid:SHIFT + F5,keylabel:"@3D@Configure",keytoolti
p:KEYTOOLTIP_O
NLINE,callback:navigate_transaction,args:[strWorkOrderTCode,2,true]},
{type:SCR_ONAPPKEY,keyid:SHIFT + F2,keylabel:"@0H@Download",fieldname:
"g_dwnldwkdr",keytooltip:KEYTOOLT IP_ONLINE,callback:download_wkdr},
{type:SCR_ONFKEY,keyid:EVENT_FKEY_BACK,keylabel:"Back",keytooltip:"Bac
k",callback:navigate_transaction,args:["menu",2,true]<}]);
```

The screen we created is the screen holding a table that will list all the work orders that the user can select. Art this point, we have only built an empty database - there is no data yet contained within the database. To download the data from SAP, it is necessary to build the screen into which the user can enter the parameters defining what data will be downloaded from SAP. This step is optional - if a user wishes to narrow the range of data to be downloaded from SAP, it is possible, but it is also possible to download all data without specifying any ranges at all. The screen where a user can define the parameters of the data to be downloaded is as follows:

```
objSelWkdr.addScreen([
{type:SCR_CAPTION,label:"Configuration Screen"},
{type:SCR_EDIT,label:"Work o
rder",position:{
row:1,col:1,edcol:20,edlen:10},
fieldname:"z_mobile_wo",maxtextlength:10,required:0},
{type:SCR_EDIT,label:"Plant",posi
tion:{row:2,col:1,edcol:20,edlen:10},
fieldname:"z_plant",maxtextlength:10, required:0},
{type:SCR_EDIT,label:"Work
Center",position:{
```


(c) 2026 Liquid UI | Synactive | GuiXT <dev@guixt.com> | 2026-11-02 14:01
URL: https://www.liquid-ui.com/knowledge_base/content/136/1069/en/50101-standard-implementation.html

```
row:3,col:1,edcol:20,edlen:10},
fieldname:"z_work_ctr",maxtextlength:10,required:0},
{type:SCR_EDIT,labe
l:"Functional Location",position:{
row:4,col:1,edcol:20,edlen:10},
fieldname:"z_mobile_floc",maxtextlength:10,required:0},
{type:SCR_EDIT,label:"Equipme
nt",position:{
row:5,col:1,edcol:20,edlen:10},
fieldname:"z_mobile_equipment_no",maxtextlength:10,required:0},
{type:SCR_EDIT,label:"Order
Type",position:{
row:6,col:1,edcol:20,edlen:10},
fieldname:"z_mobile_otype", maxtextlength:10,required:0},
{type:SCR_EDIT,label:"To",position:{
row:6,col:35,edcol:40,edlen:10},
fieldname:"z_mobile_otype2",maxtextlength:10,required:0},
{type:SCR_EDIT,label:"Dow
nload Rows",position:{
row:11,col:1,edcol:20,edlen:10},
fieldname:"z_download_rows",maxtextlength:10,required:1},
{type:SCR_EDIT,label:"From Da
te",position:{
row:8,col:1,edcol:20,edlen:10},
fieldname:"z_mobile_from_date",maxtextlength:10,required:0},
{type:SCR_EDIT,label:"To Date",
position:{
row:9,col:1,edcol:20,edlen:10},fieldname:"z_mobile_to_date",maxtextlen
gth:10,required:0},
{type:SCR_LABEL,label:"MM/D
D/YYYY",position:{row:8,col:31},{
type:SCR_LABEL,label:"MM/DD/YYYY",position:{row:9,col:31},
{type:SCR_ONFKEY,keyid:EVENT_FKEY_CANCEL,keylabel:"Cancel",keytooltip:
"Cancel", callback:SR3GenericFunctionKeyF12Cancel},
{type:SCR_ONFKEY,keyid:EVENT_FKEY_EXIT,keylabel:"Exit",keytooltip:"Exi
t",callback:SR3GenericFunctionKeySHIFTF3Exit},
{type:SCR_ONFKEY,keyid:EVENT_FKEY_BACK,keylabel:"Back",keytooltip:"Bac
k",callback:navigate_transaction,args:[strWorkOrderTCode,1,true]},
{type:SCR_ONFKEY,keyid:EVENT_FKEY_SAVE,keylabel:"Save",keytooltip:"Sav
e",callback:save_wkdr_criteria},
{type:SCR_FUNCTION,funcname:FUNCTION_PBO,callback:display_wkdr_criteri
a}]);
```

Downloading Data from SAP

The next piece of code that will need to be added is the actual function that

(c) 2026 Liquid UI | Synactive | GuiXT <dev@guixt.com> | 2026-11-02 14:01
URL: https://www.liquid-ui.com/knowledge_base/content/136/1069/en/50101-standard-implementation.html

performs the download of the data from SAP. The sample download function shown in the example below includes the generic download function 'SR3GenericDownloadProcessConfigurable' as defined earlier in this document. It also includes code that will refresh the onscreen table and includes variables that can hold the user's logon credential as well:

```
function download_wkdr()
{
 System.TraceOutput(">>> download_wkdr()...\n");
  //user logon credentials
  var xusr = SR3_arConnections
  [this.m_CurrentTransaction.m_strConnectionName].m_strLogonUserName;

  var x pwd = SR3_arConnections
  [this.m_CurrentTransaction.m_strConnectionName].m_strLogonPassword;

  var dpwd = System.Decrypt(xpwd);
  //generic download function
  var arDLResult  = SR3GenericDownloadProcessConfigurable.call
  (this,S1,xusr,dpwd,"wkdr",arCriteria,["g_wo_no","g_wo_activity"]);
//refreshes table if one exists
  var vTableDefinition = this.findTableDefinition();
  var objTableControl = this.m_RealTimeScreen.FindTableListControl( vT
ableDefinit    ion.position.top, vTableDefinition.position.left);
  if (objTableControl != null)
  {
  objTableControl.RemoveControl();
  objTableControl = null;
  if (this.createTableListControl(vTableDefinition) == false)
  {
   return;
  }
  }
  this.onSendScreen(false,"Downloaded: "+(arDLResult[0]) + "/" +(arDLR
esult[0]));
  }
```

Once the data is downloaded, the user can then commence to modify or otherwise work on the data in question.

Unique solution ID: #2072
Author: sarvani.kusuri@guixt.com
Last update: 2021-06-03 17:46

(c) 2026 Liquid UI | Synactive | GuiXT <dev@guixt.com> | 2026-11-02 14:01
URL: https://www.liquid-ui.com/knowledge_base/content/136/1069/en/50101-standard-implementation.html