Offline Configuration

4.02.04 Maintaining Record Structures

We have created the basic outlines for a Master Record structure. Now we will need to define the structures associated with the Master Record and with its Addendums. There are several elements concerned. These are discussed in the following section.

- **Name:** Each application and Addendum will have a unique name. In our example, the application will be named 'Work Order; the first Addendum will be 'Parts'. The code is as follows:

  ```
  g_WorkOrderAppDesc.Name = "Work Order";
  g_WorkOrderAppDesc.Addendum[0].Name = "Parts";
  ```

- **Pattern:** The Pattern, as explained previously, differentiates a Master Record from an Addendum and also identifies what kind of screen is created for that Addendum. There two possible styles. For our Master Record screen, which will contain edit fields, lookup tables, checkboxes and radiobuttons, the following code would be used:

  ```
  g_WorkOrderAppDesc.Addendum[0].Pattern = APPLICATION_PATTERN_TAB
  LE_STYLE;
  ```

  For the first Addendum, however, screen consisting of elements represented in table format, the following code would be used:

  ```
   g_WorkOrderAppDesc.Addendum[1].Pattern = ADDENDUM_PATTERN_NORMA
  L_STYLE;
  ```

- **Database Structure:** Each record and Addendum contains a data table. The Master Record and Addendum screens are created from these data tables. An example of the data table for our Master Record is shown below:

  ```
  g_WorkOrderAppDesc.MasterRecord.Database = [
  //Label Text for FieldName Definition Display width for Field le
  ngth Should the field Should this control Is this field
  //EDIT control edit control and show in table use the same line
  shared in Addendum a hidden field?
  // table column control ? as its preceding control? transaction?
  {label:"Order",fieldname:"g_wo_no",displaywidth:8,fieldlength:10
  ,showintable:1, sharedfield:1},
  {label:"Activity",fieldname:"g_wo_activity",displaywidth:8,field
  length:10,showintable:1,sharedfield:2},
  {label:"Location",fieldname:"g_wo_location",displaywidth:10,fiel
  dlength:30,showintable:1},
  {label:"Time",fieldname:"g_wo_time",displaywidth:8,fieldlength:1
  0,showintable:1},
  {label:"Status",fieldname:"g_sys_status_info",displaywidth:10,fi
  eldlength:30,hidden:1 },
  {label:"Info_2",fieldname:"g_order_info_second",displaywidth:8,f
  ieldlength:10},
  ```


(c) 2026 Liquid UI | Synactive | GuiXT <dev@guixt.com> | 2026-11-02 20:01

URL: https://www.liquid-ui.com/knowledge_base/content/130/1057/en/40204-maintaining-record-structures.html

```
{label:"Info_3",fieldname:"g_order_info_third",displaywidth:10,f
ieldlength:30,usesameline:1},
{label:"Grp_Info",fieldname:"g_planner_group_info",displaywidth:
8,fieldlength:10},
{label:"Grp_Div",fieldname:"g_planner_group_info_div",displaywid
th:4,fieldlength:10, usesameline:1},
{label:"Ctr_Info",fieldname:"g_Mn_Wk_ctr_info",displaywidth:8, f
ieldlength:10},
{label:"Ctr_Div",fieldname:"g_Mn_Wk_ctr_info_div",displaywidth:4
,fieldlength:10,usesameline:1},
{label:"Per_Info",fieldname:"g_person_responsible_info",displayw
idth:10,fieldlength:30},
{label:"Per_Div",fieldname:"g_person_responsible_info_div",displ
aywidth:10,fieldlength:10, usesameline:1},
{label:"Ntf_Info",fieldname:"g_notification_info",displaywidth:8
,fieldlength:10},
{label:"Pri_Info",fieldname:"g_primary_accType_info",displaywidt
h:3,fieldlength:10, usesameline:1},
{label:"Equipment",fieldname:"g_Equipment_info",displaywidth:8,f
ieldlength:20} ];
```

- **Screen Structure:** The screens may require pushbuttons to assist in transaction navigation, basic functionality or custom functionality. In our examples, we will enable create, change and delete functionality for each transaction in addition to the Addendum functionality already introduced. The create, change and delete functionalities we will add via these pushbuttons will create new screens for the specific transaction. These functionalities are indexed and must be in the same order. If a user wants to change these, he or she can do so by accessing the screen object. The code is as follows:

  ```
  g_WorkOrderAppDesc.MasterRecord.Screen = "Process Order|Create O
  rder|Change Order|Display Order|Delete Order";g_WorkOrderAppDesc
  .Addendum[0].Screen = "@17@Add Parts|Create Part|Change Part|Dis
  play Part";
  ```

- **Tcode Structure:** The Tcode defines the transaction code for the Master Record and for the Addendum screen. The Master Record and Addendum screens are generated from the data contained in this table. In our example, we will set the Master Record equal to the 'wkdr' tcode. The first Addendum will be set as the 'IW32' transaction. The code is as follows:

  ```
  g_WorkOrderAppDesc.MasterRecord.TCode = "wkdr";
  g_WorkOrderAppDesc.Addendum[0].TCode = "iw32";
  ```


(c) 2026 Liquid UI | Synactive | GuiXT <dev@guixt.com> | 2026-11-02 20:01

URL: https://www.liquid-ui.com/knowledge_base/content/130/1057/en/40204-maintaining-record-structures.html

Offline Configuration

Author: sarvani.kusuri@guixt.com
Last update: 2021-06-03 17:11

(c) 2026 Liquid UI | Synactive | GuiXT <dev@guixt.com> | 2026-11-02 20:01
URL: https://www.liquid-ui.com/knowledge_base/content/130/1057/en/40204-maintaining-record-structures.html