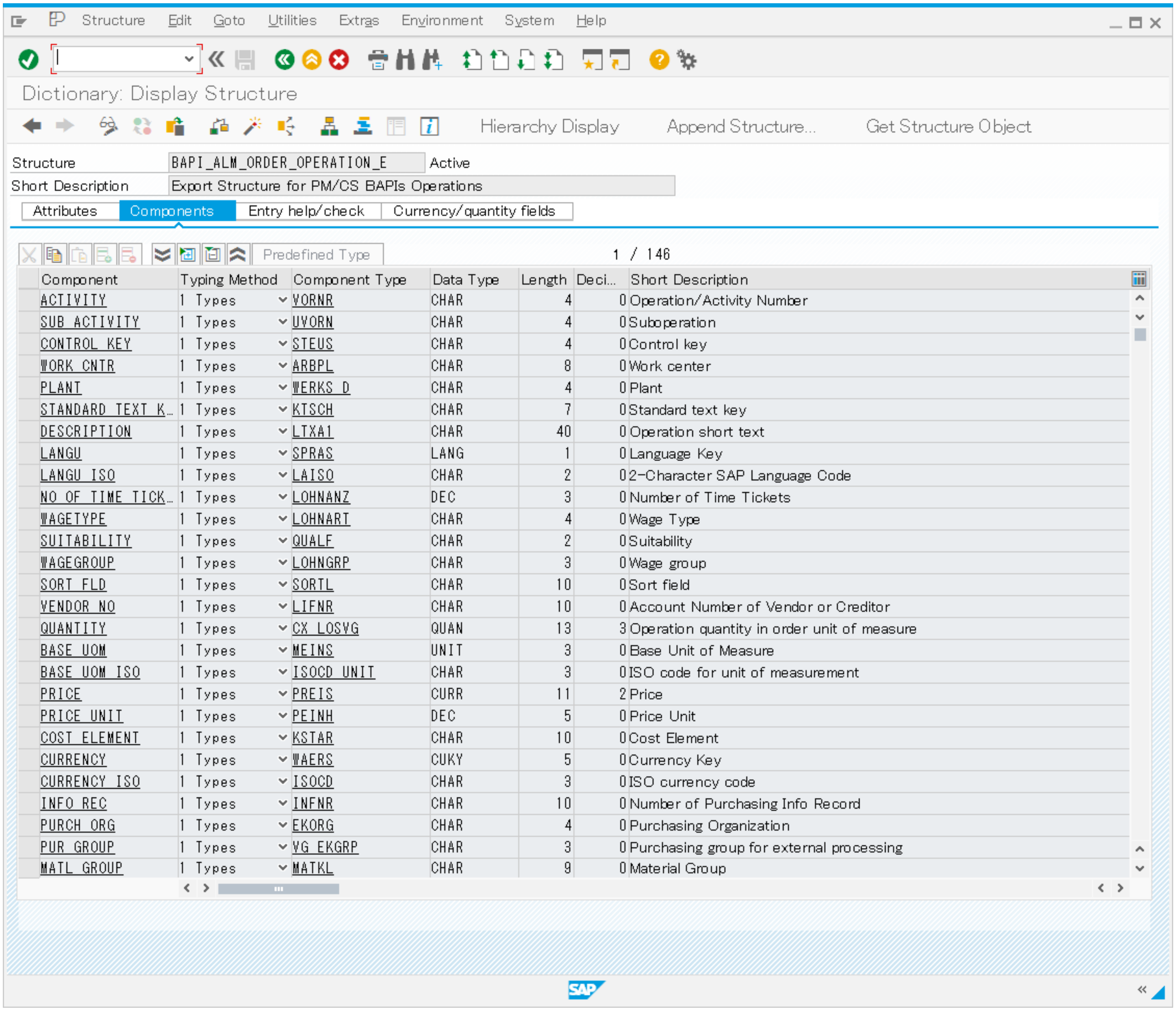
Liquid UI: Convert function module structure detail to type object

User Interface

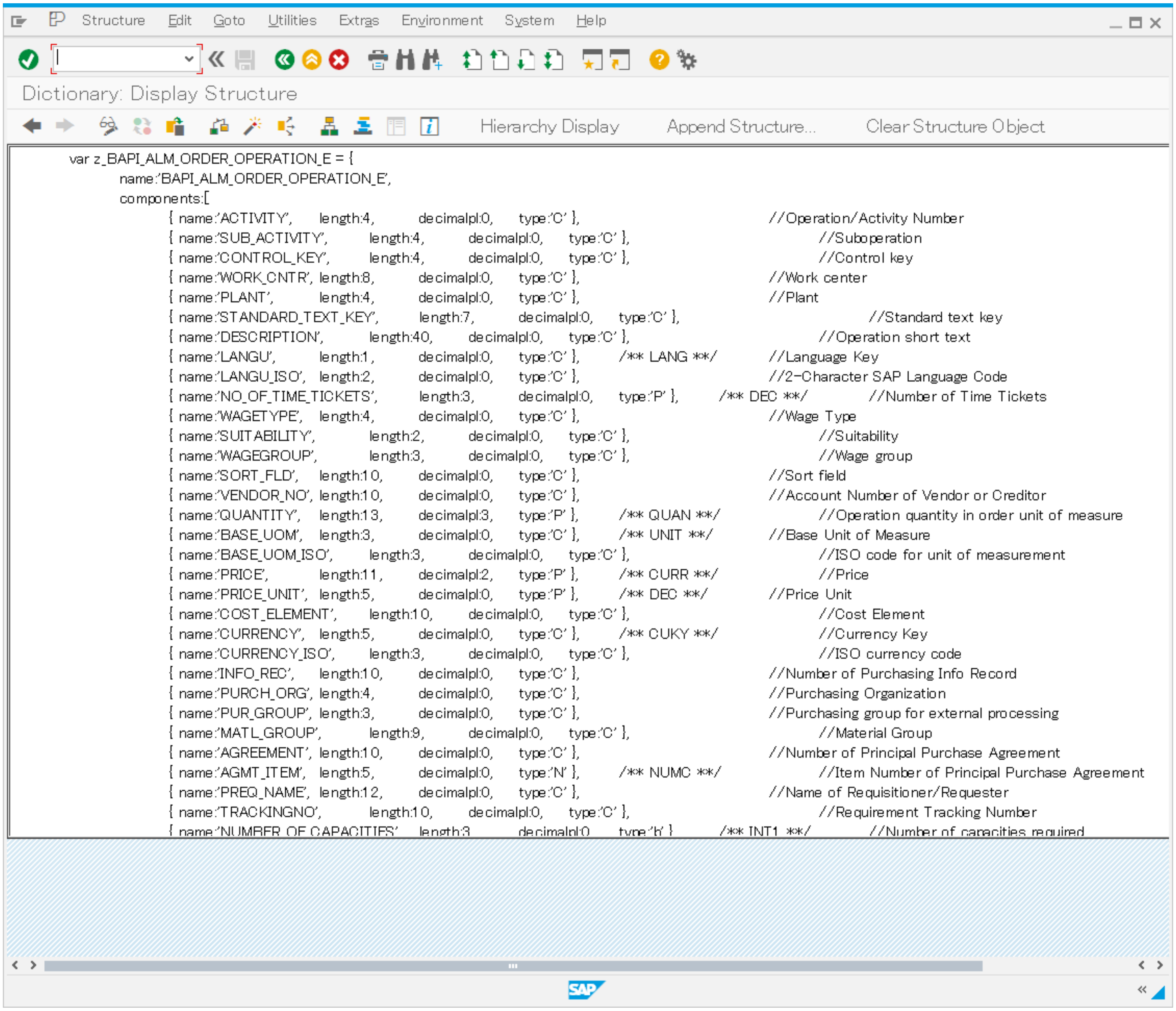


Structure	BAPI_ALM_ORDER_OPERATION_E Active
Short Description	Export Structure for PM/CS BAPIs Operations

1 / 146

Component	Typing Method	Component Type	Data Type	Length	Deci...	Short Description
ACTIVITY	1 Types	VORNR	CHAR	4	0	Operation/Activity Number
SUB_ACTIVITY	1 Types	UVORN	CHAR	4	0	Suboperation
CONTROL_KEY	1 Types	STEUS	CHAR	4	0	Control key
WORK_CNTR	1 Types	ARBPL	CHAR	8	0	Work center
PLANT	1 Types	WERKS_D	CHAR	4	0	Plant
STANDARD_TEXT_K...	1 Types	KTSCH	CHAR	7	0	Standard text key
DESCRIPTION	1 Types	LTXA1	CHAR	40	0	Operation short text
LANGU	1 Types	SPRAS	LANG	1	0	Language Key
LANGU_ISO	1 Types	LAISO	CHAR	2	0	2-Character SAP Language Code
NO_OF_TIME_TICK...	1 Types	LOHNANZ	DEC	3	0	Number of Time Tickets
WAGETYPE	1 Types	LOHNART	CHAR	4	0	Wage Type
SUITABILITY	1 Types	QUALF	CHAR	2	0	Suitability
WAGEGROUP	1 Types	LOHNGRP	CHAR	3	0	Wage group
SORT_FLD	1 Types	SORTL	CHAR	10	0	Sort field
VENDOR_NO	1 Types	LIFNR	CHAR	10	0	Account Number of Vendor or Creditor
QUANTITY	1 Types	CX_LOSVG	QUAN	13	3	Operation quantity in order unit of measure
BASE_UOM	1 Types	MEINS	UNIT	3	0	Base Unit of Measure
BASE_UOM_ISO	1 Types	ISOCD_UNIT	CHAR	3	0	ISO code for unit of measurement
PRICE	1 Types	PREIS	CURR	11	2	Price
PRICE_UNIT	1 Types	PEINH	DEC	5	0	Price Unit
COST_ELEMENT	1 Types	KSTAR	CHAR	10	0	Cost Element
CURRENCY	1 Types	WAERS	CUKY	5	0	Currency Key
CURRENCY_ISO	1 Types	ISOCD	CHAR	3	0	ISO currency code
INFO_REC	1 Types	INFNR	CHAR	10	0	Number of Purchasing Info Record
PURCH_ORG	1 Types	EKORG	CHAR	4	0	Purchasing Organization
PUR_GROUP	1 Types	VG_EKGRP	CHAR	3	0	Purchasing group for external processing
MATL_GROUP	1 Types	MATKL	CHAR	9	0	Material Group

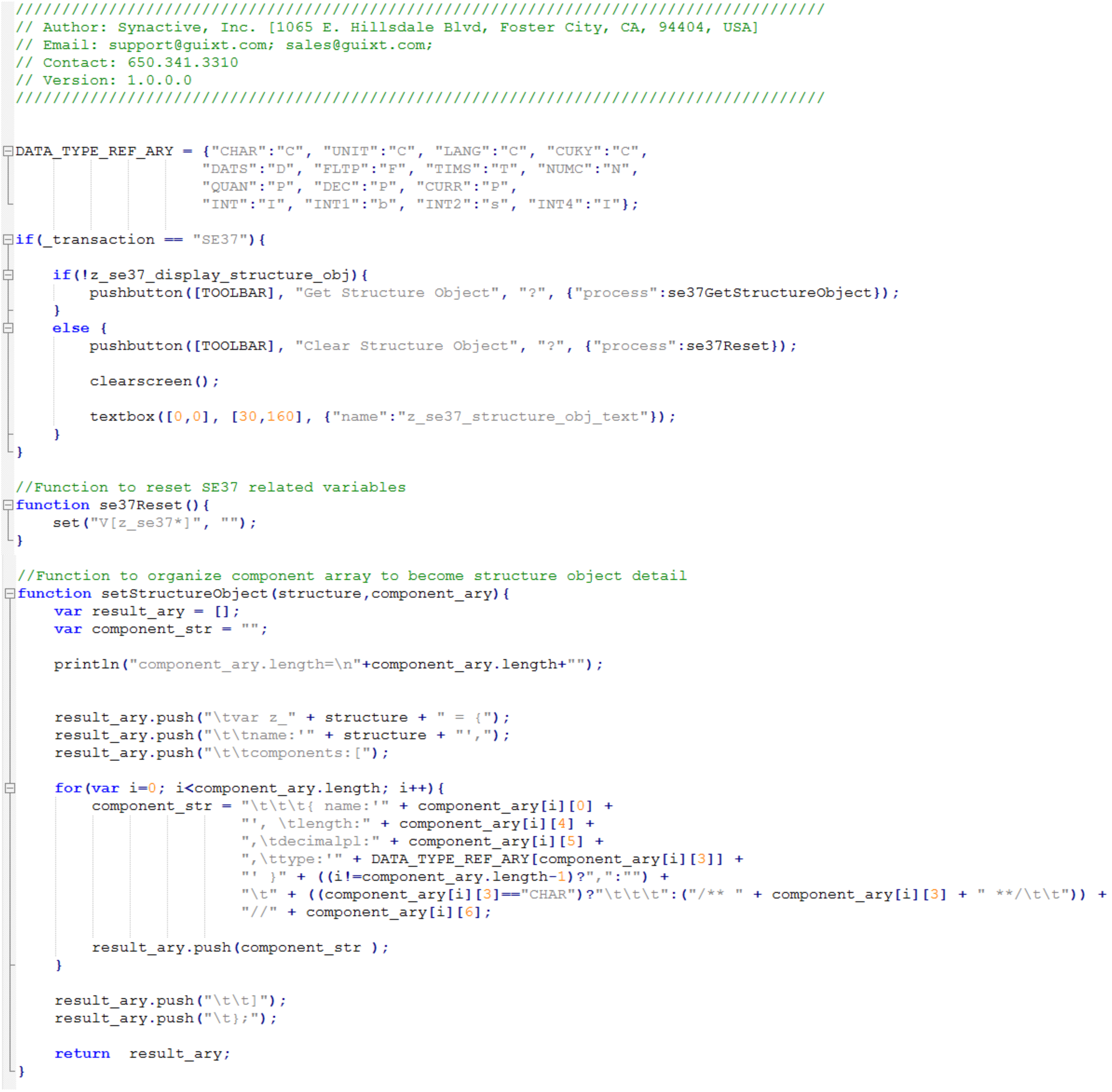
After click “Get Structure Object”



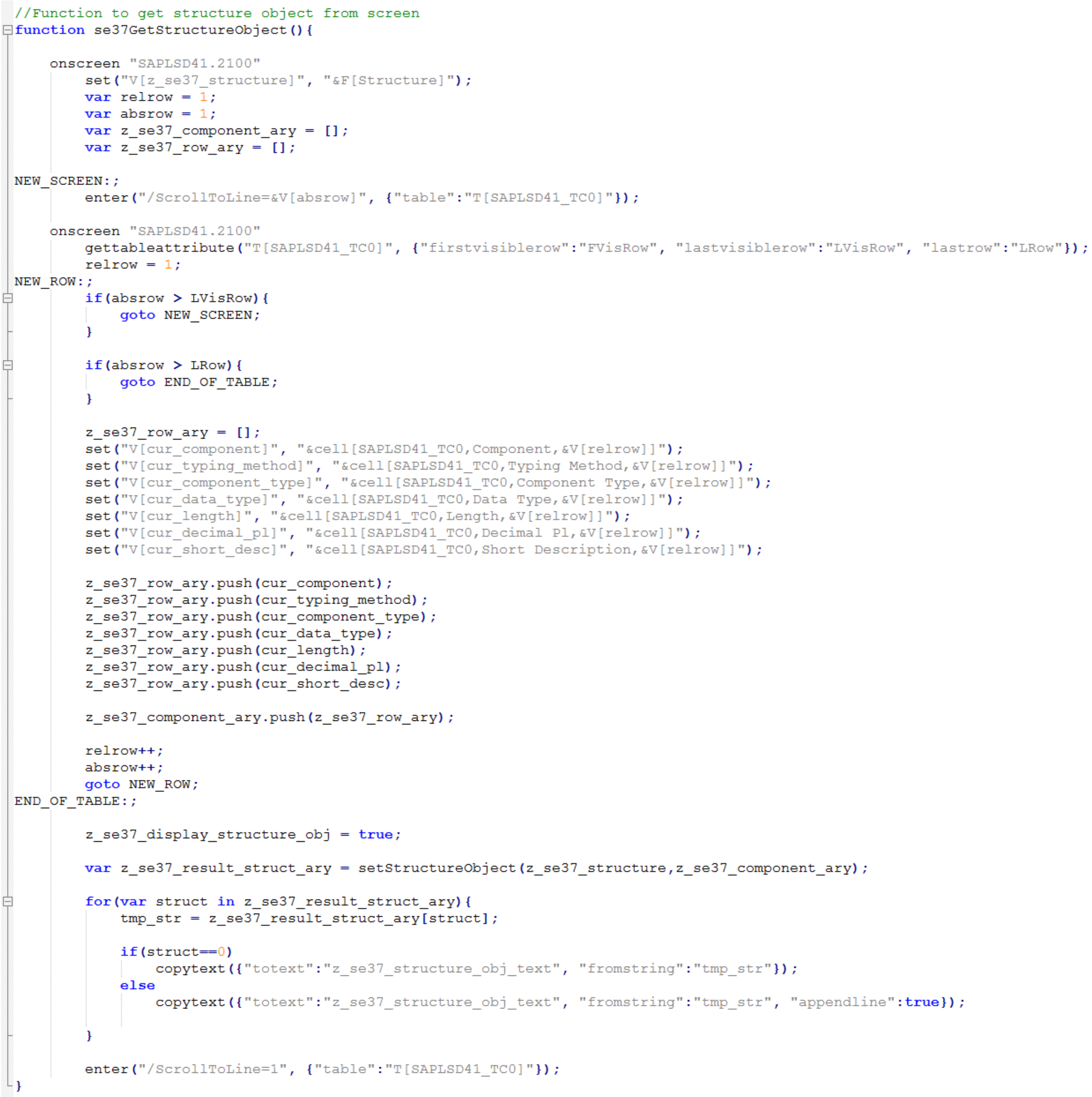
Structure Edit Goto Utilities Extras Environment System Help
Dictionary: Display Structure
Hierarchy Display
Append Structure...
Clear Structure Object

```
var z_BAPI_ALM_ORDER_OPERATION_E = {
        name:'BAPI_ALM_ORDER_OPERATION_E',
        components:[
                { name:'ACTIVITY',      length:4,       decimalpl:0,    type:'C' },                             //Operation/Activity Number
                { name:'SUB_ACTIVITY',          length:4,       decimalpl:0,    type:'C' },                             //Suboperation
                { name:'CONTROL_KEY',           length:4,       decimalpl:0,    type:'C' },                             //Control key
                { name:'WORK_CNTR', length:8,       decimalpl:0,    type:'C' },                             //Work center
                { name:'PLANT',         length:4,       decimalpl:0,    type:'C' },                             //Plant
                { name:'STANDARD_TEXT_KEY',     length:7,       decimalpl:0,    type:'C' },                             //Standard text key
                { name:'DESCRIPTION',           length:40,      decimalpl:0,    type:'C' },                             //Operation short text
                { name:'LANGU',         length:1 ,      decimalpl:0,    type:'C' },     /** LANG **/            //Language Key
                { name:'LANGU_ISO',     length:2,       decimalpl:0,    type:'C' },                             //2-Character SAP Language Code
                { name:'NO_OF_TIME_TICKETS',    length:3,       decimalpl:0,    type:'P' },     /** DEC **/     //Number of Time Tickets
                { name:'WAGETYPE',      length:4,       decimalpl:0,    type:'C' },                             //Wage Type
                { name:'SUITABILITY',           length:2,       decimalpl:0,    type:'C' },                             //Suitability
                { name:'WAGEGROUP',             length:3,       decimalpl:0,    type:'C' },                             //Wage group
                { name:'SORT_FLD',      length:10,      decimalpl:0,    type:'C' },                             //Sort field
                { name:'VENDOR_NO', length:10,      decimalpl:0,    type:'C' },                             //Account Number of Vendor or Creditor
                { name:'QUANTITY',      length:13,      decimalpl:3,    type:'P' },     /** QUAN **/            //Operation quantity in order unit of measure
                { name:'BASE_UOM',      length:3,       decimalpl:0,    type:'C' },     /** UNIT **/    //Base Unit of Measure
                { name:'BASE_UOM_ISO',          length:3,       decimalpl:0,    type:'C' },                             //ISO code for unit of measurement
                { name:'PRICE',         length:11 ,     decimalpl:2,    type:'P' },     /** CURR **/            //Price
                { name:'PRICE_UNIT',    length:5,       decimalpl:0,    type:'P' },     /** DEC **/     //Price Unit
                { name:'COST_ELEMENT',          length:10,      decimalpl:0,    type:'C' },                             //Cost Element
                { name:'CURRENCY',      length:5,       decimalpl:0,    type:'C' },     /** CUKY **/            //Currency Key
                { name:'CURRENCY_ISO',          length:3,       decimalpl:0,    type:'C' },                             //ISO currency code
                { name:'INFO_REC',      length:10,      decimalpl:0,    type:'C' },                             //Number of Purchasing Info Record
                { name:'PURCH_ORG', length:4,       decimalpl:0,    type:'C' },                             //Purchasing Organization
                { name:'PUR_GROUP', length:3,       decimalpl:0,    type:'C' },                             //Purchasing group for external processing
                { name:'MATL_GROUP',            length:9,       decimalpl:0,    type:'C' },                             //Material Group
                { name:'AGREEMENT', length:10,      decimalpl:0,    type:'C' },                             //Number of Principal Purchase Agreement
                { name:'AGMT_ITEM',     length:5,       decimalpl:0,    type:'N' },     /** NUMC **/            //Item Number of Principal Purchase Agreement
                { name:'PREQ_NAME', length:12,      decimalpl:0,    type:'C' },                             //Name of Requisitioner/Requester
                { name:'TRACKINGNO',            length:10,      decimalpl:0,    type:'C' },                             //Requirement Tracking Number
                { name:'NUMBER_OF_CAPACITIES',  length:3,       decimalpl:0,    type:'b' },     /** INT1 **/    //Number of capacities required
```

Liquid UI Code [Script]



```
////////////////////////////////////////////////////////////////////////////////////////
// Author: Synactive, Inc. [1065 E. Hillsdale Blvd, Foster City, CA, 94404, USA]
// Email: support@guixt.com; sales@guixt.com;
// Contact: 650.341.3310
// Version: 1.0.0.0
////////////////////////////////////////////////////////////////////////////////////////


DATA_TYPE_REF_ARY = {"CHAR":"C", "UNIT":"C", "LANG":"C", "CUKY":"C",
                    "DATS":"D", "FLTP":"F", "TIMS":"T", "NUMC":"N",
                    "QUAN":"P", "DEC":"P", "CURR":"P",
                    "INT":"I", "INT1":"b", "INT2":"s", "INT4":"I"};

if(_transaction == "SE37"){

    if(!z_se37_display_structure_obj){
        pushbutton([TOOLBAR], "Get Structure Object", "?", {"process":se37GetStructureObject});
    }
    else {
        pushbutton([TOOLBAR], "Clear Structure Object", "?", {"process":se37Reset});

        clearscreen();

        textbox([0,0], [30,160], {"name":"z_se37_structure_obj_text"});
    }
}

//Function to reset SE37 related variables
function se37Reset(){
    set("V[z_se37*]", "");
}

//Function to organize component array to become structure object detail
function setStructureObject(structure,component_ary){
    var result_ary = [];
    var component_str = "";

    println("component_ary.length=\n"+component_ary.length+"");


    result_ary.push("\tvar z_" + structure + " = {");
    result_ary.push("\t\tname:'" + structure + "',");
    result_ary.push("\t\tcomponents:[");

    for(var i=0; i<component_ary.length; i++){
        component_str = "\t\t\t{ name:'" + component_ary[i][0] +
                        "', \tlength:" + component_ary[i][4] +
                        ",\tdecimalpl:" + component_ary[i][5] +
                        ",\ttype:'" + DATA_TYPE_REF_ARY[component_ary[i][3]] +
                        "' }" + ((i!=component_ary.length-1)?",":"") +
                        "\t" + ((component_ary[i][3]=="CHAR")?"\t\t\t":("/** " + component_ary[i][3] + " **/\t\t")) +
                        "//" + component_ary[i][6];

        result_ary.push(component_str );
    }

    result_ary.push("\t\t]");
    result_ary.push("\t};");

    return  result_ary;
}
```



```
//Function to get structure object from screen
function se37GetStructureObject(){

    onscreen "SAPLSD41.2100"
        set("V[z_se37_structure]", "&F[Structure]");
        var relrow = 1;
        var absrow = 1;
        var z_se37_component_ary = [];
        var z_se37_row_ary = [];

NEW_SCREEN:;
        enter("/ScrollToLine=&V[absrow]", {"table":"T[SAPLSD41_TC0]"});

    onscreen "SAPLSD41.2100"
        gettableattribute("T[SAPLSD41_TC0]", {"firstvisiblerow":"FVisRow", "lastvisiblerow":"LVisRow", "lastrow":"LRow"});
        relrow = 1;
NEW_ROW:;
        if(absrow > LVisRow){
            goto NEW_SCREEN;
        }

        if(absrow > LRow){
            goto END_OF_TABLE;
        }

        z_se37_row_ary = [];
        set("V[cur_component]", "&cell[SAPLSD41_TC0,Component,&V[relrow]]");
        set("V[cur_typing_method]", "&cell[SAPLSD41_TC0,Typing Method,&V[relrow]]");
        set("V[cur_component_type]", "&cell[SAPLSD41_TC0,Component Type,&V[relrow]]");
        set("V[cur_data_type]", "&cell[SAPLSD41_TC0,Data Type,&V[relrow]]");
        set("V[cur_length]", "&cell[SAPLSD41_TC0,Length,&V[relrow]]");
        set("V[cur_decimal_pl]", "&cell[SAPLSD41_TC0,Decimal Pl,&V[relrow]]");
        set("V[cur_short_desc]", "&cell[SAPLSD41_TC0,Short Description,&V[relrow]]");

        z_se37_row_ary.push(cur_component);
        z_se37_row_ary.push(cur_typing_method);
        z_se37_row_ary.push(cur_component_type);
        z_se37_row_ary.push(cur_data_type);
        z_se37_row_ary.push(cur_length);
        z_se37_row_ary.push(cur_decimal_pl);
        z_se37_row_ary.push(cur_short_desc);

        z_se37_component_ary.push(z_se37_row_ary);

        relrow++;
        absrow++;
        goto NEW_ROW;
END_OF_TABLE:;

        z_se37_display_structure_obj = true;

        var z_se37_result_struct_ary = setStructureObject(z_se37_structure,z_se37_component_ary);

        for(var struct in z_se37_result_struct_ary){
            tmp_str = z_se37_result_struct_ary[struct];

            if(struct==0)
                copytext({"totext":"z_se37_structure_obj_text", "fromstring":"tmp_str"});
            else
                copytext({"totext":"z_se37_structure_obj_text", "fromstring":"tmp_str", "appendline":true});

        }

        enter("/ScrollToLine=1", {"table":"T[SAPLSD41_TC0]"});
}
```