“5 LINES OF CODE CAN MAKE YOUR DAY”

HANDOUTS

1. VARIABLE ERROR MESSAGE ROUTINES
2. SINGLE INSTANT WINDOWS “CHECK_FOR_WINDOW”
3. VISUAL BASIC LOG-IN ROUTINE
4. CALL DESKTOP APPLICATION “START_EXTERNAL_APPLICATION”
5. THE ‘HELP’ SYSTEM

VARIABLE ERROR MESSAGE ROUTINES

PROCEDURE FOR INIDIVIDUAL REGISTRATION

Model : HOMES LODGING ANALYSIS R01 Mar. 28, 1996 07:23
Subset: CONSTRUCT IND REG

Procedure Step: FD04_INDIVIDUAL_REGISTRATION

```
 282 | | | EXIT STATE IS aa_processing_ok
 283 | | | USE fd0431_check_hskpng_status (CALLS ACTION BLOCK
FD0431)
 284 | | |     WHICH IMPORTS: Entity View export_selected facility
 285 | | |                    Entity View export_selected saleable_unit
 286 | | |                    Entity View export_selected room
 287 | | |                    Entity View export individual_visit
 288 | | |     WHICH EXPORTS: Work View   export
aa_parameters_to_msg_handler
 289 | | |                    Entity View export_for_display saleable_unit
 290 | | | +- IF EXITSTATE IS NOT EQUAL TO aa_processing_ok (ACTION
BLOCK RETURNS                HERE)
 291 | | | | EXIT STATE IS aa_lnk_to_message_handler_normal  (ONLY
EXECUTED IF FD0431           RETURNS “ERROR”)
 292 | | | | SET export_message_handler_called ief_supplied flag TO "Y"
 293<------------ESCAPE
 294 | | | +- ELSE
 295 | | | | +- IF export_for_display saleable_unit housekeeping_status_code
 296 | | | | |           IS EQUAL TO "RECHECK"
 297 | | | | |       OR export_for_display saleable_unit
 298 | | | | |           housekeeping_status_code IS EQUAL TO "DIRTY"
 299 | | | | | OPEN Dialog Box fd04_housekeeping_status_warning
 300<---------------ESCAPE
 301 | | | | +--
 302 | | | +--
```

ACTION BLOCK FD0431

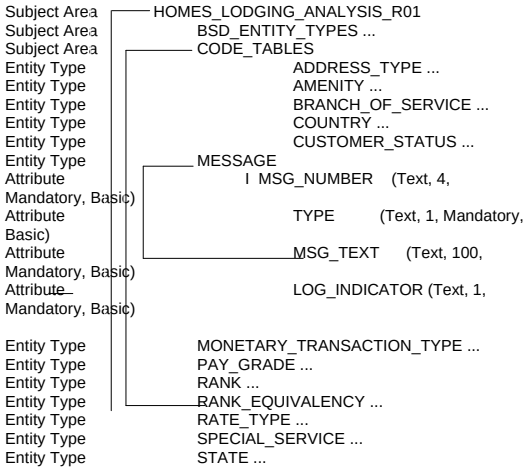
Model : HOMES LODGING ANALYSIS R01 Mar. 28, 1996 07:25
Subset: CONSTRUCT IND REG

BSD Action Block: FD0431_CHECK_HSKPNG_STATUS

```
  10 | | +- READ saleable_unit
  11 | | |      WHERE DESIRED saleable_unit number IS EQUAL TO 1
  12 | | |      AND DESIRED saleable_unit located_in SOME room
  13 | | |      AND THAT room is_contained_in SOME facility
  14 | | |      AND THAT room identification_code IS EQUAL TO import room
  15 | | |          identification_code
  16 | | |      AND THAT room system_id IS EQUAL TO import room system_id
  17 | | |      AND THAT facility facility_system_id IS EQUAL TO
  18 | | |          import facility facility_system_id
  19 | | +- WHEN successful
  20 | | | SET export saleable_unit housekeeping_status_code TO
saleable_unit
  21 | | |          housekeeping_status_code (RETURNS TO LINE 290 IN
FD04)
  22 | | +- WHEN not found
  23 | | | SET export aa_parameters_to_msg_handler msg_number TO "0025"
(***)
  24 | | | SET export aa_parameters_to_msg_handler value1 TO "Saleable
Unit"
  25 | | | EXIT STATE IS saleable_unit_nf  (RETURNS TO LINE 290 IN
FD04)
  26 | | +--
```

NOTES:
Routine locates a “saleable_unit”
If a unit is “Not found” routine sends error message number (“0025”) to
Message Handler at (***)

COMPOSER DATABASE DECLARATION



DATA BASE STRUCTURE

VARIABLE NAME	ATTRIBUTE	DESCRIPTION
msg_id	char(4)	System ID
msg_typ-cd	char(1)	'E' for Error 'W' for Warning 'I' for information
msg_tx	varchar (100)	Actual message text
msg_log_id	char(1)	'Y' or 'N'

```
/*****************************************************************
*
 *
 *                  Source Code Generated by
 *               Information Engineering Facility (tm)
 *                  Texas Instruments, Inc.
 *            Copyright (c) Texas Instruments, Inc. 1995
 *
 *    Name: CHECK_FOR_WINDOW              Date: 95/10/23
 *    Target OS:   DOS                    Time: 09:42:48
 *    Target DBMS: INFORMIX               User: OS2USER
 *
 *    Generation options:
 *    Debug trace option not selected
 *    Data modeling constraint enforcement not selected
 *    Optimized import view initialization not selected
 *

*****************************************************************
/
/*****************************************************************
**
 Data declarations

*****************************************************************
/
static char * ti_copyright = "Copyright(c) Texas Instruments Inc. 1995";
static char ief_cgen_date[] = "95/10/23";
static char ief_cgen_time[] = "09:42:48";
static char ief_cgen_ency[] = "6.0.C7";
static char ief_cgen_userid[] = "OS2USER";
static char ief_cgen_model[] = "MODELESS TEST";
static char ief_cgen_subset[] = "ALL";
static char ief_cgen_name[] = "CHECK_FOR_WINDOW";
#include <datetime.h>
#include <stdio.h>
#include <windows.h>
#include <string.h>
   struct psmgr_eab_data
       {
       long   psmgr_eabpcb_cnt;
       char * psmgr_eabpcb_entry [255];
       };

static struct psmgr_eab_data  *eabdata;
```

```
static char  *ief_runtime_parm1;
static char  *ief_runtime_parm2;
/* * * * * * * * * * * * * */
/* START OF IMPORT VIEWS */
/* * * * * * * * * * * * * */

/* Data View Group: W_IA */
/*       DVG id name: A_0007340133_IA */
struct a_0007340133_ia
{
/* Entity View: IMPORT */
/*         Type: IEF_SUPPLIED */
char command_001as;
char command_001[81] /* 80 + 1 */;
char flag_001as;
char flag_001[2] /* 1 + 1 */;
};
static struct a_0007340133_ia  *w_ia;
/* * * * * * * * * * * * * */
/* START OF EXPORT VIEWS */
/* * * * * * * * * * * * * */
/* Data View Group: W_OA */
/*       DVG id name: A_0007405670_OA */
struct a_0007405670_oa
{
/* Entity View: EXPORT */
/*         Type: IEF_SUPPLIED */
char command_002as;
char command_002[81] /* 80 + 1 */;
char flag_002as;
char flag_002[2] /* 1 + 1 */;
};
static struct a_0007405670_oa  *w_oa;
/* * * * * * * * * * * * * * * * * * * */
/* MISC DECLARATIONS AND PROTOTYPES  */
/*    FOLLOW AS NEEDED:              */
/* * * * * * * * * * * * * * * * * * * */
static void f_131093(void);
static char func_0000131093_esc_flag;
/*       +->   CHECK_FOR_WINDOW                  10/23/95  09:42             */
/*       !       IMPORTS:                                                    */
/*       !         Work View import ief_supplied (Transient, Optional,       */
/*       !         Import only)                                              */
/*       !           command                                                 */
/*       !           flag                                                    */
```

```
/*      !       EXPORTS:                                        */
/*      !         Work View export ief_supplied (Transient, Export only)     */
/*      !           command                                     */
/*      !           flag                                  */
/*      !     EXTERNAL ACTION BLOCK                                 */
/*      +---                                                  */
```

```
/* * * * * * * * * * * * * * * * * * * */
/* ACTION BLOCK FUNCTION DECLARATIONS  */
/* * * * * * * * * * * * * * * * * * * */

void CHECKFOR(in_runtime_parm1,
in_runtime_parm2,
in_psmgr_eab_data,
import_view,
export_view)
char  *in_runtime_parm1;
char  *in_runtime_parm2;
struct psmgr_eab_data  *in_psmgr_eab_data;
struct a_0007340133_ia  *import_view;
struct a_0007405670_oa  *export_view;
{
ief_runtime_parm1 = in_runtime_parm1;
ief_runtime_parm2 = in_runtime_parm2;
eabdata = in_psmgr_eab_data;
w_ia = import_view;
w_oa = export_view;

f_131093();
return;
}


static void f_131093(void)
{
HWND a;
int i;
func_0000131093_esc_flag = '\0';
for(i=79;i>=0;i--){
        if (w_ia->command_001[i] != 32){
                w_ia->command_001[i+1] = 0;
                i = -1;
        }
}
a = FindWindow(NULL,w_ia->command_001);
if (a != NULL){
        SetActiveWindow(a);
   strcpy(w_oa->flag_002,"1");
}
}
```

VISUAL BASIC LOG-IN ROUTINE

```
Sub cmbOK_click ()

On Error GoTo ErrorFound

Dim Path As String
Dim DBName As String
Dim Msg, NL, DgDef, Response
Dim msgType As Integer
Dim DirName

NL = Chr(13) + Chr(10)
msgType = MB_ICONINFORMATION

If txtUserID = "" Then
   Msg = "A User ID is required for access."
   MsgBox Msg, msgType, TITLE
   txtUserID.SetFocus
   Exit Sub
ElseIf txtPwd = "" Then
   Msg = "A Password is required for access."
   MsgBox Msg, msgType, TITLE
   txtPwd.SetFocus
   Exit Sub
Else
   If ((InStr(UCase$(INI_DBNAME), "TEST") <> 0) Or
(InStr(UCase$(INI_DBNAME), "TRAIN") <> 0)) Then
      DgDef = MB_YESNO + MB_ICONEXCLAMATION
      Msg = "Please note the database you have selected" + NL
      Msg = Msg + "is a test database. [ " & INI_DBNAME & " ]" + NL + NL
      Msg = Msg & "Do you wish to continue?"
      Response = MsgBox(Msg, DgDef, TITLE)
   End If

   If Response = 7 Then Exit Sub

   txtUserID.SetFocus
   txtUserID.SelStart = 0
   txtUserID.SelLength = Len(txtUserID.Text)
   Screen.MousePointer = HOURGLASS

   Path = INI_INFORMIXPATH
   Response = Shell(Path, 1)
   AppActivate "SETNET"
```

```
    SendKeys "{TAB}" & txtUserID & "{TAB}{TAB}{TAB}{TAB}" & txtPwd &
"~%S", True

    Path = app.Path

    If Right$(Path, 1) <> "\" Then
        Path = Path + "\"
    End If
    Path = Path & "LODGING.INI"

    default$ = "NONE"
    buffer$ = Space(50)
    buffersize% = 50
    section$ = "Database"
    itemname$ = "iqwinpath"
    x% = GetPrivateProfileString%(section$, itemname$, default$, buffer$,
buffersize%, Path)
    INI_IQWINPATH = Mid(Trim(buffer$), 1, Len(Trim(buffer$)) - 1)

    Path = INI_IQWINPATH
    If (Dir(Path & "\BATCHQ.DAT", 0) <> "") Then
        Kill Path & "\BATCHQ.DAT"
    End If

    If (Dir(Path & "\BATCHQ.TMP", 0) <> "") Then
        Kill Path & "\BATCHQ.TMP"
    End If

    Path = GetWindowsSysDir() & "LODGE.DLL"
    DirName = Dir(Path, 0)
    If (DirName = "") Then
        Err = 53
        GoTo ErrorFound
    End If

    x% = DBConnect(INI_DBNAME)

    If (x% <> 0) Then
        Screen.MousePointer = ARROW
        DgDef = MB_OK + MB_ICONSTOP
        Select Case x%
            Case -329
                Msg = "Selected database does not exist." & NL
                Msg = Msg & "(DB = " & INI_DBNAME & ")" & NL
                Msg = Msg & "(SQLCODE = " & x% & ")" + NL + NL
                Msg = Msg & "Please select an existing database or" & NL
```

```
            Msg = Msg & "contact System Administrator if you require" &
NL
            Msg = Msg & "additional assistance."
            Response = MsgBox(Msg, DgDef, TITLE)
         Case -951
            Msg = "Invalid User ID entered, access denied."
            Response = MsgBox(Msg, DgDef, TITLE)
         Case -952
            Msg = "Invalid Password entered, access denied."
            Response = MsgBox(Msg, DgDef, TITLE)
         Case 987
            Msg = "The system is currently in Maintenance mode." + NL
            Msg = Msg & "Lodging access is not permitted at this time." +
NL + NL
            Msg = Msg & "Please contact System Administrator for
assistance."
            Response = MsgBox(Msg, DgDef, TITLE)
         Case 999
            Msg = Msg & "Ploblem executing Dynamic Link Library." & NL
            Msg = Msg & "(RC = " & x% & ")" & NL & NL
            Msg = Msg & Path + NL + NL
            Msg = Msg & "Please contact System Administrator."
            Response = MsgBox(Msg, DgDef, TITLE)
         Case Is > 0
            Msg = "Informix SETNET settings may be incorrect." & NL & NL
            Msg = Msg & "Please contact System Administrator."
            Response = MsgBox(Msg, DgDef, TITLE)
         Case Else
            Msg = "Database access is currently unavailable." + NL
            Msg = Msg & "(SQLCODE = " & x% & ")" + NL + NL
            Msg = Msg & "Please contact your System Administrator for
assistance."
            Response = MsgBox(Msg, DgDef, TITLE)
      End Select

      GoTo ExitClean
   End If

   Path = app.Path
   If Right$(Path, 1) <> "\" Then
      Path = Path + "\"
   End If
   Path = Path & "LODGING.INI"
   section$ = "Application"
   itemname$ = "lodging1"
   itemvalue$ = txtUserID
```

```
    x% = WritePrivateProfileString%(section$, itemname$, itemvalue$,
Path)

    itemname$ = "lodging2"
    itemvalue$ = txtPwd
    x% = WritePrivateProfileString%(section$, itemname$, itemvalue$,
Path)

    Path = app.Path
    If Right$(Path, 1) <> "\" Then
        Path = Path + "\"
    End If

  Path = Path & INI_EXECNAME & " /DB=" & INI_DBNAME
    Response = Shell(Path, 1)
    Screen.MousePointer = ARROW

End If

ExitClean:
    Screen.MousePointer = ARROW
    Unload frmGetPwd
    Exit Sub

ErrorFound:
    Screen.MousePointer = ARROW
    ExitState = ErrorHandler(Err, Path)
    Unload frmGetPwd
    Exit Sub

End Sub
```

FILE NAME: LODGING.DOC

```
/*****************************************************************
 *
 *                  Source Code Generated by
 *               Information Engineering Facility (tm)
 *                   Texas Instruments, Inc.
 *            Copyright (c) Texas Instruments, Inc. 1995
 *
 *    Name: START_EXTERNAL_APPLICATION       Date: 95/10/23
 *    Target OS:   DOS                       Time: 14:50:15
 *    Target DBMS: INFORMIX                  User: OS2USER
 *
 *    Generation options:
 *    Debug trace option not selected
 *    Data modeling constraint enforcement not selected
 *    Optimized import view initialization not selected
 *
 *****************************************************************/
/*****************************************************************
 Data declarations
 *****************************************************************/
static char * ti_copyright = "Copyright(c) Texas Instruments Inc. 1995";
static char ief_cgen_date[] = "95/10/23";
static char ief_cgen_time[] = "14:50:15";
static char ief_cgen_ency[] = "6.0.C7";
static char ief_cgen_userid[] = "OS2USER";
static char ief_cgen_model[] = "MODELESS TEST";
static char ief_cgen_subset[] = "ALL";
static char ief_cgen_name[] = "START_EXTERNAL_APPLICATION";
#include <datetime.h>
#include <windows.h>
#include <stdio.h>
#include <string.h>
   struct psmgr_eab_data
       {
       long   psmgr_eabpcb_cnt;
       char * psmgr_eabpcb_entry [255];
       };

static struct psmgr_eab_data  *eabdata;
static char  *ief_runtime_parm1;
static char  *ief_runtime_parm2;
/* * * * * * * * * * * * */
/* START OF IMPORT VIEWS */
/* * * * * * * * * * * * */
```

```
/* Data View Group: W_IA */
/*       DVG id name: A_0007602277_IA */
struct a_0007602277_ia
{
/* Entity View: IMPORT */
/*        Type: IEF_SUPPLIED */
char command_001as;
char command_001[81] /* 80 + 1 */;
char flag_001as;
char flag_001[2] /* 1 + 1 */;
};
static struct a_0007602277_ia  *w_ia;
/* * * * * * * * * * * * * */
/* START OF EXPORT VIEWS */
/* * * * * * * * * * * * * */
/* Data View Group: W_OA */
/*       DVG id name: A_0007667814_OA */
struct a_0007667814_oa
{
/* Entity View: EXPORT */
/*        Type: IEF_SUPPLIED */
char command_002as;
char command_002[81] /* 80 + 1 */;
char flag_002as;
char flag_002[2] /* 1 + 1 */;
};
static struct a_0007667814_oa  *w_oa;
/* * * * * * * * * * * * * * * * * * * */
/* MISC DECLARATIONS AND PROTOTYPES  */
/*    FOLLOW AS NEEDED:              */
/* * * * * * * * * * * * * * * * * * * */

static void f_196629(void);
static char func_0000196629_esc_flag;
/*       +->   START_EXTERNAL_APPLICATION        10/23/95  14:50              */
/*       !       IMPORTS:                                                      */
/*       !         Work View import ief_supplied (Transient, Optional,        */
/*       !         Import only)                                               */
/*       !           command                                                  */
/*       !           flag                                                     */
/*       !       EXPORTS:                                                      */
/*       !         Work View export ief_supplied (Transient, Export only)     */
/*       !           command                                                  */
/*       !           flag                                                     */
```

```
/*      !    EXTERNAL ACTION BLOCK                                  */
/*      +---                                                        */



/* * * * * * * * * * * * * * * * * * * * */
/* ACTION BLOCK FUNCTION DECLARATIONS  */
/* * * * * * * * * * * * * * * * * * * * */

void STARTEXT(in_runtime_parm1,
in_runtime_parm2,
in_psmgr_eab_data,
import_view,
export_view)
char  *in_runtime_parm1;
char  *in_runtime_parm2;
struct psmgr_eab_data  *in_psmgr_eab_data;
struct a_0007602277_ia  *import_view;
struct a_0007667814_oa  *export_view;
{
ief_runtime_parm1 = in_runtime_parm1;
ief_runtime_parm2 = in_runtime_parm2;
eabdata = in_psmgr_eab_data;
w_ia = import_view;
w_oa = export_view;

f_196629();
return;
}


static void f_196629(void)
{
int i;
func_0000196629_esc_flag = '\0';
for(i=79;i>=0;i--){
        if (w_ia->command_001[i] != 32){
                w_ia->command_001[i+1] = 0;
                i = -1;
        }
}
WinExec(w_ia->command_001,SW_SHOWNORMAL);
}
```

#K{bmc m:\lodging\onl_help\bmps\banner.bmp} **THE HOMES LOGO**

(FIRST PAGE OF 'HELP'-- EXAMPLE of contents of the 'HELP' File.)

The Lodging System provides a timely and accurate method for managing Unaccompanied Personnel Housing (UPH), Permanent Party (PP), Temporary Duty (TDY), and Guesthouses under the control of the Lodging Office. It improves the reservation/registration processes and offers better control over customer accounts, utilization, and day-to-day operations of transient housing management.

Contents:

Deleting an Amenity

1. From the **Utilities** menu of the Utilities window, select the Amenities option.
   RESULT: The system displays the Amenities window.

2. Select the amenity to delete from the list box.

3. From the Edit menu, select the Delete option.

4. Select the YES button to confirm the deletion.
   RESULT: The system deletes the amenity from the list box at the top of the window.

Updating an Amenity

1. From the Utilities menu of the Utilities window, select the Amenities option.
   RESULT: The system displays the Amenities window.

2. Select the amenity to update from the list box.

3. From the Edit menu, select the Update option.
   RESULT: The system populates the data fields with the selected amenity.

4. Type in the changes to the amenity as required.

5. From the Action menu, select the Save option.
   RESULT: The system displays the updated amenity in the list box at the top of the window.

LODGING_MAIN_MENU
K Contents

The above information is identified by the process performed. The next page Is a sample list of windows and their ID Numbers as identified in COMPOSER.

WINDOW NAME AND ID NUMBER

```
#define AMENITY 7099
#define NIGHT_AUDIT 8000
#define NIGHT_AUDIT_REPORTS 8001
#define SUNDRY_SALES 8002
#define SUNDRY_SALES_REFUND 8003
#define BATCH_CHARGES_AND_CREDITS 8004
#define CASHIER_SHIFT 8005
#define OPEN_CASHIER_SHIFT 8006
#define CLOSE_CASHIER_SHIFT 8007
#define LODGING_MAIN_MENU 8008
#define MESSAGE_HANDLER 8009
#define COMMENTS 8010
#define FIND_GROUP_CUSTOMER 8011
#define FIND_INDIVIDUAL_CUSTOMER 8012
#define ADDRESS 8013
#define TELEPHONE 8014
#define SPONSOR 8015
#define REPORT_SELECTION 8016
#define NAF_LOCATION_DEPARTMENT_SELECTION 8017
#define REPORT_VERSIONS 8018
#define SELECT_ROOM 8019
#define FRONT_DESK_MENU 8020
#define INDIVIDUAL_RESERVATION 8021
#define ROOM_CATEGORY_SELECTION 8022
#define CUSTOMER_SPECIAL_SERVICES 8023
#define INDIVIDUAL_REGISTRATION 8024
#define UTILIZE_ROOM_CATEGO RY 8025
#define GROUP_RESERVATION 8026
#define GROUP_ROOM_CATEGORY_SELECTION 8027
#define GROUP_ROOMING_LIST 8028
#define BLOCK_ROOMS_FOR_A_GROUP 8029
#define GROUP_REGISTRATION 8030
#define GROUP_MEMBER_INFORMATION 8031
#define GROUP_ROOM_SELECTION 8032
#define GENERAL_AVAILABILITY 8033
#define CUSTOMER_LOCATOR 8034
#define ROOM_BLOCKS 8035
#define HOUSEKEEPING_STATUS 8036
#define FILTER_BLOCKED_ROOMS 8037
#define HOUSEKEEPING_MENU 8038
```

```
#define	FINANCIAL_MANAGEMENT_MENU	8039
#define	INDIVIDUAL_FOLIO	8040
#define	GROUP_FOLIO	8041
#define	PAYMENT	8042
#define	TRANSFER_FOLIO_CHARGE	8043
#define	ROOM_CHARGE_MODIFICATION	8044
#define	GROUP_CHARGE_ALLOCATIONS	8045
#define	PAYMENT_TYPES_AND_ALLOCATIONS	8046
#define	REFUNDS	8047
#define	CHARGE	8048
#define	CHECK_OUT_GROUP	8049
#define	CHECK_OUT_INDIVIDUAL	8050
#define	GROUP_FOLIO_CHARGE_DETAILS	8051
#define	GROUP_MEMBERS_NOT_CHECKED_OUT	8052
#define	FOLIO_CHARGE_DETAILS	8053
#define	INVOICE	8054
#define	FOLIO_DETAILS	8055
#define	FIND_INVOICE	8056
#define	INVOICE_PA YMENT	8057
#define	INVOICE_CREDIT	8058
```