

```

APP_RET create_r Rack( _db_t_database_handle hdb_plat )
{
    ret= _db_create_table(hdb_plat,
                          "R_RACK",
                          &hdb_r_Rack); /* 创建关系表 */
    .....;
    return APP_SUCCESS;
}

```

```

typedef struct tagJID
{
    WORD32  dwJno;           /* JOB 号 */
    WORD32  dwDevId;         /* 设备ID */
    WORD16  wModule;         /* 编号MP */
    WORD16  wUnit;           /* 单板号 */
    BYTE    ucSUnit;         /* 单板上的处理器号 */
    BYTE    ucSubSystem;     /* 子系统号 */
    BYTE    ucRouteType;     /* 路由类型，区别左右板位或者主备 */
    BYTE    ucPad;           /* 填充字节 */
} JID;

```

```

APP_RET  load_sig_table_from_sync( )
{
    CHAR tableName[_DB_TABLE_NAME_MAX]={ "r_employee", "
        r_office",...);
    WORD TableNum=2;
    _db_t_database_handle hdb_plat;
    if ( _DB_SUCCESS != _db_load_data_from_sync(dbName, tableName
        , TableNum ) )
    {
        return APP_ERROR;
    }
    return APP_SUCCESS;
}

```

```

/*数据区单元编号占用个18，从开始，最大为bit1262143*/
#define DA_INDEX_MASK      (WORD32)0x0003ffff
/*按低位0编号，实例号占用到bit18bit24bit*/
#define DA_INSTANCE_BIT    (BYTE )18
#define DA_INSTANCE_MASK   (WORD32)0x01fc0000
/*按低位0编号，数据区类型占用到bit25bit31bit*/
#define DA_TYPE_BIT        (BYTE )25
#define DA_TYPE_MASK       (WORD32)0xfe000000
/*根据获取数据区单元编号CallID*/
#define DA_GETINDEX(CallID)      (WORD32)((CallID) &
    DA_INDEX_MASK)
/*根据获取数据区类型CallID*/
#define DA_GETTYPE(CallID)      (WORD16)((CallID) & DA_TYPE_MASK
    ) >> DA_TYPE_BIT )
/*根据获取数据区所属实例号CallID*/
#define DA_GETINSTANCE(CallID) (WORD16)((CallID) &
    DA_INSTANCE_MASK) >> \DA_INSTANCE_BIT )
/*根据数据区类型，数据区所属实例号，数据区单元编号合成CallID*/
#define DA_GETCALLID(DA_TYPE, DA_INSTANCE, DA_INDEX)
    ( ((DA_TYPE) << DA_TYPE_BIT) | \
    ((DA_INSTANCE) << DA_INSTANCE_BIT) | (DA_INDEX) )

```

```

typedef struct{
    DWORD pageNo;      /* 对象所在页面 */
    WORD  objectID;    /* 页面内部对象编号 */
    WORD  reserved;    /* 保留填充字节 */
} _PACKED_1_ _db_t_object_handle;

```

```

#define USE_RET_CODE /* 说明需要使用返回结果标志 */
#define USE_DM_XXX /* 说明需要调用的接口 */
#include "dbaccess.h" /* 引入数据库头文件" dbaccess." h */

_UseDBDM_XXXDemo() /* 接口调用的函数 */
{
    /* 入参的申明 */
    DM_XXX_REQ tReq;
    /* 出参的申明（对于异步调用、异步调用无回应可以不需要） */
    DM_XXX_ACK tAck;

    /* 入参的初始化tReq */

    /* 调用接口（对于异步调用，调用者进程需要对响应消息进行相应的处理） */
    dbAccess( DM_XXX, (LPSTR) &tReq, (LPSTR) &tAck );

    /* 出参的处理（对于异步调用、异步调用无回应可以不需要） tAck */
}

```