# Day1

通过接口表调用标准表，

ERP

大概工作范围:

了解技术的实现掌握--技术

功能业务层面

表架构

流程

通用化 粗

系统支持 定制专用化 细

----

1.改名：8位字符 CUXPOVDR

2.画布

3.窗口

4.数据块

窗口：改标题

画布：堆叠画布 CANBAS\_STACKED

注意子类的选择

1. 建表和视图

把rowid拿出来 记得给别名row\_id

在后面记得修改row\_id属性:ROW\_ID：子类信息选择ROW\_ID,唯一标识（三个属性：名称，最大长度18，子类属性）

2.双击窗口：修改标题

3三个属性：标题，子类属性,画布

# 5.新建画布

FOLDER\_FIXED 子类属性：CANVAS\_STACKED\_DIXED\_FIELD

画布：MAIN 大画布 1

自定义FOLDER\_FIXED --包含FOLDER\_STK和滚动条和光标指示条 2

FOLDER\_STK 放数据的画布 -- 堆叠在FOLDER\_FIXED 3

有层次结构

FOLDER\_STK 子类属性:CANVAS\_STACKED

# 6.数据源

右键FOLDER-》数据块向导-》表或视图-》选择全部属性

在FOLDER的字段选择FOLDER\_STK画布

在FOLDER\_STK的字段也选择FOLDER\_STK画布，可初始化列名：初始值

# 7.FOLDER

把需要字段组合在一起拖到上面，修改子类属性为TEXT\_ITEM属性，滚动条属性 选择显示滚动条：是，带滚动条画布：FOLDER\_FIXED，修改Y轴：0.25

还有个指示条：CURRENT\_RECORD\_INDICATOR 选择FOLDER\_FIXED画布

需要模板的触发器，最好不自己创建，如新创记得复制触发器过去，不然会识别到之前的文件夹（FOLDER）

# 8.FOLDER\_PROMPT

把需要提示的字段放进FOLDER\_PROMPT

修改Y轴0，修改子类属性FOLDER\_PROMPT\_Mu，删除F1字段，不需要选择画布

初始值：改列名

# 9.修改表格触发器

注册，初始化表格块

When-new-form-instance:

app\_folder.define\_folder\_block(object\_name => 'CUXPOVDR21084',

folder\_block\_name => 'FOLDER',

prompt\_block\_name => 'FOLDER\_PROMPT',

folder\_canvas\_name => 'FOLDER\_STK',

folder\_window\_name => 'MAIN',

disabled\_functions => '',

tab\_canvas\_name => '',

fixed\_canvas\_name => 'FOLDER\_FIXED');

app\_folder.event('INSTANTIATE');

# 10.FTP：

把模板放进

/u01/dev121/apps/apps\_st/appl/cux/12.0.0/forms/ZHS

# 11.Putty

命令行

先cd $AU\_TOP/forms/ZHS

再编译

frmcmp\_batch userid=apps/apps module=/u01/dev121/apps/apps\_st/appl/cux/12.0.0/forms/ZHS/CUXTEVDR21080.fmb output\_file=/u01/dev121/apps/apps\_st/appl/cux/12.0.0/forms/ZHS/CUXTEVDR21080.fmx

---

frmcmp\_batch userid=apps/apps module=/u01/dev121/apps/apps\_st/appl/cux/12.0.0/forms/ZHS/CUXPOVDR21080.fmb output\_file=/u01/dev121/apps/apps\_st/appl/cux/12.0.0/forms/ZHS/CUXPOVDR21080.fmx

# 12.IE

http://gzdc01.haasgz.hand-china.com:30000/OA\_HTML/

到应用开发员

注册表单

注册功能 说明 特性 。。

菜单

统一放到CUX\_TRAIN\_XXXX

用户菜单不需要再创建

3、创建这样一个菜单

应用开发员->应用产品

（1）表单：表单（表单的标识 CUXPOVDR21084），应用（），用户表单名（外部展示的表单名）

（2）功能：说明，特性，表单（选择上面创建的外部展示表单名）

（3）菜单：（可以包含多个功能）菜单名，用户菜单名，

序号，提示，功能（选择刚才建立的功能）

（4）先自定义一个菜单

将菜单挂在CUX\_TRN\_MENU培训菜单下

# Day2

# 1.增加增删改功能：

## 1.自动生成包头包体，添加到程序说明

对FORM的增删改查

BEGIN

hand\_plsql\_autocreate.form\_view\_iud(p\_block\_name =>'FOLDER',

p\_table\_name =>'CUX\_PO\_VENDORS\_21080',

p\_owner =>'CUX',

p\_primary\_key =>'VENDOR\_ID');

END;

复制内容

程序说明，添加包头包体

在FOLD数据块下增加触发器：ON-INSERT ON-LOCK ON-UPDATE ON-DELETE

在触发器添加代码，例如：FOLDER\_PRIVATE.insert\_row;

## 2.增加日历

日期字段：值列表：ENABLE\_LISE\_LAMP

从列表中验证：否

添加触发器：KEY-LISTVAL

编辑：calendar.show;

## 3.添加值列表

（1）在值列表LOV右键选择LOV向导（sql，LOV窗口标题，列标题，宽度）

SELECT cpc.contact\_name,

cpc.contact\_number,

cpc.contact\_id

FROM cux\_po\_contacts cpc

LOV窗口高度6000，宽度4000

列宽度，设置为0相当于隐藏该列

（2）将值列表分配给对应的FOLDER数据

## 4.检验数据

### 1.日期的校验

在日期属性下增加WHEN-VALIDATE-ITEM触发器

### 编辑错误信息代码：

IF :FOLDER.END\_DATE\_ACTIVE < :FOLDER.START\_DATE\_ACTIVE THEN

fnd\_message.set\_name('HXT','HXT\_39092\_STDSTART\_L\_STOP');

fnd\_message.error;

RAISE FORM\_TRIGGER\_FAILURE;

END IF;

去应用开发员找错误信息代码 ‘HXT’之类的

还有一种自定义错误信息

fnd\_message.set\_string(‘error’);

fnd\_message.error;

RAISE FORM\_TRIGGER\_FAILURE;

用when-calidate-item更好

注册信息：看自定义消息

消息：查找-》HXT\_39092\_STDSTART\_L\_STOP

-》复制应用产品-》注册-》查找应用产品-》简称

### 2.插入之前校验

触发器：PRE-INSERT

app\_folder.event('PRE-QUERY');

IF :parameter.g\_query\_find = 'TRUE' THEN

:folder.vendor\_id := :query.vendor\_id;

:folder.contact\_id := :query.contact\_id;

:folder.telephone := '%'||:query.telephone||'%';

app\_find.query\_date\_range(:query.start\_date\_active\_from,

:query.start\_date\_active\_to,

'FOLDER.START\_DATE\_ACTIVE');

END IF;

这里是传值用的吧？把上面的参数传到下面，和查找按钮结合，相当于按了F11

### 3.查询之后修改格式

Folder下触发器POST-QUERY

案例：

app\_folder.event('POST-QUERY');

--folder.Populate\_Fields;

app\_globe.event('POST-QUERY');

IF :folder.end\_date\_active < trunc(SYSDATE) THEN

Set\_Item\_Instance\_Property( 'FOLDER.VENDOR\_NUMBER', CURRENT\_RECORD, update\_allowed,property\_false);

Set\_Item\_Instance\_Property( 'FOLDER.VENDOR\_NAME', CURRENT\_RECORD, update\_allowed,property\_false);

Set\_Item\_Instance\_Property( 'FOLDER.CONTACT\_NAME', CURRENT\_RECORD, update\_allowed,property\_false);

Set\_Item\_Instance\_Property( 'FOLDER.TELEPHONE', CURRENT\_RECORD, VISUAL\_ATTRIBUTE,'RED');

Set\_Item\_Instance\_Property( 'FOLDER.START\_DATE\_ACTIVE', CURRENT\_RECORD, update\_allowed,property\_false);

Set\_Item\_Instance\_Property( 'FOLDER.END\_DATE\_ACTIVE', CURRENT\_RECORD, update\_allowed,property\_false);

set\_block\_property('FOLDER', delete\_allowed,property\_false);

ELSE

Set\_Item\_Instance\_Property( 'FOLDER.VENDOR\_NUMBER', CURRENT\_RECORD, update\_allowed,property\_true);

Set\_Item\_Instance\_Property( 'FOLDER.VENDOR\_NAME', CURRENT\_RECORD, update\_allowed,property\_true);

Set\_Item\_Instance\_Property( 'FOLDER.CONTACT\_NAME', CURRENT\_RECORD, update\_allowed,property\_true);

Set\_Item\_Instance\_Property( 'FOLDER.TELEPHONE', CURRENT\_RECORD, update\_allowed,property\_true);

Set\_Item\_Instance\_Property( 'FOLDER.START\_DATE\_ACTIVE', CURRENT\_RECORD, update\_allowed,property\_true);

Set\_Item\_Instance\_Property( 'FOLDER.END\_DATE\_ACTIVE', CURRENT\_RECORD, update\_allowed,property\_true);

set\_block\_property('FOLDER', delete\_allowed,property\_true);

END IF;

## 5.调整画布

把FOLDER\_PROMPT除数据的属性归在FLODER\_FIXED画布下

在MAIN窗口才可以用堆叠画布查看

调整物理的和视图的宽高和Y轴位置

# 6.查找功能

选取需要的字段给QUERY数据块

把需要展示的字段子类属性设置好，画布选择MAIN，调整好位置

日期不需要从列表中验证

新建LOV，赋值

给LOV属性

## 模糊查询

Folder块的PRE-QUERY触发器：

app\_folder.event('PRE-QUERY');

添加：

IF :parameter.g\_query\_find = 'TRUE' THEN

:folder.vendor\_id := :query.vendor\_id;

:folder.contact\_id := :query.contact\_id;

:folder.telephone := '%'||:query.telephone||'%';---这里

app\_find.query\_date\_range(:query.start\_date\_active\_from,

:query.start\_date\_active\_to,

'FOLDER.START\_DATE\_ACTIVE');

END IF;

## 查询按钮

QUERY数据块下：

按钮：子类属性BUTTON

触发器：when button press

添加相关的代码：

:parameter.g\_query\_find := 'TRUE';

app\_find.find('FOLDER');

:parameter.g\_query\_find := 'FALSE';

### 清除内容按钮

app\_find.clear;

### 查询按钮

:parameter.g\_query\_find := 'TRUE';

app\_find.find('FOLDER');

:parameter.g\_query\_find := 'FALSE';

folder数据块:

pre-query:

-- 名称: PRE-QUERY

-- 功能描述: 对参数的判断

IF :parameter.g\_query\_find = 'TRUE' THEN

:folder.vendor\_id := :query.vendor\_id;

:folder.contact\_id := :query.contact\_id;

:folder.telephone := :query.telephone;

app\_find.query\_date\_range(:query.start\_date\_active\_from,

:query.start\_date\_active\_to,

'FOLDER.START\_DATE\_ACTIVE');

END IF;

查找小电筒，联系查找窗口

在FOLDER数据块下自定义触发器：QUERY\_FIND

:parameter.g\_query\_find := 'TRUE';

app\_find.query\_find('SUMMARY\_MAIN',--block\_window

'QUERY',--find\_window

'QUERY'--find\_block

);

:parameter.g\_query\_find := 'FALSE';

## 清除相关项内容：

在相关项添加触发器WHEN-VALIDATE-ITEM

app\_field.clear\_dependent\_fields('QUERY.CONTACT\_NUMBER'

,'QUERY.CONTACT\_NAME'

,'QUERY.CONTACT\_ID');

清空lov的id

IF :query\_find.vendor IS NULL THEN

:query\_find.vendor\_id := NULL;

:query\_find.vendor\_number := NULL;

END IF;

也可以

app\_field.clear\_dependent\_fields('CONTRACT\_FIND.CUSTOMER\_NAME',

'CONTRACT\_FIND.CUSTOMER\_ID');

设置属性

一列的属性：set\_item\_property 有Set\_Item\_Instance\_Property是一块属性，行列交叉

整块的属性：set\_block\_property

一行的属性：set\_record\_property

**清除项值**（LOV选过之后，手动删除相关项的值）

使用触发器

在相关项添加触发器WHEN-VALIDATE-ITEM

例子：

app\_field.clear\_dependent\_fields('QUERY.CONTACT\_NUMBER'

,'QUERY.CONTACT\_NAME'

,'QUERY.CONTACT\_ID');

**实现失效产品不能更新不能删除功能**

判断条件：失效时间 < SYSDATE(系统当前时间)

使用post-query触发器，在查询数据后，将失效产品的记录设置为不可更新状态。

IF :FOLDER.END\_DATE\_ACTIVE < TRUNC(SYSDATE) THEN

set\_item\_instance\_property('FOLDER.ITEM\_NUMBER',CURRENT\_RECORD,UPDATE\_ALLOWED,PROPERTY\_FALSE);

set\_item\_instance\_property('FOLDER.ITEM\_NAME',CURRENT\_RECORD,UPDATE\_ALLOWED,PROPERTY\_FALSE);

set\_item\_instance\_property('FOLDER.UOM\_CODE',CURRENT\_RECORD,UPDATE\_ALLOWED,PROPERTY\_FALSE);

set\_item\_instance\_property('FOLDER.START\_DATE\_ACTIVE',CURRENT\_RECORD,UPDATE\_ALLOWED,PROPERTY\_FALSE);

set\_item\_instance\_property('FOLDER.END\_DATE\_ACTIVE',CURRENT\_RECORD,UPDATE\_ALLOWED,PROPERTY\_FALSE);

END IF;

使用when-new-record-instance触发器，在点击记录行时，若该产品已失效，设置为不可删除状态。

IF :FOLDER.END\_DATE\_ACTIVE < TRUNC(SYSDATE) THEN

set\_block\_property('FOLDER',DELETE\_ALLOWED,PROPERTY\_FALSE);

ELSE

set\_block\_property('FOLDER',DELETE\_ALLOWED,PROPERTY\_TRUE);

END IF;

注意：set\_block\_property将**整个数据块**都设置为不可删除，所以要加ELSE，实现在点击未失效产品时，恢复为可删除状态。

**帮助->联机帮助，可以查看相关的系统函数api**

# 程序的出入口

打开MAIN入口修改 PRE-FORM

退出在调用程序说明主体的CLOSEWINDOW的上

可以设置关闭时隐藏窗口

# 行号的实现

)ji-k—: (81Jhl/hitik) 
i2HYk: HEL APPROVE LINESMt'±: Query All Records: Yes; 
HEL APPROVE LINES.LINE NUM MAX; Number 
2, 
HEL APPROVE LINES.LINE NUM 
Calculation: 
Calculation Mode: Summary; 
Summary Function:Max; 
Summarized Block: HEL APPROVE LINES.LINE NUM; 
WHEN-CREATE-RECORD 
:HEL APPROVE LINES.LINE NUM:- 
APPROVE LINES.LINE NUM 
: 38 

# 弹性域

根据ID

新增两个ITEM 子类属性TEXT ITEM

例如：

EXPENSE\_ACCOUNT\_FLEX\_DESC 费用账户说明

EXPENSE\_ACCOUNT\_FLEXFIELD 关键性弹性域实现的ITEM

/\*==================================================

名称: get\_chart\_of\_accounts\_id

功能描述: 获取弹性域上下文值

历史记录: 1.00 2016-08-15 Jim 创建

==================================================\*/

FUNCTION get\_chart\_of\_accounts\_id(p\_set\_of\_books\_id IN NUMBER)RETURN NUMBER IS

l\_chart\_of\_accounts\_id NUMBER;

BEGIN

SELECT gsob.chart\_of\_accounts\_id

INTO l\_chart\_of\_accounts\_id

FROM gl\_sets\_of\_books gsob

WHERE gsob.set\_of\_books\_id = p\_set\_of\_books\_id;

        RETURN l\_chart\_of\_accounts\_id;

EXCEPTION

        WHEN OTHERS THEN

         RETURN NULL;

END get\_chart\_of\_accounts\_id;

初始化关键性弹性域：

fnd\_key\_flex.define(BLOCK => 'HEADERS',

field => 'EXPENSE\_ACCOUNT\_FLEXFIELD',

description => 'EXPENSE\_ACCOUNT\_FLEX\_DESC',

appl\_short\_name => 'INV',

code => 'MTLL',

data\_set => ':PARAMETER.ORG\_ID',

id => 'LOCATOR\_ID',

required => 'N',

usedbflds => 'N',

validate => 'FULL',

where\_clause => '(DISABLE\_DATE > SYSDATE or DISABLE\_DATE IS NULL) AND(NVL(SUBINVENTORY\_CODE,''Z''))=NVL(:HEADERS.SUBINV\_CODE,''Z'')',

Num => 101);

## 注册说明性弹性域：

DECLARE  
x\_return\_status VARCHAR2(1);  
x\_msg\_count NUMBER;  
x\_msg\_data VARCHAR2(3000);  
BEGIN  
  cux\_pub\_importdff\_pub.import\_dff(x\_return\_status              =>x\_return\_status,  
                                   x\_msg\_count                  =>x\_msg\_count,  
                                   x\_msg\_data                   =>x\_msg\_data,  
                                   p\_appl\_short\_name            =>'CUX',  
                                   p\_descriptive\_flexfield\_name =>'CUX\_OM\_HEADERS',  
                                   p\_application\_table\_name     =>'CUX\_OM\_HEADERS',  
                                   p\_flexfield\_title            =>'CUX10000:销售订单头');  
END;

## 需要添加的的触发器（form级）：

PRE-QUERY  
PRE-UPDATE  
PRE-INSERT  
POST-QUERY  
WHEN-NEW-ITEM-INSTANCE (ITEM级触发器)在DESC\_FLEX下  
WHEN-VALIDATE-ITEM  
WHEN-VALIDATE-RECORD

**定义说明性弹性域**

HEADERS-----------------------------

fnd\_descr\_flex.define(BLOCK => 'HEADERS',

field => 'DESC\_FLEX',

appl\_short\_name => 'CUX',

desc\_flex\_name => 'CUX\_OM\_HEADERS\_21084');

增加一个说明性弹性域字段

应用开发员-》弹性域-》说明性-》段

# 自定义消息（报错信息）

应用开发员 -> 应用产品 -> 消息

名 称 
I TEM LO T 
I TEM LO T 
I TEM LO T 
吡 物 料 受 吡 次 崆 制 ， 必 须 录 入 吡 次 信 
COX 
COX 
UX 
COX 
COX 
COX 
COX 
INV 
INV 
INV 
EXP NULL 2 1 0 8 4 
MI S S 2 1 0 8 4 
NULL 2 1 0 8 4 
I r_ 尺 Y 一 BILL 一 NI 卫 L21084 
I r_ 尺 Y 一 DATE 一 21084 
PO I TEM 2 1 0 8 4 DATE 
PO I TEM 2 1 0 8 4 DATE 
ZHS 
ZHS 
ZHS 
ZHS 
ZHS 
ZHS 
应 用 产 品 
c ： 客 户 化 开 发 应 用 
c ： 客 户 化 开 发 应 用 
c ： 客 户 化 开 发 应 用 
c ： 客 户 化 开 发 应 用 
c ： 客 户 化 开 发 应 用 
c ： 客 户 化 开 发 应 用 
c ： 客 户 化 开 发 应 用 
当 前 信 本 

要实现适应多语言环境，可以建立不同语言的消息，//US，ZHS

创建消息后，还需要执行一个请求才能在EBS系统里使用。

执行 生成消息 请求：

应用开发员职责 -> 查看 -> 请求 -> 提交新请求 -> 执行单个请求

运 行 请 求 ． 
名 实 老 
称 言 
的 中 
应 用 产 品 C 皿 ： 客 户 化 开 发 应 用 
R_IO_RUKIINE 
运 行 时 司 (.msb 文 件 ） 
文 件 名 
确 定 
耵 消 

参数设置如上图。（语言，应用产品，名称）

提交请求（不执行另一项请求）

点击查看，可以看到系统所有请求。（点击更新数据，可以看到最新的请求执行进度）

# TAB页

增加表级触发器：WHEN-TAB-PAGE-CHANGED

修改lines和LINES\_PROMPT项下的需要在TAB分页的显示画布

去程序说明包增加TAB页初始化的代码（DTL\_TAB,DTL\_ATTA）和关于触发器的控制代码，画布没有触发器

初始化代码：

app\_folder.define\_folder\_block(object\_name => 'CUXOMORD10001\_DTL',

folder\_block\_name => 'LINES',

prompt\_block\_name => 'LINES\_PROMPT',

folder\_canvas\_name => 'DTL\_STK,DTL\_ATTA',

folder\_window\_name => 'DTL\_WIN',

disabled\_functions => '',

tab\_canvas\_name => 'DTL\_TAB',

fixed\_canvas\_name => 'DTL\_FIXED');

-----

app\_folder.event('INSTANTIATE');

show\_view('DTL\_FIXED');

show\_view('DTL\_STK');

触发器代码：ELSIF (event = 'WHEN-TAB-PAGE-CHANGED') THEN

IF :system.tab\_new\_page = 'LINES\_DTL' THEN

show\_view('DTL\_STK');

go\_item('LINES.ITEM\_NUMBER');

ELSE

show\_view('DTL\_ATTA');

go\_item('LINES.ADDITION1');

END IF;

调整画布大小和位置，DTL\_ATTA应和DTL\_STK的大小位置一样，切换的效果较好，记得把DTL\_ATTA的水平滚动条显示出来。

定义关键性弹性域

fnd\_key\_flex.define(BLOCK => 'HEADERS',

field => 'LOCATOR\_FLEXFIELD',

appl\_short\_name => 'INV',

code => 'MTLL',

num => '101',

id => 'LOCATOR\_ID',

description => 'LOCATOR\_FLEX\_DESC');

菜单的定义

加自定义触发器SPECIAL1—FORM级

去表级的NEW FORM INSTENCE初始化

--工具栏的菜单的实现

--给菜单的名字

fnd\_message.set\_name('CUX',

'CUX\_OM\_SUBMIT\_ORDER\_MENU');

l\_menu1 := fnd\_message.get;

--初始化菜单

app\_special.instantiate('SPECIAL1',

l\_menu1||'(&M)'); l\_menu1||'(&M)');

Default where条件的动态实现

在表格触发器

set\_block\_property('HEADERS',default\_where,'cohv.org\_id = '||fnd\_global.org\_id);

注意事项

记得初始化表格和 其他需要初始化的东西

# 注册快速编码 ：

职责-》快速编码

# 环境信息

haasgz.hand-china.com

账号：appl121

密码：appl121hand201

:parameter.g\_query\_find := 'TRUE';

app\_find.query\_find('SUMMARY\_MAIN',--block\_window

'QUERY',--find\_window

'QUERY'--find\_block

);

:parameter.g\_query\_find := 'FALSE';