

做过哪些东西（应该是从这些里面抽的问）

BDC RFC WEBSERVICE 增强（user exit /bapi） query smartform FIORI CDS VIEW 批导程序

ALV程序 导入一些外部库（abap git） sap odata 暴露service 然后通过外部接口

RFC

a.同步RFC

同步RFC（sRFC， synchronous RFC）是RFC的第一个版本，它要求连接的双方是同步的工作方式，即都是在可用状态才能够实现成功调用。

b.异步RFC

异步RFC（aRFC， asynchronous RFC）这种RFC可以实现异步的RFC调用方式，它可以进行多个并发调用，并且不要求被调用系统的可用状态。发出调用系统会一直尝试直到获得被调用系统的应答。它通常用于当你需要提高系统并行调用多个RFC的效率，相对于强制等待程序的结果，它的效率更高。

c.并行

是一种特殊的RFC，它是aRFC的一种扩展类型。因为它改善了系统的性能，在执行大量的aRFC时。SAP使用它在MRP里面提高速度。但是它只能执行在同一个系统和同一个client里。

还可以去设置队列和分组

```
CALL FUNCTION 'ZPPDE100'  
  STARTING NEW TASK l_taskid  
  DESTINATION IN GROUP l_servgpname  
  PERFORMING result_return "内表去存储是否成功  
  ON END OF TASK
```

假如数据量很大通过RFC 接收/送出，如何处理？ 思路 怎么能快速一些，怎么阔以避免数据异常

接收数据可以考虑通过zip的方式放到文件服务器

然后解压 解压需要解zip 然后 csv 文件的话 再拆文件记录多少一包

发送数据的话 拆包 + 并行call rfc + log 表记录 + 记录错误的记录进行重送

增强

PP release MO 判断mo是否符合条件

SD release dn 如果dn 存在自定义表里 再做特殊处理

MM 创建po 的时候检查日期

增强的种类及相关事务代码

增强就是ERP系统标注程序出口，在该出口中由用户根据企业实际需求编写客户化逻辑代码。

查找：

USER exit:SE37 以EXIT*开头

Customer exit: SMOD（查看增强组件）/CMOD（建立项目后，实现增强）

BADI增强：SE18（查找接口）/SE19（对接口进行实施）

####

- a. 通过查资料 chat gtp 查找
- b.通过 看程序内容 通过相应t-code查找
- c. 测试程序debug 是否跑进去

查看 BAPI 的详细信息: 查看BAPI的接口参数和示例代码。

方法2：使用 SE37 和 SE80

1. **SE37 (Function Builder):**

- 打开事务代码 SE37。
- 在 Function Module 输入框中输入你猜测的关键字（如 BAPI_SALESORDER_CREATEFROMDAT2 ），然后点击搜索按钮。

2. **SE80 (Object Navigator):**

- 打开事务代码 SE80。
- 在 Repository Browser 中选择 BAPI ，然后输入相关的对象名称或关键字进行搜索

使用 SE93 查找 User Exits

1. **查找相关事务代码的程序名称:**

- 使用事务代码 SE93 ，输入事务代码并点击 "Display"。
- 记下显示的程序名称。

2. **搜索程序中的 User Exits:**

- 使用事务代码 SE38 ，打开报告程序。
- 搜索关键字 USEREXIT 或 CALL CUSTOMER-FUNCTION 或 CALL CUSTOMER-SUBSCREEN 。

增强介绍 （必问 都看）

<https://blog.csdn.net/XLevon/article/details/127632234>

ALV

FM ALV 和 OO ALV 的比较

特点	函数模块 (FM) ALV	面向对象 (OO) ALV
实现方式	基于函数模块	基于类和方法

特点	函数模块 (FM) ALV	面向对象 (OO) ALV
易用性	简单易用，适合简单报表	需要更多的代码和理解，适合复杂报表
灵活性和可扩展性	较低，难以实现复杂功能和自定义需求	高，可实现复杂功能和自定义需求
代码组织	代码较为集中，适合小型项目	代码模块化，适合大型项目和复杂业务逻辑
维护性	维护较为困难，代码可读性较低	维护较为容易，代码可读性高

FM ALV

流程： 设置style 设置栏位 设置 数据

如果需要添加什么事件 或者 处理 都需要在call fun 的时候传入参数

ALV刷新

首先找到ALV显示Function的I_CALLBACK_USER_COMMAND参数所对应FROM，然后把FROM所传进来参数的构造SELFIED的REFRESH参数设置为“X”就可以实现

修改颜色

(1) 列颜色：LVC_S_FCAT-EMPHASIZE = 'C610' - 红色

(2) 行颜色：在ALV内表中加入字段：CLR TYPE CHAR4

在LVC_S_LAYO-INFO_FNAME = 'CLR'

在需要控制颜色的行：GS_DATA-CLR = 'C610'

(3) 单元格颜色：在ALV内表中加入结构：SCOL TYPE LVC_T_SCOL，在LVC_S_LAYO-CTAB_FNAME = 'SCOL'

BDC

销售订单多行实现

TCODE:SHDB

步骤：

- 1.SHBD 点击生成程序
- 2.从记录获取数据 然后添加数据
- 3.然后会生成相应代码 复制到 代码中 并且调整相应的栏位

4. 然后调用事务码
5. 进行BDC 消息处理 然后进行 后续的开发 对结果进行验证等等

N模式用来使用 A模式用来debug
可能会有lock 数据情况 做延时处理

准备数据：从外部系统或文件中获取数据，通常以文本文件或 Excel 文件的形式存在。

记录事务（Transaction Recording）：使用事务代码 SHDB 记录用户在 SAP 系统中手动执行的操作，生成记录脚本。

生成 BDC 程序：

使用录制的脚本生成 ABAP 程序。

将外部数据映射到录制的脚本中。

执行 BDC 程序

显示模式DISMODE应设置为“A”或“E”。

A：代表无论出错与否每个画面都显示；

E：只是在出错时显示；

N：代表无论出错与否每个画面都不显示。

BDC两种处理方式 Session 方式 Call transaction方式

a) BDC_OPEN_GROUP

b) BDC_INSERT

c) BDC_CLOSE_GROUP

CALL FUNCTION ‘BDC_OPEN_GROUP’在SE35生成一个session，可以在SE35里手动执行，或在程序里使用命令“SUBMIT RSBDCBTC AND RETURN WITH QUEUE-ID = KEY的ID”来直接执行。

PDATE 当时只记得都填S，并不太记得是什么意思，面试后去查了一下A表示异步，S表示同步，L表示本地

做过的开发任务 列举

MM PP SD FI

MM: User 需要去获取一些关于PO 和 GR 的数据和未来时间内 的库存量的数据，然后需要串多个表去发送数据，

User 通过传输数据到SFTP来更新相应内表

User 在开PO 单的过程中 要对数据进行检测 如果不在相应的范围 就跳过hub id检查 等等
me23n

query 给user 拿到相应的数据

PP: Release mo 生成alv 从内表选数据 去dispaly 然后让user 勾选相应的alv 勾选完后release 然后提示相应的问题

还可以删除release 数据 上传表格去release

smartform 来实现user 的产品单子和需要的原料 遇到问题：翻页 打印机 先拿数据 然后拿到数据之后再去画table 画完之后

SD: Customer 相关的创建 通过外部接口实现 然后最后通过call bapi 来实现

传送customer 的全部资料生成xlsx 资料来实现

修改customer 的相关资料 通过bapi 和webservice 实现

####

table 和 structer 的区别

特性 表 (Table) 结构 (Structure)

定义 用于存储多条记录的数据库表。每个表由多个字段组成，字段数据可以从数据库中查询、插入、更新或删除。用于定义一个单一的数据记录或数据单元，包含多个字段。通常用于数据处理和传输。

数据存储 存储在数据库中，可以持久化存储。 不存储数据，仅在程序运行时存在于内存中。

用途 存储和管理应用程序中的数据，如用户信息、交易记录等。 定义数据的结构，以便在程序中处理数据，例如传递参数或返回值。

访问 可以通过ABAP语句（如SELECT、INSERT、UPDATE、DELETE）访问和操作。 主要用于内存中的数据处理，没有直接的数据库操作。

维护 可以在SAP系统中维护，包含在数据字典中。 只能在程序中定义，不能在数据字典中维护。

at new at end 注意事项

排序内表：

事件触发条件：

嵌套使用：

性能考虑：

数据一致性：

接口/交互

ODATA

a. 建立相应structer

b. segw 建项目

c. 导入structer 导入数据元素

d. 生成类 生成service

e. 重写对应的方法 read=get create/deep=post batch

f. 写具体的代码逻辑

WEB service

a. 功能：方便外部系统进行调用 调用SAP 的RFC BAPI INTERFACE WEBSERVICE 通过web http 协议进行调用

消费webservice。Netweaver通过创建代理能够通过Internet直接调用采用标准协议封装的其他系统提供的webservice

可以java/abap 开发

- b. 配置：RZ10 配置接口参数文件
- c. 创建 相应的function/function model 生成 相应的web service
- d. 配置: ***SOAMANAGER*** webservice 配置 然后创建服务 创建endpoint
- E. 产生soa 管理文件 提供外部地址 然后 在相应的function里面写数据
- F. 在se80 创建代理 配置端口 然后再调用对象方法

SICF restful 接口

1. 创建 ICF 服务：

- 进入事务代码 `SICF`。
- 在路径 `/sap/bc` 下创建一个新的 ICF 服务。
- 配置服务的处理程序类。

2. 创建处理程序类：

- 进入事务代码 `SE24`，创建一个新的 ABAP 类，继承自 `CL_HTTP_EXT_SERVICE`。
- 重定义 `IF_HTTP_EXTENSION~HANDLE_REQUEST` 方法。

3. 实现处理逻辑：

- 在 `HANDLE_REQUEST` 方法中实现你的业务逻辑，处理 HTTP 请求并返回响应

```
`CLASS zcl_my_rest_service DEFINITION
PUBLIC
INHERITING FROM cl_http_ext_service
FINAL
CREATE PUBLIC.

PUBLIC SECTION.
    INTERFACES if_http_extension.
ENDCLASS.

CLASS zcl_my_rest_service IMPLEMENTATION.

METHOD if_http_extension~handle_request.
    DATA: lv_method TYPE string,
           lv_response TYPE string.
```

```
" 获取 HTTP 方法
lv_method = server->request->get_header_field( '~request_method' ).

CASE lv_method.
    WHEN 'GET'.
        " 处理 GET 请求
        lv_response = 'Hello, this is a GET response!'.
    WHEN 'POST'.
```

```

" 处理 POST 请求
lv_response = 'Hello, this is a POST response!'.
WHEN OTHERS.
" 处理其他请求
lv_response = 'Unsupported HTTP method!'.
ENDCASE.

" 设置响应
server->response->set_cdata( lv_response ).
server->response->set_status( i_code = 200 i_reason = 'OK' ).

ENDMETHOD.

ENDCLASS.`

```

Debug 技巧

如果数据很多 debug 错误原因? 如何快速定位

- a. 设置条件断点
- b. 或者修改代码 加if 条件判断

1. Odata service deep 类型 如何进行关联 , 实际上的创建步骤

通过建立data set 然后通过两层数据的某个栏位进行关联 , 也可以进行一对一或者一对多的关联
建立关联之后 再xxx

Report事件

- ① LOAD-OF-PROGRAM 程序执行时, 自动触发该EVENT
- ② INITIALIZTION 对程序变量进行初始化
- ③ AT SELECTION-SCREEN OUTPUT 选择屏幕PBO事件, 显示选择屏幕之前触发
- ④ AT SELECTION-SCREEN 选择屏幕PAI事件, 用户在选择屏幕上执行某些功能
- ⑤ START-OF-SELECTION 选择开始事件
- ⑥ END-OF-SELECTION 选择结束事件, 一般在该事件中输出列表
- ⑦ TOP-OF-PAGE 页眉控制
- ⑧ END-OF-PAGE 页角控制
- ⑨ AT LINE-SELECTION 行选择控制, 用户选择某列表行时触发

FTP文件的上传下载操作是什么？

```
FTP_CONNECT  
FTP_COMMAND  
OPEN DATASET  
FTP_R3_TO_SERVER  
FTP_DISCONNECT
```

模块表的主档数据和关联关系？

财务会计 (FI)

客户主数据表：

KNA1：客户主数据（一般数据）

关键字段：KUNNR（客户编号）

KNB1：客户主数据（公司代码数据）

关键字段：KUNNR（客户编号），BUKRS（公司代码）

供应商主数据表：

LFA1：供应商主数据（一般数据）

关键字段：LIFNR（供应商编号）

LFB1：供应商主数据（公司代码数据）

关键字段：LIFNR（供应商编号），BUKRS（公司代码）

总账科目表：

SKA1：总账科目主数据（科目表）

关键字段：SAKNR（总账科目编号）

SKB1：总账科目主数据（公司代码数据）

关键字段：SAKNR（总账科目编号），BUKRS（公司代码）

物料管理 (MM)

物料主数据表：

MARA：物料主数据（一般数据）

关键字段：MATNR（物料编号）

MARC：物料主数据（工厂数据）

关键字段：MATNR（物料编号），WERKS（工厂）

MARD：物料主数据（库存数据）

关键字段：MATNR（物料编号），WERKS（工厂），LGORT（库存地点）

采购信息记录表：

EINA: 采购信息记录 (一般数据)
关键字段: INFNR (信息记录编号)
EINE: 采购信息记录 (采购组织数据)
关键字段: INFNR (信息记录编号), EKORG (采购组织)

销售与分销 (SD)

客户主数据表:

KNA1: 客户主数据 (一般数据)
关键字段: KUNNR (客户编号)
KNVV: 客户主数据 (销售数据)
关键字段: KUNNR (客户编号), VKORG (销售组织), VTWEG (分销渠道), SPART (产品组)
销售订单表:

VBAK: 销售订单头数据
关键字段: VBELN (销售订单编号)
VBAP: 销售订单行项目数据
关键字段: VBELN (销售订单编号), POSNR (行项目编号)

生产计划 (PP)

物料主数据表:

MARA: 物料主数据 (一般数据)
关键字段: MATNR (物料编号)
MARC: 物料主数据 (工厂数据)
关键字段: MATNR (物料编号), WERKS (工厂)
生产订单表:

AUFK: 生产订单头数据
关键字段: AUFNR (生产订单编号)
AFPO: 生产订单行项目数据
关键字段: AUFNR (生产订单编号), POSNR (行项目编号)

SAP bapi 里为什么不能commit

那种东西

然后commit 是对的

但是 你call 他一个bapi 他 一整个流程 是一个最小流程

然后只做了一半

就commit 了 就会导致后续 没法rollback 等等

ABAP的三层结构

表示层：AP GUI（Graphical User Interface）、Web Dynpro、SAP Fiori、SAP Business Client

应用层：ABAP程序、功能模块（Function Modules）、类（Classes）、方法（Methods）、BAPIs（Business Application Programming Interfaces）、事务（Transactions）等。

业务规则和计算逻辑，是实现系统核心功能的地方。可以将其进一步划分为多个应用服务器，来分担负载和提高系统的可靠性。

数据库：SAP HANA、SAP ASE、Oracle、Microsoft SQL Server等数据库系统。

透明表和池表的区别

在 SAP ABAP 中，数据库表分为几种类型，其中透明表和池表是两种主要类型。它们各自具有不同的用途和特点。以下是透明表和池表的详细区别：

透明表（Transparent Tables）

定义：透明表用于存储应用数据，与数据库表有一一对应关系。每个透明表在 ABAP 数据字典中都直接对应一个底层数据库中的表。

池表

池表用于存储控制数据、配置信息和其他不常访问的数据。多个池表被组合在一起，存储在一个称为表池（Table Pool）的数据库表中。

特点	透明表 (簇表)	池表
定义	与数据库表一一对应	多个池表存储在一个表池中
数据存储	数据按照定义直接存储在数据库表中	不同表的数据组合存储在一个表池中
索引	支持主索引和辅助索引	索引功能有限
性能	性能较好，直接映射	由于存储/检索的复杂性，性能可能较差
使用场景	事务数据和主数据	配置信息、控制数据、不常访问的数据
直接对应	是	否

数据分类

数据类别	特征	示例
主数据	相对稳定，变化较少	物料主数据、信息记录、BOM
业务数据	经常变化，记录业务活动和交易	物料凭证、会计凭证
组织数据	定义企业结构和业务流程，变化较少	工厂、公司代码、销售组织
系统数据	由系统使用，用于系统配置和管理	系统配置数据、系统日志数据

面向对象和面向过程

特征	面向过程	面向对象
编程范式	过程式编程	面向对象编程
代码组织	功能模块、子程序	类、对象
数据处理	内表、工作区	对象的属性和方法
复用和扩展	功能模块、子程序	继承、多态、接口
调试工具	标准 ABAP 编辑器和调试工具	支持对象视图和方法调用的高级调试功能
设计和维护	设计复杂	设计清晰，维护方便

性能

a. 代码层面进行 管控

- a. select 尽量写在外面不要嵌套再loop
 - b.避免是用select * 和select end select
 - c. 如果有必要进行并行处理
 - d. 避免不必要的循环 及时退出
 - e.对数据进行筛选 尽量提前添加条件
 - f. 使用 cds view 获取数据
 - g. read table 用二分法 sort 排序
 - h. 使用hash 表进行read

- b. 流程层面 利用缓存机制 集中数据处理
- c. DB 层面 对于一些常用的数据量大的表 建立表 的时候允许缓存
- d.使用相应 t-code 进行校验sat se30 查看具体的程序运行情况

ST22 异常追踪

ST05 SQL追踪

SE30 SAT 追踪程序的处理时长 以及各个function

常用函数

文件操作： GUI_UPLOAD GUI_DOWNLOAD

解析xls： ALSM_EXCEL_TO_INTERNAL_TABLE

解析csv: DATA:lo_csv TYPE REF TO cl_rsda_csv_converter. CALL METHOD
cl_rsda_csv_converter=>create

文件在服务器上用DATASET, 在客户端用函数, 下载用WS_EXCEL, 上传可以用
ALSM_EXCEL_TO_INTERNAL_TABLE。

常用T-code

SE38 程序创建

SE37 函数创建

SE11 表、元素、域、结构、搜索帮助、锁对象、表类型、表视图创建

SE80 函数组、类等查询创建

ST05 查询性能

SM59 ABAP链接 RFC连接

SE24 创建类

SE93 查询创建事务代码

SE91 查询消息

SM12查询锁表情况

SM36 SM37 job 管理

SU20 21 用于权限对象检查 authoritic check

BDC:SHDB 性能bapi 更快

BAPI:BAPI

MM模块的主要t-code:

MMBE 库存总览

ME51N 采购申请

ME21N 采购订单

MIGO 收货

MIRO 后勤发票校验

SD模块的主要t-code:

VA11 询价

VA21 报价

VA01 销售订单

VL01N 出库

VF01 开具发票

Form perform中using、changing、using value、changing value的区别？

	USING	USING VALUE	CHANGING	CHANGING VALUE
参数传递方式	应用传递	值传递	应用传递	值传递
值改变与否	改变	不改变	改变	当正常结束后改变

in background task 和 starting new task :

- (1) in background task 需要等到commit work 显示提交时才执行，
- (2) starting new task 是异步执行

锁

首先要在 ABAP 字典中创建锁对象，然后才能在 ABAP 程序中设锁。创建锁对象时，系统会自动生成两个 FM 来进行锁管理。

用于设锁的 FM 为：ENQUEUE_<锁对象名>。它用于在锁表（Lock Table）中生成一个锁项（Lock Entry）。若设锁不成功的话，就会在 Return 中反映出来。

用于释放锁的 FM 为：DEQUEUE_<锁对象名>。它用于从锁表中删除一个锁项。在 ABAP 程序中，只需使用 "CALL FUNCTION ..." 语句就可以调用它们。

这两个锁 FM 是在 SAP 系统的一个特殊工作进程中执行的，专门进行锁管理。它运行在一个单独的服务器上，而该服务器专门用于维护整个 SAP 系统的主锁表（Central Lock Table）。

有两种锁类型：

1. 共享锁——只读锁，一个用户正在读数据时，阻止其他用户更改该数据。
2. 独占锁——可写锁，一个用户正在修改数据时，阻止其他用户更改该数据。

使用for all entries in 的注意事项

- (1) IN条件所在内表不能为空（空值将取出所有数据）
- (2) IN条件内表数据量大于5000时不建议使用，一定要使用请每次使用3000，循环的方式使用
- (3) 该语句自动去重，使用时尽量多的带出主键（不清楚的时候全带出来就行）

Dialog 对话框的事件和逻辑流

首先出发PBO->子画面PBO，然后显示dynpro画面，当在dynpro点击按钮时，会触发PAI->触发子PAI，触发完之后又会执行PBO->子PBO

PROCESS ON HELP-REQUEST.(F1帮助)

PROCESS ON VALUE-REQUEST. (F4帮助)

Function ALV如何做到单击跳转事务

- (1) 首先在fieldcat中设置 hotspot = 'X'
- (2) 然后再USER_COMMAND事件对应Form中 加入

CASE R_UCOMM.

WHEN '&IC1'.

然后调用

SET PARAMETER ID 'BLN' FIELD LS_ALV-VBLNR.

SET PARAMETER ID 'BUK' FIELD LS_ALV-ZBUKR.

SET PARAMETER ID 'GJR' FIELD LV_GJAHR.

CALL TRANSACTION 'FB03' AND SKIP FIRST SCREEN.

SMARTFROMS中

换页

- (1) 在循环中插入命令
- (2) 在命令条件中加入触发的条件

table 和 template 的区别

tabel比较灵活，我比较喜欢用这个，只需要设置格子的宽度，高度不固定；设置好宽度之后如果一行存不下会自动换行。

template需要把格子的宽度和高度都设置好，如果内容放不下会自动截断。

CDS View

core data service view hana 得虚拟数据模型

SAP 官方API 使用

get material 的API 用来暴露

Inventory 的内容来实现

