

作业需求：利用 IDOC 自动生成销售订单

目录：

1. 作业目的和安排	1
2. 作业简单需求	1
3. 作业要求	1
4. 作业过程	2
5. 时间统计	10
6. 知识和经验总结	11

1. 作业目的和安排

本作业的目的是：

1. 为了进一步熟悉 ABAP 的编程技巧；
2. 更主要的目标的：熟悉接口的业务场景；
3. 熟悉 IDOC 的使用流程；

2. 作业简单需求

开发一个 IDOC：

实现客户在电子商务网站上的订单（.CSV 格式存放在本地）通过 IDOC 转化为 SAP 系统里的销售订单，同时在 VA01 里面做一个抬头增强，增加一个字段存放客户的采购订单编号。

大致的需求是：http://res.yitai.edu.com/sap/abap/Case/02_xiaoshen/SAPABAP_IDoc_Dev_01.htm

3. 作业要求

时间要求：

1. 郭裕的要求

6 天左右开发完成毛坯程序；

2. 一般的项目要求

这种开发需求，一般的项目会基于 15-30 天不等的的时间，基于实际项目的 IDOC 的接口数量和复杂度的不同。比如固安捷的项目，基本上作为一个 6 个月的项目在做，因为电子商务网站和 SAP SD 模块的存在比较多的 IDOC 的接口。

4. 作业过程

1. 充分分析需求:

刚开始接到老师的作业, 完全一头雾水, 不知道想要实现的结果是什么样的, 不知道用哪些东西来实现, 目前: 我们只是从视频里学习到了一个 ALE 的完全的配置过程, 简单理解了配置的原理, 但是直接用到系统自带的 message type, 我以为老师会先教我们一下怎么自己开发一个 IDOC, 怎么使用, 看看实现的结果是什么, 老师说这个可以通过完成这次的作业来学习, 而且印象更加深刻, 所以就开始硬着头皮查资料了

首先是理解老师的需求:

因为没有头绪, 所以在网上找到一个配置 idoc 的文档:

<http://www.mamicode.com/info-detail-454486.html>

写的非常详细, 完全是自己创建一个 idoc/message type, 自己写发送 IDOC 的程序, 自己在发送端配置, 在接收端配置, 以及自己写处理 IDOC 的 function, 和自己创建处理代码等等:

步骤如下: (800 端发送一个 IDOC 给 810 端)

800 端配置:

WE31 创建段类型: YPOHEAD4 YPOITEM4

WE30 创建 IDOC: YPOIDOC4

WE81 创建 Message Type: YPO_MESS_TYPE4

WE82 关联消息类型和 IDOC:

SM59 创建一个到 810 的物理连接: ZTO8104 (RFC 目的地)

WE21 创建端口: TO810PORT4

SALE 创建逻辑系统并分配: 800: Z800LS4

SALE 创建逻辑系统并分配: 810: Z810LS4

WE20 创建 partner profile: Z810LS4

通过 ABAP 程序发送 IDOC

```
REPORT Z_SWJ4.
```

```
DATA:is_head TYPE YPOHEAD4,  
      is_item TYPE YPOITEM4,  
      is_EDIDC TYPE EDIDC,  
      it_EDIDC TYPE STANDARD TABLE OF EDIDC,  
      it_EDIDD TYPE STANDARD TABLE OF EDIDD WITH HEADER LINE.
```

"通过以下四句话可以与 WE20 的配置关联起来

```
CLEAR is_EDIDC.
```

```
is_EDIDC-MESTYP = 'YPO_MESS_TYPE4'.
```

```
is_EDIDC-IDOCTP = 'YPOIDOC4'.
```

```
is_EDIDC-RCVPRN = 'Z810LS4'.
```

```
is_EDIDC-RCVPRT = 'LS'.
```

```
CLEAR it_EDIDD.
```

```
it_EDIDD-SEGNAM = 'YPOHEAD4'. "头节点
```

```
it_EDIDD-DTINT2 = 0. "VARC 字段的长度字段
```

```

CLEAR is_head.
is_head-EBELN = '5000000021'.      "采购订单号
is_head-BUKRS = '1000'.           "公司代码
is_head-BEDAT = '20100116'.       "日期
it_EDIDD-SDATA = is_head."节点内容:ls_pohead 结构中的数据最后被拼接成字符串再赋值给 lt_edidd-sdata,
最大长度不能超过 1000
APPEND it_EDIDD.

CLEAR it_EDIDD.
it_EDIDD-SEGNAM = 'YPOITEM4'.     "行项目节点
it_EDIDD-DTINT2 = 0.              "VARC 字段的长度字段

CLEAR is_item.
is_item-EBELN = '5000000021'.     "采购订单号
is_item-EBELP = '1'.              "行项目号
is_item-MATNR = 'MM01'.           "物料号
is_item-MENGE = '10'.             "单位
is_item-MEINS = 'ST'.             "数量
it_EDIDD-SDATA = is_item.
APPEND it_EDIDD.

CLEAR it_EDIDD.
it_EDIDD-SEGNAM = 'YPOITEM4'.     "行项目节点
it_EDIDD-DTINT2 = 0.              "VARC 字段的长度字段

CLEAR is_item.
is_item-EBELN = '5000000021'.     "采购订单号
is_item-EBELP = '2'.              "行项目号
is_item-MATNR = 'MM02'.           "物料号
is_item-MENGE = '20'.             "单位
is_item-MEINS = 'ST'.             "数量
it_EDIDD-SDATA = is_item.
APPEND it_EDIDD.

CALL FUNCTION 'MASTER_IDOC_DISTRIBUTE'
  EXPORTING
    MASTER_IDOC_CONTROL          = is_EDIDC
*   OBJ_TYPE                     = ''
*   CHNUM                        = ''
  TABLES
    COMMUNICATION_IDOC_CONTROL  = it_EDIDC
    MASTER_IDOC_DATA            = it_EDIDD
  EXCEPTIONS
    ERROR_IN_IDOC_CONTROL       = 1
    ERROR_WRITING_IDOC_STATUS   = 2

```

```

ERROR_IN_IDOC_DATA                = 3
SENDING_LOGICAL_SYSTEM_UNKNOWN    = 4
OTHERS                             = 5
.
IF SY-SUBRC <> 0.
  MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
    WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.

ELSE.
  commit WORK.

  WRITE: 'IDOC sent:'.
  LOOP AT it_EDIDC into is_EDIDC.
    WRITE:/ 'IDOC number is :', is_EDIDC-DOCNUM,
      /'接收者伙伴是: ', is_EDIDC-RCVPRN,
      /'发送者伙伴是: ', is_EDIDC-SNDPRN.

  ENDLOOP.
ENDIF.之后保存激活

```

BD87 检查, 测试

刚开始发送的状态是黄灯, 点击‘处理’之后, 变成绿灯了, 这一步从第一次做到最后一次做都是没有问题的, 成功!!

然后: 在 810 端设置:

因为 800 和 810 在一个系统里面, 并且段类型, IDOC 类型, 消息类型这些都是跨客户端的, 所以在这里不需要在 810 端再配置一次 Segment、IDoc Type、Message Type, 这些是共享的, 但如果不是在同一个服务器上, 就需要重新配置一下:

直接:

SM59: 创建 RFC 目的地: ZFROM8004

WE21 创建端口: FRM800POR4

创建入站处理函数: Y_IDOC_PO_PROCESS4 (先创建一个函数组: Z_RFC_SWJTEST1) (写参数, 代码):

```

UNCTION Y_IDOC_PO_PROCESS4.
*-----
*""Local interface:
*  IMPORTING
*    VALUE (INPUT_METHOD) LIKE  BDWFAP_PAR-INPUTMETHD
*    VALUE (MASS_PROCESSING) LIKE  BDWFAP_PAR-MASS_PROC
*  EXPORTING
*    VALUE (WORKFLOW_RESULT) LIKE  BDWF_PARAM-RESULT
*    VALUE (APPLICATION_VARIABL) LIKE  BDWF_PARAM-APPL_VAR
*    VALUE (IN_UPDATE_TASK) LIKE  BDWFAP_PAR-UPDATETASK
*    VALUE (CALL_TRANSACTION_DONE) LIKE  BDWFAP_PAR-CALLTRANS
*  TABLES
*    IDOC_CONTRL STRUCTURE  EDIDC
*    IDOC_DATA STRUCTURE  EDIDD
*    IDOC_STATUS STRUCTURE  BDIDOCSTAT
*    RETURN_VARIABLES STRUCTURE  BDWFRETVAR

```

```

*"      SERIALIZATION_INFO STRUCTURE  BDI_SER
*"      EXCEPTIONS
*"      WRONG_FUNCTION_CALLED
*"      OTHERS
*"-----

DATA: ls_thead TYPE ypothead4,
      ls_citem TYPE ypoitem4.
CLEAR idoc_contrl.
READ TABLE idoc_contrl INDEX 1.
IF idoc_contrl-mestyp <> 'YPO_MESS_TYPE4'.
  RAISE wrong_function_called.
ENDIF.
LOOP AT idoc_contrl.
  LOOP AT idoc_data WHERE docnum = idoc_contrl-docnum.
    CASE idoc_data-segnam.
      WHEN 'YPOHEAD4'.
        "直接将字符赋值给结构，赋值过程中会按照结构中的字段长度来划分各字段
        ls_thead = idoc_data-sdata.
        WRITE: / 'Head',ls_thead.
      WHEN 'YPOITEM4'.
        ls_citem = idoc_data-sdata.
        WRITE: / 'Item',ls_citem.
      WHEN OTHERS.
    ENDCASE.
  ENDLLOOP.
  "根据数据处理情况设置当前 IDoc 处理的状态
  IF sy-subrc = 0.
    CLEAR idoc_status.
    idoc_status-docnum = idoc_contrl-docnum."当前正处理的 IDoc
    idoc_status-status = '53'. "IDOC 处理成功
    APPEND idoc_status.
  ELSE.
    CLEAR idoc_status.
    idoc_status-docnum = idoc_contrl-docnum.
    idoc_status-status = '51'. "IDOC 不成功
    idoc_status-msgty = 'E'. "错误信息
    idoc_status-msgid = 'YMSG'.
    idoc_status-msgno = '001'.
    APPEND idoc_status.
  ENDIF.
ENDLLOOP.

ENDFUNCTION.

```

BD51 注册入站处理函数:

将入站函数与 IDOC Type/Message Type 关联 (WE57)

创建入站处理代码 Inbound Process Code (WE42)

创建发送端合作和伴配置文件 Partner profile (WE20)

BD87 测试, 但是测试的结果一直是红灯, 状态显示 56 或者 51 等等

Z800LS6		28
出站处理的 IDocs		6
进站处理的 IDoc		22
申请文件未过帐	51	10
ZSO11		10
IDoc 添加错误	56	12
ZSO11		12

做了很多遍一直显示错误, 因为对 IDOC 了解得不透彻, 所以也不知道在哪里做错了, 都是一些配置的操作, 对代码 debug 也找不到问题, 后来, 终于把代码里修改了一个语句:

```
ENDLOOP.
```

```
"根据数据处理情况设置当前IDoc处理的状态
```

```
IF 1 = 0.
```

```
  CLEAR idoc_status.
```

```
  idoc_status-docnum = idoc_contrl-docnum."当前正处理的IDo
```

```
  idoc_status-status = '53'. "IDOC处理成功
```

```
  APPEND idoc_status.
```

```
ELSE.
```

```
  CLEAR idoc_status.
```

```
  idoc_status-docnum = idoc_contrl-docnum.
```

```
  idoc_status-status = '51'. "IDOC不成功
```

```
  idoc_status-msgty = 'E'. "错误信息
```

```
  idoc_status-msgid = 'YMSG'.
```

```
  idoc_status-msgno = '001'.
```

```
  APPEND idoc_status.
```

```
ENDIF.
```

```
ENDLOOP.
```

```
ENDFUNCTION.
```

这个是错误的, 后来改成了这样的:

```
37 | WHEN 'YPOITEM4'.
38 |   ls_citem = idoc_data-sdata.
39 |   WRITE: / 'Item', ls_citem.
40 | WHEN OTHERS.
41 | ENDCASE.
42 | ENDOLOOP.
43 | "根据数据处理情况设置当前IDoc处理的状态
44 | IF sy-subrc = 0.
45 |   CLEAR idoc_status.
46 |   idoc_status-docnum = idoc_contrl-docnum."当前正处理的IDo
47 |   idoc_status-status = '53'. "IDOC处理成功
48 |   APPEND idoc_status.
49 | ELSE.
50 |   CLEAR idoc_status.
51 |   idoc_status-docnum = idoc_contrl-docnum.
52 |   idoc_status-status = '51'. "IDOC不成功
53 |   idoc_status-msgty = 'E'. "错误信息
54 |   idoc_status-msgid = 'YMSG'.
55 |   idoc_status-msgno = '001'.
56 |   APPEND idoc_status.
57 | ENDIF.
58 | ENDOLOOP.
59 |
60 | ENDFUNCTION.
```

结果成功了!!!

ALE 消息的状态监视器		
IDocs	IDoc 状态	数量
- IDoc 选择		
- 更改日期 的范围从 10.02.2017 到 10.02.2017		
- Z810LS6		5
- 进站处理的 IDoc		5
- 应用凭证已过帐	53	5
- YPO_MESS_TYPE4		5
- (000) : (无错误消息)		5

总结:做成功之后才发现,这个问题太小太小了!!!也太简单了!!!!想想之前这么没有头绪,真实感慨,碰到一个问题,不知道怎么解决,就一步一步的去理解它,再 看看哪里有问题!!这是最简单最笨的方法了!!!

至此,我们对 IDOC 有了一个最基本的理解,根据网上的文档能开发一个最简单的 IDOC 了!

然后再回过头来,看看老师的作业需求:利用 IDOC 将一个文件转换从销售订单,威力避免作业过于复杂,老师说只需要自己在本地创建一些数据(一张普通的采购订单,如果不知道有哪些字段,我们可以去超市买一些东西,看看小票就知道了)存在 CSV 格式的表里,用 idoc 把这些数据生成销售订单,老师说 SAP 系统自带有 IDOC,可以创建销售订单,我们只需要找到它,再 inbound 处理一下,就能生成销售订单了,但是怎么把本地数据上传,并且变成 IDOC 文档呢?

这个问题困扰了我们很久,因为我们在网上找到一个文档:

<http://www.docin.com/p-1519577065.html> (通过电子数据交换(EDI)内向处理销售订单)

很详细的讲了怎么把一个客户的采购订单利用 IDOC 的形式转换成另一个客户的销售订单,因为对这里面的逻辑还不是特别懂,所以大概看了一下配置过程,就模仿着梳理他的流程,看能不能做出来,可以做到后来,发现需要配置的东西太多了,比如在 800 系统需要确定一个采购组织/供应商/物料/销售组织/分销渠道/产品组/项目类别组/工厂/公司代码等等,然后在接收端也要做对应的配置,关键是我们的系统 800 还好,很多配置已经做好了,但是 810 系统因为用得比较少,里面很多事务都不存在,各种采购/销售的配置也没有,我们需要从头开始配置,因为工作量太大了,而且这个文档是用业务顾问的角度来做到,机会没有 ABAP,所以我们后来还是放弃按照这个文档来做了,不过在做的过程中,也理解到了,如果我们想要把这个作业做完,只是写写代码,配置 ALE 还是不够的,需要把详细的客户主数据,供应商主数据,物料主数据都配置好,在两个公司之间把数据打通,并且要弄清楚 800 创建销售订单的时候,会自动生成 IDOC 文档,自动发送给 810,接收端自动生成销售订单,需要哪些字段也要先配置好,否则容易出错(之前对这一点的理解不够深刻,模模糊糊的),总之,因为对 IDOC 的概念理解得不清晰,所以兜兜转转的,也不知道正确的思路是什么,看这个文档感觉有点像,看那个文档又觉得能做出来,但是每个都做不出来

下面是练习的过程:

2月10日练习：尝试着按照网上的业务顾问的一个文档配置：

发送端配置：

- 创建 LS, 分配 LS(注意设置一下)
- 创建 RFC 目的地 (两个系统都设置, 并且都测试一下是否成功)
- 创建端口 WE21
- XK02 修改供应商 (在 WE31 里查看段类型: E1EDK18, 他的段定义 E2EDK18 里面有个字段: QUALF, IDOC 资格: 付款条件)
- 选择供应商 1001, 公司代码 1000, 选择公司代码数据里的“支付交易”, 进去可以看到付款条件, 修改一下 (不同的供应商和公司代码不统一)
- 选择供应商 1001, 公司代码 1000, 选择公司代码数据里的“信函”, 进去可以看到: 带供应商科目, 修改一下, 填写“1001”
- 注意: 这里 1001 就是接收端创建的售达方的代码, 需要在接收端创建一个: 某个销售范围下的一个客户编码
- 创建 partner profile 之供应商
- 创建 partner profile 之逻辑系统

接收端配置：

- 创建逻辑系统并分配
- 创建 RFC 目的地
- 创建端口
- 定义伙伴参数
- 查看生成的 IDOC

XD02 客户设置：

创建销售订单需要的字段：

- 订单类型 VBAK-AUART 'SO'
- 销售组织 VBAK-VKORG '1000'
- 分销渠道 VBAK-VTWEG '12'
- 产品组 VBAK-SPART '00'
- 售达方 KUAGV-KUNNR '1001'
- 物料 VBAP-MATNR 'M-05'
- 订单数量 RV45A-KWMENG '10/9/8...'
- 增强的一个字段: 采购订单号:

200client 是 001client 的客户 (编号 1001, 公司代码 1000, 销售组织 1000, 分销渠道 12, 产品组 00)


001client 是 200client 的供应商 (编号 200, 公司代码 2100)

XD02 输入客户 1001, 进入公司代码数据-信函-客户处我们编号“2002”,

已针对公司代码 1000 销售区域 2300 10 00 建立客户 100002

需要设置以下这个物料“M-05”在这个工厂+销售组织+分销渠道里面的状态

process code : ORDE (ORDERS 创建销售订单), function : IDOC_INPUT_ORDERS



供应商 0000002002 对于公司代码 1000 采购组织 1000 已被创建 (账户组: 0001)

创建物料: ZSL-007

Vendor 0000002003 has been created for company code 2300 purchasing organization 2300

swi

因为 810 当中各种物料, 供应商, 客户的数据要配置的太多了, 所以放弃使用这个方法做

http://wenku.baidu.com/link?url=2ylqdlmKxBe3FAZ_-j15nMiCD9zU03jVguSlvsFJhikQ40Fj3PcbvMUy4zZR890zIumBqwapYazF56DTNr3e-xW6iUHO71lppmxZFbzn9e

现在重新整理思路和方法：

把数据上传到 810 端, 变成 IDOC 文档, 发送到 800 端 (自家创建 message type, IDOC type 等, 自己写 ABAP 代码发送 IDOC)

在 800 端处理 IDOC, 让他自动变成 SO

步骤：

首先设置 810

最后, 在我们跟老师确定了一下我们的思路之后, 得到了他的肯定回答我们才知道正确的道路是什么: 首先用我们之

前学习的上传文档的代码，把本地的 CSV 文件上传，变成一个内表，然后用我们学到的发送 IDOC 的程序，还有函数把这些数据发送出去，自己创建 IDOC，message type，RFC 目的地，partner profile，端口，发送端就可以了，然后在接收端处理，老师确定说系统里面有自带的 message type 和 process code 可以将进来的 IDOC 自动处理成销售订单，我们只需要找到它，使用他就可以了!!（有了老师认可的思路，我们就觉得有头绪了，不像之前一直对每个知识点都迷迷糊糊的）

最后我们按照如下步骤来做的：

- 1、 创建 segment (WE31): zher, (字段: EBELN BUKRS AEDAT),, ZITEM,, (字段: EBELP MATNR MENGE MEINS)
- 2、 创建 IDOC Type (WE30): zpotoso
- 3、 创建 Message Type (WE81) : zso11
- 4、 关联 Message Type 和 IDOC Type (WE82)
- 5、 创建接收端 RFC Destination (SM59) ZTO8006, 并测试
- 6、 创建到收端的端口 (WE21) : TO800PORT6
- 7、 创建发送端 Logical System 并分配 (SALE) : Z810LS6
- 8、 创建接收端 Logical System (SALE): Z800LS6
- 9、 创建接收端合作和伴配置文件 Partner profile (WE20) : Z800LS6

找到需要用到的几个数据：

系统原有的进站处理函数: **IDOC_INPUT_ORDERS**

系统原有的进站 process code: **ORDR**

系统原有的 message type: **ORDERS**

接收端配置：

创建逻辑系统并分配

创建 RFC 目的地

定义伙伴参数（**进站处理函数: IDOC_INPUT_ORDERS, process code: ORDR, message type: ORDERS**）

查看生成的 IDOC

结果：

- 1, 系统报错：发送 IDOC 不成功，后来发现是我们上传文件的时候，字段不匹配，所以文件没有上传上来，没法生成 IDOC，因为创建的 IDOC 默认一个 IDOC 编号对应一个采购订单，但是一个采购订单可能有多个行项目，所以给 IDOC 节点的时候，需要 loop 上传的行项目的内表，一个一个给数据！**这个是属于比较小儿科的问题了，因为我们很早的时候就学习了怎么用 ABAP 上传一个 CSV 文档，数据存在一个内表里就可以处理或者使用了！因为这方面的练习少，所以才轻易出错!!!**解决之后，800 端 IDOC 总算可以成功发送出去了！
- 2, 系统报错，说 message type 和 IDOC 不匹配，找到原因：我们发送端使用的 message type 是自己创建的，字段也是

自己创建的，接收端的 message type 是使用系统原来有的，在发送 IDOC 的时候，如果两边都是 IDOC，那么两边的 message type 要是一样的，至少 segment type 里面的字段要是一样的啊！解决办法：在接收端使用我们自己创建的 message type，然后把这个 message type 和系统原有的处理函数：IDOC_INPUT_ORDERS 关联，然后使用这两个数据重新创建一个 process code，使用这个 process code 在 partner profile 里面处理进站 IDOC，最后我们看到 IDOC 数据过来了

至此，接收端收到 IDOC 了，但是用它处理成 SO 还有一些问题：因为出现错误消息：销售组织，分销渠道，产品组等等都没有确定

那么我们需要创建一个标准销售订单，看看他需要哪些数据，然后把这些数据传给 IDOC，再来处理

订单类型：OR VBAK-AUART
 Sales Organization: 1000 VBAK-VKORG
 Distribution Channel: 10 VBAK-VTWEG
 Division: 00 VBAK-SPART
 售达方: 2130 KUAGV-KUNNR
 物料号: R-1141 VBAP-MATNR E1EDP01
 数量: 4 RV45A-KWME

此方案失败，因为系统原有的 IDOC: orders05 里面的所有的字段都没有，我们创建销售订单需要的这几个字段呀（此 IDOC 里面字段超级多，段类型很很多，用的最笨的办法，一个一个打开找的，还是没有找到，不知道出了什么问题???)

学习 BAPI:

根据网上的文档自己创建一个 BAPI

file:///C:/Users/swj/Desktop/SAP 中创建 BAPI 笔记(2)小明下载的.pdf

5. 时间统计

完成该练习的学员：*号的时间是必须填写的

项目	内容	工作时间
1*	业务需求理解和分析	大概 8 天（1/19，1/20，2/8-2/13）过年中间有 20 天休息，没有学习
2*	编写程序前的逻辑图（程序逻辑快划分）	大概 1 天，因为前期理解业务花了特别多时间，而且因为不动的东西太多了，所以各种网上查资料，只好一边照着做配置练习，一边理解
3*	编写和调试程序	大概 1 天，之前也调试了很多次，所以最后一次调试用时较少
4	编写程序后的总结的逻辑图	

5*	知识和经验总结	1 天，未完待续
6	其他问题和难题的解决时间	全部一起总共大概 11 天

练习效果评估：结对评分，学员之间相互结成对子，彼此给对方评分。

项目	内容	分值	学员自评	结对评分	老师评分
1	业务需求理解和分析				
2	程序结果正确性				

错误集锦：

- 1, 不理解什么是对象类型, BAPI 名称, 业务对象名称……这其中的各种名词
- 2, 忘记了怎么创建数据元素, 之前学习过的, 重新做了一遍
- 3, 知道了为什么一个函数组里面只能放一个函数了
- 4, pass value 的使用
- 5, BAPI 的“释放”是什么意思?

6. 知识和经验总结

花了这么多时间做 IDOC 的练习, 最后的结果也没有完全做出来, 只是学会了自己创建 IDOC, 发送它, 在接收端处理, 确保能够在接收端里看到 IDOC 的数据, 因为自动创建 SO 的话还需要配置很多主数据, 所以我们没能在 VBAK, VBAP 里面看到自动传过来的数据 (当然, 最理想的结果, 就是我们上传一个 CSV 的数据, 然后在接收系统的 VA03 里面能显示出来有这个订单,) 但是在做练习的过程中, 还是实实在在的学到的 IDOC 的原理, 真是如老师所说, 这些知识需要自己各种摸爬滚打, 探索出来, 自己才能印象深刻, 如果只是看看视频, 一学而过, 估计过几天自己也会忘掉了, 所以还是踏踏实实的按照老师的学习思路一步一步做!

用本文档记录自己学习 IDOC 的过程!