|  |
| --- |
| 自定义IDOC发送与接收实例 |
|  |
| 自定义IDOC发送与接收实例 |

idoc是基于sap自己的类似xml格式的文档数据交换的方式。idoc基于文档，可以实现异步的。

该实例使用800发送端向810接收端发送Idoc进行实验

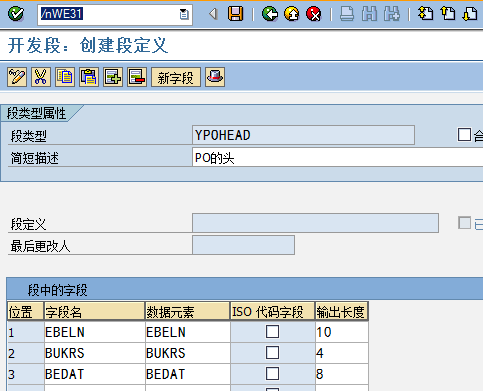
### 在发送端（outbound）中配置

以下步骤都是在Client 800进行设置

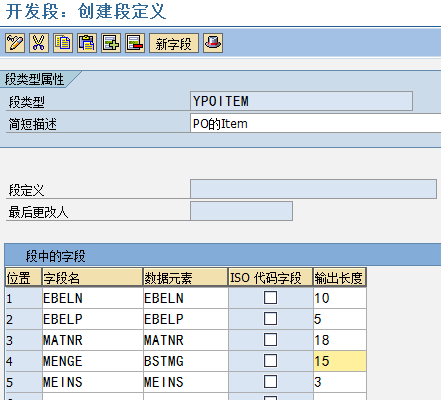
#### 1、创建segment（WE31）

segment，类似于创建XML的节点及节点属性，即定义XML文档中的节点及节点属性。

这里先输入YPOHEAD，点击创建，在接下来的屏幕中，录入EBELN, BUKRS, BEDAT等字段及他们对应的data element：



接着创建YPOITEM，输入EBELN, EBELP, MATNR, MENGE, MEINS等字段及他们对应的data element：



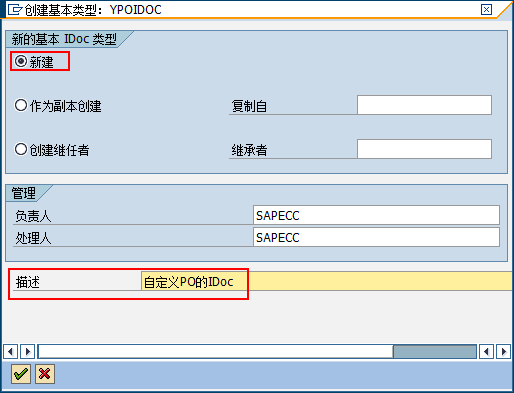
保存后用SE11查看你将发现，系统自动添加了YPOHEAD和YPOITEM两个结构，

每个字段都成了CHAR类型，长度就是WE31中的EXPORT LENG（输出长度）。

#### 2、创建IDOC Type（WE30）

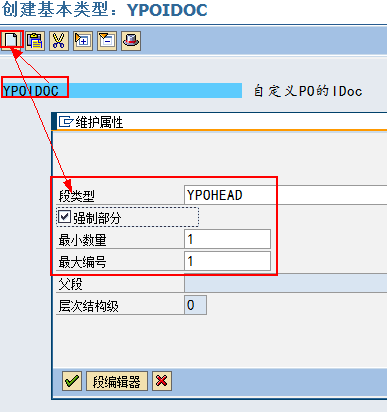
创建IDOC Type，定义结点间的相互逻辑关系

先输入YPOIDOC，然后点击创建，紧跟着点击create new进入：

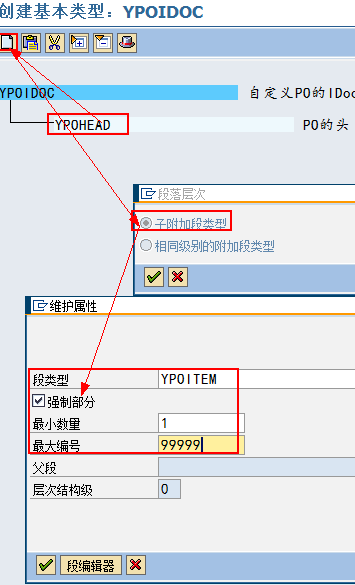


在主界面中，先点击创建按钮，将YPOHEAD添加，设置Mandatory seg打勾，

min ＝ 1， max ＝ 1，代表我们每个IDOC仅包含一张采购订单：



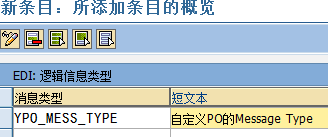
然后在YPOHEAD下添加YPOITEM，同样的Mandatory seg打勾，min ＝ 1， max ＝99999：



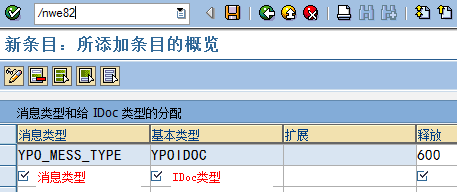


#### 3、创建Message Type（WE81）

 先切换到编辑状态，然后点击New Entries，输入YPO\_MESS\_TYPE即可。



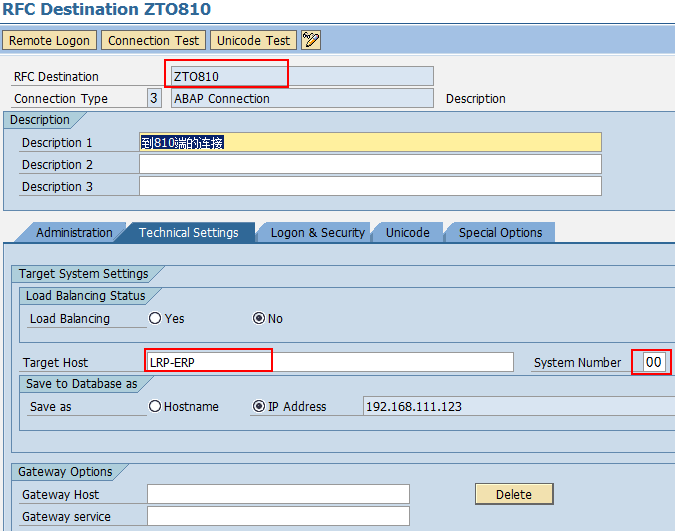
#### 4、关联Message Type和IDOC Type（WE82）

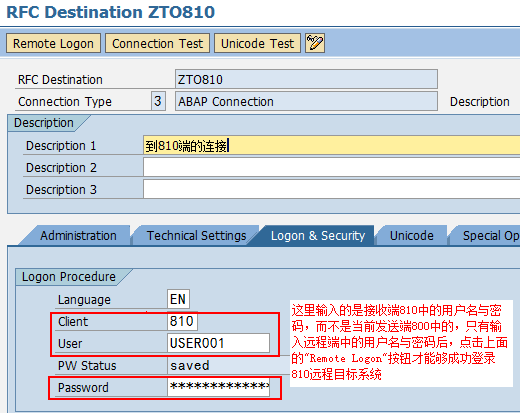


#### 5、创建接收端RFC Destination（SM59）

创建一个到接收端810的物理连接，由于是该实例是在同一个SAP系统内部进行实验，

所以“连接类型”选择的是ABAP Connection连接，名为ZTO810：





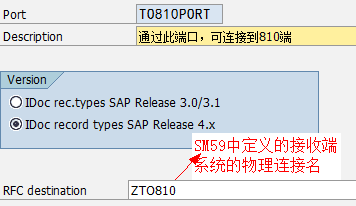
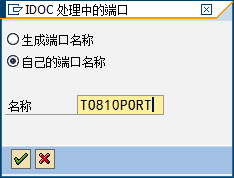
#### 6、创建到收端的端口（WE21）

注：这里讲的端口不是单纯的指定Socket端口号，而是指连接到RFC目标系统的统称，

包括IP、端口等信息，实质上是在SM59创建的物理连接基础之上创建的另一种逻辑名而已

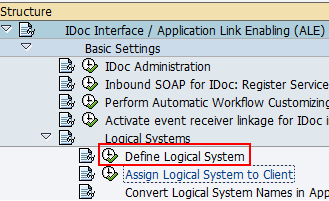
基于上面第5步创建的RFC Destionation，创建端口Port，类型选TRANSACTIONAL RFC，

名为TO810PORT，RFC destination则填写[ZTO810](file:///D:\Users\Administrator\Desktop\%E6%96%B0%E5%BB%BA%E6%96%87%E4%BB%B6%E5%A4%B9%20(2)\%23_第三步_维护RFC目标（SM59）)：



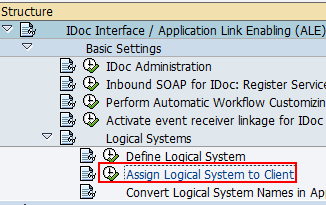
#### 7、创建发送端Logical System并分配（SALE）

为发送端800创建逻辑系统 Z800LS：





并将逻辑系统分配到发送端800：



注：这里不需在本端（Client 800）创建此逻辑系统的合作和伴配置文件，

但需要在810配置

#### 8、创建接收端Logical System（SALE）

为接收端810创建逻辑系统 Z810LS，这将在下一步创建到接收端的合作伙伴配置文件Partner profile（WE20）时用到：



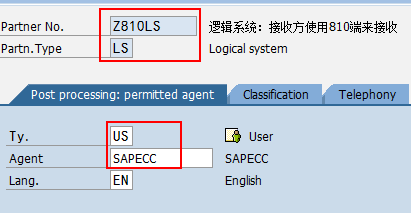
但与上面创建发送端逻辑系统不一样的是，在发送端系统800中是不需要将它分配给Client 810，

但此分配操作需要在接收端810进行，这一分配操作请参考后面接收端（Inbound）配置章节

#### 9、创建接收端合作和伴配置文件Partner profile（WE20）

合作和伴配置文件将Message Type消息类型、receiver port RFC目标端口、IDoc类型关联起来

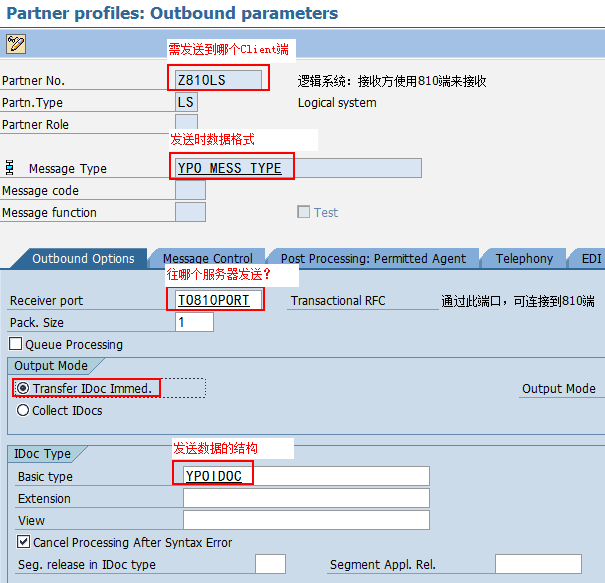
创建一个patner no为Z810LS的合作和伴配置文件，该配置文件描述了将IDoc发往何处：



上图中合作伙伴编号填上一步创建的接收端逻辑系统，类型选择LS。Ty.选择是“用户”类型，代理人为本系统（发送端800）中的用户，如果在Idoc发送过程中出现什么问题，会向此用户发送邮件，这里要注意的是：这不是目标系统（接收端810）中的用户

当上面信息填好后，点击保存（保存之后才可以对Outbound parmtrs进行设置）。

然后，点击outbound下方的加号，创建一个outbound parameter。Message Type为YPO\_MESS\_TYPE，receiver port为TO810PORT，output mode选择Transfer idoc immed.，Basic Type填写YPOIDOC，保存即可：



#### 10、通过Abap程序发送IDOC

**程序的思路就是，把每个IDOC节点按字符串形式逐个添加，而字符串的添加次序自然也体现了IDOC节点间的逻辑关系。代码如下：**

**DATA: ls\_pohead TYPE ypohead,"IDoc数据段：头**

**ls\_poitem TYPE ypoitem,"IDoc数据段：Item**

**ls\_edidc TYPE edidc,"IDoc的控制记录**

**lt\_edidc TYPE TABLE OF edidc,**

**lt\_edidd TYPE TABLE OF edidd WITH HEADER LINE."IDoc的数据记录**

**CLEAR ls\_edidc.**

**\*系统根据下面4行即可与WE20(合作和伴配置文件)设置关联起来**

**ls\_edidc-mestyp = 'YPO\_MESS\_TYPE'. "Message Type**

**ls\_edidc-idoctp = 'YPOIDOC'. "IDOC Type**

**ls\_edidc-rcvprn = 'Z810LS'. "partner Number of Recipient接收方合作伙伴**

**ls\_edidc-rcvprt = 'LS'.      "partner Type of Receiver接收方类型为逻辑系统**

**\*添加IDOC节点**

**CLEAR lt\_edidd.**

**lt\_edidd-segnam = 'YPOHEAD'."头节点**

**lt\_edidd-dtint2 = 0.**

**CLEAR ls\_pohead.**

**ls\_pohead-ebeln = '4001122334'."采购单号**

**ls\_pohead-bukrs = '1000'."公司代码**

**ls\_pohead-bedat = '20090630'."日期**

**lt\_edidd-sdata = ls\_pohead. "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 lt\_edidd.**

**CLEAR lt\_edidd.**

**lt\_edidd-segnam = 'YPOITEM'."Item节点**

**lt\_edidd-dtint2 = 0.**

**CLEAR ls\_poitem.**

**ls\_poitem-ebeln = '4001122334'."采购单号**

**ls\_poitem-ebelp = '0001'."Item行号**

**ls\_poitem-matnr = '000000000000004527'."物料号**

**ls\_poitem-menge = '3'."数量**

**ls\_poitem-meins = 'ST'."单位**

**lt\_edidd-sdata = ls\_poitem.**

**APPEND lt\_edidd.**

**CLEAR lt\_edidd.**

**lt\_edidd-segnam = 'YPOITEM'."Item节点**

**lt\_edidd-dtint2 = 0.**

**CLEAR ls\_poitem.**

**ls\_poitem-ebeln = '4001122334'."采购单号**

**ls\_poitem-ebelp = '0002'."Item行号**

**ls\_poitem-matnr = '000000000000009289'."物料号**

**ls\_poitem-menge = '5'."数量**

**ls\_poitem-meins = 'M'."单位**

**lt\_edidd-sdata = ls\_poitem.**

**APPEND lt\_edidd.**

**CALL FUNCTION 'MASTER\_IDOC\_DISTRIBUTE'"发送IDoc**

**EXPORTING**

**master\_idoc\_control            = ls\_edidc "IDoc控制记录**

**TABLES**

**communication\_idoc\_control     = lt\_edidc "接收：用来接收IDoc发送情况**

**master\_idoc\_data               = lt\_edidd "IDoc数据记录**

**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 lt\_edidc INTO ls\_edidc.**

**NEW-LINE.**

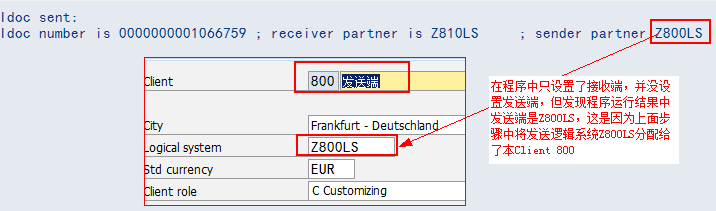
**WRITE: 'Idoc number is', ls\_edidc-docnum,**

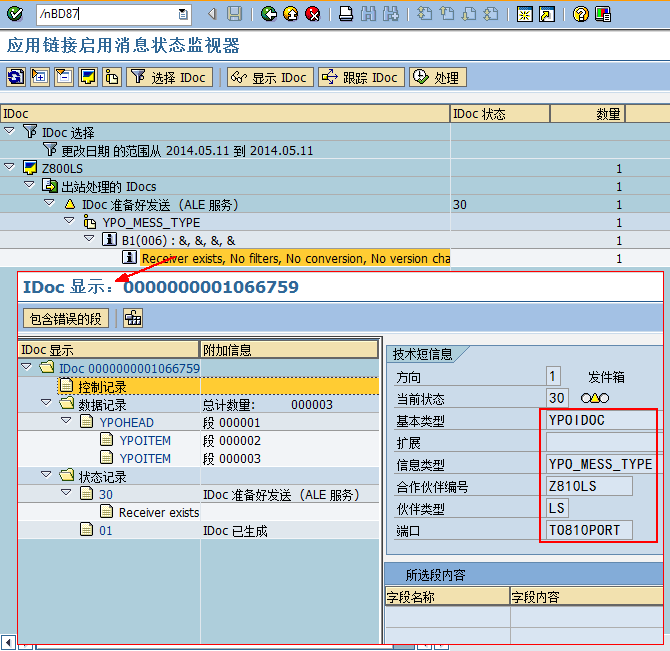
**'; receiver partner is', ls\_edidc-rcvprn,**

**'; sender partner',ls\_edidc-sndprn.**

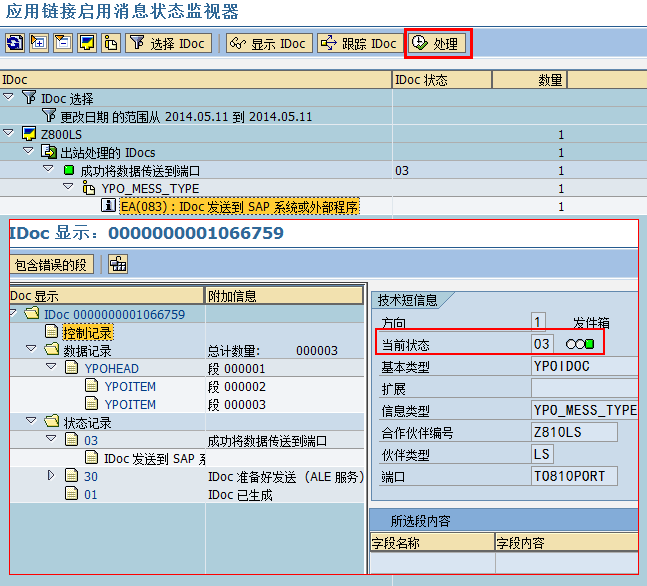
**ENDLOOP.**

**ENDIF.**

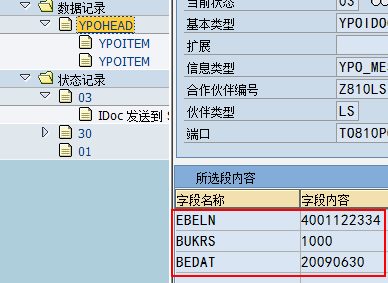




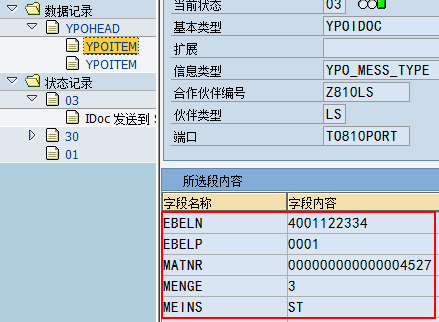
选中消息，并点击“处理”按钮后，该消息就会发往目的客户端810，状态从准备发送到成功发送：



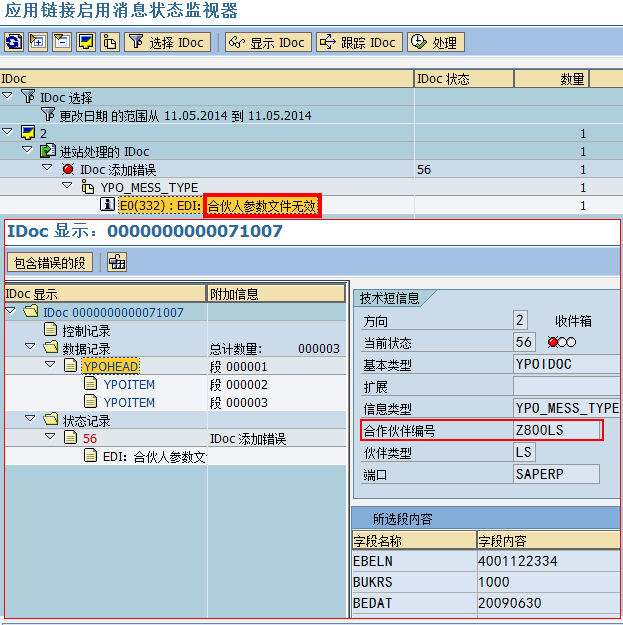
发送的数据可以选择数据记录节点来查看：







以上都是在发送端800进行的，下面登录到接收端**810**去看看IDoc接收情况：



由于在接收端810未配置到发送端800的合作伙伴配置文件，所以出错。

下面章节在接收端810中进行发送端800的相关配置

### 在接收端（Inbound）中配置

以下步骤都是在Client 810进行设置

由于该实例是在同一服务器的同一实例中进行的，又由于Segment、IDoc Type、Message Type这些都是跨Client的，所以上面1、2、3、4步就不需要在810端再次配置了，这些是共享的（但如果不是在同一服务器上，则需要像上面那样进行配置）

#### 1、创建发送端RFC Destination（SM59）

创建一个到发送端810的物理连接

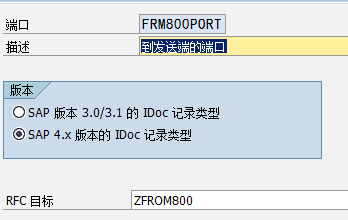




#### 2、创建发送端的端口（WE21）

基于上面创建的RFC Destination，创建端口Port，类型选TRANSACTIONAL RFC，

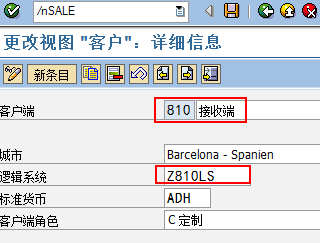
名为FRM800PORT，RFC destination则填写上面创建的RFC远程目标[ZFROM800](file:///D:\Users\Administrator\Desktop\%E6%96%B0%E5%BB%BA%E6%96%87%E4%BB%B6%E5%A4%B9%20(2)\%23_第三步_维护RFC目标（SM59）)：



#### 3、将接收端Logical System分配到Client 810（SALE）

如果是在不同的服务器中，则这里需要像在发送端那样：发送端逻辑系统Z800LS与接收端逻辑系统Z810LS都需要被创建，并且还需要将接收端逻辑系统Z810LS分配到Client 810，并且还需要以发送端逻辑系统Z800LS为基础创建发送端合作伙伴配置文件

由于是在同一服务器的同一实例中，所以在发送端中创建的发送端逻辑系统Z800LS与接收端逻辑系统Z810LS在此端是通用共享，这里不需要再次创建（outbound中已经创建了这两个逻辑系统了），但（outbound章节中并未分配到具体的Client）Z810LS逻辑系统没有分配到相应的Client，这一步操作需要在接收端完成，所以需要在此进行分配：



#### 4、创建入站处理函数

创建一个function：Y\_IDOC\_PO\_PROCESS.

当IDOC设置完毕之后，SAP可以自动调用该Funtion Module处理IDOC，所以这个函数的参数接口是有规范的，可以从IDOC\_INPUT\_BBP\_IV这些标准函数拷贝参数接口部分：

"ypohead\ypoitem实为上面定义的IDoc类型

DATA: ls\_chead TYPE ypohead,

      ls\_citem TYPE ypoitem.

  CLEAR idoc\_contrl.

  READ TABLE idoc\_contrl INDEX 1.

  IF idoc\_contrl-mestyp <> 'YPO\_MESS\_TYPE'.

    RAISE wrong\_function\_called.

  ENDIF.

  LOOP AT idoc\_contrl.

    LOOP AT idoc\_data WHERE docnum = idoc\_contrl-docnum.

      CASE idoc\_data-segnam.

        WHEN 'YPOHEAD'.

          "直接将字符赋值给结构，贬值过程中会按照结构中的字段长度来划分各字段

          ls\_chead = idoc\_data-sdata.

          WRITE: / 'Head',ls\_chead.

        WHEN 'YPOITEM'.

          ls\_citem = idoc\_data-sdata.

          WRITE: / 'Item',ls\_citem.

        WHEN OTHERS.

      ENDCASE.

    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.

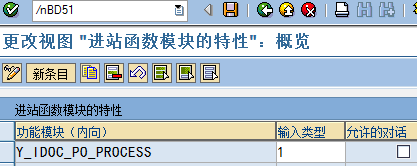
    ENDIF.

  ENDLOOP.

ENDFUNCTION.

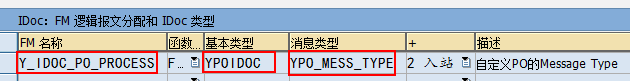
#### 5、注册入站处理函数（BD51）

填入函数名Y\_IDOC\_PO\_PROCESS，Input Type＝1



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

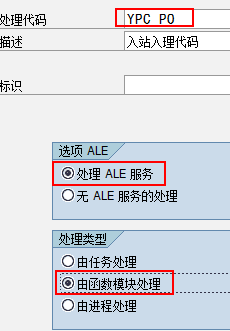
Function Module输入Y\_IDOC\_PO\_PROCESS，其下的Type填写F；IDOC Type下的Basic Type填写YPOIDOC；Message Type填写YPO\_MESS\_TYPE；Direction填写2(Inbound)



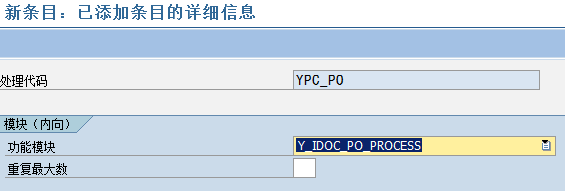
#### 7、创建Inbound Process Code（WE42）

Process Code输入YPC\_PO，在Option ALE下选择Processing with ALE service，

在Processing Type下选择function module：



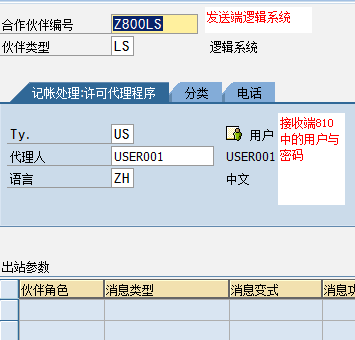
保存后，在随后的窗口中，输入Inbound Module为Y\_IDOC\_PO\_PROCESS。



#### 8、创建发送端合作和伴配置文件Partner profile（WE20）

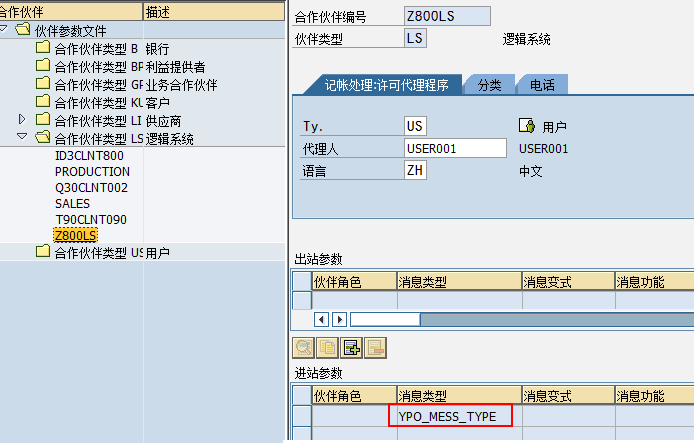
由于在发送端800中已创建了发送端逻辑系统Z800LS ，所以在此端不需要在创建（发送端逻辑系统Z800LS），

只是需要以此为基础创建和伴配置文件：



保存后，增加进站参数，点击cid:90cb0797501543d0636d037f5a63d52d.png:

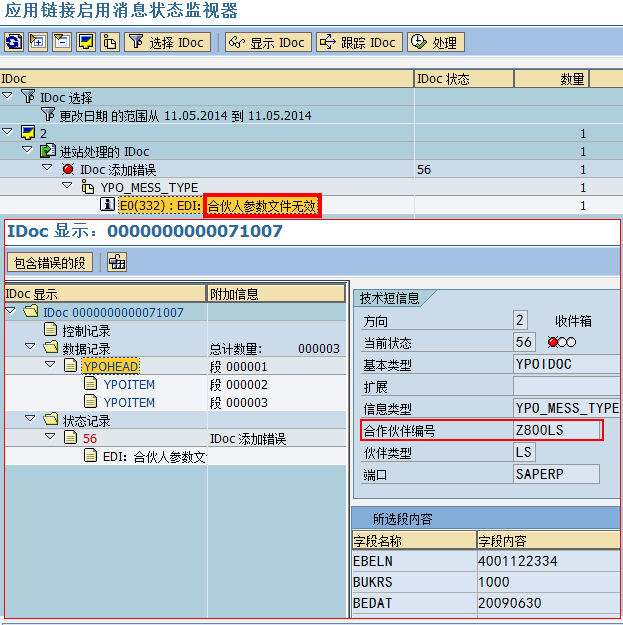




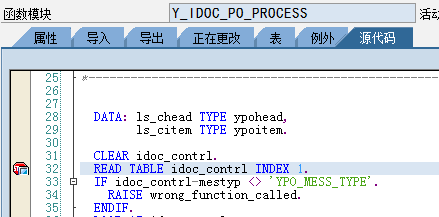
#### 9、测试 BD87

在接收端810使用BD87，登录进去后，会看到发送端发送过来的IDoc，但由于先前还没有配置发送伙伴配置文件，

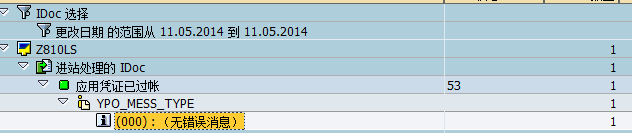
所以失败了：



现在在入站处理理函数中设置断点：



再执行BD87事务，继续处理出错的消息，最后发现处理成功：



调试程序时，发现数据也传递过来了：

