Step by Step procedure for creation of IDOC

Introduction
The main objective of an implementation of the SAP R/3 ERP is to group all the functions of the company together, in a single system. But it is very unlikely that this philosophy is applied so strictly.

Indeed, for such various reasons as load-balancing, task segmentation or also risk distribution, it is common to meet with a software landscape with more than one implementation of SAP R/3. It is even more common that these implementations of SAP R/3 run together with other heterogeneous systems, like AS/400 mainframes (IBM iSeries).

To communicate with each other, SAP has designed for R/3 systems its own communication tool: IDocs. These Intermediate Documents are the basis of every interface between R/3 systems. It is even possible, using a middleware EDI system, to have a R/3 system communicate by IDocs on its side, with an open system by XML files on the other side.

1 Structure of an IDoc interface
1.1 Structure of an IDoc
1.1.1 Segment
A segment is a record, defined as such in the vocabulary of databases. Indeed, as a line of a database, a segment is a sequence of fields of different length. There is no hierarchical structure within a segment, every fields are at the same level.

1.1.2 Section
An IDoc is made of three sections: control, data and status. Each section is named following the name of the one or many segments that composed it. Thus, the control section contains a single control segment only, the data section contains one or many data segments, the status section contains one or many status segments.

It is important to notice that when exchanging IDocs between systems, whether they are SAP R/3 or not, only control and data sections are sent. Indeed, the status section remains system specific. Nevertheless, the status section is conceptually associated with the IDoc, so it is systematically represented as being a part of the IDoc.

1.1.2.1 Control section
The particularity of the control section is that it contains only one single segment. This section represents the header of the IDoc, it contains an identifier of the IDoc, along with data concerning the sender system and the receiver system.

1.1.2.2 Data section
The particularity of the data section is that it contains one or many segments which are organized in a hierarchical way. There is a concept of parent segment and of child segment, a concept in which the child segment could not exist if the superior parent segment does not exist in the hierarchy.

This section is the most important, because as it is implied in its name, it is this section which contains the application data to transmit.

1.1.2.3 Status section
The particularity of the status section is that it is specific to the system where the IDoc is displayed. Indeed, after its transfer into an other system, the IDoc is actually rebuilt by copy, segment by segment. The status segments only are not transferred: there are specific to each one of the systems.

This status section is composed of one or many status segments. These segments are organized in a sequential way, so that only the last segment has a real importance. In consequence, what is refered as being the IDoc’s status is actually copied from the status mentioned in the last segment of the IDoc’s status section.

1.2 Structure of a segment
1.2.1 Control segment
The most important fields of the control segment are the following:

IDoc number: number of the IDoc in the local system
Direction: direction of the IDoc, from the point of view of the local system; 1 stands for outbound, 2 for inbound
Status: current status of the IDoc in the local system
Basic type: basic IDoc type
Extension: type of extension, if applicable
Message type: message type
Sender or Recipient information: details concerning the sender or the receiver
Port: port
Partner number: number of logical system
Partn.Type: partner type; most of the time LS which stands for Logical System
SAP Release: version number of the IDoc
Output Mode: output mode; 2 stands for immediate sending, 4 for collected sending

1.2.2 Data segment
The structure of a data segment depends on the segment type. Each segment type has a different number of fields, each field having a different length.

The only common characteristic between all the segment types is the format of the data. Indeed, every fields of every data segments have the character format, whatever the way the fields are represented in the IDoc display.

1.2.3 Status segment
The structure of a status segment is the following:

Status: the status reported by the segment
Message: the text describing the status
The other fields of the status segment are used to create a message following a SAP standard structure.

1.3 Structure of an interface
An interface is made of the following elements:

A view of the distribution model (optional)
No, one or many filters
A sending system
A receiving system
A message type
For outbound interfaces:
A port to emit to
An output mode
A packet size
A basic IDoc type
For inbound interfaces:
A process code

1.3.1 View of the distribution model
The creation of a view in the distribution model is optional. Indeed, it is strictly obligatory only in a case where the application of a filter is necessary.

After the creation of a view in the distribution model, it is possible to distribute this view to the partner system, as it is also possible to generate the partner profiles for this view.

1.3.2 Message type and basic IDoc type
A message type represents the group in which every basic IDoc types must belong. A message type is conceptually the nature of the data transmitted within an IDoc.

For example, the MATMAS message type is related to material master data, Material Master. To this MATMAS message type is associated several basic IDoc types: MATMAS01, MATMAS02, MATMAS03, MATMAS04 et MATMAS05. Each basic IDoc type is able to contain essential data on material master data. MATMAS01 is the first version, each subsequent version increments the sequence number (MATMAS02, etc.) and adds fields comparing with the previous version.

By convention, each basic IDoc type which follows the ascending numbering can not remove a field or change the applicative meaning of a field comparing with the previous version of the basic IDoc type. The standard message types and basic IDoc types in SAP R/3 respect this convention, and it is preferable to do so for all the new types to be created.

1.3.3 Partner profiles
Dealing with every interfaces which do not have any filter, only the partner profiles are effective. Indeed, the views of the distribution model are there only to simplify and conceptualize the partner profiles, without replacing them though.

The parameters in the partner profiles are displayed with a reference frame which is the local system. For exemple, the parameters LS / REMSYST / Outbound deal with the outbound interfaces with the REMSYST logical system.

These parameters are different whether they affect an outbound or an inbound interface.

For outbound interfaces:
A port to emit to
An output mode
A packet size
A basic IDoc type
For inbound interfaces:
A process code

2 Customizing of an outbound interface
This chapter approaches the creation of a new outbound interface entirely specific. It uses no one of the standard SAP R/3 structures.

2.1 Creation of a new message type
2.1.1 Creation of a new segment type
Launch the WE31 transaction (Development segments: Initial screen).

Fill the Segment type field with the value Z1VISTAPM then press F5 (Create).

Fill the Short Description field and add as much lines as wished fields in the segment. For each field, give it a name and a Data Element to refer to.

Save.

2.1.2 Creation of a new basic IDoc type
Launch WE30 transaction (Develop IDoc Types: Initial Screen).
Fill the Obj. name field with the value ZVISTAPM01 then press F5 (Create).

In the popup window which appears, choose Create new and fill the Description field.

Select the ZVISTAPM01 header line then choose the Create segment... button. In the popup window which appears, fill the fields like the following:

Segm.type: Z1VISTAPM
Mandatory seg.: checked
Minimum number: 1
Maximum number: 999999

Save.

2.1.3 Creation of a new message type
Launch the WE81 transaction (Display View "EDI: Logical Message Types": Overview).

Choose the Display -> Change button then the New Entries button.

In the Message type column, type the value ZVISTAPM then in the Short text column type a short description.

Save.

2.1.4 Association of the basic IDoc type to the message type
Launch the WE82 transaction (Display View "Output Types and Assignment to IDoc Types": Overview).

Choose the Display -> Change button then the New Entries button.

In the Message Type column, type the value ZVISTAPM. In the Basic Type column, type the value ZVISTAPM01. In the Release column, type the value 620.

2.2 Customizing of a new interface
2.2.1 Creation of a new distribution model view
Launch the BD64 transaction (Display Distribution Model).

Press the Switch between display and edit mode button, then the Create model view button.

In the popup window which appears, enter in the Short text field a short description and in the Technical name field the value YVISTAPM.

Place on the newly created entry then press the Add Message Type button.

In the popup window which appears, enter in the Sender field the name of the local logical system, in the Receiver field the name of the receiving logical system and in the Message type field the name of the message type of the interface.

Validate then save.

2.2.2 Distribute the distribution model view
This step is necessary only if the partner system is a SAP R/3 system. It allows to centralize the changes to make on the interface. Be careful!
The partner profiles also have to be generated and managed in the partner system.

Select the recently created view then choose in the menu Edit / Model view / Distribute. In the popup window which appears, the partner system is already selected, there is no need to make other further selection. Validate.

2.2.3 Generate and manage the partner profiles
From the distribution model view (BD64 transaction), place on the message type then choose in the menu Environment / Generate partner profiles, or launch the BD82 transaction (Generating partner profile).

Enter in the Model view field the value YVISTAPM, in the Partner system field the value TIBCO, and choose as Output mode the Collect IDocs and transfer radio button. Press the Execute button.

Check the partner profiles. To do so, launch the WE20 transaction (Partner profiles). Select LS / TIBCO / Outbound parmtrs. / ZVISTAPM.

The port should be checked first, because it is not specified at the generation of the partner profiles level, so it is the first port which is selected by default.

If needed, manage the partner profiles then save.

2.3 Creation of an outbound interface program
2.3.1 Creation of an IDoc generation program
The following code extract contains everything needed to generate an IDoc from data contained in a table.

```
FORM F_110_SEND_IDOC.
CONSTANTS: C_MESTYP TYPE EDIDC-MESTYP VALUE 'ZVISTAPM', C_DOCTYP TYPE EDIDC-IDOCTP VALUE 'ZVISTAPM01',
C_SEGNAM TYPE EDIDD-SEGNAM VALUE 'Z1VISTAPM'.

DATA: I_ZVISTA_PM TYPE ZVISTA_PM_T OCCURS 6000, I_EDIDC TYPE EDIDC OCCURS 0, I_EDIDD TYPE EDIDD OCCURS 0,
WA_ZVISTA_PM TYPE ZVISTA_PM_T, WA_EDIDC TYPE EDIDC, WA_EDIDD TYPE EDIDD, WA_Z1VISTAPM TYPE Z1VISTAPM,
```
Retrieve the maximum number of segments in the basic IDoc type:

```
SELECT MIN(OCCMAX) FROM IDOCSYN INTO V_OCCMAX
WHERE IDOCTYP EQ C_DOCTYP AND SEGTYP EQ C_SEGNAM.
```

Create a data segment for each line of I_ZVISTA_PM:

```
LOOP AT I_ZVISTA_PM INTO WA_ZVISTA_PM.
    MOVE CORRESPONDING WA_ZVISTA_PM TO WA_Z1VISTAPM.
    CLEAR WA_EDIDD.
    MOVE C_SEGNAM TO WA_EDIDD-SEGNAM.
    MOVE WA_Z1VISTAPM TO WA_EDIDD-SDATA.
    APPEND WA_EDIDD TO I_EDIDD.
    CLEAR WA_ZVISTA_PM.
    ENDLOOP.
```

Count the number of data segments:

```
DESCRIBE TABLE I_EDIDD LINES V_NBSEG.
```

If the number of data segments exceeds the maximum allowed number, then edit a message in the spool, then display an error message (quit the program):

```
IF V_NBSEG GT V_OCCMAX.
    WRITE:/ TEXT-003, V_OCCMAX.
    MESSAGE E751.
ENDIF.
```

Call the IDoc creation function:

```
CALL FUNCTION 'MASTER_IDOC_DISTRIBUTE' EXPORTING
    MASTER_IDOC_CONTROL = WA_EDIDC TABLES
    COMMUNICATION_IDOC_CONTROL = I_EDIDC
    MASTER_IDOC_DATA = I_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 there was an error, display a message (quit the program):

```
IF SY-SUBRC NE 0.
    MESSAGE E746.
ENDIF.
```

ENDFORM.

### 2.3.2 Running of the IDoc sending program

The interface having been customized for a collected mode output, the program created IDoc is not sent immediately to the receiving system. The IDoc stays waiting for processing. It can be viewed in the BD87 transaction (Status Monitor for ALE Messages).

It is possible to press the Process button to send the IDoc which waits for processing. However, if there were several IDocs to be sent in collected mode, pressing the Process button would have sent them one by one instead of sending them in a batch.

Actually, the RSEOUT00 program has to be executed with adequate parameters: Logical message set to the value ZVISTAPM and Output mode set to the value 4.

### 3 Additional customizings

#### 3.1 Filter

##### 3.1.1 Principle

A filter can be put in place on an interface. A filter is always placed at the distribution level (BD64 transaction), on a particular message type.

A filter is a series of values for one or many fields. When this filter is applied on a message, the segments are created only if the filtered fields contain those values.

**Example:**

- **Filter 1**: field MATNR = "123", "456"; field ATWRT = "FOO1".
- Basic IDoc type ZIDOC containing a single segment Z1SEG.
- The segment (MATNR = "123"; ATWRT = "FOO1"; FIELD1 = "VALUE") will be created.
- The segment (MATNR = "123"; ATWRT = "VAL2"; FIELD1 = "VALUE") will NOT be created.

So a filter acts the following way: for an IDoc segment to be created, each filtered field has to contain one of the set values. It is a logical "OR" between each value of a field and a logical "AND" between each filtered field.

Be careful! If a segment has a field which do not pass the filter, it is deleted. Then, every segments of a lower hierarchical level are deleted too, as the higher hierarchical level segment also if the deleted segment has been set to be mandatory in the basic IDoc type.

A filter is often used to reduce the volume of sent data, because they are not all relevant for the interface to be built.

##### 3.1.2 Customizing

##### 3.1.2.1 Creation of an ALE object

First of all, an "ALE object" has to be created to be able to filter on it. An object which type will be identical to the Data Element of the field to be filtered has to be created.

Launch the BD95 transaction (Change View "ALE Object Type": Overview).

In the ALE Object Type field, type the value MATNR. In the Table name field, type the value MAKT. In the Field name field, type the value MATNR. A table field which the Data Element is the same like the field to be filtered should be chosen.

Validate then save.

##### 3.1.2.2 Association of an ALE object type to a message type

Launch the BD59 transaction (Change View "Assignment of Object Type to Message": Overview).

In the popup window which appears, in the Message type field, type the message that should be filtered, ZVISTAPM here, then validate.

Press the New Entries button. In the ALE Object Type field, type the value MATNR. In the Segm.type field, type the value Z1VISTAPM. In the No. field, type the value 1. In the Field type, type the value MATNR.

Validate and save.

##### 3.1.2.3 Creation of the filter in the distribution model
Launch the BD64 transaction (Display Distribution Model).

Press the Switch between display and edit mode button, develop the previously created YVISTAPM view and double-clic on the No filter set line.

In the popup window which appears, press the Create filter group button. Develop Data filtering then double-clic on Material.

In the popup window which appears, press the Insert row button and type a value for this field. Do it again for all the values to let pass for this field.

Validate twice then save. Here is your new filter active.

3.2 IDoc reduction

3.2.1 Principle
Standard basic IDoc types are often adapted to all imaginable interfaces. However, it sometimes happens that the sent volume is too big compared with the needs. A filter can applied (see above), which represents a dynamical method.

The number of segments or the number of fields in a segment can also be reduced in a static way, while still keeping the structure of the basic IDoc type. Indeed, for all basic IDocs, SAP had envisaged their reduction.

A reduced IDoc has less segments than the basic IDoc, provided that the deleted segments are not mandatory. The segments of a reduced IDoc can also have less fields than the corresponding segments of the basic IDoc.

Dealing with the standard basic IDoc types, their integration function module already takes into account the possibility of a reduction. The advantage to use reduced IDocs seems to be obvious then: to profit from standard functions while reducing the volume of exchanged data.

3.2.2 Customizing
Launch the BDS3 transaction (IDoc Reduction Maintenance: Initial Screen).

In the Reduced message type field, type the value ZVISTAPM_REDUCED, then press the Create button.

In the popup window which appears, type in the Message type reference field the value MATMAS.

Validate. In the next popup window, type a short description for the new reduced message type. Validate.

In the following window, select the segments and the fields needed for the new reduced message type.

Save.

3.3 Integration of an inbound IDoc

3.3.1 Development
Below is the prototype of the function module to develop to integrate an inbound IDoc.

```
**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_VARIABLE) 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 ERROR_MESSAGE
```

3.3.2 Customizing
Below is the list of transactions to launch in order to associate the integration function module with the corresponding message type.

BD51 (Display View "Characteristics of Inbound Function Modules": Overview)
This transaction is used to register a function module as being able to be used for the integration of an inbound IDoc.

WE57 (Display View "Idoc: Assignment of FM to Log. Message and IDoc Type": Overview)
This transaction is used to associate the previously declared function module to the message type to be integrated.

WE42 (Display View "Inbound process code": Overview)
This transaction is used to declare the process codes for integration. These are those codes which will be specified at the partner profiles level for inbound interfaces.

Appendices

4.1 Useful transactions for IDocs
BD87 : Status Monitor for ALE Messages
SALE : Display ALE Customizing
WE02 : Display IDoc
WE05 : IDoc Lists
WE09 : Search for IDoc in Database
WE19 : Test tool

4.2 Standard statuses of IDoc
4.2.1 Outbound IDocs statuses
Statut Description
0 Not used, only R/2
1 IDoc generated
2 Error passing data to port
3 Data passed to port OK
4 Error within control information of EDI subsystem
5 Error during translation
6 Translation OK
7 Error during syntax check
8 Syntax check OK
9 Error during interchange handling
10 Interchange handling OK
11 Error during dispatch
12 Dispatch OK
13 Retransmission OK
14 Interchange Acknowledgement positive
15 Interchange Acknowledgement negative
16 Functional Acknowledgement positive
17 Functional Acknowledgement negative
18 Triggering EDI subsystem OK
19 Data transfer for test OK
20 Error triggering EDI subsystem
21 Error passing data for test
22 Dispatch OK, acknowledgement still due
23 Error during retransmission
24 Control information of EDI subsystem OK
25 Processing despite syntax error (outbound)
26 Error during syntax check of IDoc (outbound)
27 Error in dispatch level (ALE service)
28 Not used
29 Error in ALE service
30 IDoc ready for dispatch (ALE service)
31 Error - no further processing
32 IDoc was edited
33 Original of an IDoc which was edited
34 Error in control record of IDoc
35 IDoc reloaded from archive
36 Electronic signature not performed (timeout)
37 IDoc added incorrectly
38 IDoc archived
39 IDoc is in the target system (ALE service)
40 Application document not created in target system
41 Application document created in target system
42 IDoc was created by test transaction

4.2.2 Inbound IDocs statuses
Statut Description
50 IDoc added
51 Application document not posted
52 Application document not fully posted
53 Application document posted
54 Error during formal application check
55 Formal application check OK
56 IDoc with errors added
57 Test IDoc: Error during application check
58 IDoc copy from R/2 connection
59 Not used
60 Error during syntax check of IDoc (inbound)
61 Processing despite syntax error (inbound)
62 IDoc passed to application
63 Error passing IDoc to application
64 IDoc ready to be transferred to application
65 Error in ALE service
66 IDoc is waiting for predecessor IDoc (serialization)
67 Not used
68 Error - no further processing
69 IDoc was edited
70 Original of an IDoc which was edited
71 IDoc reloaded from archive
72 Not used, only R/2
73 IDoc archived
74 IDoc was created by test transaction

4.3 List of standard basic IDoc types
The skills of a good IDoc interfaces manager are, for the most, linked to his good knowledge of standard basic IDoc types. This knowledge allows him or her to put in place interfaces very quickly, without having to develop a new specific type.
Pricing Procedure (only in 40c)
Condition Exclusion Groups
Condition type
Conditions: Exclusion indicator:
ABSEN1 Attendance/Absence in CC1
ACCONF01 Confirmation of IDoc processing from the application
ACC_ACT_ALLOC01 Accounting: Post activity allocation
ACC_ACT_ALLOC02 Accounting: Post activity allocation
ACC_ACT_ALLOC03 Accounting: Post Activity Allocation
ACC.Asset TRANSFER01 Accounting: Post Acquisition from Transfer
ACC.Asset TRANS.ACC.POST01 Accounting: Post Acquisition from Transfer
ACC.BILLING01 Accounting: Post Billing Document (OAG: LOAD RECEIVABLE)
ACC.BILLING02 Accounting: Post Billing Document (OAG: LOAD RECEIVABLE)
ACC.BILLING.Reverse01 Accounting: Post Billing Doc.Reversal (OAG: LOAD RECEIVABLE)
ACC.DOCUMENT01 Accounting: Posting
ACC.DOCUMENT02 Accounting: Posting
ACC.DOCUMENT.Reverse01 Accounting: Post Reversal
ACC.EMPLOYEE.EXP01 FI/CO: HR Posting GL(AcctngEmployeeExpnses)
ACC.EMPLOYEE.EXP02 FI/CO: HR posting GL(AcctngEmployeeExpnses)
ACC.EMPLOYEE_PAY01 FI/CO: HR Posting AP(AcctngEmployeePaybles)
ACC.EMPLOYEE_PAY02 FI/CO: HR posting AP(AcctngEmployeePaybles)
ACC.EMPLOYEE_REC01 FI/CO: HR Posting AR(AcctngEmployeeRcvbles)
ACC.EMPLOYEE_REC02 FI/CO: HR posting AR(AcctngEmployeeRcvbles)
ACC.GL_POSTING01 Accounting: General G/L Account Posting
ACC.GL_POSTING.Reverse01 Accounting: Post General G/L Posting Reversal
ACC.GOODS_MOVEMENT01 Accounting: Post Goods Movement (OAG: POST JOURNAL)
ACC.GOODS_MOVEMENT02 Accounting: Post Goods Movement (OAG: POST JOURNAL)
ACC.GOODS_MOV.Reverse01 Accounting: Post Goods Movement Reversal (Acctng Goods Mvt)
ACC.INVOICE.PYMNTBLK01 Accounting: Change TS posting (AcctngInvoiceReceipt)
ACC.INVOICE.RECEIPT01 Accounting: Post Invoice Receipt (OAG: LOAD PAYABLE)
ACC.INVOICE.RECEIPT02 Accounting: Post Invoice Receipt (OAG: LOAD PAYABLE)
ACC.INVOICE.RECEIPT03 Accounting: Post Invoice Receipt (OAG: LOAD PAYABLE)
ACC.INVOICE.Reverse01 Accounting: Post Invoice Reversal (Acctng Invoice Receipt)
ACC.MAN_ALLOC01 Accounting: Post Manual Cost Allocation
ACC.PRIM_COSTS01 Accounting: Post primary costs
ACC.PRIM_COSTS02 Accounting: Post Primary Costs
ACC.PURCHASE.ORD01 Accounting: Post Purchase Order
ACC.PURCHASE.REQU01 Accounting: Post Purchase Requisition
ACC.REVENUES01 Accounting: Post revenues
ACC.REVENUES02 Accounting: Post Revenues
ACC.SALES.ORD01 Accounting: Post Sales Order
ACC.SALES.QUOTA01 Accounting: Post Customer Quotation
ACC.SENDER.ACTIVITIES01 Accounting: Post Sender Activities
ACC.STAT.KEY.FIG01 Accounting: Post statistical key figures
ACC.TRAVEL01 Accounting: Post trip
ACLPAY01 Posting in Accounting: Incoming Invoice
ACLREC01 Posting in accounting: Billing document
ACPJOU01 Posting in Accounting from Materials Management
ACTIV3 Units in KK3
ACTIV4 Units in KK4
ACTIVITYTYPEGROUP_ADDNODE01 Object BUS1115 (Activity Type Group) - Method AddNode
ACTIVITYTYPEGROUP_CREATE01 Object BUS1115 (Activity Type Group) - Method Create
ADR2MAS01 BAPI for inbound distribution of private addresses
ADR2MAS02 BAPI for inbound distribution of private addresses
ADR3MAS01 BAPI for inbound distribution of contact person addresses
ADR3MAS02 BAPI for inbound distribution of contact person addresses
ADRMAS01 BAPI for inbound distribution of organizational addresses
ADRMAS02 BAPI for inbound distribution of organizational addresses
ALEAUD01 Confirmations of the processing status of inbound IDocs
ALEREO01 General request - Basis IDoc type
ARTMAS01 Create and Change Material Master Data (Retail)
ARTMAS02 Create and Change Material Master Data (Retail)
ARTMAS03 Create and Change Material Master Data (Retail)
ARTMAS04 Create and change material master data (retail)
ASSET.ACQUISITION.POST01 Post Asset Acquisition
ASSET.POSTCAP.POST01 Post post-capitalization
ASSET.RETIREMENT.POST01 Post asset retirement
ASSMOD01 Assortment (manual modules)
ASSORTMENT01 Maintenance of assortments
ATT.Abs01 BAPI ext. Att.absence: Insert in IFT (w/o acct assgnmnt)
ATT.Abs.WITH.ACTIVITY01 Ext. Att.absence (PTManagerExtAttAbs)
ATT.Abs.WITH.COST01 BAPI ext. Att.absence: Insert in IFT (acct assgnmnt)
BANK.Change01 Change bank
BANK.CREATE01 Create bank
BANK.SAVE.REPLICA01 Duplicate individual banks (ALE)
CONF32 Confirmations in KK3, wage slips
CONF41 Confirmations in KK4, time events
CONF42 Confirmations in KK4, wage slips
CONF51 CC5 PDC Interface KANBAN
CONF_ACT1001 Enter activity confirmation
COPAGN01 CO-PA entry
COPA_DERIVE_ADD_VALUES01 Derivation Rule: AddValues
COPA_DERIVE_REMOVE01 Derivation Rule: RemoveValues
COPA_WW_ADD_VALUES01 Individual Characteristic: AddValues
COPA_WW_GET_VALUES01 Characteristic: GetValues
COPA_WW_REMOVE_VALUES01 Individual Characteristic: RemoveValues
COPCPA01 Transfer product costing CO-PC -> CO-PA
COPCPA02 Copy product cost estimate
COSCRO01 Core master cost center
COSMAS01 Master cost center
COSTCENTERGROUP_ADDNODE01 Object BUS1112 (Cost Center Group) - Method AddNode
COSTCENTERGROUP_CREATE01 Object BUS1112 (Cost Center Group) - Method Create
COSTELEMENTGROUP_ADDNODE01 Object BUS1113 (Cost Element Group) - Method AddNode
COSTELEMENTGROUP_CREATE01 Object BUS1113 (Cost Element Group) - Method Create
COTOTL01 IDoc for CO totals records
CO_ABC_BPROC_CREATE01 Create one or more business processes
CO_ABC_BPROC_SETSTRUC01 Enter template in business process master data
CRECOR01 Vendor master data distribution ALE Core master data
CREMAS01 Vendor master data distribution ALE
CREMAS02 Vendor master data distribution ALE
CREMAS03 Vendor master data distribution
CREMAS04 Vendor master data distribution
CREMDM01 MDM: Mass Processing CREMAS (Vendor Master)
CREMDM02 MDM: Mass Processing CREMAS (Vendor Master + Addresses)
CREMDM03 MDM: Mass Processing CREMAS (Vendor Master + Addresses)
CREMDM04 MDM: Mass Processing CREMAS (Vendor Master + Addresses)
CRESTA01 Copy credit status (DebtorCreditAccount)
CUSTOMERRETURN_CONFIRMDLIVE01 Delivery Confirmation (CustomerReturn)
CUSTSCHEDULEAGREEMENT_CONFDEL01 Delivery Confirm. (CustScheduleAgreement)
CUSTSCHEDULEAGREEMENT_CONFIRM01 Delivery Confirmation (CustScheduleAgreement)
DANGEROUSGOOD01 Dangerous goods: Save replicated instances
DANGEROUSGOOD02 Dangerous goods: Save replicated instances
DD01L_01 Domains
DD03L_01 Table Fields
DEBCOR01 Core master - customer
DEBMAS01 Customer master
DEBMAS02 Customer master
DEBMAS03 Customer master
DEBMAS04 Customer master data distribution
DEBMAS05 Customer Master Data Distribution
DEBMAS06 Customer master data distribution
DEBMDM01 MDM: Mass-enabled DEBMAS (Customer Master)
DEBMDM02 MDM: Mass-enabled DEBMAS (Customer Master)
DEBMDM03 MDM: Mass Processing DEBMAS (Customer Master + Addresses)
DEBMDM04 MDM: Mass Processing DEBMAS (Customer Master + Addresses)
DEBMDM05 MDM: Mass Processing DEBMAS (Customer Master + Addresses)
DEBMDM06 MDM: Mass Processing DEBMAS (Customer Master + Addresses)
DELFOR01 Delivery schedule/JIT schedule
DELFOR02 Delivery Schedule/JIT Schedule
DELIVERYPROCESSING_EXECUTE01 Delivery processing
DELVRY01 Delivery interface
DELVRY02 Delivery interface with dangerous goods data
DELVRY03 Delivery interface
DEPNET Master data of dependency net
DEPNET02 Master Data of a Dependency Net + E1UPSLINK
DESADV01 External service agent processing (notifications)
DES_ID01 Shipping notification
DIFFE2 Differences in KK2
DIFFE3 Differences in KK3
DIFFE4 Differences in KK4
DISPUTE_ATTRIBUTESCHANGE01 FSCM-DM: Change Dispute Case Attributes
DISPUTE_CREATE01 FSCM-DM: Creation of Dispute Case
DISPUTE_PROCESS01 FSCM-DM: Process Dispute Case
DISPUTE_STATUSCHANGED01 FSCM-DM: Change Status of Dispute Case
DISPUTE_WRITEOFF01 FSCM-DM: Dispute Cases to be Written Off Automatically
DISTU2 Reasons for problems KK2
DOCMAS01 Master document
DOCMAS02 Document 02
DOCMAS03 Document
FTR_CONDITION_DELETE01 RFC for Method Condition.Delete
FTR_CONDITION_GETLIST01 RFC for Method Condition.GetList
FTR_COUNTERCONFIRM01 Counterconfirm financial transaction
FTR_CP_CHANGE01 Change Commercial Paper
FTR_CP_CREATE01 Create Commercial Paper
FTR_CP_REVERSE01 Reverse Fixed-Term Deposit per BAPI
FTR_DAN_CHANGE01 Change fixed-term deposit per BAPI
FTR_DAN_CREATE01 Create fixed-term deposit
FTR_DAN_GETDETAIL01 Details of a fixed-term deposit
FTR_DAN_GIVENOTICE01 Roll Over Fixed-Term Deposit per BAPI
FTR_DAN_MAINFLOWCHANGE01 FTR: RFC for Method 'FinancialTransaction.MainFlowCreate'
FTR_DAN_MAINFLOWCREATE01 FTR: RFC for Method 'Paymentdetail.Create'
FTR_DAN_MAINFLOWDELETE01 FTR: RFC for Method 'Paymentdetail.Delete'
FTR_DAN_MAINFLOWGETLIST01 FTR: RFC for Method 'Paymentdetail.Create'
FTR_DAN_REVERSE01 Reverse Fixed-Term Deposit per BAPI
FTR_DAN_ROLLOVER01 Roll Over Fixed-Term Deposit per BAPI
FTR_FTD_CHANGE01 Change fixed-term deposit per BAPI
FTR_FTD_CREATE01 Create fixed-term deposit
FTR_FTD_MAINFLOWCHANGE01 FTR: RFC for Method 'FinancialTransaction.MainFlowCreate'
FTR_FTD_MAINFLOWCREATE01 FTR: RFC for Method 'Paymentdetail.Create'
FTR_FTD_MAINFLOWDELETE01 FTR: RFC for Method 'Paymentdetail.Delete'
FTR_FTD_MAINFLOWGETLIST01 FTR: RFC for Method 'Paymentdetail.Create'
FTR_FTD_REVERSE01 Reverse Fixed-Term Deposit per BAPI
FTR_FTD_ROLLOVER01 Roll Over Fixed-Term Deposit per BAPI
FTR_FXOPTION_CHANGE01 Change an FX Option
FTR_FXOPTION_CREATE01 Create Currency Option
FTR_FXOPTION_EXERCISE01 Exercise FX Option
FTR_FXOPTION_EXPIRE01 Allow FX Option to Expire
FTR_FXOPTION_GETDETAIL01 Details on FX Options
FTR_FXOPTION_KNOCKIN01 Knockin FXOPTION
FTR_FXOPTION_KNOCKOUT01 Knockout FXOPTIONS
FTR_FXOPTION_REVERSE01 Reverse FX Option
FTR_FXT.Change01 Change forex transaction
FTR_FXT.CREATE01 Create forex transaction
FTR_FXT.REVERSE01 Reverse forex transaction
FTR_PAYDET_CHANGE01 FTR: RFC for Method 'Paymentdetail.Create'
FTR_PAYDET_CREATE01 Create Payment Details
FTR_PAYDET_DELETE01 FTR: RFC for Method 'Paymentdetail.Delete'
FTR_PAYDET_GETLIST01 FTR: RFC for Method 'Paymentdetail.Create'
FTR_IRATE.Change01 Change interest rate instrument
FTR_IRATE.CREATE01 Create interest rate instrument
FTR_IRATE.DELETE01 FTR: RFC for Method 'Paymentdetail.Delete'
FTR_IRATE.GETDETAIL01 FTR: RFC for Method 'Paymentdetail.Create'
FTR_IRATE_MAINFLOWCHANGE01 FTR: RFC for Method 'FinancialTransaction.MainFlowCreate'
FTR_IRATE_MAINFLOWCREATE01 FTR: RFC for Method 'Paymentdetail.Create'
FTR_IRATE_MAINFLOWDELETE01 FTR: RFC for Method 'Paymentdetail.Delete'
FTR_IRATE_MAINFLOWGETLIST01 FTR: RFC for Method 'Paymentdetail.Create'
FTR_IRATE_REVERSE01 Reverse interest rate instrument
FTR_PAYDET.Change01 Change Payment Details
FTR_PAYDET.DELETE01 FTR: RFC for Method 'Paymentdetail.Delete'
FTR_PAYDET.GETLIST01 FTR: RFC for Method 'Paymentdetail.Create'
FTR_SECURITY.Change01 Change security transaction
FTR_SECURITY.CREATE01 Create security transaction
FTR_SECURITY.GETDETAIL01 Security transaction details
FTR_SECURITY.REVERSE01 Reverse Security Transaction
FTR_SECURITY_SETTLE01 Settle Security Transaction
FTR_SETTLE01 Settle financial transaction
FUNC_LOC.Change01 PM BAPI: Change Functional Location
FUNC_LOC.CREATE01 PM BAPI: Create Functional Location
FUNDMA01 IDoc: Funds Management Fund
FUNDCTR FundsManagement: Funds Center
GLCORE01 Documentation deleted
GLDCMT01 IDoc type for GLX rollups
GLMAST01 Master data G/L accounts: Max. IDoc
GOODSMVT SAPCREATE01 SAP Internal Cross-System Flow of Goods
GSVERF01 IDoc input credit memo procedure
GSVERF02 Cred. memo procedure
GSVERF03 Cred. memo procedure
HRCC1DNATTAB01 CC1: Download permitted att./absence reasons
HRCC1DNBALAN01 CC1: Download Employee Balances
MBGMCR01 BAPI Posting of goods mvts with MB_CREATE_GOODS_MOVEMENT
MBGMCR02 Post goods movements with MB_CREATE_GOODS_MOVEMENT
MC1_01 Condition tables
MDFDB01 MDM: Feedback Idoc (Object status)
MDMMETA MDM Metadata
MDMRECEIPT01 MDM Receipt
MDMREQUEST01 MDM Request
MDMVERA MDM Verification SFLIGHT
MDM_EQUIPMENT_SAVEREPLICA01 BAPI for Mass Maintenance of Equipment
MDM_FUNCLOC_SAVEREPLICA01
MMADDI01 Create and change additional (retail)
MRESCR01 Create reservation
OBDLV_PROCOUTP01 Process delivery output without list
OPERA2 Operations in KK2
OPERA3 Processes in KK3
OPERA4 Operations in KK4
OPERS3 Operation status in KK3
OPERS4 Operation status in KK4
ORDERS01 Purchasing/Sales
ORDERS02 Purchasing/Sales
ORDERS03 Purchasing/Sales
ORDERS04 Purchasing/Sales
ORDERS05 Purchasing/Sales
ORD_ID01 Req.for quot./quotation/purchase order/order change
ORGMASTER01 BAPI for replication of HR organizational data
OSTAT2 Process status CC2
PALMAT01 Plant assignment to material BOM
PAYMENTREQUEST_CANCEL01 Cancelation of a Payment Request
PAYMENTREQUEST_CREATE01 Creation of a Payment Request
PAYMENTREQUEST_POST01 Posting a Parked Payment Request
PAYMENTREQUEST_RELEASE01 Payment Request Released for Payment
PAYMENTREQUEST_STARTPAYMENT01 Start Payment of Payment Request
PCHEAD01 Replicate basic and structure data of a product catalog
PCITEMS01 Replicate item data of a product catalog
PDTCO_POSTPRIMARY01 Transfer of Planning Data: Post Primary Costs
PEROP2
PERSO1 Personnel master records in CC1
PERSO2 Personnel master in CC2
PERSO3 Personnel master in CC3
PERSO4 Personnel master in CC4
PEXR2001 Payment/payment advice note/credit memo/debit advice
PEXR2002 Payment/payment advice note/credit memo/debit advice
PHRMAS01 EHS: Saving Replicated Phrases
PIECEOEQUIPMENT_DISMANTLEAT01 Dismantle equipment at functional location
PIECEOEQUIPMENT_INSTALLATFU01 Install equipment at functional location
PKHD5 CC5 kanban control cycles
PKPS5 CC5 Kanban container
PKST5 CC5 status for kanban container
PLANDATATRANSFERCO_POSTACTIO01 Planning Data Transfer: Post Activity Planning
PLANDATATRANSFERCO_POSTACTIV01 Planning Data Transfer: Post Activity Input
PLANDATATRANSFERCO_POSTKEYFI01 Transfer of Planning Data: Post Stat. Key Figures
PLANT3 Plants in CC3
PLANT4 Plants in CC4
PORDCH01 Change purchase order
PORDCR01 Create purchase order
PORDCR02 Create purchase order
PORDCR03 Create purchase order
PORDCR04 Create purchase order
PORDCR05 Create Purchase Order
PORDCR101 Create Purchase Order
PORDDG01 Display Purchase Order Details
PO_DELETEPOHISTORY01 Delete Purchase Order History
PO_UPDATEPOHISTORY01 Update Purchase Order (Update, Insert)
PPCC2PREEVENT01 Transfer time event confirmations from PDC system
PPCC2PRETICKET01 Transfer time ticket confirmations from PDC system
PPCC2RECORDER01 Send production orders to PDC system
PPCC2RECWORKCENTER01 Send work centers to PDC system
PPCC2REQCONF01 Send upload request to PDC system
PPE_ACT01 Maintain or Create Process Structure
PPE_BAL01 Maintain or Create Line Balances
PPE_CMP01 Maintain or Create Product Structure
PPE_FLO01 Maintain or Create Factory Layout
PPE_GEN01 Create and Change Generic Engineering Node
PPE_PRM01 Create or Change iPEE Production Model
PPE_RES01 Maintain or Create iPPE Object Resources Nodes
WMTRID01 Transfer requirement
WORKC2 Workcenters in CC2
WORKC3 Work centers in CC3
WORKC4 Workcenters in CC4
WORKSCHED_WITH_COST01 Insert Operative Planned Working Time (With Cost Assignment)
WPDBBY01 POS outbound: bonus buy conditions
WPDCUR01 POS interface: Download exchange rates
WPDNA01 POS interface: Download products
WPDREB01 POS interface: Download special offer discounts
WPDSET01 POS interface: Download set assignments
WPDTAX01 POS interface: Download tax rates
WPDWGR01 POS interface: Download material group master
WPUBON01 POS interface: Upload sales docs (transctns), non-aggregated
WPUERR01 POS interface: Upload messages SRS/POS/SCS
WPUIB01 POS interface: Upload Fin.Acc. interface SRS/POS
WPUKSR01 POS Interface: Inbound processing cashier data for POS stats
WPUTAB01 POS interface: Upload day-end closing POS
WPUUMS01 POS interface: Upload sales data (compressed)
WPUUWW01 POS interface: Upload goods movements
WP_EAN01 POS interface: Upload/Download EAN assignments
WP_PER01 POS interface: Upload/Download person data
WP_PLU01 POS interface: Upload/Download article master
WP_PLU02 POS interface: material and condition (inbound and outb.)
WRMA_REV_DATA RMA revaluation data from BW
WRMA_REV_REQUEST Data request for revaluation in R/3 system
WTADDI01 Additionals
WVINVE01 Store phy.inv.: phy.inv. docs outbound; count data inbound
WVINVE02 Store phy.inv.: phy.inv. docs outbound; count data inbound
WVINVE03 Store phy.inv.: phy.inv. docs outbound; count data inbound
W_WGRP01 Distribution of material groups