**The Outbound Process**



**The Inbound Process**



**An Overview of IDocs**

An *IDoc* is a container that is used to exchange data between any two processes.

**IDoc Types**

An IDoc type structure can consist of several segments, and each segment can consist of several data fields. The IDoc structure defines the syntax of the data by specifying a list of permitted segments, an arrangement of the segments, and any mandatory segments. Segments define a set of fields and their format.

**Instantiated IDocs**

An IDoc is an instance of an IDoc type and consists of three types of records



**Multiple Messages per IDoc Type**

A *message* represents a specific type of document transmitted between two partners.

Example: Orders, order response, material master, and customer master

In SAP, an IDoc can be used to represent several logically related messages or business documents.

Example: the orders IDoc (ORDERS05) is used for several messages, including orders (ORDERS), order response (ORDRSP), and order change (ORDCHG).



**The Outbound ALE Process**

Distributed SAP systems exchange three types of data for achieving a distributed yet integrated environment.

**Transactional data**. Sales orders, purchase orders, contracts, invoices, G/L postings

**Master data**. The material master, customer master, vendor master, employee master

**Control data**. Company codes, business areas, plants, sales organizations, distribution channels, divisions.

**Control data is transferred using the regular CTS (*Correction and Transport System*) process.**

**An Overview of the Outbound Process Components**

**The Customer Model**

A customer model is used to model a distribution scenario. In a customer model, you identify the systems involved in a distribution scenario and the messages exchanged between the systems.

**Message Control**

Message control is a cross−application technology used in pricing, account determination, material determination, and output determination. The output determination technique of Message control triggers the ALE and EDI outputs for a business document.

**Change Pointers**

The change pointers technique is based on the change document technique, which tracks changes made to key documents in SAP, such as the material master, customer master, and sales orders. Changes made to a document are recorded in the change document header table CDHDR, and additional change pointers are written in the BDCP table for changes relevant to ALE.

**IDoc Structure**

A message is defined for data that is exchanged between two systems. The message type is based on one or more IDoc structures. For example, if you're going to send material master information to another system, a message type MATMAS is already defined in the system. This message is based on MATMAS01, MATMAS02, and MATMAS03. IDocs form a major component of the ALE and EDI interfaces.

**Selection Programs**

Selection programs, typically implemented as function modules, are designed to extract application data and create a master IDoc. A selection program exists for each message type. A selection program's design depends on the triggering mechanism used in the process.

**Filter Objects**

In a distributed environment, each recipient of data can have different requirements for the data being distributed. Filter objects remove unwanted data for each recipient of data.

**The Port Definition**

A port is used in an outbound process to define the medium in which documents are transferred to the destination system. ALE uses a tRFC (*Transactional Remote Function Call*) port, which transfers data in memory buffers.

Note The EDI process uses file ports to transfer data to the subsystem in a standard text file format

**The RFC Destination**

The RFC (*Remote Function Call*) destination is a logical name used to define the characteristics of a communication link to a remote system on which a function needs to be executed. In ALE, the RFC specifies information required to log on to the remote SAP system to which an IDoc is being sent.

**The Partner Profile**

A partner profile specifies the components used in an outbound process (the logical name of the remote SAP system, IDoc type, message type, and tRFC port), an IDoc's packet size, the mode in which the process sends an IDoc (batch versus immediate), and the person to be notified in case of errors. A partner profile is created for each SAP system with which you communicate, and a record exists for each message sent and received from a system. For example, if you are sending two outbound messagespurchase order (ORDERS) and material master (MATMAS)to the SHIPPING system, a partner profile will exist for the SHIPPING system. Two outbound recordsone for each message type (ORDERS and MATMAS)will exist in the partner profile.

**The Process Flow for Distributing Transactional Data**

Transactional data is distributed using two techniques:

with Message control and

without Message control.

The technique that a process uses greatly depends on the application area, but the core logic of the selection programs remains the same. The SD (*Sales and Distribution*) and MM (*Materials Management*) applications use Message control to trigger the ALE process. For example, programs for generating IDocs for sales order responses or purchase orders start via Message control.

Applications in other areas, such as FI (*Financials*), PP (*Production Planning*), and HR (*Human Resources*), do not use Message control for transactional data. The core logic of generating IDocs is part of the application logic itself.

**The Process Flow for Distributing Master Data**

Master data between SAP systems is distributed using two techniques: stand−alone programs and change pointers. The process flows are the same for these two processes, except for the triggering mechanism that starts the IDoc selection programs.





**Triggering the Outbound Process via Stand−Alone Programs**

Stand−alone programs are started explicitly by a user to transmit data from one SAP system to another. Standard programs for several master data objects exist in SAP. For example, the material master data can be transferred using the RBDSEMAT program or transaction BD10.

Note Stand−alone programs for the master data objects are grouped under the Master Data Distribution area menu (BALM).

**The Inbound ALE Process**

The inbound process for any kind of transactional or master data has two distinct paths

for posting the documents from the IDocs.

Via a function module

Via workflow

**An Overview of the Inbound Process Components**

An inbound process uses IDoc structure, posting programs, filter objects, conversion rules, a partner profile,

service programs, and configuration tables to post an application document from an IDoc.

**Posting Programs**

Posting programs, which are implemented as function modules, read data from an IDoc and create an

application document from it. A posting program exists for each message.

For example, the posting program for message type MATMAS is IDOC\_INPUT\_MATMAS. A

four−character process code, MATM, has been assigned to this function module.

**Workflow**

A workflow represents a sequence of customized steps (dialog and background) to be carried out for a

process.

**The Partner Profile**

A partner profile specifies the components used in an inbound process (partner number, message type, process code), the mode in which IDocs are processed (batch versus immediate), and the person to be notified in case of errors.

A partner profile is created for each SAP system with which you communicate, and a record exists for each inbound message received from a remote SAP system. For example, if two inbound messages, a purchase order (ORDERS) and material master (MATMAS), are received from the SALES system, a partner profile

**Process Flow for the Inbound Process via a Function Module**

****



**Configuring the ALE Infrastructure**

**Basic Settings for IDocs**

**IDoc Administration**

**Transaction: WE46**

The IDoc Administrator parameter specifies the administrator for the IDoc interface at run time. This person or organization is responsible for the integrity of the overall IDoc interface. The workflow system uses this parameter when a technical error occurs in the ALE interface layer, such as an error in reading the partner profile for an incoming IDoc.

**Communication Settings**

**Maintaining a Logical System**

**Path**: From the ALE customizing in IMG, choose Sending and Receiving Systems,

Logical Systems, Define Logical Systems.



**Allocating Logical Systems to the Client**

**Path**: From the ALE customizing in IMG, choose Sending and Receiving Systems,

Logical Systems, Assign Client to Logical System.



**Setting Up an RFC Destination**

**Transaction:** SM59

**Path**: From the ALE customizing in IMG, choose Communication, Define RFC Destination.

In this step, you create an RFC destination on the local system for each remote SAP system with which you want to communicate. In the RFC destination, you specify all the information necessary to log on to the remote system to execute functions remotely, including the host name, the user ID, the password, the client, the system number, and additional communication settings.

**The Port Definition**

**Transaction:** WE21

A port defines the medium in which data is exchanged between the two systems. In the ALE process, IDocs

are transferred via memory. As of release 4.6, six port types are available.

Transactional RFC (tRFC) ports used for ALE communication

File ports used by EDI

CPI−C ports used to communicate with R/2 systems

Internet ports used to connect with Internet applications

ABAP−PI ports used to process data via the ABAP programming interface

XML ports used to process files in XML format



**Port**. Any meaningful name to identify the port uniquely. During automatic generation, SAP assigns

it using sequential numbers prefixed with the letter A.



**Description**. Any meaningful description of the port. This attribute is for documentation only.

**Version**. This attribute defines whether IDocs are in the version 3.x format or version 4.x format.

**RFC Destination**. This attribute is the RFC destination maintained in the "Setting Up an RFC

Destination" section.

**Distributing Master Data**

Commonly Used Master Data in ALE

Message Type Description

COACTV Activity price of cost center and cost element combination

COAMAS Activity type

COGRP5 Activity type group

BOMMAT Bill of material (materials)

BOMDAT Bill of material (documents)

CLSMAS Classes

CHRMAS Characteristics

CLFMAS Classification

COSMAS Cost center

COGRP1 Cost center group

COGRP2 Cost element group

COELEM Cost element

DEBMAS Customer master

GLMAST G/L account

HRMD\_A HR master and PD−ORG data

MATMAS Material master

PRCMAS Profit center

COGRP6 Profit center group

INFREC Purchasing information record

SRVMAS Services master

SRCLIST Source list

CREMAS Vendor master



**How Is Master Data Distributed?**

**Push original copy**. Master data is sent explicitly from one system to another using stand−alone

programs. The selection screen allows you to specify object values, the destination system, and the

message type. For example, BD10 is used for sending material master data to another system.

**Push changes**. This is an automated process in which changes made to certain fields of a master data

object trigger the process of distributing the object to remote systems. You can customize the fields

for which changes are to be recorded, which enables you to distribute only the necessary changes.

This method provides an automatic and efficient means of keeping the master data synchronized

between the systems.

**Pull master data**. A system sends a request for a specific master object to be transferred to the

requesting system. Because the request is asynchronous, the system does not wait for the object to be

transferred. The request is delivered to the sending system, which initiates the process of sending the

master data to the requesting system

**The Basic Configuration for Distributing Data**

Before you can distribute any data between the systems via ALE, you must put in place the basic infrastructure described in Chapter 22, "Configuring the ALE Infrastructure," and execute the following steps.

1. Maintain a distribution model.

2. Generate a partner profile.

3. Distribute the distribution model.

4. Maintain the workflow settings.

**Maintaining the Distribution Model**

**Transaction:** BD64

A distribution model is used to model a distributed environment in which you specify messages exchanged between sending and receiving systems.

Here are some important points to keep in mind about the distribution model.

A distribution model is maintained on only one system. It is distributed to other systems for use.

Two models cannot distribute the same message between the same set of senders and receivers.

By default, the client on which you create a model becomes the model's owner. This approach

prevents the model from being changed on another client or system.

**Generating Partner Profiles**

**Transaction:** BD82

Partner profiles can be generated automatically for your partner systems, using transaction BD82.

You can execute transaction. The distribution model and settings in the ALE

tables TBD52 and EDIFCT are read to generate partner profiles and port definitions.

WE20 to view the partner profiles generated by this transaction

**Distributing the Model**

**Transaction:** BD64

When the maintenance of a distribution model is complete, you can distribute the model to the systems

involved in the distributed process.

Proceed from the transaction BD64 tree display by clicking the model to

highlight it. Then select Edit, Model View, Distribute from the pop−up menus

**Techniques for Distributing Master Data**

**The Push Approach**

**Executing the Process**

**Transaction:** BD10

**Distributing Changes**

In this section you will learn about the change pointer technique for distributing master data. First, you will learn how the process works, and then you will learn the steps to configure the system.

**The Change Pointer Technique**

**Configuration**

Besides creating the basic infrastructure described in Chapter 22, "Configuring the ALE Infrastructure," and performing the steps described earlier in "The Basic Configuration for Distributing Data," you must carry out the following configuration steps to enable master data distribution based on changes to the object.

**Enable Change Pointers Globally**

**Transaction:** BD61

**Enable Change Pointers for a Message Type**

**Transaction:** BD50

**Specify the Fields for Which Change Pointers Are to Be Written**

**Transaction:** BD52

**Testing and Executing the Configuration**

After you do all the necessary configuration settings, you should carry out sanity tests and process flow tests for the inbound and outbound process." The tests help you pinpoint any problems in the configuration or with missing elements. You are now ready to execute the process, which involves changing a material master and verifying that an IDoc is generated.

**Changing a Field in the Master Data**

Change a field in the master data object for which the change pointer is enabled. For example, if you change the net weight of a material in the material master data, a change pointer is written.

Tip You can verify a change document and change pointer by viewing entries in tables CDHDR and

BDCP, respectively.

**Executing Program RBDMIDOC to Process Change Pointers**

**The Fetch Process**

**Executing the Process**

**Transaction:** BD11

**Testing the ALE Interface**

**Prerequisites for Testing Outbound Processes**

The outbound process is composed of programs that are linked to run one after the other. To test each

component separately, you have to disconnect them. In the outbound record of the partner profile for your

message, set the Collect IDocs flag. The flag prevents the RSEOUT00 program from running immediately

after an IDoc is created in the system.











**Configuring an Inbound Process for New IDocs**

**Process Configuration**

1. Create a new message type.

2. Link the message type to the IDoc type.

3. Allocate the function module to the logical message type.

4. Define the attributes for the inbound function module.

5. Create a new process code.

6. Assign input methods

7. Create (or change) a partner profile.

**Create a New Message Type**

Transaction: WE81

Inbound process requires a message type assigned to the data contents transferred in the IDoc.

**Link the IDoc Type to the Message Type**

Transaction: WE82

You assign the new message type to the IDoc type.

**Allocate the Function Module to the Logical Message**

Transaction: WE57

**Define the Settings for the Inbound Function Module**

Transaction: BD51

This configuration step tells the ALE system how your function module has been implemented. The ALE layer uses these settings to invoke the function module with correct parameters.

**Mass processing (0).** This option is for function modules that use a direct input method to create an application document. The ALE layer can pass multiple IDocs to the function module at the same time.

**Individual input (1).** This option is for function modules that use the call transaction method to post an application document.

**Individual input with IDoc lock in call transaction (2).** This option is for function modules that use ALE−enabled transactions to post an application document.

**Create a New Process Code**

Transaction: WE42

This step defines a process code that points to the function module developed for the inbound process.

**Create or Change the Partner Profile**

Transaction: WE20

A partner profile is created for every partner system with which you exchange IDocs. In the partner profile, a record is created for every incoming message.