

# THE ADVANCED GUIDE TO IMPLEMENTING SAP R/3's APPLICATION LINK ENABLING (ALE)

## APPLICATION LINK ENABLING

## TABLE OF CONTENTS

<b>1</b>	<b>INTRODUCTION TO ALE .....</b>	<b>1-1</b>
<b>1.1</b>	<b>What is ALE?.....</b>	<b>1-2</b>
1.1.1	A Summary .....	1-2
1.1.2	The Detail as described by SAP.....	1-2
1.1.3	Mass Processing Of IDocs .....	1-5
1.1.4	Cross-System Document Trace (Audit) .....	1-5
1.1.5	Error Handling .....	1-6
1.1.6	What can be distributed?.....	1-6
<b>1.2</b>	<b>Why ALE? .....</b>	<b>1-7</b>
1.2.1	The SAP Solution - ALE .....	1-7
1.2.2	In SAP's words: .....	1-8
<b>1.3</b>	<b>Phases in an ALE project .....</b>	<b>1-8</b>
1.3.1	Phase 1: The Assessment and Project strategy phase .....	1-9
1.3.2	Phase 2: The Design phase .....	1-9
1.3.3	Phase 3: The Configuration phase.....	1-9
1.3.4	Phase 4: The Development phase.....	1-9
1.3.5	Phase 5: The Operations phase.....	1-9
<b>2</b>	<b>ALE ASSESSMENT .....</b>	<b>2-1</b>
<b>2.2</b>	<b>Project team tasks .....</b>	<b>Error! Bookmark not defined.</b>
2.2.1	Assess business requirements.....	2-1
2.2.2	Assess ALE and determine gap .....	2-1
2.2.3	Confirm data / function distribution requirement .....	2-1
<b>2.3</b>	<b>Project team deliverables .....</b>	<b>2-1</b>
2.3.1	Client strategy document.....	2-2
2.3.2	Detailed Business Requirements Document.....	2-2
2.3.3	Business Process Impact Analysis.....	2-2
2.3.4	Technical Impact Analysis.....	2-2
<b>2.4</b>	<b>Obtain programme sign-off .....</b>	<b>2-2</b>
<b>3</b>	<b>ALE DESIGN .....</b>	<b>3-1</b>
<b>3.1</b>	<b>Project team tasks .....</b>	<b>3-1</b>
3.1.1	Develop standards & procedures early .....	3-1
3.1.2	Distribution settings .....	3-1
3.1.3	Design for flexibility .....	3-1
3.1.4	Audit .....	3-1
3.1.5	Technical impact.....	3-2

3.1.6	Sign-off .....	3-2
<b>3.2</b>	<b>Deliverables.....</b>	<b>3-2</b>
3.2.1	ALE Naming Standards .....	3-2
3.2.2	ALE Design Document .....	3-2
3.2.3	ALE Change Management Procedure .....	3-2
3.2.4	ALE Migration Strategy .....	3-2
3.2.5	ALE Project Deliverables.....	3-2
<b>4</b>	<b>ALE CONFIGURATION.....</b>	<b>4-1</b>
<b>4.1</b>	<b>Project team tasks .....</b>	<b>4-1</b>
4.1.1	Confirm business requirements.....	4-1
4.1.2	Configure ALE basis area .....	4-1
4.1.3	Functional configuration .....	4-1
4.1.4	QA & testing .....	4-2
4.1.5	Sign-Off .....	4-2
<b>4.2</b>	<b>The HOW of the ALE Technical Configuration .....</b>	<b>4-2</b>
4.2.1	Create background ALE user .....	4-2
4.2.2	ALE Administrator position .....	4-2
4.2.3	Set up logical system .....	4-2
4.2.4	Maintain number range for ports .....	4-3
4.2.5	Maintain number range for IDocs .....	4-4
4.2.6	Maintain number ranges for IDoc types and segment version.....	4-5
4.2.7	Maintain number range for change pointers .....	4-5
4.2.8	Set up ISO codes .....	4-6
4.2.9	Make basic settings for Workflow.....	4-7
4.2.10	Deleting change pointers copied from source client .....	4-7
4.2.11	Archiving IDocs copied from source client .....	4-7
4.2.12	Activating change pointers .....	4-7
4.2.13	Define RFC destination .....	4-7
<b>4.3</b>	<b>The HOW of the ALE Functional Configuration .....</b>	<b>4-8</b>
4.3.1	Control Data .....	4-8
4.3.2	Reduction of IDocs (BD53) <i>Client Independent</i> .....	4-8
4.3.3	Setting up global organisational units .....	4-9
4.3.4	Configuring the Distribution Model (BD64).....	4-10
4.3.5	Changing the owner of the distribution model.....	4-11
4.3.6	Set up Filter Objects (BD64) .....	4-11
4.3.7	Procedure .....	4-11
4.3.8	Maintain sending system partner profile automatically (BD82).....	4-11
4.3.9	Maintain sending system partner profile manually (WE20).....	4-12

4.3.10	Distribute customer model (BD71) .....	4-13
4.3.11	Maintain receiving system partner profile automatically (BD82) .....	4-14
4.3.12	Maintain receiving system partner profile manually (WE20) .....	4-14
4.3.13	ALE Audit.....	4-14
4.3.14	Conversion rules .....	4-17
4.3.15	Customer extensions.....	4-21
4.3.16	Defining settings for IDoc conversion (BD55) .....	4-21
4.3.17	Setting up workflow .....	4-21
4.3.18	Listings .....	4-22
4.3.19	Schedule active monitoring .....	4-27
4.3.20	Change pointers .....	4-27
4.3.21	Mass processing .....	4-28
4.3.22	Schedule IDoc dispatch.....	4-29
4.3.23	Schedule IDoc processing.....	4-30
4.3.24	Schedule Status conversion after successful RFC .....	4-31
4.3.25	Serialisation (BD57) <i>Client Independent</i> .....	4-32
4.3.26	Object types.....	4-32
4.3.27	Segment Filtering (BD56) .....	4-33
4.3.28	Transaction scenarios (BDM5) .....	4-33
<b>4.4</b>	<b>Deliverables.....</b>	<b>4-33</b>
4.4.1	ALE Configuration Sign-Off Procedure .....	4-33
4.4.2	ALE Configuration Procedure.....	4-33
4.4.3	ALE Detailed Configuration Guide .....	4-34
4.4.4	ALE Basis Configuration Procedure.....	4-40
4.4.5	ALE Workflow Configuration Procedure.....	4-40
<b>4.5</b>	<b>Other issues .....</b>	<b>4-40</b>
4.5.1	Which scenarios? .....	4-40
4.5.2	Control Data Implications .....	4-40
4.5.3	Other Master Data Implications.....	4-40
4.5.4	Distribution Strategy .....	4-40
4.5.5	Information Distributed .....	4-41
4.5.6	Field conversion .....	4-41
4.5.7	Conditions of distribution .....	4-41
4.5.8	Transfer Initiation Strategy .....	4-42
4.5.9	Mass processing .....	4-42
4.5.10	Serialization .....	4-42
4.5.11	ALE performance .....	4-42
4.5.12	Error notifications .....	4-42

<b>5</b>	<b>ALE DEVELOPMENT</b>	<b>5-1</b>
<b>5.1</b>	<b>Project team tasks</b>	<b>5-1</b>
5.1.1	Functional specifications	5-1
5.1.2	Technical specifications	5-1
5.1.3	Development	5-1
5.1.4	Quality assurance and testing	5-2
5.1.5	Business sign-off	5-2
<b>5.2</b>	<b>Deliverables during this phase</b>	<b>5-2</b>
5.2.1	ALE Functional Specification	5-2
5.2.2	ALE Development Guide	5-2
<b>6</b>	<b>ALE OPERATIONS SUPPORT</b>	<b>6-1</b>
<b>6.1</b>	<b>Project team tasks</b>	<b>6-1</b>
6.1.1	ALE Administrator role	6-1
6.1.2	Capacity plan	6-1
6.1.3	ALE Operations role	6-1
6.1.4	ALE Failure recovery	6-1
6.1.5	ALE Support organisation structure and process	6-1
<b>6.2</b>	<b>Other considerations</b>	<b>6-1</b>
6.2.1	Client copies	6-1
6.2.2	Client specific configuration	6-2
6.2.3	Error handling	6-2
6.2.4	ALE specific scenarios	6-2
<b>6.3</b>	<b>Deliverables</b>	<b>6-2</b>
6.3.1	ALE Administrator Troubleshooting Procedure	6-2
6.3.2	ALE Archiving strategy and procedure	6-6
6.3.3	ALE Capacity Planning Guide	6-11
6.3.4	ALE Failure and recovery procedure	6-11
6.3.5	ALE Foreground Processing - By the user	6-19
6.3.6	ALE Background Job Scheduling Procedure	6-23
6.3.7	ALE Performance Testing & Tuning Guide	6-27
6.3.8	ALE Production client cleanup procedure	6-31
<b>7</b>	<b>GENERAL</b>	<b>7-1</b>
<b>7.1</b>	<b>Glossary</b>	<b>7-1</b>
<b>7.2</b>	<b>ALE Diagrams of use</b>	<b>7-3</b>
7.2.1	ALE Configuration setup diagram	7-3
7.2.2	ALE Message Type Flow Diagram	7-3

<b>7.3</b>	<b>Setting up ALE to send messages from 1 client to itself .....</b>	<b>7-3</b>
7.3.1	Setting up the dummy client logical system .....	7-3
7.3.2	Maintain Logical System .....	7-3
7.3.3	Define RFC Destination.....	7-4
7.3.4	Define RFC Port .....	7-4
7.3.5	Setting up model messages .....	7-4
7.3.6	Automatic Generation of Partner Profiles.....	7-4
7.3.7	Creation of Filter Messages .....	7-5
7.3.8	Create Partner Profiles .....	7-5
7.3.9	Standard ALE Client.....	7-5
7.3.10	Dummy Client.....	7-5
<b>7.4</b>	<b>ALE User Guide .....</b>	<b>7-6</b>
<b>7.5</b>	<b>Solution Pack .....</b>	<b>7-6</b>
<b>7.6</b>	<b>ALE Related SAP programs .....</b>	<b>7-6</b>
<b>7.7</b>	<b>ALE Related Tables .....</b>	<b>7-11</b>
<b>7.8</b>	<b>ALE Transaction Codes .....</b>	<b>7-13</b>

**INDEX .....** **INDEX-ERROR! BOOKMARK NOT DEFINED.**

## FIGURES

Figure 1: ALE Outbound Processing .....	1-3
Figure 2: ALE Inbound Processing .....	1-4
Figure 3: ALE project team tasks during ALE Assessment phase .....	2-1
Figure 4: ALE project team tasks during ALE Design phase.....	3-1
Figure 5: ALE project team tasks during ALE Configuration phase .....	4-1
Figure 6 : ALE Configuration Checksheet.....	4-34
Figure 7: Material Master Message Flow Diagram .....	4-35
Figure 8: Distribution Strategy Options .....	4-41
Figure 9: ALE project team tasks during ALE Development phase.....	5-1
Figure 10: ALE Development components pictorial.....	5-2
Figure 11: ALE Scenario model .....	5-3
Figure 12: Example Purchasing & Selling scenario.....	5-4
Figure 13: IDoc type ZINVRV01 .....	5-5
Figure 14: Outbound processing example code .....	5-9
Figure 15: Inbound processing example code .....	5-13
Figure 16: ALE project team tasks during ALE Operations phase .....	6-1
Figure 17: Overview of the archiving process.....	6-7
Figure 18 Message flow between systems .....	6-17
Figure 19: Describing the ALE Process .....	6-19

Figure 20: ALE background job description .....	6-23
Figure 21 : Sending ALE messages from a client to itself .....	7-3

## TABLES

Table 1: ALE team roles and time dependencies .....	<b>Error! Bookmark not defined.</b>
Table 2 : ALE Performance Test Options .....	6-30
Table 3 : ALE Performance Test Results.....	6-30

*CHAPTER 1*  
**AN INTRODUCTION TO ALE**

---



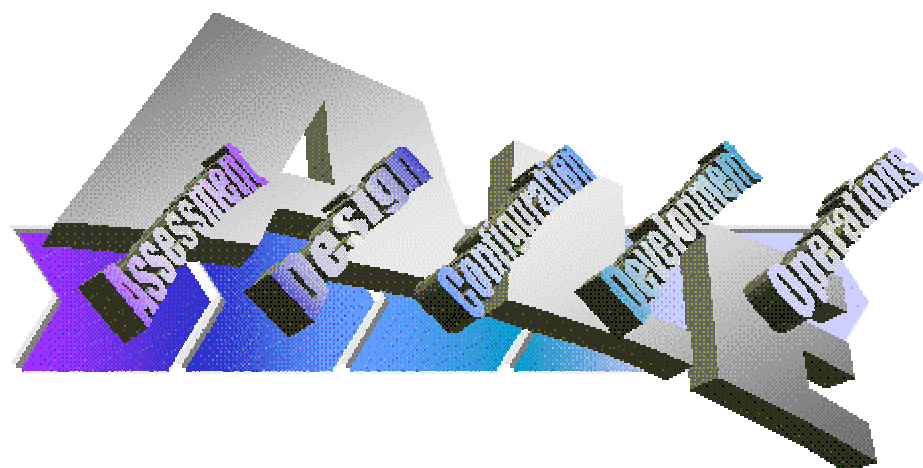
## 1 INTRODUCTION TO ALE

The purpose of this whitepaper is to provide you with a methodology on implementing SAP R/3's ALE technology. For the executive who is looking into deploying ALE the first chapter will be of benefit as well as the summary for each of the remaining chapters which will give a good indication as to the level of effort that will be required in order to implement ALE.

It highlights the areas to look at; it gives example documentation and describes the implementation process in detail. It is not an idiot's guide to implementing ALE. Detailed knowledge of the ALE functionality together with project management skills will ensure a smooth ALE implementation.

The Guide is divided into the following chapters:

- Chapter 1. Introduction to ALE;**  
*What is ALE? Why ALE?*
- Chapter 2. ALE Assessment;**  
*Putting a project team together. Deliverables of the team. Setting standards. Investigation and position papers.*
- Chapter 3. ALE Design;**  
*Design phase. Design document. Performance and technical implications.*
- Chapter 4. ALE Configuration;**  
*Configuration & related procedures.*
- Chapter 5. ALE Development;**  
*Describing ALE development. Development example.*
- Chapter 6. ALE Operations Support; and**  
*Production support procedures. Production support planning. Monitoring and archiving. Disaster recovery.*
- Chapter 7. ALE general issues.**  
*ALE transaction codes, tables, programs... Glossary.*



**ASSESSMENT -> DESIGN -> CONFIGURATION -> DEVELOPMENT -> OPERATIONS**

## 1.1 What is ALE?

The ALE (Application Link Enabling) concept available in R/3 (Release 3.0) supports the development of applications across different SAP systems. It incorporates the exchange of business information across these systems whilst ensuring consistency and integrity of the data. This functionality is achieved with the use of IDocs (Information Document) as opposed to the use of a centralized database.

### 1.1.1 A Summary

ALE allows the user to perform an SAP transaction in the sending system, after-which the following steps occur:

- 1 or more communication IDocs (intermediate documents: container for the application data) are created in the sending system database. An ALE distribution model, that needs to have been configured, determines which systems the IDocs are to be sent
- These communication IDocs, that contain the relevant application data of the transaction that was performed, are then passed to the ALE communication layer
- This layer performs an RFC call, using the port definition and an RFC destination determined through the customer model
- The IDocs are then transferred to the respective receiving systems. These could be SAP R/3, R/2 or external systems
- If the receiving system is an SAP system then:
  - In the case of master data distribution the same transaction that was performed on the sending system is again performed on the receiving system with the data contained in the IDoc. This allows the data to go through the SAP checks before posting occurs
  - In the case of transaction scenarios the relevant data is passed to the respective transactions in order to load the required application document. E.g., a PO is loaded on the sending side, yet a SO is created on the receiving system
- Master data has another difference:
  - It can be set up in such a way that any changes made to specific fields in master data tables can automatically trigger off the ALE distribution process for that particular master data object
  - If a master data object is created or changed on a sending system and distributed to another system the respective transaction is used to either create or change that respective master data object on the receiving system

In general, if standard SAP can't perform the task required then neither can ALE. It doesn't add functionality; it merely de-couples it and allows you to distribute it onto other remote systems.

### 1.1.2 The Detail as described by SAP

In the outbound processing one of the function modules of the application creates an IDoc, the so-called master IDoc. This IDoc is sent to the ALE layer where the following processing steps are applied:

#### 1.1.2.1 Outbound processing

This area describes how ALE is used to process a transaction from the sending system to the communication layer. The inbound processing will show how ALE is used to take an IDoc from the communication layer to the receiving system and post it.

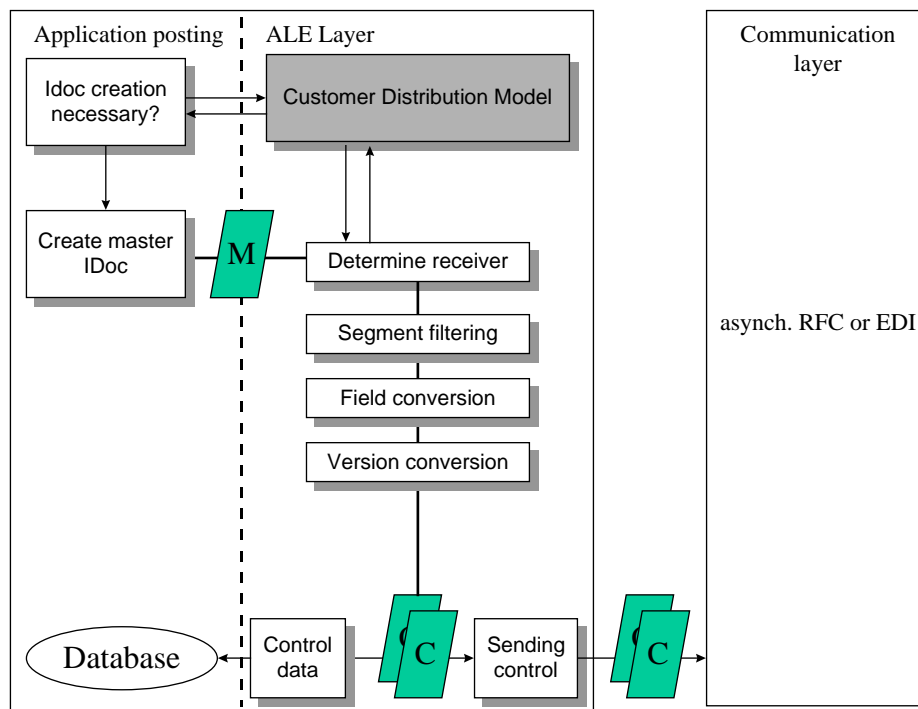


Figure 1: ALE Outbound Processing

In the outbound processing, one of the function modules of the application creates an IDoc, the so-called master IDoc. This IDoc is sent to the ALE layer where the following processing steps are applied:

#### Receiver determination

An IDoc is similar to a normal letter in that it has a sender and a receiver. If the receiver has not been explicitly identified by the application, then the ALE layer uses the customer distribution model to help determine the receivers for the message.

The ALE layer can find out from the model whether any distributed systems should receive the message and, if so, then how many. The result may be that one, several or no receivers at all are found.

For each of the distributed systems that have been ascertained to be receiver systems, the data that is specified by the filter objects in the customer distribution model is selected from the master IDoc. This data is then used to fill an IDoc, and the appropriate system is entered as receiver.

#### Data selection

##### Segment filtering

In ALE Customising, segments can be specified for each recipient that should not generally be sent to that recipient. This information is used at this point to filter out those segments. The appropriate setting depends on the sending and receiving logical R/3 System.

##### Field conversion

Receiver-specific field conversions are defined under Functions for the IDoc processing -> Conversions.

General rules can be specified for field conversions; these are important for converting data fields to exchange information between R/2 and R/3 Systems. For example, the field "plant" can be converted from a 2-character field to a 4-character field.

The conversion is done using general EIS conversion tools (Executive Information System).

##### Version change

SAP ensures that ALE functions between different R/3 System releases. By changing the IDoc format you can convert message types of different R/3 releases. SAP Development uses the following rules when converting existing message types:

- *Fields may be appended to a segment type;*
- *Segments can be added;*

ALE Customising keeps a record of which version of each message type is in use for each receiver. The correct version of the communication IDoc is created in the ALE output.

### Dispatch control

The resulting IDocs (it is possible that several IDocs could be created in the receiver determination) are referred to as communication IDocs and are stored in the database. The dispatch control then decides which of these IDocs should be sent immediately. These are passed to the communications layer and are sent either using the transactional Remote Function Call (RFC) or via file interfaces (e.g. for EDI).

Controlling the time of dispatch:

- The IDocs can either be sent immediately or in the background processing. This setting is made in the partner profile.
- If the IDoc is to be dispatched in the background, a batch job has to be scheduled. The dispatch cycle may be chosen at will (daily, weekly, etc.)

Controlling the amount of data sent:

- The packet size can be defined. The packet size is the number of IDocs that are sent in a single operation.

If an error occurs in the ALE layer, the IDoc containing the error is stored and a workflow is created. The ALE administrator can use this workflow to process the error.

### 1.1.2.2 Inbound processing

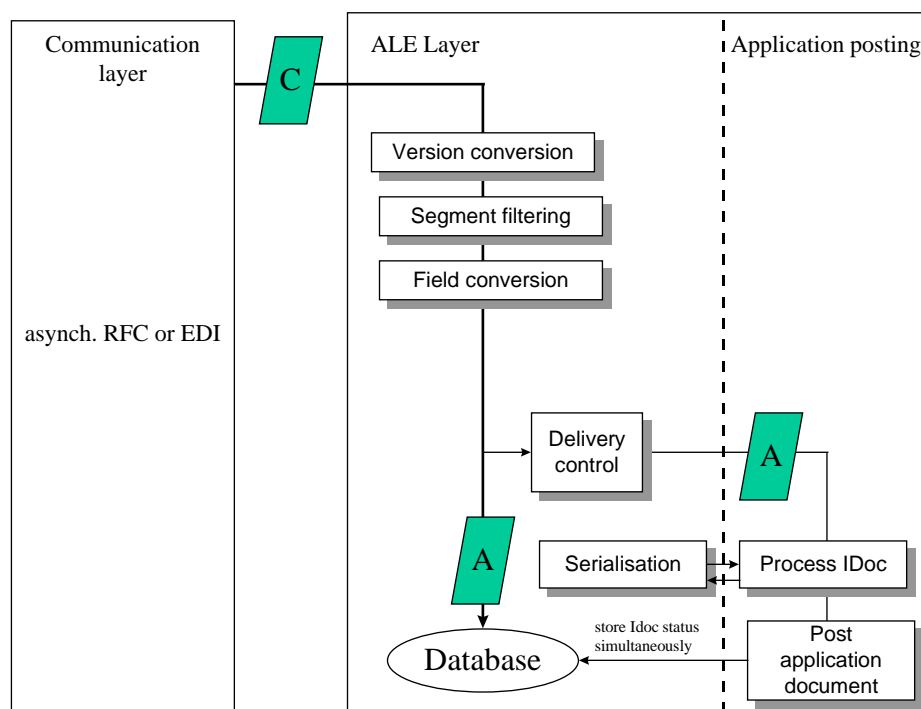


Figure 2: ALE Inbound Processing

After an IDoc has been successfully transmitted to another system, inbound processing is carried out in the receiver system, involving the following steps in the ALE layer:

## Segment filtering

Segment filtering operates the same way in inbound processing as in outbound processing.

## Field conversion

Specific field conversions are defined in ALE Customising.

The conversion itself is performed using general conversion tools from the EIS area (Executive Information System).

Generalised rules can be defined. The ALE implementation guide describes how the conversion rules can be specified.

One set of rules is created for each IDoc segment and rules are defined for each segment field.

The rules for converting data fields from an R/2-specific format to an R/3 format can be defined in this way. An example of this R/2 - R/3 conversion is the conversion of the plant field from a 2 character field to a 4 character field.

## Data transfer to the application

### Input control

When the IDocs have been written to the database, they can be imported by the receiver application.

IDocs can be passed to the application either immediately on arrival or can follow in batch.

### You can post an inbound IDoc in three ways:

1. By calling a function module directly: A function is called that imports the IDoc directly. An error workflow will be started only if an error occurs.
2. By starting an SAP Business Workflow. A workflow is the sequence of steps to post an IDoc.
3. By starting a work item a single step performs the IDoc posting.

The standard inbound processing setting is that ALE calls a function module directly. For information about SAP Business Workflow alternatives refer to the online help for ALE programming.

### 1.1.3 Mass Processing Of IDocs

The option of performing mass processing is provided for reasons of performance. This involves sending packets consisting of several IDocs to an application workflow.

The IDoc packets are put together using collective processing in the output processing before dispatch takes place. Each packet contains IDocs of a single message type intended for a single recipient.

These IDocs are then all communicated in a single step.

There are some scenarios that cannot deal with mass processing of incoming documents. This is always the case if the application that imports the data has to execute a batch input one batch at a time.

In such a case the packet size has to be set to 1 in the output processing.

### 1.1.4 Cross-System Document Trace (Audit)

The technique of using IDocs to distribute data guarantees the security of the data transfer between systems, since the transfer is always repeated should a transmission error occur or the connection get broken. The purpose of ALE Audit is to check the cross-system IDoc flow and posting in the partner system.

ALE Audit provides an automatic and seamless tracing facility for individual documents in distributed applications. Documents that have been sent from one application to another can also be traced from start to finish. Another feature of ALE Audit is that the processing status of a document can be queried from a central point.

ALE Audit periodically sends information on the processing status of messages in the target system back to the sender system. This allows cross-references to be created in the sending system. Based

on this data, the following information can then be ascertained for any application document communicated via ALE, even if the data containers (IDocs) have been reorganised in the meantime:

- *the system to which the document was sent,*
- *the processing status of the IDoc in the target system,*
- *the ID of the application document in the target system, if known.*

The second main component of ALE Audit is a statistics database for IDocs that is maintained on the sending system. Key figures for the current status of the IDocs in the sending system and recipient system are updated daily (based on the creation date of the IDocs) at the level of the message type, sending system and recipient system of the IDocs.

Confirmations are sent asynchronously using an IDoc of the ALEAUD message type. The recipient system is responsible for initiating confirmation messages, since the time at which the update is made is of great importance. If the sending system were to request a confirmation too soon, the error queue would quickly overflow with IDocs in the target system that is waiting to be imported by the application. In order to improve performance, the confirmation messages are sent as periodic batch jobs rather than being sent immediately that the IDoc is imported by the application. Therefore if the IDoc is part of a time-critical action, the batch job for this message type should be scheduled with a suitably short repeat interval. The statistics database is updated at the same time as the confirmation is sent.

The ALE distribution model is responsible for determining the sending systems to which a confirmation should be sent and the message types for which a confirmation is required.

Statistics reports can be called up with the IDoc monitor. The statistics are updated using a push-button. If necessary, an RFC is started which fetches up-to-date information from the target system. There is also the option of completely regenerating all the statistics, which may be of use in "emergencies" - if changes to the system time or date have rendered the existing statistics inconsistent, for example.

ALE Audit supports all possible sender-recipient pairs of logical systems. In each case, the recipient has to note the IDoc number of the sending system.

### **1.1.5 Error Handling**

The following is a description of how an error that occurs during ALE processing is handled:

- The processing of the IDoc causing the error is terminated.
- An event is triggered.
- This event starts an error work item:
  - *The employees responsible will find a work item in their workflow inboxes.*
  - *An error message is displayed when the work item is processed.*
  - *The error is corrected in another window and the IDoc can then be resubmitted for processing.*
  - *If the error cannot be corrected, the IDoc can be marked for deletion.*
  - *Once the IDoc has been successfully imported, an event is triggered that terminates the error work item. The work item then disappears from the inbox.*

### **1.1.6 What can be distributed?**

#### **1.1.6.1 Control data**

The control data includes all objects that describe how the system is organised. These include Customising data like company codes, languages, purchasing organisations, plants and user maintenance.

The customer details his specific distribution in the customer distribution model. Once the control data is distributed the control data cannot be changed on the receiving systems. All changes are made to the central system and transported to the receiving systems.

### 1.1.6.2 Master data

The distribution scenarios for the master data are contained in the reference model. Rather than distributing the entire master data information, views on it are distributed (for example, sales views on the material master).

By configuring the views, the customer can select the master data fields to be distributed.

### 1.1.6.3 Transaction data

The distribution scenarios for the transaction data are stored in the distribution reference model.

Examples of transaction data are customer order, distributed contracts, purchase order, shipping notification and invoice.

## 1.2 Why ALE?

ALE is business solutions to a very real need emerging in the SAP market. This is the need for businesses to move towards tighter integration between systems, yet, at the same time, provide independence to the various business units within the company.

In the past the move was towards centralised systems...

Standardisation of business processes accompanied by ever-tighter integration within the central system no longer represents a practicable approach to this problem. The following are some of the most commonly encountered difficulties:

- *technical bottlenecks,*
- *upgrade problems,*
- *the effect of time zones on international corporations,*
- *excessively long response times in large centralised systems.*

How do you achieve this "loosely coupled applications" without compromising data integrity, without committing yourself to a specific technology, without adding to the technical issues mentioned above?

What if you want to link in to external systems as seamlessly as possible?

### 1.2.1 The SAP Solution - ALE

SAP has provided ALE (Application Link Enabling) as the solution to these issues. It allows you to:

- *distribute your applications across several SAP systems, such that centralized functions, as well as decentralized functions can operate in the same company arena,*
- *maintain and distribute master data elements from a central system, thus maintaining unique number ranges across several systems,*
- *maintain and distribute control data objects from a central system, thus synchronizing important configuration data. This is important when trying to decentralize functions, yet keep them integrated*
- *couple R/2 and R/3 systems, in some instances,*
- *couple SAP and external systems, via IDocs (Intermediate documents) and an external translation mechanism.*

ALE also provides you with tools that allow you to:

- *manage the ALE process seamlessly. These tools include the reprocessing of communication errors, business process and technical error handling and routing via workflow, archiving of data*
- *keep audit trails of all the ALE communication*
- *configure the ALE solution in many different ways in order to tailor make a solution for the user*

### 1.2.2 In SAP's words:

*"The ALE (Application Link Enabling) concept available in R/3 Release 3.0 supports the construction and operation of distributed applications. It incorporates the controlled exchange of business data messages whilst ensuring data consistency across loosely coupled SAP applications. The integration of various applications is achieved by using synchronous and asynchronous communication, rather than by means of a central database.*

*ALE comprises three layers:*

- *the applications services,*
- *the distribution services,*
- *the communications services.*

*In order both to meet the requirements of today's customers and to be open for future developments, ALE must meet the following challenges:*

- *Communication between different software releases.*
- *Continued data exchange after a release upgrade without special maintenance.*
- *Independence of the technical format of a message from its contents.*
- *Extensions that can be made easily even by customers.*
- *Applications that are decoupled from the communication.*
- *Communications interfaces that allow connections to third party applications.*
- *Support for R/3-R/2 scenarios."*

### 1.3 Phases in an ALE project

Normal SAP projects should include the following areas of focus:

#### **Phase 1: What should the focus be?**

- The Steering Committee
- Identify Modules
- Project Leader
- Structure the Project Teams
- Integrate the Team Findings and Decisions
- Set Objectives
- Develop Guiding Principles
- Develop a Detailed Project Plan

#### **Phase 2: Create As Is Picture**

- Analyze Current Business Processes
- Map Business Processes into R/3
- Identify Data and System Interfaces
- Inventory Existing Hardware/Software
- Install R/3



- Begin Project Team Training

### **Phase 3: Create *To Be* Design**

- Develop High-Level Design
- The R/3 Hierarchy
- Gain User Acceptance
- Detailed Design
- Scenarios, Scripts, and Tables
- Iterative Prototyping Using Tables
- User Communications
- Gain Final Acceptance

### **Phase 4: Construction and Testing**

- Develop a Comprehensive Configuration
- Populate the Test Instance with Real Data
- Build and Test Interface Programs
- Write and Test Reports
- Test the System
- User Testing

### **Phase 5: Actual Implementation**

- Operation mode
- After Implementation
- Moving into the Future

In terms of ALE there are basically 5 phases of the ALE project that need to be completed sequentially:

#### **1.3.1 Phase 1: The Assessment and Project strategy phase**

Mapping the business needs to ALE, determine gap and implications, prototype if necessary, agree and sign-off scope. This equates to the setting the focus phase.

#### **1.3.2 Phase 2: The Design phase**

In this phase the overall system design needs to be planned and documented with the following considerations: client strategies, naming standards, distribution settings, future flexibility, audit requirements, technical impact, performance, DRP and archiving strategies.

#### **1.3.3 Phase 3: The Configuration phase**

Take the design and confirm the business requirements, before commencing with the basis configuration, functional configuration, QA and testing and sign-off.

#### **1.3.4 Phase 4: The Development phase**

Functional specifications need to be drawn up to cover any gaps discovered during the design phase. These need to be developed by the ALE/ABAP team making use of technical specifications, these documents is then used to develop and configure the new solution, followed by QA, testing and sign-off.

#### **1.3.5 Phase 5: The Operations phase**

Lots of considerations needs to be taken when moving your solution into production including the following: Implement support organisation structure, define support roles, capacity planning, job monitoring, failure recovery and problem resolution processes.

Each of these phases is discussed in more detail in the following chapters.

*CHAPTER 2*  
**THE ALE ASSESSMENT PHASE**

---

## 2 ALE ASSESSMENT

This section describes how to plan and set up an ALE project.

### 2.1



Figure 3: ALE project team tasks during ALE Assessment phase

#### 2.1.1 Assess business requirements

The business requirements for all inter-SAP functionality and / or distribution of data need to be documented. Conducting workshops with the business specialists is the best way to accomplish this.

Map out what the business needs, both present and future, into:

- *Distributable functions;*
- *Common master data;*
- *Control data implications;*
- *Technical: Up-time, response time; and*
- *Audit & legal requirements.*

Document each individual distributable requirement individually.

#### 2.1.2 Assess ALE and determine gap

An investigation to verify the use of standard ALE scenarios to satisfy the business requirements or to identify the extent of the gap between standard ALE scenarios and the required business functionality needs to be undertaken.

The performance impact of ALE must be assessed and fed into the architecture team where the sizing will be done in conjunction with the hardware team.

The impact of the ALE scenario / configuration on the business processes must be evaluated and communicated to the business champions. The gaps, especially where functionality is lost or the process has been changed, needs to be documented and included in the solution pack for that particular distribution need.

#### 2.1.3 Confirm data / function distribution requirement

The business requirements confirmed earlier will be worked to a lower level of detail, i.e. the identification of records and data elements for distribution. These lower level requirements will then be confirmed.

### 2.2 Project team deliverables

During this phase the following are seen as the necessary deliverables of the team:

### **2.2.1 Client strategy document**

When implementing ALE you need to make decisions up front, in the following areas, in order to smooth the way down the road:

- *Overall client instance strategy. IE. A naming standard to group like clients logically*
- *Do you have a separate ALE reference client or not? Does it need a server of its own? (Rule of thumb: Large volumes of transactions = separate server. Where at all possible keep the reference client separate from production clients that store transaction data)*
- *Which client will own the distribution models? (You need to keep this standard especially if you have many clients)*
- *How will you keep your master / control data in synch between all of the clients? (ALE Control data distribution or CTS? Either is OK, but stay away from the CDM for versions prior to V4.x)*
- *How will client creations, refreshes and copies be handled? The ALE team will be affected by these changes. (Client refreshes leave data that is redundant in the new client... You need to clean it up.)*

### **2.2.2 Detailed Business Requirements Document**

- Describe each functional scenario to:
  - *Distribution options: E.g., maintain on reference client and distribute, maintain on local clients and distribute upwards, etc...*
- Signed off by the process champions and business owners

### **2.2.3 Business Process Impact Analysis**

Defining the implications of the gap between the ALE scenarios and the business processes as mapped out by the functional specialists

### **2.2.4 Technical Impact Analysis**

High level overview of the volumes and performance impact to be introduced by ALE. Any impact will be mapped on to the particular hardware solution provided by the hardware vendor.

**ALE Whitepaper** - Business requirements assessment, ALE functionality mapping, impact assessment (functional and technical). Gap analysis. Scope.

- *Describe the relationship with EDI*
- *Performance implications / estimated volumes of data, together with their frequencies are required*
- *ALE Administrator position to be considered*
- *Data conversion / clean up*
- *Impact of upgrades on the various systems*

**ALE Team Approach** - Resource & skill requirements, deliverables and deadlines.

**Sign-off of scope** - The agreed scope of the ALE Implementation project must be signed-off.

## **2.3 Obtain programme sign-off**

Before commencing with the detailed design of the ALE project you need to get approval from the programme office agreeing on the scope of the ALE project.

*CHAPTER 3*  
**ALE DESIGN PHASE**

### 3 ALE DESIGN

#### 3.1 Project team tasks

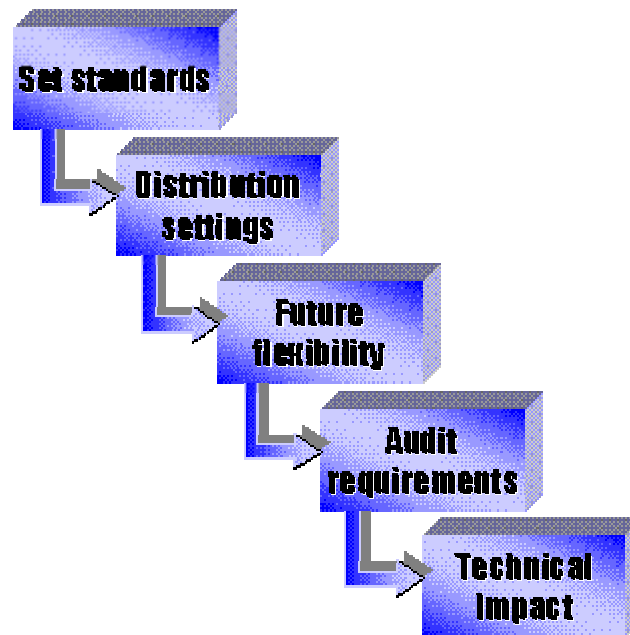


Figure 4: ALE project team tasks during ALE Design phase

##### 3.1.1 Develop standards & procedures early

- Naming standards: Logical systems, reduced message types, custom object types, custom function modules; and
- Change management procedures: Determine how you are going to manage the various clients and their different configurations. Define a process for configuration changes.

##### 3.1.2 Distribution settings

Document the various settings, per scenario, that are required to implement the solution. These include:

- Message types and fields;
- Filter objects;
- Serialization;
- Push Vs Pull;
- Immediate Vs background processing;
- Conversion rules;
- Centrally managed on which system?; and
- ...

##### 3.1.3 Design for flexibility

Future flexibility must be catered for in the design. This is especially important when considering the organization structure, as certain scenarios require certain control data to be in synch across the various clients. This is difficult to change once implemented...

##### 3.1.4 Audit

Audit requirements for transaction scenarios must be noted and designed for. Consider archiving, audit message types.

### **3.1.5 Technical impact**

Consider the potential technical impact of the proposed ALE solution its disk & processor utilization, network usage, dialog & background processing. Archiving strategy as well as disaster recovery planning

### **3.1.6 Sign-off**

Sign-off the design together with the business.

## **3.2 Deliverables**

### **3.2.1 ALE Naming Standards**

Set the standards for all configuration and development objects, including: Logical system, Dummy Logical system, RFC Destinations, Global company codes, Global business areas, Customer Distribution Model, Reduction of IDocs, Conversion rules, Object types, Class types, ALE Background jobs: Housekeeping, Active monitoring, IDoc jobs.

For development consider: Process codes, Function modules, Object types, IDoc types and message types.

### **3.2.2 ALE Design Document**

Details business process, functional impact, technical impact boundaries, workflow requirements, timing of jobs, etc....

### **3.2.3 ALE Change Management Procedure**

Schematic diagram of up to date client configuration status, configuration change request procedure, ALE configuration checklist.

### **3.2.4 ALE Migration Strategy**

How are you going to put the various clients into production, timing, impact on business processes.

### **3.2.5 ALE Project Deliverables**

Short description of the deliverables that the team must produce during the life cycle of the project.



*CHAPTER 4*  
**ALE CONFIGURATION PHASE**

## 4 ALE CONFIGURATION

This section contains documents describing how to configure ALE. These documents range from generic configuration guides to specific configuration Templates.

### 4.1 Project team tasks

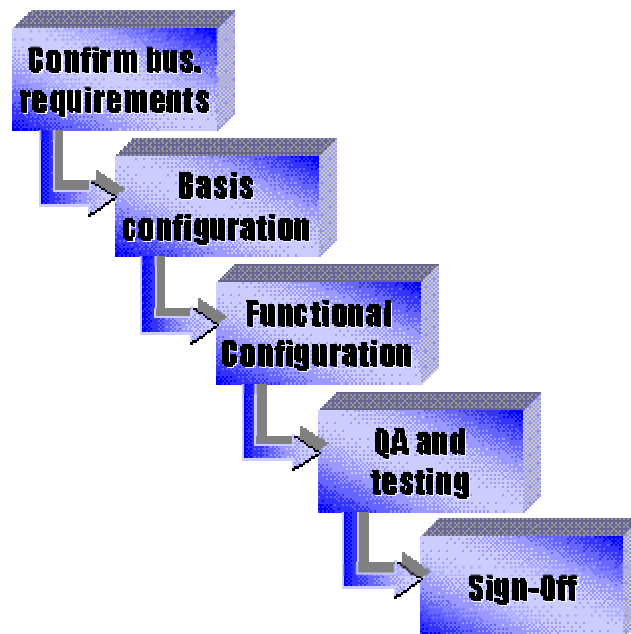


Figure 5: ALE project team tasks during ALE Configuration phase

The following steps need to be noted when configuring your ALE solution:

#### 4.1.1 Confirm business requirements

Before commencing configuration ensure that you have the detail of the business requirements for the whole scope of the implementation.

#### 4.1.2 Configure ALE basis area

Configure the ALE basis portion to enable ALE in the development environment. This involves the following steps:

- Create Logical System (LS) for each applicable ALE-enabled client;
- Link client to Logical System on the respective servers;
- Create background user, to be used by ALE, on each client (with sufficient rights to do the various ALE postings);
- Create RFC destination for each destination client; and
- Generate partner profiles for sending system. (Can only do this if at least 1 message type exists against the sending system's LS). This automatically generates the port if the LS and RFC name are the same.

The details of configuring the basis area are described in section 4.2.

#### 4.1.3 Functional configuration

Following the basis configuration is the functional configuration:

- Create a Customer Distribution Model (CDM);
- Add appropriate message types and filters to the CDM;
- Generate outbound partner profiles;
- Distribute the CDM to the receiving systems; and
- Generate inbound partner profiles on each of the clients.

Keep your ALE configuration diagram up to date and follow change management procedures rigorously, as you begin to configure. Areas you need to document well include the finer details involved with an ALE implementation:

- Listings / classes;
- Object types / Workflow tasks;
- Conversion rules;
- Filters & Segment filters;
- Change pointers / Fields activating change pointers;
- Serialization;
- Partner profiles: Immediate Vs Background;
- Background job definition and scheduling;
- Control data required to be in synch; and
- ...

See section 4-8 for the detail on how to configure each of these areas.

#### **4.1.4 QA & testing**

Demonstrate / workshop this solution with the business. Document and draw up functional specifications (done by the business) of any extensions to existing scenarios or additional scenarios required.

#### **4.1.5 Sign-Off**

Sign-off each client configuration with the business owners.

### **4.2 The HOW of the ALE Technical Configuration**

This section is intended for the use of the Basis Support Team, to enable them to configure the Basis side of the ALE solution.

#### **4.2.1 Create background ALE user**

An ALE **background** user **ALE-BATCH** needs to be created on **EVERY** client that needs to be ALE enabled. The ALE-BATCH user needs **SAP ALL** rights.

#### **4.2.2 ALE Administrator position**

The ALE administrator position needs to have been created, on each client, and the responsible persons linked to it. Please refer to the workflow procedures as to how to do this. These positions should then be documented on the respective Visio diagram for the particular client series currently being implemented.

#### **4.2.3 Set up logical system**

The distribution of systems makes it necessary to be able to identify every system individually within a network. The "logical system" is used to do this.

A logical system is an application system within which the applications are co-ordinated to work in one database. In the SAP sense of the word, a logical system corresponds to a client.

A logical system needs to be set up for each client that the system, on which you are working, needs to communicate to via ALE.

In the following steps, you must define every client as a logical system by first of all defining logical systems and then assigning the clients in question to the corresponding logical systems.

**Note:**

Assignments must be unique (that is, a client may only be assigned to one logical system).

Several clients must never be assigned to the same logical system.

*The same applies to pre-production systems and production systems: the pre-productive system must be assigned to a different logical system than the productive system.*

#### **4.2.3.1 Maintain logical systems (Client independent)**

1. In the IMG, choose *Cross-Application Components -> Distribution (ALE) -> Basic Configuration -> Set up logical system -> Maintain logical systems*
2. To create a logical system, choose *Edit -> New Entries*.
3. Enter a name for the logical system according to the ALE Naming Standards document
4. Enter a clear description for the logical system. If you want to change this short text for a logical system, please proceed as follows:
  - a) Select the appropriate line.
  - b) Choose *Edit -> Change field entries*.
5. Enter the new short text.
6. Choose *Replace*.
7. Save your entries.
8. Repeat steps 2-4 for the different clients.

#### **4.2.3.2 Allocate logical systems to the client**

This step only needs to be performed for the clients that reside on the box where you have just created the logical systems. For those logical systems that refer to clients that exist on other systems, the logical system will be linked to the client on its respective system. E.g., ALExxxCyyy will be linked to client yyy on system xxx.

1. In the IMG, choose *Cross-Application Components -> Distribution (ALE) -> Basic Configuration -> Set up logical system -> Allocate logical system to the client*
2. Select the relevant client.
3. Choose: *Goto -> Detail*. You branch into the detail screen.
4. In the field *Logical system*, specify the name of the logical system to which you want to assign the selected client, as per the naming conventions.
5. Save your entries.

#### **4.2.4 Maintain number range for ports**

The ALE interface generates ports automatically. The EDI interface assigns numbers internally for these ports. Hence the ports can be identified explicitly.

The system can only generate the numbers if a number range is entered for the number range object EDIPORT in number range 01.

Check first whether a number range in the number range 01 exists for the number range object EDIPORT. Proceed as follows:

#### **4.2.4.1 Procedure**

1. In the IMG, choose Cross-Application Components -> Distribution (ALE) -> Basic Configuration -> Maintain number range for ports
2. Perform function. The screen for number range object maintenance appears. The object name EDIPORT has already been pre-set.
3. Choose Goto -> Number ranges -> Interval -> Display. An interval must be entered for number range number 01. If no interval is entered, proceed as follows:
4. Go Back and choose Interval -> Maintain.
5. Enter a number range for number range number 01
6. Choose Interval -> Check to check whether your entry is error-free.
7. Save your entry.

#### **Notes on the transport**

You transport number range objects as follows:

In the initial screen, choose "Interval - Transport".

Note that all intervals for the selected number range object are deleted in the target system first. After the import, only the intervals you export are present. The number statuses are imported with their values at the time of export.

Dependent tables are not transported or converted.

#### **4.2.5 Maintain number range for IDocs**

Numbers are assigned internally for sent and received Intermediate Documents. This way the Intermediate Documents can be identified explicitly.

The system can only generate the numbers if a number range is entered for the number range object EDIDOC in number range 01.

#### **4.2.5.1 Procedure**

Check first whether a number range exists in number range 01 for the number range object EDIDOC. Proceed as follows:

1. Perform function for EDIDOC. The screen for number range object maintenance appears. The object name EDIDOC has already been pre-set.
2. Choose Goto -> Number ranges -> Interval -> Display. An interval should be entered for number range number 01. If no interval is entered, proceed as follows:
3. Go Back and choose Interval -> Maintain.
4. Enter a number range for number range number 01.
5. Choose Interval -> Check to check whether your entry is error-free.
6. Save your entry.

#### **Notes on the transport**

You transport number range objects as follows:

In the initial screen, choose "Interval - Transport".

Note that all intervals for the selected number range object are deleted in the target system first. After the import, only the intervals you export are present. The number statuses are imported with their values at the time of export.

Dependent tables are not transported or converted.

#### **4.2.6 Maintain number ranges for IDoc types and segment version**

Settings which are defined in the EDI area are to be maintained here.

##### **4.2.6.1 Number ranges for Intermediate Document types**

- If you made customer enhancements to Intermediate Document types and a new version of R/3 is imported, automatic merging of the new versions of the basic IDOC types (from SAP) and your customer enhancements is carried out.
- For this merging, the EDI interface assigns numbers internally for Intermediate Document types, for unique identification.
- The system can only generate the numbers if a number range is entered for the number range object EDIDDOCTYP in number range 01.

##### **4.2.6.2 Procedure**

Check first whether a number range interval in the number range 01 exists for the number range object EDIDDOCTYP. Proceed as follows:

1. In the IMG, choose Cross-Application Components -> Distribution (ALE) -> Basic Configuration -> Maintain number ranges for IDocs types and segment
2. Perform function.
3. The screen for number range object maintenance appears. The object name EDIDDOCTYP has already been pre-set.
4. Choose Goto -> Number ranges -> Interval -> Display. An interval must be entered for the number range 01.
5. If no interval has been entered, proceed as follows:
6. Go back and choose Interval -> Maintain.
7. Enter a number range interval for the number range 01.
8. Choose Interval -> Check to check whether your entry is error-free.
9. Save your entry.

##### **Notes on the transport**

You transport number range objects as follows:

In the initial screen, choose "Interval - Transport".

Note that all intervals for the selected number range object are deleted in the target system first. After the import, only the intervals you export are present. The number statuses are imported with their values at the time of export.

Dependent tables are not transported or converted.

#### **4.2.7 Maintain number range for change pointers**

In this section, the number ranges are maintained for the change pointers.

The change pointers are needed to generate Intermediate Documents from the application documents. For these pointers, internal numbers are assigned for unique identification. The system can only generate the numbers if a number range is maintained in number range 01.

#### **Example**

Number range	From number	To number
01	0000100000	1000000000

#### **Standard settings**

SAP delivers the number range for change pointers.

#### **4.2.7.1 Procedure**

If you must create number range 01, proceed as follows:

1. In the IMG, choose Cross-Application Components -> Distribution (ALE) -> Basic Configuration -> Maintain number range for change pointers
2. Perform function. The screen for number range maintenance appears. With Interval -> Display, you can see whether a number range was already maintained.
3. If no number range is maintained, choose Interval -> Maintain.
4. In the "Maintain Intervals" screen, choose Edit -> Insert interval.
5. Enter the number of interval 01 and a number range. Finish the entry with Insert.
6. Choose Interval -> Check to check whether your entry is error-free.
7. Save your entry.

#### **Notes on the transport**

You transport number range objects as follows:

In the initial screen, choose "Interval - Transport".

Note that all intervals for the selected number range object are deleted in the target system first. After the import, only the intervals you export are present. The number statuses are imported with their values at the time of export.

Dependent tables are not transported or converted.

#### **4.2.8 Set up ISO codes**

Currencies, units of measurement and country codes are held as international standardised ISO codes so that all messages sent can be understood on any system. This means that the ISO codes have to be assigned to the SAP-internal units as necessary.

**Note that a functional team usually performs this procedure.**

#### **4.2.8.1 Procedure**

Define the ISO codes for the units in use here.

Use the following IMG functions for this:

- In the IMG, choose Cross-Application Components -> Distribution (ALE) -> Basic Configuration -> Set up ISO codes

OR

- Currencies:

Global Settings -> Currencies -> Check currency codes

- Units of measurement:

Global Settings -> Check units of measurement

- Countries:

Global Settings -> Set countries -> Define countries

#### **4.2.9 Make basic settings for Workflow**

The workflow runtime environment has to be set up in each client. Workflow will be used for error handling in ALE. Technical errors are distinguished from application errors and each type of error (technical/functional) can be sent to a different person/role in the organisational structure.

The setting up of the workflow runtime environment will be covered in a procedure that explains how to set up workflow on SAP. When the actual document becomes available, it will be referenced from here.

#### **4.2.10 Deleting change pointers copied from source client**

When a client is created using a client copy, some open change pointers may be copied from the source client that is actually not needed in the target client. These “unwanted” change pointers can be deleted by running a program **RBDPCCLR**, or executing transaction **BD22**.

#### **4.2.11 Archiving IDocs copied from source client**

Because IDocs are client dependent, they get copied to the target client from the source client with a client copy. Whenever a client copy is done, these copied IDocs need to be archived and deleted to remove them from the newly created client.

Refer to the IDoc archiving strategy and procedure for the procedure to archive IDocs. When archiving and deleting IDocs in a newly copied client, all IDoc statuses must be enabled for archiving (to ensure that all IDocs are archived and deleted), refer to section 2.2 of the IDoc archiving strategy and procedure document (transaction **WE47**).

#### **4.2.12 Activating change pointers**

If you want to be able to distribute changes to master data, you must write change pointers at the same time as the change documents.

If you want change pointers to be written by the system, you must maintain a number range for them. This can be done in the section Number ranges -> Maintain number range for change pointer.

Once you’ve activated the change pointer for the particular message type then any changes made to this master data will have an IDoc automatically generated and forwarded according to the distribution model.

##### **4.2.12.1 Procedure**

In this section it is possible to activate the writing of change pointers. This activation must be done at two levels (general, per message type).

- Execute function in IMG: Cross-Application components -> Distribution (ALE) -> Distribution Scenarios -> Master data distribution -> Activate Change pointer -> Activate change pointer (generally).
- Check the box to activate.

#### **4.2.13 Define RFC destination**

In this section, you define the technical parameters for the RFC destinations.

This must be set up for all the logical destinations manually. RFC destinations are client independent.

The Remote Function Call is controlled via the parameters of the RFC destination.

The RFC destinations must be maintained in order to create an RFC port.



The name of the RFC destination should correspond to the name of the logical system in question. EG. ALExxxCyyy

#### **4.2.13.1 Procedure**

1. Execute **SM59**
2. Click on R/3 links and choose Edit -> Create;
3. The RFC destination name must be the same as the logical system name in order for the port to be automatically generated.
4. The type of RFC destination is 3.
5. Enter the required parameters dependent on the type.
  - For an R/3 link, that is, for example, the name of the RFC destination, the name of the partner machine, logon parameter ALE-BATCH with password init.
6. Select the Destination -> TRFC options function from the menu.
7. Enter the value 'X' into the 'Suppress backgr. job in case of comms. error' field.
8. Save and exit.

#### **Notes on the transport**

The maintenance of the RFC destination is not a part of the automatic transport and correction system. Therefore the setting has to be made manually on all systems.

#### **4.3 The HOW of the ALE Functional Configuration**

This section will describe the generic steps needed to configure the SAP system for ALE. An attempt will be made to explain the configuration steps in the order in which they must be executed as far as possible.

##### **4.3.1 Control Data**

Control data that needs to be in sync between the distributed systems needs to be configured. We will not be using the control data distribution provided by ALE due to its instability and various other issues. We will, instead, be using the transport and correction system to keep the control data in sync between the various clients.

Use transaction BDM5 - Consistency check (Transaction scenarios) and the control data section for your applicable ALE Message Type to determine what the applicable control data objects are.

##### **4.3.2 Reduction of IDocs (BD53) Client Independent**

Maximal intermediate document types for master data are supplied in the standard SAP System. The business would, in some cases, require that certain information not be distributed, and in these cases the respective IDoc would need to be reduced. As part of the reduction you can select the segments and fields of these intermediate document types which you wish to distribute. To do this you must activate the segments and fields that you require and generate a new message type, which satisfies your requirements.

Reduction is carried out for logical message types behind which an Intermediate Document type exists.

Mandatory segments and mandatory fields cannot be deactivated.

##### **4.3.2.1 Procedure**

1. Enter Transaction **SALE**
2. Execute the function: **Distribution Scenarios -> Master data distribution -> Reduce IDoc types for master data.**

3. To reduce a maximum message type, create a new logical message type. To do this, **enter a new name beginning with Z** followed by 5 meaningful characters that will allow the user to identify the initial IDoc. E.g.: MATMAS reduced to ZMATMS.

4. Enter the **reference message type** (for example, MATMAS).

You can use the following message types as reference message types for the various master data objects:

<u>Reference</u>	<u>Master data objects</u>
DEBMAS	Customer master
CREMAS	Vendor master
GLMAST	G/L account master
MATMAS	Material master

4. Enter a **short text** for the reduced message type.

5. The structure of the reference message type is copied to the new message type and displayed.

6. Double clicking on it can explode the structure.

7. Activating a segment:

- Place the **cursor on the segments** required.
- Choose **Edit -> Activate** in the menu.

8. Activating a field:

- Place the cursor on the active segment and click on '**Select**' push-button.
- **Select the fields** required in the relevant segment.
- Click on the **Activate** button.

9. After activating the required fields in all active segments, **save** the new reduced message type. Make a note of the number of the transport request used.

10. The reduced message type was created in a system independent of any particular client, but it cannot yet be used. Before the reduced message type can be used, it must be **activated** for each client in the initial screen of the reduction. Note also that the writing of change pointers is activated generally. This is done in the Activate change pointers generally section of this IMG.

### 4.3.3 Setting up global organisational units

*The functional expert in this area would typically perform this section.*

#### 4.3.3.1 Global company codes (OB72, OBV7 & OBB5)

Global company codes are set up and linked to organisational company codes. These global company codes can be seen across all organisational company codes and are used for example to set up filters for distribution of messages in the different systems.

They are simply an interface table used to minimise the impact of changes in systems not linked to yours. E.g., if Company X is referred to as company code X in your system, you give it a global company code name like GCX. Any system receiving IDocs from your system that refers to GCX can be mapped to their own company code. If they decide to change their company code it still remains the same on your system.

##### 4.3.3.1.1 Procedure:

1. Enter the **IMG: Distribution Scenarios -> Global Organisational Units -> Set Up Global Company Codes**.
2. Choose (by double clicking) the **Cross-system company codes** option.
3. From the new screen click on the **New Entries** button.
4. Enter the **relevant global company codes**.
5. Save the global company codes by clicking the **Save** button.
6. Go back to the previous screen by clicking on the **Back** button.
7. Choose (by double clicking) the **Assign company code -> chart of accts** option
8. Enter **CAEK** next to each global company and **save**. Choose **back**
9. Choose (by double clicking) the **Assign company code -> Cross-system company code** option.
10. Enter the **global company codes next** to the organisational company codes as described in the table.
11. Save the global company codes by clicking the **Save** button.

#### **4.3.3.2 Global business areas (OB93 & OBB7)**

Follow a similar procedure as the company codes described above for each of the business area except the chart of accounts section does not apply. Name the global business area the **SAME** as the business area. Ensure that it is transported to the group systems as well as the ALE reference client.

#### **4.3.4 Configuring the Distribution Model (BD64)**

The customer distribution model is the area where you link your message types to the logical systems. It is also the area where you enable the filters per message type.

##### **4.3.4.1 Procedure**

1. Log into the respective ALE reference client **4xx**, as this is the reference client where we maintain our customer model for that particular client series.
2. Perform **SALE**, and **Distribution Customer Model**
3. Perform the 'Maintain customer distribution model directly' function (**BD64**).
4. Specify the customer model you want to maintain and the logical system that is to be the sender of the messages.
5. Choose the receiving systems to which the sending system must forward message types.
6. For each receiving logical system allocate the message types necessary for communication to the receiving systems.
  - Click your cursor on the logical system where the message type is to be sent. Start with **ALEyyyC4xx** (where yyy is the system name).
  - Click on the **Create Message Type** button.
  - Type **the required message type** in the Log. Message type pull-down box.
  - Click on the **Transfer** button. The message type should then be added to the 4xx logical system.
  - Repeat the above procedure for all the required logical systems.
  - Save the model by clicking the **Save** button.
7. Create filter objects (vendor number, customer number, global company code, and so on) for the message types.

8. Save the entries.

- You cannot maintain a message type between the same sender and receiver in more than one customer distribution model.
- Except for the 'SUBSYSTEMS' customer model, which serves to model the local message flows for each customer distribution model as a system owner. Only the owner is authorised to modify the model.
- To transport the customer distribution model you should use the Distribute customer model function of the IMG.

#### 4.3.5 Changing the owner of the distribution model

##### 4.3.5.1 Procedure

1. Choose the 'Maintain ownership of customer distribution model' function.
2. Change the owner logical system to the required one for the required customer distribution model.
3. Make sure that all changes will be distributed to all systems that know the corresponding model. To do so, you can use the correction and transport system.

#### 4.3.6 Set up Filter Objects (BD64)

If filter objects are required then you need to set them up in the customer distribution model. To create a filter objects see the section on Object types.

##### 4.3.7 Procedure

1. Enter the **IMG -> Distribution Customer Model -> Maintain Customer Distribution Model Directly**.
2. Choose (by double clicking) the **Maintain message flow to subsystems** option.
3. Select **the sending system** as the Logical and **the required Customer model**.
4. Click on the Change Icon. After expanding all the logical systems to see the message types proceed as follows,
5. Click your cursor on the required message type in the logical system where the message type is to be sent
6. Click on the **Create Filter object** button.
7. Type **the required field or hit F4 for a lookup** in the Object type pull-down box and **the required value for that specific field** in the object box. (EG: BUKRS for company code is the object type and GC3000 is the global company code object)
8. Click on the **Transfer** button. The filter object and the value should then be added to the message type in the receiving logical system.
9. Repeat the above procedure for the other message types requiring filters for this receiving system.
10. Save the model by clicking the **Save** button.
11. Repeat the above procedure for all the other logical systems' models.

#### 4.3.8 Maintain sending system partner profile automatically (BD82)

With this function, you define the partner profiles for all outbound and inbound messages on the basis of the customer distribution model.

After you have defined and distributed the customer model, you will have to run this report locally in every system.

#### Requirements

- The customer model must be maintained.
- RFC destinations must be maintained.
- The customer model must be distributed.
- To ensure that the appropriate persons in charge are informed if a processing error occurs, you must make settings in: Error processing Maintain organisational units.

#### 4.3.8.1 Procedure

1. Log into sending system client
2. Enter the **IMG -> Cross-Application Components -> Distribution (ALE) -> Communication -> Generate Partner Profiles**. Execute the function (BD82)
3. Enter the required **customer model**
4. Enter the following for Receiver of notifications:
  - **S** as the type for position
  - **XXXXX**, where XXXXX is the number related to the ALE Administrator role (e.g. 50000085).
5. Enter the **output mode** (background, immediately) and the **package size** for outbound processing.  
**Recommendation:**
  - Output mode = Collect IDocs
  - Package size = 50
6. Enter the output mode (Immediate processing) for inbound parameters
7. Click on the **execute** icon.
8. If the profiles were not generated successfully, follow the steps provided in the guide to set up the respective areas that are highlighted as being faulty:
  - RFC destinations;
  - Generate sending partner profiles; and
  - Define logical systems (*ALE Technical Configuration Procedure*)

#### 4.3.9 Maintain sending system partner profile manually (WE20)

For ALE communication to take place, the partners must be defined with the inbound or outbound parameters.

##### Requirements

- The logical systems must be known to the system.
- A number range must be maintained for the ports. Select Basic configurations -> Maintain number range for ports.
- The receiving ports must be defined for outbound processing.

#### 4.3.9.1 Procedure

1. Execute the transaction **WE20 in the sending client**. Enter the number of the **logical system** as the partner number.
2. Use **LS** (logical system) as the partner type.
3. Choose the menu path Partner -> **Create**. If the partner already exists then choose the outbound parameter button.

4. Make entries in the Partner class and Partner status fields.
5. The inbound or outbound parameters must be created according to the direction in which communication takes place. Maintain outbound parameters for the respective partner system in the sending system and inbound parameters for the respective sender in the receiving system.
6. Inbound parameters:
  - Enter the **message type** appropriate to the scenario (see also Distribution scenarios).
  - Enter the **process code** for the message type.
  - Enter the **processing type**. We suggest **batch no override with express flag** unless volumes are extremely small in which case **immediately** can be used.
  - Enter the position of ALE Administrator in order to be informed in case of error.
7. Outbound parameters:
  - Enter the **message type** according to the scenario;
  - Enter the **receiving port** or create a new one;
  - Set **output mode** to **Transfer IDoc immediately** or **Collect IDocs**. See performance section of this document to decide on what setting to make. Default to 'immediately' until further investigation.
  - Set package size (number of Intermediate Documents processed at any one time). See performance section of this document to decide on what setting to make. Default to 1, but set to 50.
  - Enter the Intermediate Document type;
  - Enter the segment release in the IDoc type (As a rule, you only need this function when communicating with non-SAP systems and R/2 Systems)
  - Enter the position of ALE Administrator in order to be informed in case of error.
8. Store these settings.

#### Further notes

Output mode determines when the IDocs are dispatched in relation to when they were created. Set the output mode that best suits the scenario. Remember that collecting and sending IDocs in packets considerably reduces system load than immediately dispatching them. The dispatch of collected IDocs is done in batch processes. Select Periodic Work -> Schedule dispatch of IDocs.

#### 4.3.10 Distribute customer model (BD71)

Transport the customer model to all the relevant receiving systems.

##### 4.3.10.1 Procedure

1. Enter the IMG: **Distribution customer model -> Distribute customer model**. Execute the function. (BD71)
2. Double click on the **Distribute customer model** box.
3. Type **the required distribution model** into the Customer model box
4. Click on the **execute** icon.
5. The following screen should then indicate that the model data has been sent correctly to the relevant clients.
6. If the model was not sent successfully to some of the systems, follow the steps provided in the guide to set up the respective areas that are highlighted as being faulty:

- RFC destinations;
- Generate sending partner profiles; and
- Define logical systems (*ALE Technical Configuration Procedure*)

#### **4.3.11 Maintain receiving system partner profile automatically (BD82)**

Follow the same procedure as per Section: 4.3.8 on Page 4-11.

#### **4.3.12 Maintain receiving system partner profile manually (WE20)**

Follow the same procedure as per Section: 4.3.9 on Page 4-12.

#### **4.3.13 ALE Audit**

ALE Audit enables the sender system to monitor the processing status of dispatched IDocs in the receiver system. This section describes what ALE Audit is and how it is used.

ALE Audit enables the sender system to monitor the processing status of dispatched IDocs in the receiver system. The receiver system periodically sends confirmations back to the sender system, which are logged in IDoc status records and in individual audit tables.

The R/3 sender system has an analysis program for the audit database, which provides an overview of all dispatched messages and for any defective IDocs, details of current status, IDoc number and error message on receiver system.

Status records generated from the confirmations contain the following values:

- 39 - IDocs in receiver system (ALE Service) and still being processed
- 40 - Application document processing not completed successfully in receiver system.
- 41 - Application document processed in the receiver system. Processing completed independently of the processing status in the receiver system. The message in the status record is the same as the corresponding message in the IDoc status record in the receiver system.

##### **4.3.13.1 Dispatch of Audit confirmations**

For performance reasons confirmations are dispatched periodically for a packet of IDocs rather than for each one. Confirmations have a special IDoc ALEAUD01 with message type ALEAUD. The RBDSTATE report dispatches the audit confirmations. You can also start this report from the BALE transaction by selecting Periodic processing -> Schedule audit confirmations.

Confirmations can only be dispatched if a message flow for the message type ALEAUD has been maintained in the distribution model. (Follow the section Configuring the Distribution Model (BD64) to set up the ALEAUD message type with the relevant filters). The filter object MESTYP provides a more detailed specification. You can use the filter object to set the message types for generating Audit confirmations. As a rule, you do not have to send confirmations for all message types. For example, it is quite likely that confirmations will not be necessary for master data. If the MESTYP filter object is not used all IDocs are confirmed by ALEAUD in the receiver system.

##### **4.3.13.1.1 Parameters of RBDSTATE**

The selection parameters allow you to regulate in which systems and for which application message types that confirmation should be generated in the current report run. The report generates IDocs from the message type ALEAUD containing information about the processing state of inbound IDocs in your own system. An Audit IDoc contains confirmations for up to 500 IDocs. If there are more IDocs to be confirmed, several IDocs will be generated. The output consists of a list of generated IDocs. If an IDoc is not generated or an error occurs an appropriate message is displayed.

The report selects IDocs whose statuses have recently changed. Because almost every IDoc action (e.g. creation, successful posting/ unsuccessful posting in an application) alters the status of the IDoc, it is

these IDocs which have in some way or other been processed. You can specify the repeat interval for these selections with the last selection parameter. If the report is to be executed online then the parameter is mandatory. This is not the case with batch processing where there are two processing modes:

- Daily or Less Frequent Dispatch of Confirmations

This processing mode is preferred for non-time critical confirmations. You can dispatch confirmations for IDocs generated the previous day or for changes that go back even further. When you maintain the variants for the batch processing, choose a date selection variable (i.e. a date variable) for the date change selection parameter. You can flexibly set the days (e.g. yesterday) from which IDoc status changes should be confirmed.

When scheduling the batch job (e.g. changes made the previous day), make sure you do not run the job exactly at 00.00 hours. Run it at least a few minutes later, otherwise the previous day's IDocs, which were not posted by 00.00 hours, will not be confirmed.

- More Frequent Dispatch of Confirmations

This processing mode is used for time critical confirmations, e.g. those which should be done every hour. Leave the date change selection parameter empty, as the report verifies the date and time of the last run, or in the case of a first run, the date and time it was scheduled. The time less 5 minutes is the left selection limit for the IDoc changes. The batch job start time less 4 minutes is the right selection limit for the IDoc changes.

#### **4.3.13.2 Procedure to schedule audit confirmations**

##### **4.3.13.2.1 Define variant**

You create variants for the RBDSTATE report in this section. You can specify the systems and message types to which Audit confirmations should be made. The variants are always created on the receiver system of the original message.

##### **Example**

An ALE Audit confirmation is required for the 'ORDERS' message, which is sent from System A to System B. This means that a variant of the RBDSTATE report with logical sender system 'A' and message type 'ORDERS' must be created in System B.

##### **4.3.13.2.1.1 Procedure**

1. You can maintain various values for your variants:
  - logical sender system,
  - message type,
  - message variant,
  - message function,
  - Creation dates of the Intermediate Documents.
2. Create appropriate variants for the RBDSTATE report.
3. To do this, execute the function, enter "RBDSTATE", click on variants and choose Change.
4. Enter a name for the variant and choose Create.
5. Maintain the attributes for the variant.
  - The logical sending system is the system to which Audit confirmation should be sent.



- Message type, message variant and message function specifies the messages for which there should be confirmations.
  - If the fields are not maintained in the variant, then the system uses the possible settings from the customer distribution model.
  - The creation date specifies the selection time range. If nothing is specified here, then all messages since the last time the report was executed will be covered.
6. Save the variant.
  7. Create as many variants as needed to take account of all message flows for which ALE Audit should be performed.

#### 4.3.13.2.2 Schedule job

##### Requirements

The variants of the RBDSTATE report, which are to be used, must have been created.

##### 4.3.13.2.2.1 Procedure

1. Define a job, as per naming standards, with the **RBDSTATE** step and a **variant you have created**.
2. Schedule the job as a **periodic** job (daily).
3. **Save** the job.
4. Create a job for **each of the previously defined variants**.

##### Further notes

If possible arrange for there to be a small time buffer between the end of the selection period of the variant and the time at which the job is executed. This should help prevent the queue of Intermediate Documents not yet processed from overflowing.

#### 4.3.13.3 Analysis of Audit database

The report RBDAUD01 evaluates the Audit database. To initialise it, select Monitoring -> Audit analysis from the ALE menu path. To get to the ALE menu from the R/3 initial screen select Logistics -> Central functions -> Distribution. In this transaction you can check the processing status in your own system or in the partner systems. Details about all IDocs being processed are shown.

##### 4.3.13.3.1 Overview of All Statistics

Here you can see an overview of all selected Audit statistics. Statistics are recorded by receiver system, message type, and IDoc creation date. The column Last IDoc shows whether all the current day's IDocs are already included in the statistics. They will be included if the value of the entry is "24:00:00". If an earlier time is indicated then only those IDocs created up until this time will be included in the statistics database. The column IDocs total contains the total number of IDocs in the statistics database. The column Queue Outbound contains the number of IDocs still being processed in your own system, i.e. they have not yet been dispatched. The column Queue Inbound contains the number of IDocs received by the target system by the time of the last confirmation but still being processed. If you double click on a statistic a status overview is displayed.

##### 4.3.13.3.2 Overview of Individual Statistics

All IDocs for the selected statistic are grouped by status and listed separately for outbound and inbound IDocs are listed in status groups. If you double click on a status value you will get an IDoc list for this status value. The IDoc list only shows status values that are not yet finalised, so it does not show, for

example, status 53 - "application document posted". This is because ALE Audit only has details of IDocs that are still being processed

#### 4.3.13.3.3 IDoc List for a Status Value

All of the IDocs for a given status value of a statistic are displayed. If there are any linked application objects these are also shown. If the IDocs are already in the receiver system the IDoc number of the receiver system is also displayed. By double clicking on a row you can display the corresponding local IDoc.

#### 4.3.13.3.4 IDoc Display

This displays individual IDocs that are also used in the IDoc monitor. You can call up all the information that exists in your own system for this IDoc. Status records with values 39 and 40 indicate error messages in the receiver system.

#### 4.3.13.3.5 ALE Audit IDocs

The Audit IDoc ALEAUD01 contains three segments: E1ADHDR, ESTATE and E1PRTOB.

One IDoc contains a maximum of 500 IDocs confirmations. The segments comprise the following:

- **Segment E1ADHDR**

E1ADHDR contains message type, variants and function for the ensuing confirmations. Each confirmation comprises exactly one segment E1STATE and one segment E1PRTOB.

- **Segment E1STATE**

E1STATE contains the IDoc number of the sender system (Field DOCNUM), the current status in the receiver system and the message fields from the current status record. In a remote system the DOCNUM value depends on how the IDocs were previously dispatched to the R/3 System. The sending system IDoc number is the number assigned when the INBOUND\_IDOC\_PROCESS in field EDI\_DC-DOCNUM was called. The R/3 System saves this number unless it is the initial value.

**Note:**

The DOCNUM field must contain a valid R/3 IDoc number. If there is not an IDoc for the given number this entry is ignored. The Status field must contain a valid R/3 status value for inbound IDocs.

- **Segment E1PRTOB**

E1PRTOB contains the receiver's IDoc number and, if existing, the application object generated in inbound processing. The application object is in the format of the BOR link tools. I.e. it comprises a BOR object type, Object ID and logical owner system.

#### 4.3.14 Conversion rules

There are various methods of field conversion:

- Allocation of constants (=> SET),
- Allocation of ranges in the sender field for target values in the receiver field (=> GROUP),
- Combination of various sender fields for target values in one receiver field (=> COMBINE),
- Specification for all non-defined GROUP or COMBINE cases (=> SET\_R).
- If no conversion rules have been defined for a given field then the default action is to copy the contents of the sender field into the receiver field (=> MOVE).

**NOTE**

- When referring to company codes or business areas in the rules you must specify the **global equivalent** of the company code or business area.

This section defines the conversion rules.

This entails the following steps being performed:

1. Rule definition:

The rules are always defined per segment.

2. Rule maintenance:

Rule maintenance specifies the conversion on field level with defined rules.

3. Allocating rules to IDoc:

The allocation specifies when the rule should be applied. This is sender, receiver and message type specific.

#### **4.3.14.1 Define rules (BD62)**

The first work step for the conversion rule is processed in this section.

Here you define a conversion rule for each segment type.

##### **Example**

Definition of a conversion rule E1MARRG for the segment type E1MARAM.

##### **4.3.14.1.1 Procedure**

1. Execute the transaction **BD62**.
2. In the menu, choose Edit -> **New entries**.
3. Enter a **name** for the conversion rule according to the naming standards.
4. Enter the **segment type** to which the conversion rule is to apply. To determine what segments are in what IDocs use the IDoc documentation transaction WE60.
5. **Save** the entries.

#### **4.3.14.2 Maintain rules (BD79)**

The second work step for the conversion rule is processed in this section. The conversion rules are specified a per field basis.

This involves changing the contents of a segment field (receiver field) in a way that depends on the contents of segment fields (sender fields). The sender field and the receiver field may be the same.

##### **Requirements**

The conversion rule must be defined in the first step.

##### **Example**

You define a specific field conversion for the E1MARRG conversion rule or assign a constant to a field.

##### **Example "SET"**

- In the listing of all segment fields
- You can allocate a constant to a field. That can be a value or, for example, "/" to prevent a receiver field being overwritten in batch input.

##### **4.3.14.2.1 Procedure "SET"**

1. Enter the name of the conversion rule in the input field.
2. Click on the field for which you want to define a constant.
3. Select the menu option Edit -> Assign constant.
4. Enter the constant and save your entry.

#### **Example "GROUP"**

- You can assign a specific value for the recipient field in the case of a specific value or value interval of the sender field.
- That can, for example, be the conversion of a plant from the R/2 System (from 2 characters) to the R/3 format (plant 01 then becomes 0001).

#### **4.3.14.2.2 Procedure "GROUP"**

1. Enter the name of the conversion rule in the input field.
2. Double click on the recipient field for which a rule is to be defined.
3. In the following screen define:

- The conversion rule GROUP;
- The name of the sender field;

This can be the same field as the receiver segment or a different one.

- The offset;

Here you define the place in the field you want the system to read from. Therefore, in a 6-digit field (positions 0 to 5), it is possible to analyse the values from position 2. If nothing is entered here, the system reads from position zero.

- The length;

Here you define the length of the field part to be analysed. If nothing is entered here, the length is taken up to the end of the field.

- The field "constant" is of no significance to the GROUP rule. It is required for SET\_R.

4. Now double-click on the "rule" line.

In the "characteristic" field on the screen that appears, you can enter the target value in the receiver field. In our example, this would be "0001" for the plant.

5. Double-click on the "characteristic" line

6. Here you enter a value or an interval that the sender field can have so that the receiver field receives the characteristic value. In our example, this would be the value of the sender plant "01".

7. You can define several rules, but they must not overlap.

8. You can also define an "Otherwise" case with the conversion rule SET\_R.

9. Save your maintenance.

#### **Example "COMBINE"**

- For a particular value or value interval of various sender fields, you can assign a particular value for the receiver field.
- An example is the conversion of plants with storage locations, which have a different description in the receiver system than in the sender system.

- Thus, storage locations 0001 to 0009 of plant 0001 in the sender system can be converted to storage location 0100 in plant 0001 in the receiver system.

**Note**

- You must define rules for each field that do not overlap.
- In our example, for instance, it is not possible to convert storage locations of a second plant to the same target storage location 0100.

#### 4.3.14.2.3 Procedure "COMBINE"

1. Enter the name of the conversion rule in the entry field.
2. Double-click on the receiver field, for which the rule is to be defined.
3. On the next screen, define:
  - The conversion rule COMBINE;
  - The names of the sender fields;
  - The offset and the length;

Read the explanation for this in the explanation of the GROUP rule.

- The "constant" field is of no significance to the COMBINE rule. It is required for the "SET\_R" rule.
4. Now double-click on the 1st "rule" line. In the example, this would be the plant field.
  5. In the "characteristic" field on the screen that appears, enter the target value in the receiver field. In our example, this would be "1000" for the plant.
  6. Now double-click on the "characteristic" line
  7. Specify a value or interval here that the sender field can have so that the receiver field receives the characteristic value. In our example, this would be the value of the sender plant "0001".
  8. Go back to the screen with the rules.
  9. Now double-click on the 2nd "rule" line. In the example, this would be the storage location field.
  10. In the "characteristic" field on the screen that appears, enter the target value in the receiver field. In our example, this would be "0100" for the storage location.
  11. Now double-click on the "characteristic" line and enter the sender value or interval. In the example, this would be the lower limit "0001" and the upper limit "0009".
  12. Perform this function repeatedly until all the rules are defined.
  13. The rules must not overlap.
  14. You can also define an "Otherwise" case with the conversion rule SET\_R.
  15. Save your maintenance.

#### **Example SET\_R**

- If you determined a GROUP or a COMBINE, a constant can be defined for all values, which do not fall into the value interval.
- Consider the GROUP example: you could for example define that the plant "0002" should be specified for all values other than "01".

#### **4.3.14.3 Procedure SET\_R**

1. Enter the name of the conversion rule in the entry field.

2. Double-click on the field for which GROUP or COMBINE was already defined in order to complete the SET\_R rule.
3. Assign the field and the constant.
4. Save your work.

#### 4.3.15 Customer extensions

For customer extensions, there are two customer exits for segment conversion.

The exits are contained in the extension KKCD0001. Once activated, the customer exits are called for all segment conversions. It is recommended that you code a CASE statement within a customer exit, in which various cases can be accessed depending on the conversion rule (REPID).

The customer exits can be found in module pool 'SAPFKCIM'. If you want to use these customer exits, you can create a project and activate the customer exits with transaction 'CMOD'. The extension you must use for this is called KKCD0001.

#### Notes on the transport

If you want to transport the rule that you have maintained, then select the following menu function: Rules - > Transport and specify the number of the request to be used.

#### 4.3.16 Defining settings for IDoc conversion (BD55)

The third work step for the conversion rule is processed in this section.

The conversion rules are defined for a message type. The allocation of the conversion rule to segment type is carried out for each partner system.

#### Example

For the message types MATMAS, you allocate, for example, the conversion rule E1MARRG for the segment type E1MARAM with a sender and receiver system.

#### Requirements

The rules must be created and maintained with the first two steps.

##### 4.3.16.1 Procedure

1. Execute the transaction **BD55**. Enter the **message type** that is to be allocated to the rule.
2. Enter the **sender and receiver system with partner type**. The partner type is LS for almost all scenarios.
3. Enter the **segment type** and the **conversion rule** defined for it.
4. **Repeat steps two and three** for all further rules.
5. **Save** your allocations.

#### 4.3.17 Setting up workflow

The workflow runtime environment has to be set up in each client before ALE will be able to utilise workflow for error notifications. The setting up of the workflow runtime environment will be detailed in a workflow set-up procedure.

ALE can use workflow to notify users/positions in the organisation structure about errors. These errors can be put into two categories: technical- and application errors. Currently both these types of errors will be monitored by the ALE administrator position in the organisation, but it is possible to notify different positions of technical- and application errors.

The following procedures will explain how to set up ALE so that the position/user gets notified of firstly technical- and secondly application errors.

#### **4.3.17.1 Setting up technical error notification (WE20)**

When manually maintaining partner profiles (outbound and inbound), there is a section of fields in each partner profile called “receiver of notifications”. These fields pertain to the specific partner profile and who will receive notifications for the specific partner profile. In the case of this configuration, these fields in the specific partner profiles will be left blank.

In the general partner profile, a receiver of notifications can also be specified. This general receiver of notifications will then be used to send error notifications to, if the specific partner profile is left blank. This general partner profile specifies the attributes for every partnership between two partners, as opposed to the specific partner profiles that specify attributes for a specific message type.

To enter the receiver of notifications at the general partner profile level, enter transaction WE20, select a partner and click on the change button. Here the category (user, position etc.), language and ID of the user/position that has to receive notifications of technical errors can be specified.

This user/position will then be notified via the workflow inbox of any technical ALE layer errors for the specific partner profile details (message type, partner etc.).

#### **4.3.17.2 Setting up application error notification (PFTC)**

ALE sets up links between process codes and inbound function modules. These links are used (in the partner profile where the process code is specified) to determine how to process each IDoc that is received on the inbound side. The process code / inbound function module links can be viewed through transaction BD67. For each link an IDoc object type and events are specified (can be viewed by selecting the line and clicking on the view button). The IDoc object type and start event should be noted down for all process codes that application error notifications are needed.

The next step is to run transaction SWU0. The IDoc object type and start event should be filled in on this screen and the function executed. On the event simulation screen, a workflow task will be shown under the “potential tasks to be started” (if a task is available). This is the workflow task that will be started if an application error occurs for this specific IDoc object on inbound processing. This task needs to be noted down.

Lastly this task needs to be activated and the correct user/position for error notifications needs to be linked to this task. This can be done through transaction PFTC. On this screen the task type (TS) and the task number noted down in the previous step needs to be entered on the screen. Next click on the display button to display the task’s general properties. From the menu select additional data -> agent assignment -> maintain. If the task is specified as a general task, select the task and click on the general task... button. Deselect the general task check box, and press enter. Now select the task and click on the create button (or choose agent assignment -> create from the menu structure). Choose position and select the ALE administrator position from the organisational structure.

To activate the task, go back one screen level from the previous step. Click on the triggering events button and select goto -> event linkage from the menu structure. Scroll to the right of the screen and double click on “deactivated”, this will activate the task and “activated should now be displayed on the screen.

This procedure has now activated and the task and assigned a position that will receive error notifications.

#### **4.3.18 Listings**

Listings are used in master data management when distribution rules cannot be modelled using organisational units only. Listings exist for material, customer and vendor master data.

Cost centers and G/L accounts do not have a classification and it is therefore not possible to use listings for them. Since the quantities of data in these cases are relatively small, compared to, say, the material master, it is no problem to distribute all the data to all the systems if a more refined distribution cannot be modelled using organisational units.

Listings can be used to define master data objects that a system is interested in.

For each object type a (customer-defined) class type can be specified as ALE-relevant. All classes belonging to this class type can be used in the listings.

Since listings are used as classifications, they are maintained in the master data transactions. A master data object can be added to a list by classifying it.

The listing groups together all the objects belonging to a single class, and this class belongs to the ALE class type for the object.

Listings are maintained within the classification. To do this, the following steps have to be taken:

- For an object (for example, the table MARA for material), you must create a class type, marked as a distribution class type.
- You define the listings for the distribution class type.
- Afterwards, the listings (with class type) are allocated for the logical systems.
- The distribution via listings must be recorded in the customer distribution model.

#### **4.3.18.1 Maintain class code (O02D)**

In this section, you define class types as a requirement for the listings.

Either an available class type can be used or a new one created for each object (for example, material).

##### **4.3.18.1.1 Procedure**

1. Perform transaction O02D. A list of the objects defined by SAP is displayed.
2. For the master data, you need the following objects:

<b>Master data</b>	<b>Object</b>
Material	MARA
Vendor	LFA1

1. By double clicking on the object, you open up the suboptions for the object.
2. Go to the structure item Class Types and create a new class type. If the class already exists, return to the menu and choose "Display/change class type parameters" or use transaction O02E. Proceed to step 4.
3. Enter a description. (MAT for material & use 010 for Vendor)
4. Characterise it as a distribution class type. For each object, one class type can be indicated as distributable.
5. Set the 'Multiple Classification allowed' indicator.

**NB!** The creation or modification of class types requires "Repository Access" which should be arranged with the Basis/Environment Management team.

#### **Further notes**

If you have allocated listings, which belong to a distribution class, to a logical system, you cannot transfer the distribution indicator to another class type, since otherwise inconsistencies occur in the configurations.

Class types, which can be distributed, can be defined for all objects available in the R/3 classification. If such a class type is created, you can create listings for distribution of this object.



#### 4.3.18.2 Maintain status for class type (O02F)

A status has to be maintained for each distribution class type, so that classes can be created for the class type.

The status controls:

- whether the class can be maintained
- whether objects can be assigned to the class
- whether a search can be performed for objects in this class

#### Requirements

The distribution class type must already have been created.

##### 4.3.18.2.1 Procedure

1. Select the object type for the class type:

<b>MARA</b>	<b>Material</b>
<b>LFA1</b>	<b>Vendors</b>

2. Select the **distribution class type** for which you want to create the status.

3. Create the status 1.

4. Enter a **text**.

5. Select the following options for this status:

- Maintenance is allowed for the class (Cl.Maintce.).
- The classification of objects is allowed (Classif.).
- Selections can be made for objects in the class (Sel. alwd.).

6. Make this status the default status (Default).

#### Example

The following status has been defined for MARA, class type MAT:

Status	Text	Cl.Maintce	Classif	Sel. alwd.	Default
<b>1</b>	<b>Released</b>	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>

The following statuses are defined for LFA1, class type 010:

Status	Text	Cl.Maintce	Classif	Sel. alwd.	Default
<b>0</b>	<b>In Preparation</b>	<b>X</b>			
<b>1</b>	<b>Released</b>	<b>X</b>		<b>X</b>	<b>X</b>
<b>2</b>	<b>Locked</b>	<b>X</b>			<b>X</b>

#### Further notes

You may also define several statuses for a single class type (see example above).

#### 4.3.18.3 Maintain classification status (O02I)

In this step, you can define classification statuses. The classification status is used to determine whether an allocation can be used to find objects. Classification statuses are maintained separately for each class type.

The statuses are assigned when you classify an object in a class.

Use the classification status to define:

- Whether the allocation is released. This status is set if all characteristics with the required entry indicator have correct values assigned to them.
- Whether the allocation is locked. This allows you to determine, for example, whether allocations are shown in the search result.
- Whether the allocation is incomplete. If, for whatever reason, you could not assign values to all the characteristics with the required entry indicator when classifying an object, you can still save the classification data by setting "incomplete" status.
- Whether the allocation was recognised by the system as incomplete. This status is set automatically by the system if you allocate an object to a class, but do not assign values to it, although the class contains required characteristics.

In the find objects function, you can restrict the search result according to classification status. You can use the View -> Scope function to decide whether locked allocations and incomplete allocations are displayed in the search result.

### **Requirements**

The class type must exist in the system.

#### **4.3.18.3.1 Procedure**

Create classification statuses according to your requirements.

You can also copy classification statuses to create new ones. You can copy from the classification statuses of any class type.

To do this, proceed as follows:

1. Choose step Maintain classification statuses -> Copy classification status.
2. Select the object type you require.
3. Select the class type whose classification statuses you want to copy.
4. In the dialog box, enter the target class type for the classification statuses. You can display the possible entries for the class type and select one.
5. Select Continue to edit your copy.

### **Note**

- For each status, you can only set one parameter.
- You cannot use a parameter in more than one status - only in exactly one status.
- There must be a status for which the parameter "incomplete" is set.

#### **4.3.18.4 Maintain lists (CL01)**

After you created the class types, the listings are defined in this section. In the R/3 System, every listing is regarded as a class.

### **Requirements**

- The logical systems must have been made.
- The class type for distribution must be maintained.

#### **4.3.18.4.1 Procedure**

1. Perform transaction **CL01**. Choose the **distribution class type** and enter a **name for the listing** (Class).
2. In the detail screen, enter:
  - a listing **description**,
  - The **class status**, which describes whether a class may be managed.
  - A **class group**, if an allocation to a certain group is to be made.
  - You must define this group beforehand.
3. **Save** your settings.

#### **Notes on the transport**

Since listings cannot be distributed, you will have to create listings (classes) on all systems. It is not possible to use the transport system.

#### **4.3.18.5 Allocate listings to a logical system (BD68)**

In this section you assign the listings that are to be used for the distribution of master data to the combination of message type and logical recipient system.

##### **Requirements**

- The logical systems must have been defined.
- The class type for distribution must be maintained.
- The listings must be defined.

##### **4.3.18.5.1 Procedure**

1. Execute the transaction **BD68** and enter the **logical system** that is to receive the listings.
2. In the menu, choose **Edit -> New entries**.
3. Enter the **message type** to be distributed.
4. Enter the **listing** (class) for this message type.
5. Set the Push/Pull value to **'2'** (Push).
6. **Save** your entries.

##### **Further notes**

The allocation of the master data to the listings is made in the ALE task level menu or in master data maintenance.

#### **4.3.18.6 Include lists in the distribution model (BD64)**

In this section the dispatch is entered into the distribution model with listings. Make these entries in the PC Tool.

##### **Requirements**

Each listing must be assigned to a logical system.

##### **4.3.18.6.1 Procedure**

1. Maintain the Customer distribution model as described in the section earlier.
2. Add the **object type** 'LISTING' (filter) for the appropriate message and select the **object value** 'YES' for the system connection that is to use a listing.

#### 4.3.19 Schedule active monitoring

Active monitoring is usually scheduled as a periodic background job. The system determines whether the number of IDocs in a status group exceeds a particular threshold. If this is the case, a message is sent to the integrated inbox of the specified receiver.

During selection, IDocs, which meet the selection criteria, are analysed. The possible status values of the IDocs are assigned to status groups. If the current status value of an IDoc is assigned to one of the status groups passed to the report via parameters, the IDoc is regarded as critical. If the number of critical IDocs exceeds the specified value, the system will regard this as a problem situation and sends a message to the specified Workflow receiver.

You can schedule report RSEIDOCM to carry out the active monitoring periodically.

##### 4.3.19.1 Define variant (SE38) Client Independent

In this section, variants are defined for report RSEIDOCM.

You must make valid entries for the receiver, receiver type, status group and number of critical IDocs fields for the selection run to work. You can also limit the IDoc selection further using additional parameters.

###### 4.3.19.1.1 Procedure

1. Execute the transaction **SE38** and enter "**RSEIDOCM**". Click on **Variants** and choose **Change**
2. Enter a **name** for the variant and choose **Create**
3. Fill in the **input fields** as required
4. Choose the **Continue** button (or F6)
5. **Maintain** the variant attributes
6. **Save** the variant

###### 4.3.19.1.2 Schedule Job (SM36) Client Independent

In this step, the job for the active consistency check is scheduled.

##### Requirements

Variants for report RSEIDOCM must exist.

###### 4.3.19.1.3 Procedure

1. Define a job containing the step **RSEIDOCM** and your **own variant**.
2. Schedule the job to run **periodically**, specifying a period which suits your needs (ex. hourly, daily etc.)
3. **Save** the job.
4. **Create** a job for **each defined variant**.

#### 4.3.20 Change pointers

The following 2 steps need to be performed to get activate a change pointer for a particular message type.

##### 4.3.20.1 Activate generally (BD61)

1. Execute the transaction **BD61**
2. **Check** the active box and **save**

#### 4.3.20.2 Activate for message type (BD50)

1. Enter the **IMG: Distribution Scenarios -> Master data distribution -> Activate change pointer -> Activate change pointer for message types**
2. Activate the change pointers for the message types you need by **checking the active box**
3. **Save** your entries.

So that the master data changes can be distributed, a report must be scheduled. This report reads the change pointers and generates IDocs from it.

To do this, the following steps are necessary:

- Define variants for the report,
- Schedule jobs, each with the report and a variant as the step.

#### 4.3.20.3 Define variant (SE38) Client Independent

1. Create a variant for the **RBDMIDOC** report for every message type (for example, for material master, customer master and so on).
2. To do this, execute the function and enter "**RBDMIDOC**", click on **variants** and choose **Change**.
3. Enter a **name** for the variant and choose **Create**.
4. Enter the **message** type (for example, MATMAS).
5. **Maintain** the attributes for the variant.
6. **Save** the variant.
7. Repeat the process until variants have been created for all master data message types.

#### 4.3.20.4 Schedule Job (SM36) Client Independent

If, for example, the material master is to be distributed, a job must be planned for the message type (for example, MATMAS), which periodically (for example, once a day) checks the changes and generates Intermediate Documents.

1. Define a job with the step **RBDMIDOC** and the **variant maintained by you**.
2. Schedule the job as a **periodic** job, where the period must be adapted to your requirements (for example, hourly, daily, and so on).
3. **Save** the job.
4. Create a job for every master data message type.

#### 4.3.21 Mass processing

Mass processing refers to bundles of IDoc packets, which are dispatched and processed by the receiving R/3 System. Only one RFC call is needed to transfer several IDocs. Performance is considerably better when transferring optimal packet sizes.

To define a mass processing parameter, select Communication -> Manual maintenance of partner profile -> Maintain partner profile. For a message type the parameter packet size and output mode can be defined.

If the output mode is set to "Collect IDocs", outbound IDocs of the same message type and receiver are sent in a scheduled background job or in the BALE transaction in appropriately sized IDoc packets. The IDocs can be dispatched in batch or in the BALE transaction code.

Some distribution scenarios cannot support mass processing of inbound IDoc packets. This is especially true if the application sending the IDocs uses the ABAP/4 command CALL TRANSACTION USING. In this case the outbound parameter PACKETSZ must be set to "1".

To get a list of function modules that can be mass processed, select Enhancements -> Inbound -> specify inbound module in ALE Customising. INPUTTYP is "0".

#### 4.3.22 Schedule IDoc dispatch

Some IDocs are not sent immediately, but are collected and then sent all together. The report RSEOUT00 must be scheduled in order that these IDocs can be sent.

The following steps are necessary for this:

- Define variants for the report.
- Schedule jobs, each with the report and a variant as the step.

#### Further notes

Numbers are assigned internally for IDocs that have been sent. Number ranges have to be maintained for this purpose. You do this under menu entry Basic settings -> Maintain number ranges for IDocs.

##### 4.3.22.1 Define variant (SE38) Client Independent

In this section, variants are defined for the report RSEOUT00.

###### 4.3.22.1.1 Procedure

1. You can maintain different values for sending:

- port,
- partner type, partner role
- partner number,
- Message type.

2. SAP recommends variants with partner number and message type.

3. Create corresponding variants for the report **RSEOUT00**.

4. To do this, execute the transaction **SE38** and enter "**RSEOUT00**", click on **Variants** and choose **Change**.

5. Enter a **name** for the variant and choose **Create**.

6. Enter the **message type** (for example, MATMAS).

7. **Maintain** the attributes for the variant.

8. **Save** the variant.

9. **Repeat the process** until variants have been created for all message types.

##### 4.3.22.2 Schedule Job (SM36) Client Independent

The jobs for each message type are scheduled in this step.

#### Requirements

Variants for each message types must be maintained.

###### 4.3.22.2.1 Procedure

1. **Define a job** with the step **RSEOUT00** and a **variant** that you have maintained.

2. Schedule the job as a **periodic** job, where the period must be adapted to your requirements (for example, hourly, daily, and so on).

3. **Save** the job.
4. **Create a job for every variant** defined beforehand.

#### **4.3.23 Schedule IDoc processing**

Jobs for inbound processing are scheduled in this section. These jobs must be scheduled for the following situations:

- Processing the IDocs that are not imported immediately on arrival, but periodically in the background (settings in partner profile). Use Report RBDAPP01.
- Importing IDocs that are stored in the system, but which were not transferred to the application due to exceptional situations. (Report RBDMANIN). With selected parameters you can also schedule a job to collect IDocs that have not yet been processed, for instance, because of a locking problem.

To do this, the following settings are necessary:

- Define variants for a report.
- Schedule jobs, with the report and a variant as the step.

#### **Further notes**

The report RBDMANIN allows intermediate documents which have already been cancelled once with errors (e.g. Status: 51 error in transfer to application) to be read into the application again. It is also possible to schedule a job with selected parameters, in order to collect up intermediate documents which could not be processed, e.g. because of a locking problem.

Numbers are assigned internally for intermediate documents that have been sent. Number ranges have to be maintained for this purpose. You do this under menu entry Number ranges -> Maintain number ranges for intermediate documents.

#### **4.3.23.1 Define variant (SE38) Client Independent**

In this step you create variants for report RBDAPP01 or for report RBDMANIN.

##### **4.3.23.1.1 Procedure**

1. Define variants with the following values or a subset:
  - logical message type
  - message variant
  - message function
  - partner type of the sender
  - partner number of the sender
  - partner role of the sender
2. SAP recommends the definition of variants with the logical message type and the partner number. However, which IDocs are imported and how frequently depends on your requirements.
3. For all partner and message types for which no immediate processing is carried out you must create a variant of the report RBDAPP01.
4. SAP recommends that you define a variant without values in order to enter the remaining IDocs.

#### **4.3.23.2 Schedule Job (SM36) Client Independent**

##### **Requirements**

Variants must be maintained for each message type.

#### 4.3.23.2.1 Procedure

1. **Define a job** with the step **RBDAPP01** or **RBDMANIN** and a variant maintained by you.
2. Schedule the job as a **periodic** job where the period must be adapted to your requirements (for example, hourly, daily, and so on).
3. **Save** the job.
4. **Create a job** for every variant defined beforehand.
5. Create a job with the variant **without values** to import the IDocs that were not transferred to the application due to exceptional situations.
6. This job must have a period that is **larger than the maximum period of the other jobs**.

#### 4.3.24 Schedule Status conversion after successful RFC

When outbound IDocs have been successfully passed to the communication layer, they are assigned the status "data passed to port OK".

This status does not, however, indicate that the communication via a transactional RFC was successful. Therefore a report should regularly be started that checks whether the communication was successfully completed and if so, changes the status of the IDoc.

To schedule the RBDMOIND report, the following must be done:

- Define variants for the job,
- Schedule job, with the report and one variant in one step.

##### 4.3.24.1 Define variant (SE38) Client Independent

In this step, you create a variant for report RBDMOIND.

#### 4.3.24.1.1 Procedure

1. Perform the transaction **SE38** and enter "**RBDMOIND**"; Click on **Variants** and choose **Change**
2. Enter a **name** for the variant and choose **Create**
3. **Delete the date** in the field '**IDoc creation date (from)**'
4. Specify a value in the field '**IDocs per Commit Work**'
5. (SAP recommends the number 100)
6. **Maintain** the attributes for the variant
7. **Save** the variant

##### 4.3.24.2 Schedule Job (SM36) Client Independent

In this step, you schedule the job for changing the status of intermediate documents that have been successfully dispatched.

#### Requirements

A variant for report RBDMOIND must be maintained.

#### 4.3.24.2.1 Procedure

1. Define a **job** with the step **RBDMOIND** and the **variant you have maintained**.
2. Schedule the job as a **periodic** job; adapt the period to meet your requirements (e.g. hourly, daily etc.).



3. **Save** the job.

#### **4.3.25 Serialisation (BD57) Client Independent**

With maintenance of the serialisation types, the system can check whether overtaking occurred during arrival of the IDoc types. This way, it is recognised if an older purchase order change arrives before a current change arrives, for example.

##### **4.3.25.1 Procedure**

1. Execute transaction **BD57**
2. When making an assignment to a new message type, you should use the switch new entries.
3. For each of your **new message types** you should enter an ALE object type as a **link type** and also as a **serialisation type**.
4. If you change the assignment, then you should overwrite the entries for the corresponding message type.
5. **Save** your entries

#### **4.3.26 Object types**

An ALE object type specifies the field of an IDoc in which an object, for example, plant, is stored. An ALE object type specifies the field of an IDoc in which an object, for example, plant, is stored.

The ALE objects are used to create links between IDocs and applications objects, to control the serialisation, to filter messages in the customer model and to use listings.

##### **4.3.26.1 Maintain object types (BD59) Client Independent**

You must maintain these object types if you have developed your own message types and/or you're own IDoc types or if you want to define new ALE objects for existing message types.

The assignment is carried out for each message type for segments and fields of an IDoc type.

This assignment defines those segments and fields of an intermediate document from which information must be read for the object types.

##### **Notes**

- A custom object type must start with Z. we suggest you complete it with the field that you wish to use. E.g. ZBUKRS

##### **4.3.26.2 Assign object types (Links and serialisation) (BD57) Client Independent**

If you have developed your own message types and/or IDocs then you must maintain **object types for the links**.

If you want to check the serialisation for a message type, then you must maintain **object types for the serialisation**. If no serialisation object has been maintained for a given message type, then the serialisation will not be checked for this message type.

##### **4.3.26.3 Allocate application object to a logical message Client Independent**

The system automatically creates links between IDocs and the applications objects that are connected with the message.

If you have created a new logical message type, then you have to tell the system which application object type the new message type should be linked with. (E.g. The logical message type MATMAS is linked to the application object type BUS1001 (material).)

#### 4.3.27 Segment Filtering (BD56)

In this section, you can determine IDoc type segments that are not to be transferred for each message type and receiver.

##### 4.3.27.1 Settings for filtering

###### Requirements

The partner systems must be defined.

###### 4.3.27.1.1 Procedure

1. Perform function **BD56**. Enter the **message type**.
2. Generate new entries for the respective **partner system**.
3. **Save** your work.

###### Further notes

- You cannot filter any required segments out.

You must ensure that all segments, which are in the hierarchy under a segment, are also filtered out if you filter the "parent segment".

#### 4.3.28 Transaction scenarios (BDM5)

*The functional expert in this area would typically do this section.*

If the ALE scenario is considered to be a transaction scenario then the consistency check for that scenario needs to be run against the respective message type. All errors in this consistency check needs to be resolved functionally.

###### Requirements

The message types for the transaction need to have been maintained in the customer distribution model.

##### 4.3.28.1 Procedure

1. Run transaction **BDM5**
2. Click on consistency of application
3. Double click on the require message type for the transaction scenario you need to set up
4. Correct the red lines indicating errors.

#### 4.4 Deliverables

##### 4.4.1 ALE Configuration Sign-Off Procedure

Agree on the process to follow during QA and Production phases of the project. During production, transactions cannot be posted to test the correctness of the solution...

##### 4.4.2 ALE Configuration Procedure

Details *how* to perform the configuration. (See section: 4.2)

This section should also include an **ALE configuration check sheet**. This indicates the various steps that need to be completed when performing a configuration on a new set up. An example would look as follows:

ALE Config Checksheet									
Series: _____		Date: _____		By: _____		Reviewed By: _____			
		Client System							
		400		500					
		600		700					
		800		900					
		1000		1100					
		1200		1300					
		1400		1500					
		1600		1700					
		1800		1900					
		2000		2100					
		2200		2300					
		2400		2500					
		2600		2700					
		2800		2900					
		3000		3100					
		3200		3300					
		3400		3500					
		3600		3700					
		3800		3900					
		4000		4100					
		4200		4300					
		4400		4500					
		4600		4700					
		4800		4900					
		5000		5100					
		5200		5300					
		5400		5500					
		5600		5700					
		5800		5900					
		6000		6100					
		6200		6300					
		6400		6500					
		6600		6700					
		6800		6900					
		7000		7100					
		7200		7300					
		7400		7500					
		7600		7700					
		7800		7900					
		8000		8100					
		8200		8300					
		8400		8500					
		8600		8700					
		8800		8900					
		9000		9100					
		9200		9300					
		9400		9500					
		9600		9700					
		9800		9900					
		10000		10100					
		10200		10300					
		10400		10500					
		10600		10700					
		10800		10900					
		11000		11100					
		11200		11300					
		11400		11500					
		11600		11700					
		11800		11900					
		12000		12100					
		12200		12300					
		12400		12500					
		12600		12700					
		12800		12900					
		13000		13100					
		13200		13300					
		13400		13500					
		13600		13700					
		13800		13900					
		14000		14100					
		14200		14300					
		14400		14500					
		14600		14700					
		14800		14900					
		15000		15100					
		15200		15300					
		15400		15500					
		15600		15700					
		15800		15900					
		16000		16100					
		16200		16300					
		16400		16500					
		16600		16700					
		16800		16900					
		17000		17100					
		17200		17300					
		17400		17500					
		17600		17700					
		17800		17900					
		18000		18100					
		18200		18300					
		18400		18500					
		18600		18700					
		18800		18900					
		19000		19100					
		19200		19300					
		19400		19500					
		19600		19700					
		19800		19900					
		20000		20100					
		20200		20300					
		20400		20500					
		20600		20700					
		20800		20900					
		21000		21100					
		21200		21300					
		21400		21500					
		21600		21700					
		21800		21900					
		22000		22100					
		22200		22300					
		22400		22500					
		22600		22700					
		22800		22900					
		23000		23100					
		23200		23300					
		23400		23500					
		23600		23700					
		23800		23900					
		24000		24100					
		24200		24300					
		24400		24500					
		24600		24700					
		24800		24900					
		25000		25100					
		25200		25300					
		25400		25500					
		25600		25700					
		25800		25900					
		26000		26100					
		26200		26300					
		26400		26500					
		26600		26700					
		26800		26900					
		27000		27100					
		27200		27300					
		27400		27500					
		27600		27700					
		27800		27900					
		28000		28100					
		28200		28300					
		28400		28500					
		28600		28700					
		28800		28900					
		29000		29100					
		29200		29300					
		29400		29500					
		29600		29700					
		29800		29900					
		30000		30100					
		30200		30300					
		30400		30500					
		30600		30700					
		30800		30900					
		31000		31100					
		31200		31300					
		31400		31500					
		31600		31700					
		31800		31900					
		32000		32100					
		32200		32300					
		32400		32500					
		32600		32700					
		32800		32900					
		33000		33100					
		33200		33300					
		33400		33500					
		33600		33700					
		33800		33900					
		34000		34100					
		34200		34300					
		34400		34500					
		34600		34700					
		34800		34900					
		35000		35100					
		35200		35300					
		35400		35500					
		35600		35700					
		35800		35900					
		36000		36100					
		36200		36300					
		36400		36500					
		36600		36700					
		36800		36900					
		37000		37100					
		37200		37300					
		37400		37500					
		37600		37700					
		37800		37900					
		38000		38100					
		38200		38300					
		38400		38500					
		38600		38700					
		38800		38900					
		39000		39100					
		39200		39300					
		39400		39500					
		39600		39700					
		39800		39900					
		40000		40100					
		40200		40300					
		40400		40500					
		40600		40700					
		40800		40900					
		41000		41100					
		41200		41300					
		41400		41500					
		41600		41700					
		41800		41900					
		42000		42100					
		42200		42300					
		42400		42500					
		42600		42700					
		42800		42900					
		43000		43100					
		43200		43300					
		43400		43500					

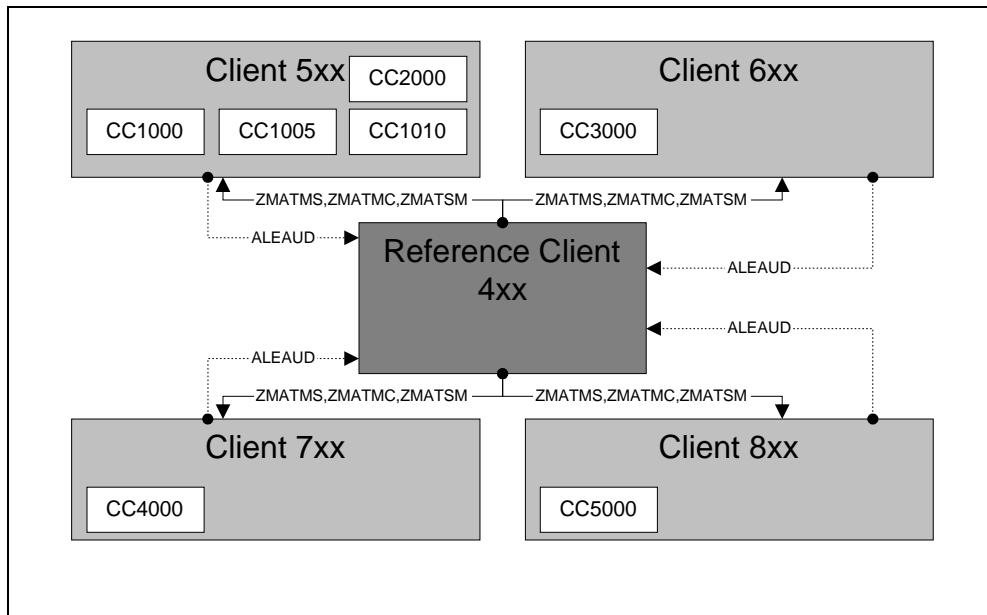


Figure 7: Material Master Message Flow Diagram

## 2. IDoc reduction

The Material master solution requires the reduction of the MATMAS Message Type into 3 separate message Types, ZMATMS, ZMATMC and ZMATSM as follows:

### **ZMATMS**

This Message Type is used to distribute new Material master items to the group clients, 5xx to 8xx, and contains the following fields:

#### **E1MARAM Segment:**

- Material Number
- Creation Date
- Name of user who created the object
- Date of last change
- Name of user who changed record
- Maintenance status
- Material Type
- Industry Sector
- Material Group
- Old Material Number
- Base Unit of Measure

#### **E1MAKTM Segment:**

- Language Key
- Material Short Description

**E1MTXHM Segment:**

- Function
- Texts: application object
- Text Name
- Text Id
- Language Key
- SAPScript: Format of a text

**E1MTXLM Segment:**

- Function
- Tag Column
- Text Line

**ZMATMC**

This Message Type is used to distribute the changes to Material master items to the group clients, 5xx to 8xx, and contains the following fields:

**E1MARAM Segment:**

- Material Number
- Creation Date
- Name of user who created the object
- Date of last change
- Name of user who changed record
- Maintenance status
- Material Type
- Industry Sector
- Material Group
- Old Material Number
- Base Unit of Measure (deactivate)

**E1MAKTM Segment:**

- Language Key
- Material Short Description

**ZMATSM**

This Message Type is used to distribute the Material \ services used in the internal charges solution, to the group clients, 5xx to 8xx, and contains the following fields:

**E1MARAM Segment:**

- Function
- Material Number
- Creation Date
- Name of user who created the object

- Date of last change
- Name of user who changed record
- Maintenance status
- Deletion flag for all material data
- Material Type
- Industry Sector
- Material Group
- Base Unit of Measure

**E1MAKTM Segment:**

- Language Key
- Material Description

**E1MARCM Segment:**

- Plant
- Delete flag
- Purchasing Group

**E1MVKEM Segment:**

- Sales Organisation
- Distribution Channel
- Delete flag for material data of Sales Org. / Distr. Channel
- Material Statistics Group
- Item category group
- Delivering Plant
- Account Assignment group for this material

**E1MLANM Segment:**

- Departure Country
- Tax Category 1
- Tax Classification 1
- Tax Category 2
- Tax Classification 2

**E1MTXHM Segment:**

- All fields in the long text header segment.

**E1MTXLM Segment:**

- All fields in the long text line segment.

### 3. Filter objects

#### ZMATMS

Client: 5xx	Client: 6xx	Client: 7xx	Client : 8xx
<b>LISTING</b> YES <b>GS</b>	<b>LISTING</b> YES <b>S</b>	<b>LISTING</b> YES <b>S</b>	<b>LISTING</b> YES <b>S</b>

#### ZMATMC

Client: 5xx	Client: 6xx	Client: 7xx	Client : 8xx
<b>LISTING</b> YES <b>GS</b>	<b>LISTING</b> YES <b>S</b>	<b>LISTING</b> YES <b>S</b>	<b>LISTING</b> YES <b>S</b>

#### ZMATSM

Client: 5xx	Client: 6xx	Client: 7xx	Client : 8xx
<b>MATKL</b> INTCHARGE	<b>MATKL</b> INTCHARGE	<b>MATKL</b> INTCHARGE	<b>MATKL</b> INTCHARGE
<b>VKORG</b> 1000	<b>VKORG</b> 3000	<b>VKORG</b> 4000	<b>VKORG</b> 5000
<b>WERKS</b> 1130	<b>WERKS</b> 3110	<b>WERKS</b> 4600	<b>WERKS</b> 5150
<b>WERKS</b> 1140	<b>WERKS</b> 3120		<b>WERKS</b> 5630
<b>WERKS</b> 2600	<b>WERKS</b> 3560		

#### 4. Partner profiles

##### Sending system

- **ZMATMS & ZMATMC**

The following settings need to be set in the outbound partner profile for ZMATMS and ZMATMC:

- Collect IDocs
- Packet size: 50

##### Receiving systems

- **ZMATMS & ZMATMC**

The following settings need to be set in the inbound partner profile for ZMATMS and ZMATMC:

- Process in the background

#### 5. Conversion rules

No conversion rules apply to Material Master.

#### 6. Classes / Listings

##### ZMATMS & ZMATMC

Listings are to be activated for Message Types ZMATMS and ZMATMC as follows:

Class	Value	Logical System
MAT	Group 1	ALEyyyC5xx
MAT	Group 2	ALEyyyC6xx
MAT	Group 3	ALEyyyC7xx
MAT	Group 4	ALEyyyC8xx

#### 7. Change Pointers

- Activate: ZMATMC

- Activate: ZMATMS

## 8. Fields activating change pointers

### ZMATMS

MARA	MATKL
------	-------

### ZMATMC

DMAKT	KEY
DMAKT	MAKTX
DMAKT	SPRAS
MARA	AENAM
MARA	BISMT
MARA	KEY
MARA	LAEDA
MARA	MATNR

## 9. ALE Audit

ALEAUD is to be enabled for each of the group clients, 5xx to 8xx as follows:

- Message Type: ALEAUD; Filter Object: MESTYP; Filter Object value: ZMATMS

## 10. Jobs

The following jobs need to be set up for of the clients 4xx to 8xx. As described.

### Outbound

The following jobs are scheduled on the ALE reference client, 4xx.

- Z\_ALE\_ZMATMS\_OUT\_4xx:
  - Step 1: RBDMIDOC; Variant: Z\_ZMATMS (All IDocs created this month of type ZMATMS)
  - Step 2: RSEOUT00; Variant: Z\_ZMATMS (All IDocs of type ZMATMS)
  - Step 3: External program, **Name**: /bin/sleep; **parameter**: 60; **Target host**: system name in which the job is being scheduled.
  - Step 4: If Maestro is NOT to be used: Raise event ZA145xx (Where xx is client series where the event must be raised); **Name**: /bin/remsh; **Parameter**: recmach -l zzzadm -n ./sapevt ZA145xx -p ZMATMS pf=./profile; **Target host**: Machine you are running job on. (zzz: target host name; recmach: Receiving machine name)
  - Step 5 - 7: As above for each client.
  - Periodically: 5 minutes, Job class: B

### Inbound

The following job is to be scheduled on each of the group clients, 5xx to 8xx.

- Z\_ALE\_ZMATMS\_IN\_xxx:
  - Step 1: RBDAPP01; Variant: Z\_ZMATMS (All IDocs of type ZMATMS)
  - Periodically: Raised on event ZA14xxx (As above)



#### **4.4.4 ALE Basis Configuration Procedure**

Details how to perform the basis configuration for ALE. Used by the Basis team.

#### **4.4.5 ALE Workflow Configuration Procedure**

Details how to configure the workflow error message handling for ALE.

#### **4.5 Other issues**

Before you can commence with the ALE configuration the following issues need to be resolved.

##### **4.5.1 Which scenarios?**

An investigation needs to have been completed on the business requirements and these then need to be mapped to the existing ALE scenarios. If the requirement is there AND the scenario exists an ALE solution could then be offered to the business. If it entails the development of an ALE solution then the costs of the development need to be investigated.

##### **4.5.2 Control Data Implications**

Check that the control data that is required to be in sync between the systems, for each of the ALE scenarios, is actually in sync. I.E. ALE requires that certain control data be the same for both the sending and receiving clients.

##### **4.5.3 Other Master Data Implications**

Certain ALE scenarios, when implemented, imply that other scenarios need to be implemented as well. E.g. cost and profit centers.

##### **4.5.4 Distribution Strategy**

Different distribution strategies are supported:

1. One option is to maintain all data centrally and then to distribute it.
2. Another option is to create the master data locally. There is then always one maintenance system for each view. After the master data has been maintained it is then transferred to a central management system and is distributed from there as required.

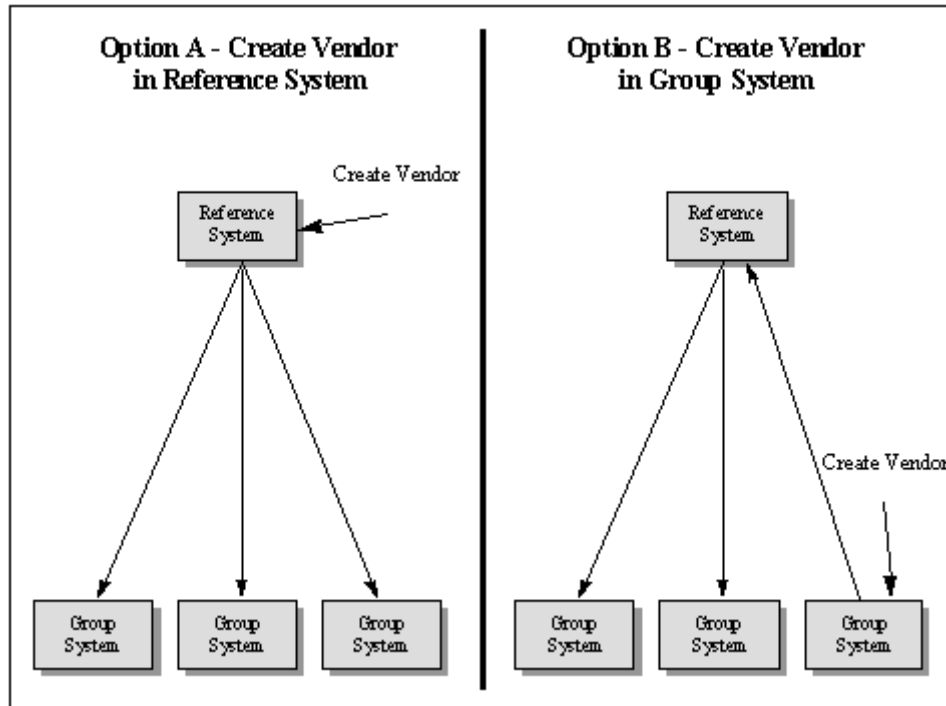


Figure 8: Distribution Strategy Options

A decision has to be made as to what strategy should be adopted for each of the required scenarios.

**NOTE:** Some ALE scenarios make use of direct table updates to input the data on the receiving side, as opposed to making use of std SAP transactions. This implies that no change pointers will be generated for Option B, make it quite unfeasible. It may also imply inconsistencies across clients. E.g., cost centers can be deleted, yet there are transactions posted against them.

#### 4.5.5 Information Distributed

The business needs to indicate clearly what data is required to be distributed and what data is not. The ALE administrator needs to ensure that all the fields that are required to be distributed are available for distribution in the applicable IDoc for that ALE scenario.

If this is not the case then an extension is required for that ALE scenario. (For any ALE development or extension please follow the ALE Scenario Development Procedure)

PS: This is not a trivial exercise. Allow for at least 2 weeks to complete. See development section for the detail.

#### 4.5.6 Field conversion

Do any of the fields need to assume different values when being distributed? This needs to be clearly documented by the user.

If so the ALE administrator needs to identify where the applicable rule would be most suited, on the inbound or the outbound side. He also needs to be aware of the conversion rule functionality to ensure that the requirement is within the bounds of the functionality provided for.

#### 4.5.7 Conditions of distribution

Does the required data need to go to a certain client based on certain information contained in the IDoc? If so, filters need to be defined and documented. The users need to specify this requirement in detail.

The ALE administrator needs to ensure that that particular field is available, as a filter object, for the ALE scenario in question, otherwise he will need to add it.

#### **4.5.8 Transfer Initiation Strategy**

2 options are available for certain of the ALE scenarios:

- Automatic creation of IDocs through change pointers; and
- Manual creation of IDocs done by the user.

The ALE administrator needs to check that change pointers exist for the required ALE scenario, as all of them are not available. If a change pointer is not available then one needs to be developed (Not documented anywhere - Would require SAP specialist help)

#### **4.5.9 Mass processing**

The number of IDocs that are to be created influences the settings made in the partner profiles for the various message types. The users need to give a clear indication of the volumes, frequency and timings of these bulk creations.

Various jobs need to be scheduled to process IDocs collected on the outbound side as well as to process them in the background on the inbound side. These settings will be defined in a separate document detailing the ALE performance tuning options and settings.

#### **4.5.10 Serialization**

Is there a need to implement this feature? Does it matter if an IDoc that was created after 1 IDoc is processed on the receiving side before the first one? If so, the ALE administrator needs to implement this feature. Areas where this may be important is for the transaction scenarios, where the ORDERS message must get to the receiving system prior to an ORDCHG coming through for that same order.

#### **4.5.11 ALE performance**

IDocs are fundamental to Application Link Enabling; optimal performance in ALE processing is synonymous with optimal performance of IDocs.

In ALE processing IDocs are:

- created in the R/3 sender system
- transferred to the communication layer in the R/3 sender system
- Used when R/3 sender and receiver systems communicate.
- updated in the receiver system

You can optimize IDoc handling in one or more of the following ways:

- Controlling IDoc Events
- Processing IDocs in Parallel
- Packet Processing
- Managing IDoc Communication

These settings will be defined in a separate document detailing the ALE performance tuning options and settings. They are also documented in the Detailed ALE Configuration Guide.

#### **4.5.12 Error notifications**

Who is going to receive the error notifications of each of the scenarios, if at all? By default we suggest the ALE administrator, but if an applicable person, in the business, has the required skills and is willing to assume the role, the functional errors can be split and forwarded to that person.

The ALE administrator would then need to set up the applicable tasks and positions, as well as perform the required configuration to enable it.

If it is decided not to implement the workflow error notification functionality the ALE administrator would need to continuously monitor WE05 for errors on each of the applicable clients.

2 Neat ways of implementing more proactive fault reporting would be to implement either of the following:

- Implement SAP workflow integrated to your MS Mailbox. This will allow all your error messages to be routed to one central mailbox. This stops the ALE administrator from having to log into every client.
- Redevelop the inbound ALEAUD function module to extract errors and either send an SAP mail message or initiate an external fault reporting system with the relevant details.

Configuration of ALE is a relatively simple task but the implications of doing it incorrectly will have a vast impact on a production system. It therefore needs careful attention to detail and well trained and competent ALE support staff available to correct any issues that may arise.

*CHAPTER 5*  
**THE ALE DEVELOPMENT PHASE**

## 5 ALE DEVELOPMENT

This section contains all generic documents related to ALE custom scenario development.

### 5.1 Project team tasks

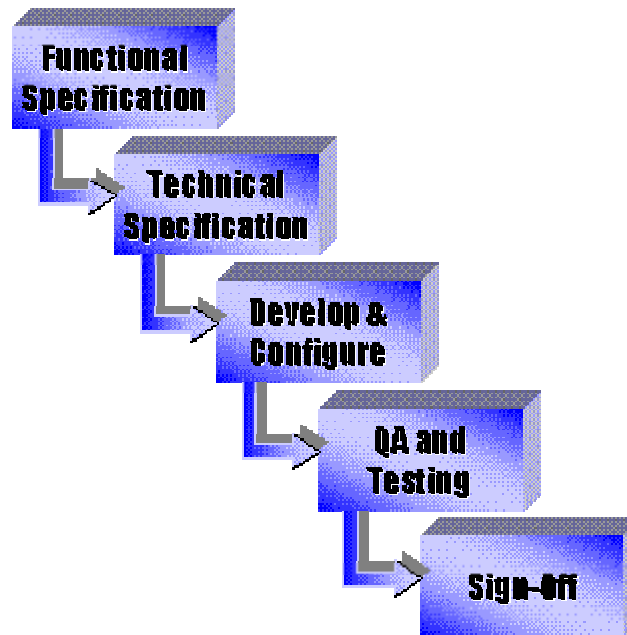


Figure 9: ALE project team tasks during ALE Development phase

#### 5.1.1 Functional specifications

Functional specifications need to be drawn up, by the business, describing exactly what is required by them, that is not available in the standard ALE scenarios.

#### 5.1.2 Technical specifications

A technical specification can then be drawn up by the ALE team. In these technical specifications the following needs to be addressed:

- New IDocs required;
- New fields added to existing IDocs;
- Program / user exits / message control for populating the fields;
- Inbound functional module details;
- ALE configuration set up requirements; and
- Error handling.

Once the specification is completed and confirmed by the business, programming can begin.

#### 5.1.3 Development

Developing an ALE scenario will include the following steps:

##### OUTBOUND

- Create Segments and IDoc type;
- Create Message Type and link to IDoc type;
- Populate IDoc using message control \ user exit or program (ABAP);
- Distribute IDoc using MASTER\_IDOC\_DISTRIBUTE;
- Create object type if required; and

- Define CDM with message type and filter object. Distribute model. Generate outbound partner profiles;

#### INBOUND

- Write inbound function module (FM) to process inbound IDoc (ABAP);
- Create process code and link to FM;
- Generate inbound partner profiles; and
- Link object type to FM for error handling.

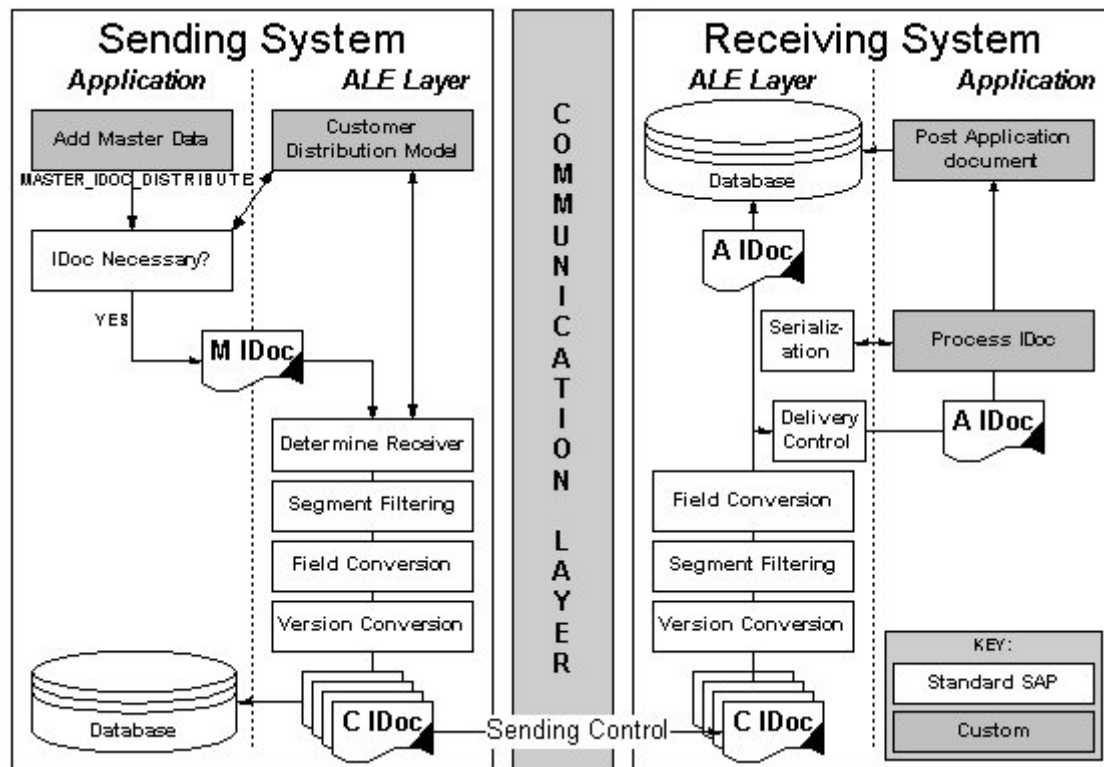


Figure 10: ALE Development components pictorial

#### 5.1.4 Quality assurance and testing

QA & Testing - Including unit and string testing. Ensure a quality solution. QA done by the ALE team leader and testing done by the business owner and ALE scenario developer.

#### 5.1.5 Business sign-off

Sign-off by the business - Ensure the development is thoroughly tested and comprehensively documented prior to sign-off against the functional specification.

### 5.2 Deliverables during this phase

#### 5.2.1 ALE Functional Specification

For each development a detailed functional requirement document is produced and confirmed by the business.

#### 5.2.2 ALE Development Guide

Details what development was done and how to configure the solution. Needs to be quality assured by someone else in the know. An example would be as follows:

### 5.2.2.1 Introduction to ALE development

To develop a new custom ALE scenario comprises 5 steps:

1. Design and develop the custom IDoc with it's segments and a new message type
2. Configure the ALE environment with the new IDoc and message type (customer model, partner profiles and linking IDoc to message type)
3. Develop the outbound process which does the following:
  - Populates the custom IDoc with control info and functional data
  - Sends the IDoc to the ALE layer for distribution
  - Updates status and handles errors
4. Configure the ALE inbound side (partner profiles with inbound process code)
5. Develop the inbound process which does the following:
  - Reads the IDoc into a BDC table; selects other data that is required
  - Runs transaction using call transaction or BDC session
  - Updates status and handles errors

Below is a pictorial representation of the flow of a complete ALE scenario from the sending system to the receiving system.

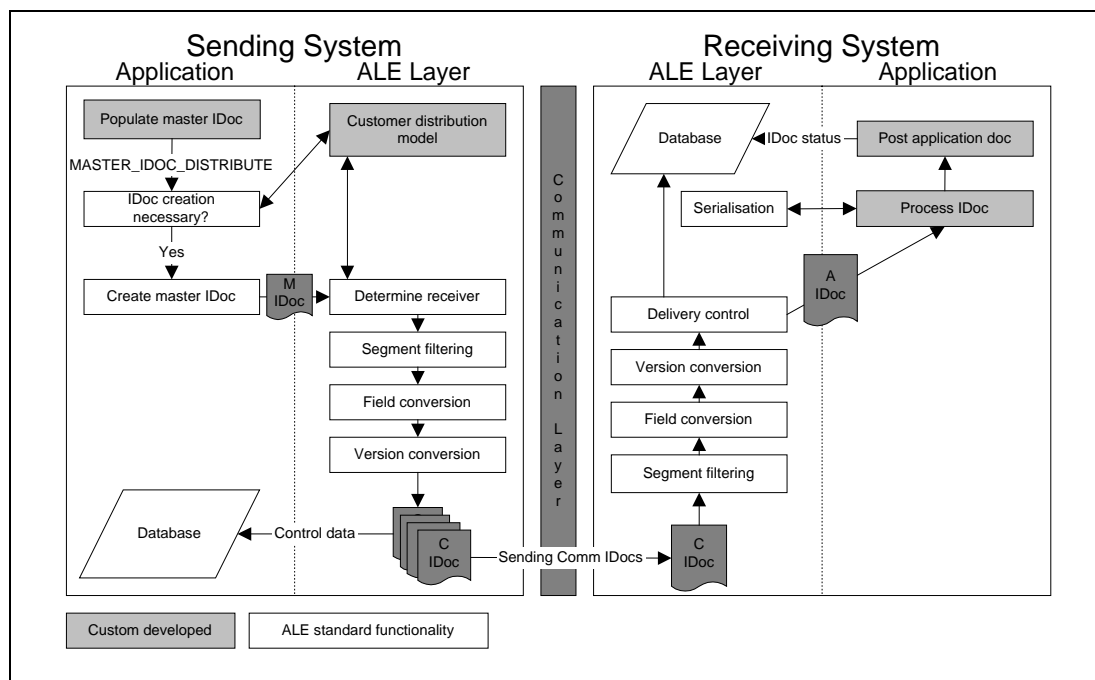


Figure 11: ALE Scenario model

### 5.2.2.2 ALE Example

For the purposes of this example we will develop a small ALE scenario. This scenario is described below.

*“The receiver of an internal service must be able to reverse (cancel) the invoice receipt which will then cancel the applicable billing document automatically on the service provider’s system.”*



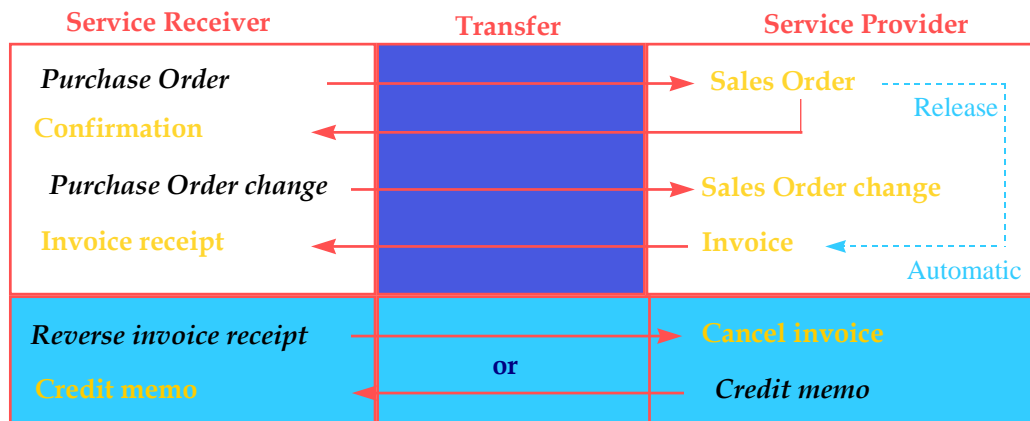


Figure 12: Example Purchasing & Selling scenario

We will develop a custom IDoc to carry the billing number from the Service Receiver's system to the Service Provider's system. We will populate the IDoc in a user exit on the sending side and we will process the transaction on the receiving side using a custom function module and a BDC transaction call.

No rule conversion, segment filtering or version conversion will be implemented in the model as described in Figure 11.

#### Requirements

- Working ALE environment - See ALE Basis Configuration Section;
- ALE scenario design together with the business requirement;
- Development access; and
- ALE configuration access.

#### NOTES:

1. All IMG references to transactions are located in the transaction **SALE** which is the ALE portion of the IMG
2. This is one way of developing a scenario where no message control exists. If message control exist (EG. On purchase orders) then NAST can be used to call an outbound function module that would create the required IDocs.
3. Extensive knowledge of IDocs and ALE basis configuration is required in order to understand this guide.

#### 5.2.2.3 Outbound processing

##### 5.2.2.3.1 Create IDoc type (WE30) Client independent

The IDoc type refers to the IDoc structure that you will require for your development. In our case the IDoc type is called **ZINVRV01**. This IDoc type will have 1 segment called **Z1INVRV** with 2 fields, **LIFNR** & **XBLNR**, in this segment. If you require many segments or nested segments then they are also created using the same procedure.

We will create the IDoc of the following structure:

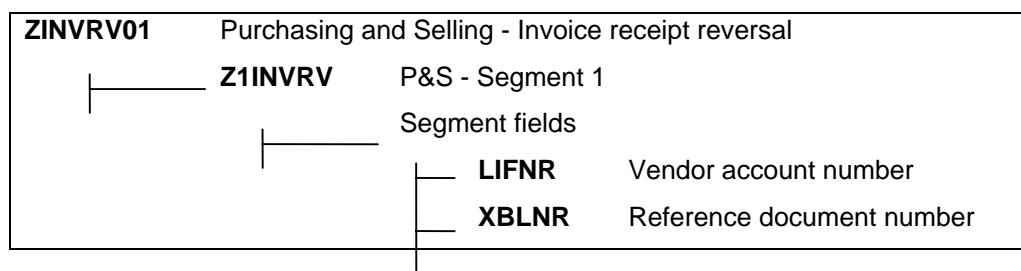


Figure 13: IDoc type ZINVRV01

To create the IDoc type, follow these next few steps:

- Enter transaction **WE30** (ALE -> Extensions -> IDoc types -> Maintain IDoc type)
- Type in **ZINVRV01** and click on **Basic IDoc type**, click the **Create** icon
- Click on **Create new** (we are creating an IDoc from scratch but you may want to copy another IDoc if it is similar to your requirements) and enter a **description**, and press **enter**
- Click on **ZINVRV01** and then on the **Create** icon
- Enter **Z1INVRV** as the segment type (must start with Z1), check mandatory if the segment must exist (in this case check it), enter **1** in minimum number and **1** as maximum number. (Make the maximum number 9999999999 if there are going to be many of these segments in each IDoc. IE. When line items are passed via IDocs), click on **Segment editor**
- Enter a **description** for your segment type and **create**
- Enter a **description** for your segment, enter each field required in your IDoc, in our case type **LIFNR** across for Field name, DE structure and DE documentation, repeat for **XBLNR** and press enter to validate.
- **Save** and **generate**, press **back**
- To release the segment choose **Goto, Release** from the menu
- **Check the box** on the line of your new segment
- **Save, back** and **enter**
- Your IDoc type structure should be displayed with your new segment
- **Save** and **back**
- To release the IDoc type choose **Extras, Release type** from the menu and **Yes**

Your IDoc is now ready for use. If you need to add fields or segments to your IDoc type, you will need to cancel the release of the IDoc type as well as the segment release using a similar procedure followed above (except now you uncheck the release box for the segment and you choose cancel release for the IDoc type).

#### 5.2.2.3.2 Create message type (WE81) Client independent

To create a new message type, follow these next few steps:

- Enter transaction **WE81** (ALE -> Extensions -> IDoc types -> Maintain message type for intermed. Structure -> Create logical message type)
- Choose **Create logical message type** by double clicking on it
- Click on **change** icon to enter change mode
- Click on **New entries** to add a new type
- Enter the required message type, in our case it is **ZINVRV** and an appropriate **description**
- **Save** and **exit**.

Your message type has now been created. The next step will be to link it to the IDoc.

#### 5.2.2.3.3 Link message to IDoc type (WE82 & BD69) Client independent

To link the message type to the IDoc type follow these next few steps:

- Enter transaction **WE82** (ALE -> Extensions -> IDoc types -> Maintain message type for intermed. Structure -> EDI: Message Types and Assignment to IDoc Types)
- Click on **change** icon to enter change mode
- Click on **New entries** to create the link

- Enter the message type **ZINVRV** and the BasicIDoc type as **ZINVRV01**
- **Save** and **exit**
- Enter transaction **BD69** (ALE -> Extensions -> IDoc types -> Maintain message type for intermed. Structure -> Assign message type to IDoc for ALE)
- Click on **change** icon to enter change mode
- Click on **New entries** to create the link
- Enter the message type **ZINVRV** and the BasicIDoc type as **ZINVRV01**
- **Save** and **exit**

Your IDoc is now linked to your message type. We still need to link object types and add the message to the model before we can use the message.

#### 5.2.2.3.4 Maintain object type for message type (BD59) Client independent

The ALE objects are used to create links between IDocs and applications objects, to control the **serialisation**, to **filter messages** in the customer model and to use **listings**.

For our own message type and IDoc you must maintain **object types for the links**.

If you want to check the serialisation for the message type, then you must maintain **object types for the serialisation**. If no serialisation object has been maintained for a given message type, then the serialisation will not be checked for this message type.

To add an object type to our message type, follow these next few steps:

- Enter transaction **BD59** (ALE -> Extensions -> ALE object maintenance -> Maintain object types)
- Type in your message type **ZINVRV** and press **enter**
- Click on **New entries**
- Enter your object type, **LIFNR** (We need to use the vendor as a filter object), the segment name where LIFNR resides, **Z1INVRV**, a number **1** for the sequence followed by the actual field name **LIFNR**
- **Save** and **exit**.

You have now created an object that we'll use as a filter object in the customer model to direct the flow of messages to the various logical systems based on the vendors in the filter of the message type ZINVRV.

We now need to add our new message type to the distribution model.

#### 5.2.2.3.5 Configuring the Distribution Model

To manually configure the customer distribution model, read the ALE configuration procedure, and follow these steps:

- Perform the **Maintain customer distribution model directly** function. (ALE -> Distribution customer model -> Maintain customer distribution model directly)
- Specify the customer model you want to maintain and the logical system that is to be the sender of the messages OR create a new model. (Create model ALE with logical system ALELS1C400)
- Choose the receiving systems to which the sending system must forward message type **ZINVRV** to.
- For each receiving logical system allocate the message type necessary for communication to the receiving systems as per ALE configuration procedure.
- Create filter objects (in our case **LIFNR** as the object type with the associated vendor number, **0000018001** with leading zeros, in the object area) for the message types.

- **Save** the entries.

#### **NOTES:**

You cannot maintain a message type between the same sender and receiver in more than one customer distribution model.

Only the owner is authorised to modify the model.

To change the owner of a model, choose the 'Maintain ownership of customer distribution model' function. Make sure that all changes will be distributed to all systems that know the corresponding model. To do so, you can use the correction and transport system.

To transport the customer distribution model you should use the Distribute customer model function of the IMG as described below.

#### 5.2.2.3.6 Distribute customer model (BD71) Client dependent

After the customer model has been created centrally, it must be distributed to the other remote systems. This entails first of all setting up the communication for the distributed systems and then sending the model.

##### 5.2.2.3.6.1 *Distribute Model (BD71) Client dependent*

This task is performed on your ALE reference client. To distribute the customer distribution model, read the ALE configuration procedure and follow these steps:

- Make the settings for the communication with the other non-central systems that you have not set them yet.
  - Define the RFC destination for R/3 connections whose names correspond to the name of the corresponding logical system.
  - Create the output partner profile.
- Distribute the customer model
  - Specify the name of the customer model.
  - You must specify the target system to which you want to distribute the customer model.
  - You must repeat this function for every distributed logical system.

#### 5.2.2.3.7 Maintain sending system partner profile (WE20) Client dependent

With this function, you define the partner profiles for all outbound and inbound messages on the basis of the customer distribution model.

After you have defined and distributed the customer model, you will have to maintain the partner profiles locally. To do this read the ALE configuration procedure.

- Enter the output mode (background, immediately) and the package size for outbound processing.

#### **Requirements**

- The customer model must be maintained.
- RFC destinations must be maintained.
- The customer model must be distributed.
- To ensure that the appropriate persons in charge are informed if a processing error occurs, you must make settings in: Error processing Maintain organisational units.

#### 5.2.2.3.8 Populate & distribute IDoc using ABAP

An IDoc consists of a control record with structure edidc and one or more data records with structure edidd. The control record contains the sender and recipient of the IDoc, as well as information on the type of message.

To be able to pass an IDoc to the ALE layer, you must set up a field string with structure edidc and an internal table with structure edidd. They are used to call function module **master\_idoc\_distribute**, which performs the save to the database and triggers the dispatch if necessary.

#### 5.2.2.3.8.1 Example code

The code displayed below does the following:

- populates our IDoc segment Z1INVR with the 2 fields XBLNR and LIFNR, populates the segment name and appends this to an internal table used to store the IDoc data;
- populates the control record info with the message type and IDoc type; and
- calls the MASTER\_IDOC\_DISTRIBUTE function module which distributes the IDoc as configured in the customer distribution model.

```
*--- Data declaration statements
DATA:                                C_INVREV_SEGNAME(7) TYPE C VALUE 'Z1INVRV',
                                      C_INVREV_MESTYPE(6) TYPE C VALUE 'ZINVRV',
                                      C_INVREV_IDOC_TYPE(8) TYPE C VALUE 'ZINVRV01',
                                      Z1INVRV LIKE Z1INVRV,
                                      C_INVREV_DOCTYPE LIKE BKPF-BLART VALUE 'YY',
                                      IDOC_CONTROL LIKE EDIDC,
                                      T_COMM_CONTROL LIKE EDIDC OCCURS 0 WITH HEADER
LINE,                                IDOC_DATA LIKE EDIDD OCCURS 0 WITH HEADER LINE.

*--- Move the document header into a structure
LOOP AT DOC_HEAD_TAB INTO DOC_HEAD.
ENDLOOP.

*--- Move the document item data into a structure
LOOP AT DOC_ITEM_TAB INTO DOC_ITEM WHERE NOT ( LIFNR IS INITIAL ).
ENDLOOP.

*--- Populate the IDoc segment's field with the required data
CLEAR Z1INVRV.
Z1INVRV-LIFNR = DOC_ITEM-LIFNR.      "Store vendor number for filter
Z1INVRV-XBLNR = DOC_HEAD-XBLNR.     "Billing number
IDOC_DATA-SEGNAM = C_INVREV_SEGNAME. "Segment name
IDOC_DATA-SDATA = Z1INVRV.          "Segment data
APPEND IDOC_DATA.                   "Populate IDoc internal table

*--- Move the control data info required for the distribution
IDOC_CONTROL-MESTYP = C_INVREV_MESTYPE.
IDOC_CONTROL-DOCTYP = C_INVREV_IDOC_TYPE.
```

```

*--- Call the distribute function with the required parameters
CALL FUNCTION 'MASTER_IDOC_DISTRIBUTE' IN UPDATE TASK
      EXPORTING
        MASTER_IDOC_CONTROL          = IDOC_CONTROL
      TABLES
        COMMUNICATION_IDOC_CONTROL =
T_COMM_CONTROL
        MASTER_IDOC_DATA            = IDOC_DATA
      EXCEPTIONS
        ERROR_IN_IDOC_CONTROL       = 1
        ERROR_WRITING_IDOC_STATUS   = 2
        ERROR_IN_IDOC_DATA          = 3
        SENDING_LOGICAL_SYSTEM_UNKNOWN = 4
        OTHERS                      = 5.

```

Figure 14: Outbound processing example code

#### NOTE:

For debugging purposes, use transaction WE05 (IDoc overview) to see check your IDoc status, or to see whether an IDoc was created/

### 5.2.2.4 Inbound processing

#### 5.2.2.4.1 Create Function Module

This function module is called when a message type, of type ZINVRV, comes into the receiving system. This needs to be configured and is dealt with later in this section. The function module is passed the IDoc as a parameter.

##### 5.2.2.4.1.1 Example parameters

Import parameters	Reference field	Opt Y/N
INPUT_METHOD	BDWFAP_PAR-INPUTMETHD	N
MASS_PROCESSING	BDWFAP_PAR-MASS_PROC	N

Export Parameters	Reference field	Opt Y/N
WORKFLOW_RESULT	BDWFAP_PAR-RESULT	N
APPLICATION_VARIABLE	BDWFAP_PAR-APPL_VAR	N
IN_UPDATE_TASK	BDWFAP_PAR-UPDATETASK	N
CALL_TRANSACTION_DONE	BDWFAP_PAR-CALLTRANS	N

Table Parameters	Reference field	Optional Y/N
IDOC_CONTRL	EDIDC	
IDOC_DATA	EDIDD	

IDOC_STATUS	BDIDOCSTAT	
RETURN_VARIABLES	BDWFRETVAR	
SERIALIZATION_INFO	BDI_SER	

Exceptions
WRONG_FUNCTION_CALLED

#### 5.2.2.4.1.2 Example code

The code displayed below does the following:

- populates a BDC table with the IDoc info;
- calls the transaction via a BDC call; and
- updates the IDoc status according to the BDC error status.

```

EXTRACT FROM: Z_IDOC_INPUT_ZINVRV

*--- Declaration of local variables
DATA: C_SEGNAM(10) TYPE C VALUE 'Z1INVRV'.

*-Loop through the IDOCs
LOOP AT IDOC_CONTRL.
*---Loop through the data for the IDOC
      LOOP AT IDOC_DATA WHERE DOCNUM = IDOC_CONTRL-
DOCNUM.
          CASE IDOC_DATA-SEGNAM.
              WHEN C_SEGNAM.
*                  Here we get the info from the idoc table
                  IT_Z1INVRV = IDOC_DATA-SDATA.
          ENDCASE.
          PERFORM REV_INV.
      ENDLOOP.
      PERFORM UPDATE_IDOC_STATUS.
ENDLOOP.

FORM REV_INV                "Reverse invoice form
*--- Local variables & constants
DATA: C_TCODE LIKE BKPF-TCODE VALUE 'VF11'. "BDC transaction code

*--- Now we can build the bdc table to call the reversal transaction start of screen 109
CLEAR BDC_TAB.
BDC_TAB-PROGRAM = 'SAPMV60A'.

```

```

BDC_TAB-DYNPRO = '109'.
BDC_TAB-DYNBEGIN = 'X'.
APPEND BDC_TAB.
*--- Document number
CLEAR BDC_TAB.
BDC_TAB-FNAM = 'KOMFK-VBELN(01)'.
BDC_TAB-FVAL = IT_Z1INVRV-XBLNR.    "Billing document number
APPEND BDC_TAB.
*--- OK Code for screen 109
CLEAR BDC_TAB.
BDC_TAB-FNAM = 'BDC_OKCODE'.
BDC_TAB-FVAL = 'SICH'.
APPEND BDC_TAB.

*--- Now we can call transaction 'VF11' with the populated bdc table. The transaction is
called inside the idoc-ctrl loop, so a transaction will be called for every idoc (journal).
the transaction is called in no-display mode ('N') because this code runs in background
as it is called by ale. The update is specified to be synchronous ('S') because we have
to wait for the result to update the idoc status correctly.
CALL TRANSACTION C_TCODE USING BDC_TAB MODE 'N' UPDATE 'S'.

*--- Store the return code for use in another form (status update)
RETURN_CODE = SY-SUBRC.

*--- Here we check the return code, if there was an error, we put the transaction in a
bdc session for the user to review and correct.
IF SY-SUBRC NE 0.
    CALL FUNCTION 'BDC_OPEN_GROUP'
        EXPORTING
            CLIENT = SY-MANDT
            GROUP  = 'ZINVRV'
            USER   = C_ALE_USER
            KEEP   = 'X'.
    CALL FUNCTION 'BDC_INSERT'
        EXPORTING
            TCODE   = C_TCODE
        TABLES
            DYNPROTAB = BDC_TAB.
    CALL FUNCTION 'BDC_CLOSE_GROUP'
        EXCEPTIONS
            NOT_OPEN = 1

```



```

                                QUEUE_ERROR = 2
                                OTHERS      = 3.
ELSE.                                "No problems
                                C_EXISTS = 'N'.
*   Select from the billing document table to get sales doc number
                                SELECT * FROM VBRP WHERE VBELN = IT_Z1INVRV-XBLNR.
*   Select from the sales document table to get user status number
                                SELECT SINGLE * FROM VBAP WHERE VBELN = VBRP-
AUBEL AND
                                POSNR = VBRP-AUPOS.
*   Select from the status table to change the user status to pending
                                SELECT * FROM JEST WHERE OBJNR = VBAP-OBJNR AND
                                STAT LIKE C_USER_STATUS.
                                IF JEST-STAT = C_US_PENDING.  "User status is pending
                                JEST-INACT = C_UNCHECKED.    "Make pending
the active status
                                UPDATE JEST.
                                C_EXISTS = 'Y'.            "I.E. An entry is already in
table
                                ELSEIF JEST-INACT = C_UNCHECKED AND JEST-STAT
NE C_US_PENDING.
                                JEST-INACT = C_CHECKED.      "Make everything
else inactive
                                UPDATE JEST.
                                ENDIF.
                                ENDSELECT.
                                IF C_EXISTS = 'N'.  "I.E. Pending has never been a status before
                                JEST-OBJNR = VBAP-OBJNR.
                                JEST-STAT = C_US_PENDING.
                                JEST-INACT = C_UNCHECKED.    "Make pending the
active status
                                INSERT JEST.
                                ENDIF.
                                ENDSELECT.  "Select from VBRP (Billing document table)
ENDIF.
ENDFORM.                                " REV_INV

FORM UPDATE_IDOC_STATUS.
*--- Now we check the CALL TRANSACTION return code and set IDOC status
                                CLEAR IDOC_STATUS.
                                IF RETURN_CODE = 0.
                                WORKFLOW_RESULT = '0'.

```

```

IDOC_STATUS-DOCNUM = IDOC_CONTRL-DOCNUM.
IDOC_STATUS-STATUS = '53'.
IDOC_STATUS-UNAME = SY-UNAME.
IDOC_STATUS-REPID = SY-REPID.
IDOC_STATUS-MSGTY = SY-MSGTY.
IDOC_STATUS-MSGID = SY-MSGID.
IDOC_STATUS-MSGNO = SY-MSGNO.
IDOC_STATUS-MSGV1 = SY-MSGV1.
IDOC_STATUS-MSGV2 = SY-MSGV2.
IDOC_STATUS-MSGV3 = SY-MSGV3.
IDOC_STATUS-MSGV4 = SY-MSGV4.
RETURN_VARIABLES-WF_PARAM = 'Processed_IDOCs'.
RETURN_VARIABLES-DOC_NUMBER   =   IDOC_CONTRL-
DOCNUM.

    APPEND RETURN_VARIABLES.
ELSE.
    WORKFLOW_RESULT = '99999'.
    IDOC_STATUS-DOCNUM = IDOC_CONTRL-DOCNUM.
    IDOC_STATUS-STATUS = '51'.
    IDOC_STATUS-UNAME = SY-UNAME.
    IDOC_STATUS-REPID = SY-REPID.
    IDOC_STATUS-MSGTY = SY-MSGTY.
    IDOC_STATUS-MSGID = SY-MSGID.
    IDOC_STATUS-MSGNO = SY-MSGNO.
    IDOC_STATUS-MSGV1 = SY-MSGV1.
    IDOC_STATUS-MSGV2 = SY-MSGV2.
    IDOC_STATUS-MSGV3 = SY-MSGV3.
    IDOC_STATUS-MSGV4 = SY-MSGV4.
    RETURN_VARIABLES-WF_PARAM = 'ERROR_IDOCS'.
    RETURN_VARIABLES-DOC_NUMBER   =   IDOC_CONTRL-
DOCNUM.

    APPEND RETURN_VARIABLES.
ENDIF.
APPEND IDOC_STATUS.
ENDFORM.               " UPDATE_IDOC_STATUS

```

*Figure 15: Inbound processing example code*

#### 5.2.2.4.2 Debugging inbound FM

Use transaction **WE19** to test inbound function module in debugging mode. Also use **WE05** to view the IDocs and their statuses.

#### 5.2.2.4.3 Maintain ALE attributes

The inbound function module needs to be linked to the message type and the message type needs to be linked to the appropriate inbound process code at the partner profile level before the scenario is enabled. These steps are described below in detail.

##### 5.2.2.4.3.1 Link Message Type to Function Module (WE57) Client independent

To link a message (ZINVRV) type to a function module (Z\_IDOC\_INPUT\_ZINVRV) follow these steps:

- Enter transaction **WE57** (ALE -> Extensions -> Inbound -> Allocate function module to logical message)
- Select an entry (EG. IDOC\_INPUT\_ORDERS) and **copy**
- Type in module name **Z\_IDOC\_INPUT\_ZINVRV**
- Type in basic IDoc type as **ZINVRV01**
- Type in message type as **ZINVRV**
- Type object type as **IDOCINVOIC** (Invoice document) - Used for workflow
- Direction should be set to **2** for inbound
- **Enter** and **save**

##### 5.2.2.4.3.2 Define FM settings (BD51) Client independent

- Enter transaction **BD51** (ALE -> Extensions -> Inbound -> Define settings for input modules)
- Click on **New entries**
- Type in the name of the new function module **Z\_IDOC\_INPUT\_ZINVRV**
- Enter **0** for mass processing in the output column
- **Save** and **Exit**

##### 5.2.2.4.3.3 Maintain process codes (WE42) Client dependent

A process code needs to be maintained on each client. It then needs to be linked to the message via the partner profiles on each client. This allows the various clients to use a unique function module for the same message type.

To maintain the process code follow these steps:

- Log on to the appropriate receiving system client
- Execute **WE42** (ALE -> Extensions -> Inbound -> Maintaining process codes inbound)
- Choose **Inbound with ALE service**
- Choose **Processing with function module**
- Click on **Processing with function module** and choose **create** icon
- Click on **New Entries**
- Type in process code **ZINR** and give it a **description** and **save**
- Now you are asked to Please maintain codes added in ALE entry methods, **enter** and choose **Z\_IDOC\_INPUT\_FIRVSL** and **copy** it. You should choose a FM similar to your one.
- Enter your process code **ZINR**
- Enter your function module **Z\_IDOC\_INPUT\_ZINVRV**

**NOTE:** The next 6 steps are used in workflow error handling.

- Enter **IDPKFIDCMT** in object type

- Enter **MASSINPUTFINISHED** in End event
- Enter **IDOCINVOIC** in IDoc object type
- Enter **INPUTERROROCCURREDFI** in IDoc start event
- Enter **INPUTFINISHEDFI** in IDoc End event
- Enter **IDOCINVOIC** in Application object type

You will need to determine the task associated with object IDOCINVOIC, and then assign the appropriate position to it. This position will then receive the application error messages via workflow.

To set up the workflow areas please consult the Workflow config guide.

#### 5.2.2.4.4 Create inbound partner profile

For each message type you need to maintain the inbound partner profiles.

##### 5.2.2.4.4.1 *Maintain receiving system partner profile (WE20) Client dependent*

To maintain inbound partner profiles read the document ALE configuration procedure:

- Add the message type **ZINVRV** with process code **ZINR**.
- Enter the output mode (background, immediately) for inbound processing and **NO** message code.
- Enter the position **S** and choose the ALE administrator **50000085**. This position will then receive all the technical ALE errors via workflow.

#### 5.2.2.4.5 Test

Once the inbound function module has been debugged the scenario should be ready to test in its entirety. If problems occur, read through the relevant areas of this document to check your configuration or code.

#### 5.2.2.5 **Test Procedures**

Unit and string testing needs to be conducted on the development together with result logging and sign-off.

*CHAPTER 6*  
**THE ALE OPERATIONAL SUPPORT PHASE**

---

## 6 ALE OPERATIONS SUPPORT

This section contains all areas related to supporting ALE in a production environment.

### 6.1 Project team tasks

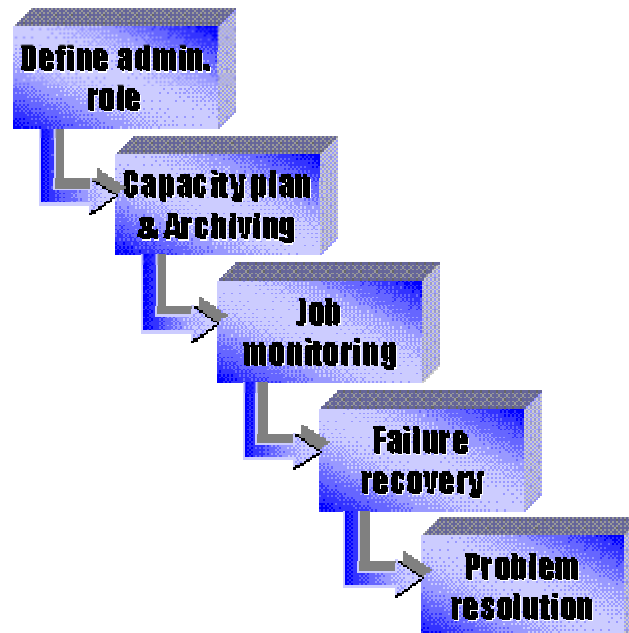


Figure 16: ALE project team tasks during ALE Operations phase

Various areas, pertaining to the Operations / Production environment need to be addressed BEFORE you go live:

#### 6.1.1 ALE Administrator role

ALE Administrator role defined and sourced. Responsible for ALE configuration and issue resolution.

#### 6.1.2 Capacity plan

Capacity plan including ALE IDoc archiving strategy - Consider the growth of the ALE solution, procedures the archiving of IDocs while considering the legal implications.

#### 6.1.3 ALE Operations role

ALE Operations role defined and sourced. Responsible for ALE archiving, background job scheduling and monitoring.

#### 6.1.4 ALE Failure recovery

ALE failure recovery procedure - When a soft failure occurs (I.E. When a system is restored to a previous point in time) a recovery procedure needs to be implemented... How?

#### 6.1.5 ALE Support organisation structure and process

Define the ALE support organization structure. The ALE administrator will receive the ALE problems when in production. Using a troubleshooting guide the problems can be resolved.

### 6.2 Other considerations

#### 6.2.1 Client copies

Never do client copies... If you do, check on:

- the change pointer table (BDGP/S);
- the IDoc table (EDIDC/S); and

- all the logical system references.

### 6.2.2 Client specific configuration

- Immediate processing on the outbound and inbound side should be avoided due to the fact that ALE takes every available dialog process, on the receiving system, when performing the send.
- Configuration needs to happen, on-line, in production systems for versions up to V4

### 6.2.3 Error handling

Workflow is used to trap error messages. If you have MS Exchange then consider implementing the integration between MS and SAP as this will help the ALE administrator with 1 in-box of errors as apposed to 1 per client that he supports

### 6.2.4 ALE specific scenarios

- The classification master and its respective master data object (vendor, material or customer) are not linked, in any way, via ALE message types up to V4. This implies that the classification master may end up on systems where it is irrelevant.
- OSS notes need to be implemented to improve the speed of the programs RBDAPP01, RBDMANIN, RBDMIDOC, RBDSTATE.
- OSS note needs to be implemented to get the ALE recovery tool. BDRC and BDRL.

## 6.3 Deliverables

### 6.3.1 ALE Administrator Troubleshooting Procedure

What to do when the unexpected happens... An example of what to produce follows:

#### 6.3.1.1 Introduction

The purpose of this document is to provide a guideline for the ALE administrator to perform the following tasks:

- perform the technical ALE configuration;
- monitor and detect errors occurring in the master data ALE process; and
- monitor and detect erros occurring in the transaction ALE process.

#### 6.3.1.2 Roles using this Procedure

Name of Role
ALE Administrator

#### 6.3.1.3 Installation and configuration

Please refer to the document: **ALE Technical Configuration Section**, in order to configure an ALE technical environment.

##### 6.3.1.3.1 Processing RFCs with errors

If errors occur in a Remote Function Call, these are processed in the standard in the single error processing. A background job is scheduled for each RFC that resulted in an error, and this background job keeps restarting the RFC until the RFC has been processed successfully. In the case that the connection to the recipient system has been broken, this can mean that a very large of background jobs gets created that will represent a considerable additional load on the sending system.

You should always use the collective error processing in productive operation so as to improve the system performance. This will not automatically re-submit the RFC immediately, but a periodically scheduled background job will collect together all the RFCs that failed and will re-start them as a

packet. This helps to reduce the number of background jobs created. This can be done both for R/3 connections and for TCP/IP connections.

To set up the collective error processing proceed as follows:

1. Change the RFC destination
2. Select the Destination -> TRFC options function from the menu.
3. Enter the value 'X' into the 'Suppress backgr. job in case of comms. error' field.

Perform the error handling as follows:

4. Start the 'Transactional RFC' monitor (menu: Logistics -> Central functions -> Distribution -> Monitoring -> Transactional RFC)
5. Select the Edit -> Select. FM execute function.

For the error handling you should schedule a periodic background job that regularly does this.

Train the error handling for errors in the Remote Function Call before the productive start.

#### **6.3.1.4 Start-up and Shut-down**

- 1 ALE will be started automatically on startup. If a new installation is being started up then the configuration in the above-mentioned section needs to be performed.

#### **6.3.1.5 Administration**

- 1 Grant the ALE administrator role the necessary rights as per the ALE authorisations developed by the basis team
- 2 Allocate the ALE administrator people to the ALE Administrator role
  - 2.1.. Follow the workflow procedures on performing this activity
- 3 Load the ALE-BATCH background user with SAP ALL rights with password init

#### **6.3.1.6 Monitoring**

- 1 IDoc overview monitoring (Transaction WE05) RED = Error
  - 1.1.. Follow the procedure in housekeeping to correct

#### **6.3.1.7 Housekeeping**

##### **6.3.1.7.1 Master data scenarios**

This section applies to the vendor master, material master, cost & profit center master, classification master, chart of accounts and cost elements master.

##### **6.3.1.7.1.1 Log on to the reference client (4XX series)**

A person belonging to the ALE administrator role needs to log onto the client.

##### **6.3.1.7.1.2 Checks in 4XX (Outbound)**

- Start transaction **BALE**
- Choose menu option: **Monitoring -> IDoc Overview**
- Type in the required message type in the Logical message type box together with a date and time from the last check. You can also leave the message type box empty to see all message types.
- Execute the function.
- If any IDoc is red, highlighting an error, double click on the IDoc line. A list of IDocs will appear. (Each one corresponding to one IDoc in error) Go through each one, by double clicking,



checking the IDoc status. Correct the error in the application and reprocess the IDoc (See section 6.3.1.7.1.4)

#### 6.3.1.7.1.3 Checks in receiving clients (Inbound)

- Start transaction **BALE**
- Choose menu option: **Monitoring -> IDoc Overview**
- Type in the required message type in the Logical message type box together with a date and time from the last check. You can also leave the message type box empty to see all message types.
- Execute the function
- If your IDoc is red with a message "Application Document not posted". Then double click on the error. A list of IDocs will appear. (Each one corresponding to an IDoc in error) Go through each one, by double clicking, checking the IDoc status. Correct the error in the application and reprocess the IDoc (See section 6.3.1.7.1.4)
- If the IDoc is white showing that your application document has been posted then the IDoc has been transferred from 400 to this client successfully and has been posted correctly.

#### 6.3.1.7.1.4 Reprocessing IDocs with errors

##### 6.3.1.7.1.4.1 Outbound (BD88)

- Once the error has been determined and corrected it is not necessary to resend the IDoc again. You simply resend the IDocs that have already been generated.
- Using the IDoc overview screen you need to take note of the following for each IDoc that was not processed:
  - Error number: 2, 4, 5, 25, 29
  - Error number: 30 (Execute Check IDoc dispatch to process)
  - IDoc number
- Using the **Error number**, the **IDoc number** and the transaction **BD88** , with the required message type you can resend the IDoc. Match the error number with this transaction and execute the function for the IDocs incorrectly processed.
- If your IDoc is still not correctly processed redo the first 3 steps.

##### 6.3.1.7.1.4.2 Inbound (BD87)

- Once the error has been determined and corrected it is not necessary to resend the IDoc again. You simply reprocess the IDocs that have already been received.
- Using the IDoc overview screen (WE05) you need to take note of the following for each IDoc that was not processed:
  - Error number: 51, 56, 61, 63, 65, 60
  - Error number: 64 (Execute Check IDoc dispatch to process)
  - IDoc number
- Using the **Error number**, the **IDoc number** and the transaction **BD87**, with the required message type you can reprocess the IDoc. Match the error number with this transaction and execute the function for the IDocs incorrectly processed.
- If your IDoc is still not correctly processed redo the first 3 steps after consulting with the functional specialist to ensure functional setup is correct.

#### 6.3.1.7.2 Transaction scenarios

#### 6.3.1.7.2.1 Log on to the sending system client

This refers to the client where the Purchase order was created, the invoice was created or the journal was created... The person needs to be a member of the ALE administrator position.

#### 6.3.1.7.2.2 Checks on sending \ outbound client

- Start transaction **BALE**
- Choose menu option: **Monitoring -> IDoc Overview**
- Type in the required message type in the Logical message type box together with a date and time from the last check. You can also leave the message type box empty to see all message types.
- Execute the function.
- If any IDoc is red, highlighting an error, double click on the IDoc line. A list of IDocs will appear. (Each one corresponding to one IDoc in error) Go through each one, by double clicking, checking the IDoc status. Correct the error in the application and reprocess the IDoc (See section 6.3.1.7.2.4)

#### 6.3.1.7.2.3 Checks in receiving \ inbound client

- Start transaction **BALE**
- Choose menu option: **Monitoring -> IDoc Overview**
- Type in the required message type in the Logical message type box together with a date and time from the last check. You can also leave the message type box empty to see all message types.
- Execute the function
- If your IDoc is red with a message "Application Document not posted". Then double click on the error. A list of IDocs will appear. (Each one corresponding to an IDoc in error) Go through each one, by double clicking, checking the IDoc status. Correct the error in the application and reprocess the IDoc (See section 6.3.1.7.2.4)
- If the IDoc is white showing that your application document has been posted then the IDoc has been transferred from 400 to this client successfully and has been posted correctly.

#### 6.3.1.7.2.4 Reprocessing IDocs with errors

##### 6.3.1.7.2.4.1 Outbound (BD88)

- Once the error has been determined and corrected it is not necessary to resend the IDoc again. You simply resend the IDocs that have already been generated.
- Using the IDoc overview screen you need to take note of the following for each IDoc that was not processed:
  - Error number: 2, 4, 5, 25, 29
  - Error number: 30 (Execute Check IDoc dispatch to process)
  - IDoc number
- Using the **Error number**, the **IDoc number** and the transaction **BD88** , with the required message type you can resend the IDoc. Match the error number with this transaction and execute the function for the IDocs incorrectly processed.
- If your IDoc is still not correctly processed redo the first 3 steps.

##### 6.3.1.7.2.4.2 Inbound (BD87)

- Once the error has been determined and corrected it is not necessary to resend the IDoc again. You simply reprocess the IDocs that have already been received.
- Using the IDoc overview screen (WE05) you need to take note of the following for each IDoc that was not processed:
  - Error number: 51, 56, 61, 63, 65, 60
  - Error number: 64 (Execute Check IDoc dispatch to process)
  - IDoc number
- Using the **Error number**, the **IDoc number** and the transaction **BD87**, with the required message type you can reprocess the IDoc. Match the error number with this transaction and execute the function for the IDocs incorrectly processed.
- If your IDoc is still not correctly processed redo the first 3 steps after consulting with the functional specialist to ensure functional setup is correct.

#### 6.3.1.7.3 Reprocessing IDocs (Description)

If the IDoc could not be passed to the application successfully (status: 51 - error on hand-over to application), then repeated attempts may be made with the RBDMANIN report.

This functionality can be accessed through the menu: Logistics -> Central functions -> Distribution and then Period. work -> IDoc, ALE input

Selections can be made according to specific errors. Therefore this report could be scheduled as a periodic job that collects IDocs that could not be passed to the applications because of a locking problem.

#### 6.3.1.7.4 Error handling in Inbound processing (Description)

The following is a description of how an error that occurs during ALE processing is handled:

- The processing of the IDoc causing the error is terminated.
- An event is triggered.
- This event starts an error workitem:
  - The employees responsible will find a workitem in their workflow inboxes.
  - An error message is displayed when the workitem is processed.
  - The error is corrected in another window and the IDoc can then be resubmitted for processing.
  - If the error cannot be corrected, the IDoc can be marked for deletion.
- Once the IDoc has been successfully imported, an event is triggered that terminates the error workitem. The workitem then disappears from the inbox.

### 6.3.2 ALE Archiving strategy and procedure

When and how are you going to archive the IDocs? Remember the legal implications. The following document can serve as an example:

#### 6.3.2.1 Introduction

The purpose of this document is to describe the ALE IDoc archiving strategy and to provide a procedure to perform the archiving of IDocs.

Whenever information gets sent from one SAP system to another using ALE, IDocs are created. These IDocs act as the conveyors of the information that is sent between systems. An IDoc is filled with the information and then physically sent from the sending SAP system to the receiving SAP system. In both the sending and receiving SAP systems, the IDoc gets stored in the SAP database. These IDoc tables can get large after a while and start affecting system performance (especially in the ALE environment). Therefore these IDoc tables need to be cleared of old IDocs periodically.

The way in which these tables will be cleared of old IDocs, is to archive and then delete the IDocs.

#### 6.3.2.1.1 Troubleshooting

- Error: Could not create Archive file XXXXX. Check tables **ARCH\_NUM** (Next number for the archive index) and table **ADMI\_RUN** (list of archive run). Check that ADMI\_RUN does not have an entry for the number contained in ARCH\_NUM. **SOLUTION:** Delete entries from ADMI\_RUN until ARCH\_NUM is reached (also including the ARCH\_NUM number).

#### 6.3.2.2 Archiving of IDOCs

SAP provides a way of archiving and deleting application objects. One of the objects that can be archived is IDocs. After IDocs have been archived, the IDocs can be deleted from the SAP database.

IDocs for master data and transaction data will be archived before the IDocs will be deleted. The IDocs for transaction data will be backed up before the archive files are deleted. The archive files can optionally be backed up as part of the general SAP archiving strategy for master data, but there are no requirements for backing up archived master data IDocs.

##### 6.3.2.2.1 Overview of the archiving process

When archiving of IDocs is performed, IDocs are selected from the R/3 database based on user defined criteria and written to an archive file by the archiving program. After IDocs have been written to the archive file, a delete program can be run that will read the archive file and delete all IDocs that have been successfully archived. This ensures that IDocs are only deleted after they have successfully been written to an archive file.

Optionally the archive file can then be backed up onto a backup device or using SAP's optical archiving facility.

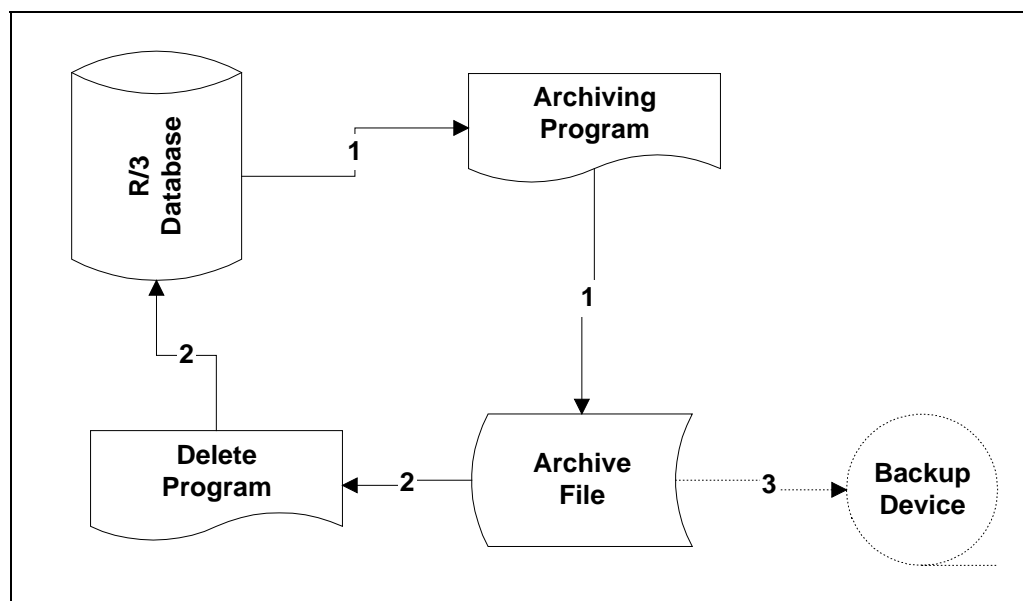


Figure 17: Overview of the archiving process

#### 6.3.2.3 Determining which IDOCs to archive

The archiving programs are responsible for reading the IDocs to be archived and writing them to an archive. The actual deletion of the archived IDocs from the SAP database is carried out by the deletion program. The archiving programs offer the user various selection parameters which can be used to determine exactly which IDocs are to be archived. The selection parameters of RSEXARCA are:

- Date of last status update of the IDoc

- Logical message, divided into message type, message variant and message function.
- Direction
- IDoc number

The selection list may also contain IDocs, which cannot be archived due to their status. The status values, which can and cannot be archived are stored in the R/3 database and can be maintained by the user. In other words, only IDocs that are in a status that has been flagged as possible for archiving, will be archived.

The following is a list of IDoc statuses that will be archived:

IDoc status	Description
12	Dispatch OK
41	Application document created in target system
53	Application document posted

These IDoc statuses have to be configured before the archiving program will archive the IDocs in the specified statuses. The transaction where the IDoc statuses are updated is **WE47**. The menu path from the SAP main menu is: Tools->**Administration->Administration->Process technology->IDoc->IDoc basis->Control->Status maintenance**. The IDoc statuses are client independent and have to be maintained on each physical SAP system. The changes can be put in a change request and transported to other environments.

The status maintenance lists all IDoc statuses, and each status can be edited by double clicking on the status, or selecting the status and choosing Goto->Details from the menu structure.

By default only status 40 and 41 are set up to be archived. Status 40 (Application document not created in target system) should be set up to suppress archiving, and the statuses mentioned above (12, 41 and 53) should be set up to enable archiving.

This set up will ensure that all IDocs that are in statuses that were enabled for archiving, will be archived.


#### 6.3.2.3.1 The archiving and deletion procedure

This section of the document will describe how to archive and delete IDocs. The SAP archiving functionality can be accessed through transaction **SARA**. The menu path from the SAP main menu is Tools-> Administration-> Administration-> Archiving.

Before archiving IDocs, transaction **WE05** can be executed to determine which IDocs will be archived for a specific selection list. ***Note that transaction WE05 uses the creation date of IDocs to select IDocs, while the archiving program uses the date of the last status change of an IDoc to determine which IDocs will be archived. This may cause some discrepancies between what IDocs are shown in transaction WE05 and what are archived for the same selection list in archiving.***

The Object name for archiving IDocs is **IDOC**. The following steps need to be performed to archive IDocs:


1. Enter IDOC as the object name from transaction SARA.
2. Select the archive button.
3. Maintain a variant for the archiving procedure. This variant will determine what selection of IDocs will be archived. A variant for archiving can only be used once for security reasons. The variant name will not be useable more than once or until the archiving job that used the variant has been deleted. The variant parameters have to be selected depending on the selection of IDocs that need to be archived:
4. Save the variant and go back to the archive screen.

5. Select the start date button to set up when the archiving job will be scheduled. For archiving purposes, the job can be scheduled to run immediately. Save the job parameters to go back to the archiving screen.
6. Select the Spool parameters button to set up the background printing for the archive job. A valid printer has to be specified. Deselect the print immediately option and ensure that the delete after print option is selected.
7. Select the Customizing button to set up customising options. Select ARCHIVE\_DATA\_FILE as the logical file name (if a specific physical location for the archive file needs to be selected, the basis setup for archiving needs to be changed using transactions FILE and SF01. These transactions can also be accessed through the Reference IMG under: Basis Components->System Administration->Platform-independent File Names->Client-independent maintenance of file names and paths or Additional client-dependent file name maintenance). Select start automatically for the Settings for delete program (this will automatically start the delete program after successful archiving of the IDocs). All other settings can be left at the default settings:
8. These settings are client independent and can be transported to other environments. The customising settings need to be configured once only.
9. Click on the generate job () button to generate the archiving job. After the archiving job has been generated, the job can be monitored by clicking on the Job overview button. After the archiving job has completed, the delete job will execute and delete all successfully archived IDocs. The spool lists for all jobs can be checked for the details of the archiving and delete jobs.

#### 6.3.2.3.2 The reload procedure

If IDocs need to be reloaded, an existing archive can be used to reload IDocs after the IDocs have been deleted. This section of the document will describe how to reload deleted IDocs. The SAP archiving functionality can be accessed through transaction **SARA**. The menu path from the SAP main menu is Tools-> Administration-> Administration-> Archiving.

The Object name for archiving IDocs is **IDOC**. The following steps need to be performed to reload IDocs:

1. Enter IDOC as the object name from transaction SARA.
2. Select the reload button.
3. Select the archive to be reloaded by clicking on the Archive selection button.
4. Select an archive from the list of archives that have been successfully deleted and continue.
5. Select the start date button to set up when the reload job will be scheduled. For reload purposes, the job can be scheduled to run immediately. Save the job parameters to go back to the reload screen.
6. Select the Spool parameter button to set up the background printing for the reload job. A valid printer has to be specified. Deselect the print immediately option and ensure that the delete after print option is selected.
7. Click on the generate job () button to generate the reload job. After the reload job has been generated, the job can be monitored by clicking on the Job overview button. After the reload job has completed, the IDocs in the selected archive(s) will be reloaded in status "IDoc reloaded from archive". The spool lists for the job can be checked for the details of the reload job.

#### 6.3.2.3.3 Housekeeping

When archiving IDocs, certain files and jobs need to be maintained on a regular basis. When archiving IDocs, a file gets created on UNIX level for every archiving session. These files have to be backed up if the IDocs archived were for transaction data but don't need to be backed up for master data IDocs. The archive files at UNIX level also need to be deleted periodically to recover disk space.

Any archiving, deletion and reloading session, creates job(s) on SAP. These jobs also need to be deleted from the job log periodically.

#### 6.3.2.3.3.1 *Deleting archive files*

The directory where archive files are stored needs to be determined before the files can be deleted.

1. The logical file name that is used for archiving can be determined by selecting the Customising button from the main archiving screen (transaction SARA).
2. From the customising screen note the logical file name for the archiving files.
3. This logical file name can then be inspected through transactions SF01 and FILE to determine the physical path of the archive files. This information can then be used as part of the Basis backup and housekeeping procedures to backup and delete archive files.

#### 6.3.2.3.3.2 *Deleting archive jobs*

Transactions **SM37** (job overview) can be used to view and delete archive jobs. All archive jobs that are finished can be deleted and do not need to be kept in the job log.

All archiving jobs (archiving, deletion and reloading) start with **ARV\_IDOC\***.

**This periodic deletion of jobs can be done manually or through some scheduled job as part of the Basis housekeeping procedures.**

#### 6.3.2.4 *Settings for Archiving*

This section will describe the specific settings that must be used in conjunction with the previous procedural section to provide an archiving solution.

##### 6.3.2.4.1 *Archiving timing*

Archiving of IDocs must be performed as part of setting up a newly copied client. Whenever a client is copied from another client, the source client's IDocs are copied to the new client and assigned the new client's client number. These "unwanted" IDocs must be archived and deleted (no need for backing up the archive files).

All change pointers are also copied from the source client to the target. These "unwanted" change pointers must also be deleted as part of creating a new client. **This procedure will be described in detail in the client copy procedure.**

Archiving of IDocs need to happen periodically once a client is being used in a development, quality assurance or production environment. Currently the proposal is that the archiving happens twice a month, during periods when the system is not too busy i.t.o. running other batch jobs (proposed to run on the 5<sup>th</sup> and 20<sup>th</sup> of each month).

During this archive run all master data IDocs older than the current date can be archived and deleted. This can be achieved by setting up the archive variant in such a way that the date in the variant is set up as a variable of the current date minus 1 day. All message types that are for master data scenarios should be selected as part of the master data variant (CHRMAS, CLFMAS, CLSMAS, COELEM, COGRP1, COGRP2, COGRP6, COSMAS, PRCMAS, ZCREIC, ZCRENE, ZDEBIC, ZDEBMS, ZGLMAS, ZMATMC, ZMATMS, ZMATSM).

All IDocs for transaction scenarios older than a month can be archived and deleted. This can be achieved by setting up the archive variant in such a way that the date in the variant is set up as a variable of the current date minus 30 days. All message types that are for transaction data scenarios should be selected as part of the transaction data variant (FIDCMT, ZFIREV, ORDERS, ORDRSP, INVOIC, ZINVRV, DESADV). **The archive file that is created by the transaction data archiving session must be backed up as part of the basis backup procedure for auditing purposes.**

If the number of IDocs in an archive session are more than the maximum number of documents in the archiving customising settings (see step 7 of the archiving procedure), the archiving date or message type in the variant can be changed to limit the number of IDocs, or the customising settings can be changed to allow for more IDocs to be archived in one session.

#### 6.3.2.4.2 Storing of archive files

**The procedure for storing transaction data archiving files on some backup medium will be detailed as part of the basis backup procedure.** Only the archive files for transaction data IDocs needs to be backed up.

### 6.3.3 ALE Capacity Planning Guide

Check the growth and plan for expansion. A short document can be produced as follows:

#### 6.3.3.1 Introduction

The aim of this document is to provide some guidelines when doing system capacity planning. This document will look at the ALE area and describe the impact of ALE data on capacity planning.

These guidelines can be used as part of the larger system wide capacity planning.

#### 6.3.3.2 ALE Related Data

ALE will impact on system sizing and capacity planning. The impact on system sizing and capacity planning can be determined by getting the volumes of transactions from the business and applying the guidelines given in this document.

##### 6.3.3.2.1 Application Documents

ALE application documents can be divided into two categories, master data and transaction data. Any ALE master data IDoc will create a master data object at application level. An ALE transaction data IDoc will create a transaction data object at application level.

These master data objects and transaction documents have to be taken into account in the overall sizing model, when doing system sizing and capacity planning.

The actual processing of the master data and transaction data also has to be taken into account.

##### 6.3.3.2.2 IDocs

ALE uses IDocs to transfer data from one system to another. The IDocs are created and stored in the database in the sending system, before being sent to the receiving system. In the receiving system the IDocs are stored in the database before being processed.

The size of the IDocs will vary according to the type of information that is sent in the IDoc.

The ALE Archive Procedure explains how to archive and delete these IDocs, but there is still some disk space needed to store the IDocs before they are archived and deleted. Depending on whether the IDocs that were archived have to be stored, some archive space will also be needed (e.g. backup tape or CD).

### 6.3.4 ALE Failure and recovery procedure

In case a system needs to be restored to a point back in time, how do you recover? An example document follows:

#### 6.3.4.1 Introduction

The aim of this document is to provide a strategy and detailed procedure for recovering ALE communications from a system failure. The various kinds of failures that can occur will be described and the procedures to recover from these failures will be described.

#### 6.3.4.2 Types of Failures

There are 3 types of failures that affect ALE to different degrees. These different types of failures will now be described before the actual recovery procedure for each type of failure will be described.

##### 6.3.4.2.1 Communication Failures



This type of error happens when the communication line between systems is down. This could be due to a network problem, a system being down, an RFC connection failing etc.

#### 6.3.4.2.2 Hardware Failures

A hardware failure happens when hardware fails, for e.g. a disk subsystem crash. The system halts when this type of error occurs.

#### 6.3.4.2.3 Soft “Failures”

This type of failure happens when important data is made useless by a user (this could be due to a table being deleted at database level etc.). The database was updated for an indefinite time after the error occurred, but the system must be restored to a status before the error occurred.

#### 6.3.4.3 Communication Failures

When a communication failure occurs (as described in section 6.3.4.2.1) the IDocs that have to be sent are waiting in the RFC queue. The ALE layer will keep on trying to send these IDocs until the communication line is up again. These IDocs will then be sent once communication is re-established.

This attempting to re-send IDocs until successful is done by scheduling batch jobs (these can be generated automatically or by scheduling the program **RSARFCEX**). When maintaining RFC destinations (**Tools->Administration->Administration->Network->RFC destinations / SM59**) and editing a RFC destination, choose **Destination->TRFC** options from the menu structure. The **Suppress background job if connection error** flag indicates whether an attempt will be made automatically to reconnect to the RFC destination or not. This option is usually flagged so that automatic attempts are not made to reconnect to a RFC destination.

Rather than attempting to reconnect to the RFC destination automatically, a batch job can be scheduled to periodically try to re-send information to a RFC destination where the communication failed. The program **RSARFCCP** needs to be run periodically to achieve this re-send of information where communication failed.

Communication errors can also be re-sent manually by using transaction **SM58**.

#### 6.3.4.4 Preconditions for Hardware and “Soft” Failures

The discussion around hardware and “soft” failures needs to be seen in the light of the following assumptions / pre-conditions for the solutions provided:

- A continuous backup concept is in place (proper online and off-line backups are taken at regular intervals, and adequate hardware redundancy is in place (hardware failures do not cause loss of data); and
- the database subsystem being used allows for point-in-time recovery of database (this should be the case for all databases that supports SAP R/3 from release 3.0D).

#### 6.3.4.5 Hardware Failures

When a hardware failure occurs, a database recovery has to be done. No data loss will take place with a continuous backup concept in operation (as will be the case in the Finesse environment). When a hardware failure occurs, the system involved will be halted until the hardware failure is recovered.

All outbound messages from the involved system will not be sent until the recovery is complete and the system becomes available again.

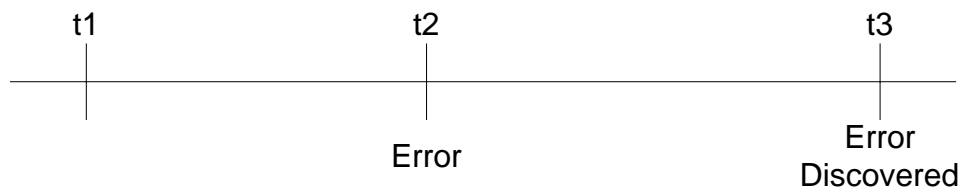
All inbound messages from other systems will be kept in the ALE queue of the sending system until the receiving system becomes available again.

No user intervention is therefore needed to ensure consistency across distributed systems for hardware failures.

#### 6.3.4.6 “Soft” Failures

These are the only types of failures that need extensive manual intervention from users to ensure consistency across distributed systems. These types of “soft” failures are absolute exceptions and are the worst case failures that can occur. This type of failure requires extensive user intervention even in single integrated systems (e.g. transactions need to be re-processed after determining what data has been lost). Even without ALE implemented in a distributed environment, this type of failure needs to be manually recovered in a distributed environment to ensure consistency across distributed systems. With ALE implemented, the problem worsens even further, with an additional component that has to be recovered across distributed systems.

##### 6.3.4.6.1 Description of the Problem

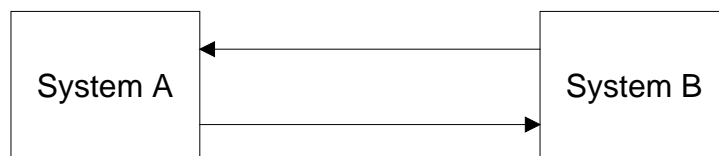


A soft error occurs when an error takes place at time t2 that makes data useless (e.g. a user deletes all orders). The error is only recognised at time t3 and the system is halted, after the state of the database has changed from t2 to t3. The error that occurred at t2 is of such a nature that the system has to be restored to a point in time (t1) before the error occurred.

All “useful” transactions between time t1 and t3 will have to be repeated. External interactions between systems will have to be synchronised with the state of the external systems.

##### 6.3.4.6.2 Soft Errors and ALE

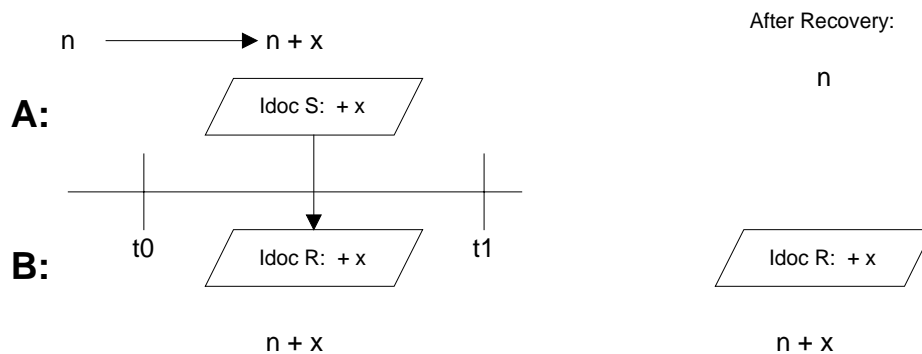
When two (or more) systems are using ALE to communicate, a soft error in one system will cause the systems to be out of synchronisation.



If an error is discovered in System A at time t2, System A has to be restored to an earlier status t1. System B will have IDocs sent from System A after time t1, and System B will have sent IDocs to System A after time t1. Cases of where data could be out of synchronisation are:

- Case 1: Transactional data. New documents will have been created in System B, but the original documents do not exist in System A any more (this affects both application documents and IDocs).
- Case 2: Master data. Copies of master data objects have been created in System B, but they differ from the originals in System A.
- Case 3: After restoring the system, IDocs will exist in both systems, but the IDocs have not been sent from System A.
- Case 4: Messages that depend on a previous message from System A to System B (e.g. order response from B->A depends on the message orders from A->B).

##### 6.3.4.6.3 Case 1: Transactional Data



System A applies delta + x to n. This information is sent using the sending IDoc on System A to the receiving system on System B, and the delta (+ x) is applied in System B. After the recovery, System A is restored to time  $t_0$ . The delta change on System A is lost, but have been applied on System B.

#### 6.3.4.6.3.1 Recovery Procedure

After System A has been recovered, all batch jobs must be put on hold to prevent transactions going into or out of System A.

**When recovering transaction data, a person from the business that fully understands the business process needs to be involved in cancelling business transactions (the cancelling of business transactions needs to be thought through carefully to determine the impact on the rest of the system from a functional point of view, and this procedure will not be explained as part of this document).**

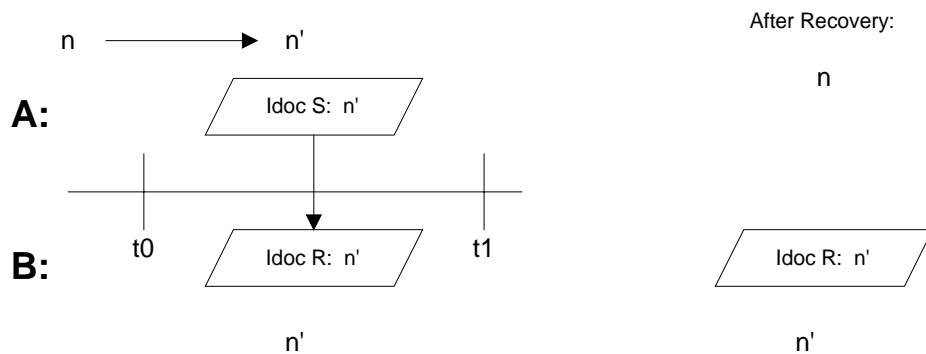
Documents in System B that depend on documents in System A, and which have been created after  $t_0$ , have to be cancelled.

All transactional IDocs in System B that have been created after  $t_0$  from sending system A have to be analysed. These IDocs can be found by using transaction **WE05 (Logistics->Central Functions->Distribution->Monitoring->IDoc Overview)** to list the IDocs that meet the criteria mentioned above.

For internal charges journals (IDoc FIDCMT), the application document number that was posted can be found in the status record of the IDoc (view IDocs through transaction **WE05**). These journals have to be reversed, and the IDocs that are created to reverse the journals on the sending system, have to be deleted (see the IDoc archiving procedure).

For internal charges purchasing & selling, all sales orders in System B has to be cancelled for purchase orders that were created in System A after time  $t_0$ . All Purchase orders in System B (for System A) that were created after time  $t_0$  also have to be cancelled. When cancelling the orders in System B, all related documents also need to be cancelled (order changes, reversals, invoices, credit notes).

#### 6.3.4.6.4 Case 2: Master Data



All master data objects that have been changed after time t0 in System A, will have a newer version in System B than in System A. These master data objects have to be determined and sent again from System A to get the systems in synchronisation. All changes to objects in System A after time t0 will have to be re-applied.

#### 6.3.4.6.4.1 Recovery Procedure

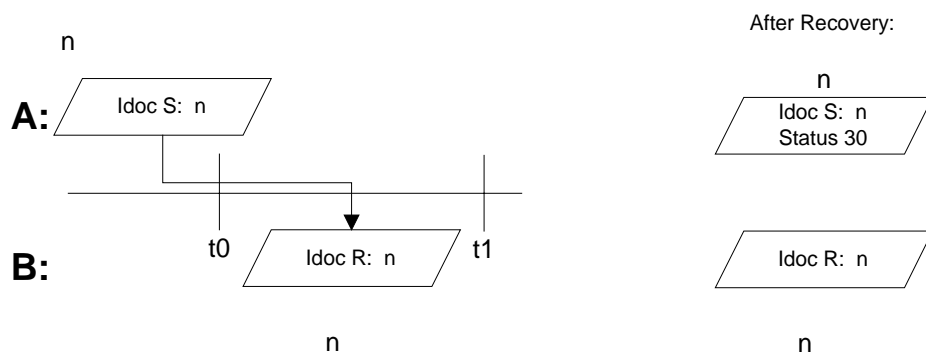
After System A has been recovered, all batch jobs must be put on hold to prevent master data objects going into or out of System A.

Documents in System B that depend on documents in System A, and which have been created after t0, have to be resent from System A.

All master data IDocs in System B that have been created after t0 from sending system A have to be analysed. These IDocs can be found by using transaction **WE05 (Logistics->Central Functions->Distribution->Monitoring->IDoc Overview)** to list the IDocs that meet the criteria mentioned above.

The master data objects can be determined by inspecting the data records of the IDocs. These master data objects have to be sent again from System A.

#### 6.3.4.6.5 Case 3: IDocs in Communication



All IDocs that have been created in System A before time t0, but sent after time t0, may not be resent. After the recovery, these IDocs will be in status 30 (IDoc ready to be sent), so the IDocs' statuses have to be changed corresponding to what happened on System B.

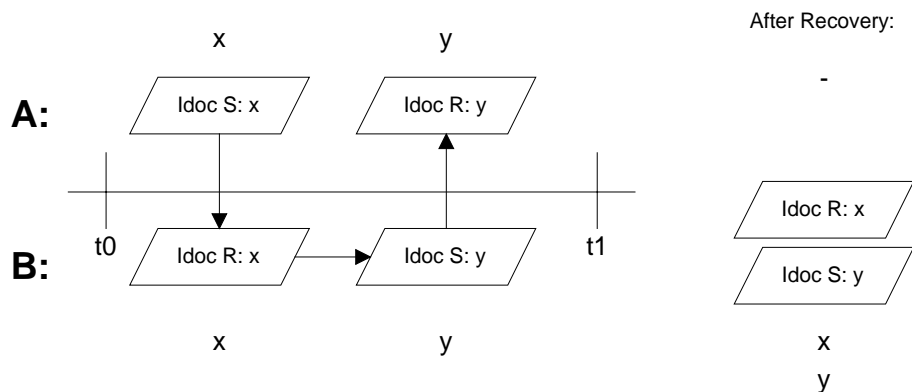
#### 6.3.4.6.5.1 Recovery Procedure

After System A has been recovered, all batch jobs must be put on hold to prevent IDocs going into or out of System A.

IDocs that have been created in System A before time t0, but were sent after time t0, may not be resent after System A has been recovered. These IDocs' statuses have to be reset corresponding to what has happened on System B.

The report **RBDMOIND** can be run to see which IDocs have not been sent to System B (IDocs in status 03). This report will change the status of all IDocs that have been sent to status 12. From transaction **BDM2 (Logistics->Central Functions->Distribution->Monitoring->IDoc Trace)** IDocs in System A can be matched to IDocs in System B after the recovery, to see which IDocs have been sent and which not.

#### 6.3.4.6.6 Case 4: Dependant Messages



IDocs that depend on messages, which are dependent on other messages sent after time t0, may not be sent/resent. These IDocs have to get a status that can be deleted (e.g. status 31) and have to be deleted from the TRFC queue id needed.

##### 6.3.4.6.6.1 Recovery Procedure

After System A has been recovered, all batch jobs must be put on hold to prevent transactions going into or out of System A.

**When recovering transaction data, a person from the business that fully understands the business process needs to be involved in cancelling business transactions (the cancelling of business transactions needs to be thought through carefully to determine the impact on the rest of the system from a functional point of view, and this procedure will not be explained as part of this document).**

Documents in System B that depend on documents in System A, and which have been created after t0, have to be cancelled.

All transactional IDocs in System B that have been created after t0 from sending system A have to be analysed. These IDocs can be found by using transaction **WE05 (Logistics->Central Functions->Distribution->Monitoring->IDoc Overview)** to list the IDocs that meet the criteria mentioned above.

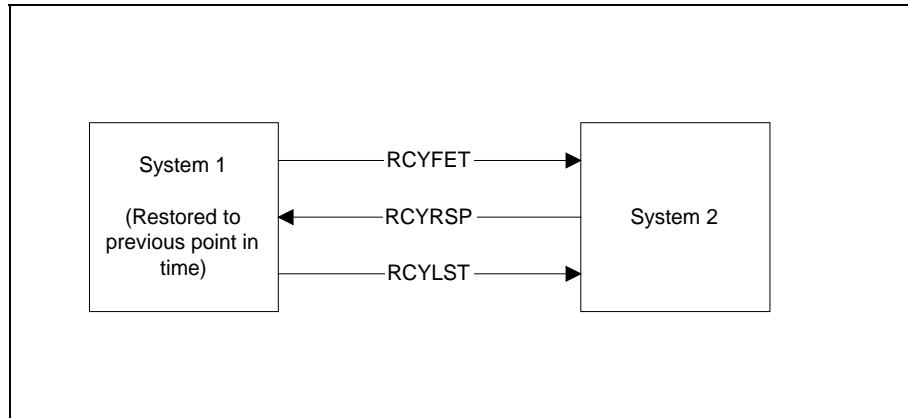
For internal charges journals (IDoc FIDCMT), the application document number that was posted can be found in the status record of the IDoc (view IDocs through transaction **WE05**). These journals have to be reversed, and the IDocs that are created to reverse the journals on the sending system, have to be deleted (see the IDoc archiving procedure).

For internal charges purchasing & selling, all sales orders in System B has to be cancelled for purchase orders that were created in System A after time t0. All Purchase orders in System B (for System A) that were created after time t0 also have to be cancelled. When cancelling the orders in System B, all related documents also need to be cancelled (order changes, reversals, invoices, credit notes).

#### 6.3.4.6.7 SAP ALE Recovery Tools

SAP has provided an ALE recovery tool to aid in the recovery of distributed system coupled by ALE. This tool is available as a path on OSS (note number 0105101).



This tool will help to compare IDocs between two systems coupled by ALE (one of which had to be restored to a previous point in time). The tool will show all IDoc discrepancies between systems, and will allow the user to delete IDocs that are unwanted. The user will still have to process the appropriate application documents as described in the rest of this document.



*Figure 18 Message flow between systems*

An ALE distribution model with message flows as shown in Figure 18 has to be set up (see the ALE configuration guide for information on how to set up a customer distribution model). This will allow the recovery tool to send information from the recovered system to another system, and also to get some information back from the other system. The recovery tool then uses this information to determine IDoc discrepancies between the two systems.

The following procedure will explain the steps that need to be taken to make use of the tool.

1. To investigate the status of IDocs between system 1 and system x, the messages in Figure 18 needs to be set up in ALE (see ALE configuration guide for details regarding ALE configuration). These messages can be set up for immediate processing.
2. Run transaction **BDRC** in system 1. This will start the transaction of the recovery tool that will transfer IDoc information between system 1 and system x.
  - The date and time fields should be filled in with the date and time that system 1 was restored to (to be safe this date and time can be entered to a few hours before the restored time). The last time field is just a margin of safety that must be built in. The two systems will be inspected for IDocs from this time and date minus the margin, for discrepancies between the systems.
  - The  **Beteiligte Systeme auswählen** button will give a list of systems that system 1 communicates with using ALE, where a specific system can be selected that will be inspected. It is suggested that one system x is selected at a time for synchronisation purposes.
  - The last button on the screen allows the user to edit ALE distribution models, which should have been set up already at this stage. The button has to be selected anyway, and the dialog box closed, before the transaction can be executed.
3. Execute transaction **BDRC** by clicking the  button. This will exchange information between system 1 and system x, to allow for later inspection of IDocs to determine discrepancies.
4. Run transaction **BDRL** in system x. This transaction will allow the user to inspect IDoc discrepancies between system 1 and system x.
  - The logical system for which IDocs must be inspected should be filled in (system 1) in the P\_PRTSYS field.

5. The **Anzeigen** button will give a list of all IDocs that were exchanged between system 1 and system x after the time and date specified in transaction **BDRC** that was run on system 1. These IDocs will be listed and can be looked at individually.
6. The **Ausführen** button will list all IDocs that are in system x, but have been lost in system 1 due to the restore. The IDocs can be inspected (IDoc Anzeigen), deleted (IDoc Bearbeiten), or the document could be reversed (Beleg Storniert) - although the reversal of documents didn't work when the tool was tested.
7. After all lost IDocs have been determined, the procedures described in this document should be used to get the systems into synchronisation again.

#### 6.3.4.6.8 General Comments

This section contains some general comments on some of the procedures described in previous sections.

##### 6.3.4.6.8.1 Synchronisation Tools Provided by SAP

SAP will provide some tools in future (general release in SAP R/3 version 4.0). These tools will automate some of the manual procedures described in previous sections. According to SAP, these tools will assist in monitoring IDocs across systems and indicate where distributed SAP systems are out of synchronization.

When these tools become available they will be included as part of the discussions in this document.

##### 6.3.4.6.8.2 Changing the Status of an IDoc

There is no standard functionality available in SAP to change the status of an IDoc. To change the status of an IDoc, a custom report will have to be written to change the status. This report can use the function module **IDOC\_STATUS\_WRITE\_TO\_DATABASE**.

It may be useful to write an error text to the new IDoc status that will be displayed in the IDoc monitoring.

Status 30 can be used to resend an IDoc, and status 31 can be used to prevent IDocs from being sent/resent.

##### 6.3.4.6.8.3 Investigating IDoc Links

IDoc links can be investigated to see the links between IDocs and application documents. When viewing IDocs (transaction **WE05**), links can be seen from the IDoc control record (**Goto->Links**).

Links are written in the table **SWW\_CONTOB** with the following information:

- OBJTYPE: type of object (e.g. IDOC);
- OBJKEY: number of the object;
- WI\_ID: ID of the link;
- Element: (e.g. SOURCE\_IDOC, OUTBOUND\_OBJECT);
- LOGSYS: logical system that owns the system.
- others...

The function module **SWW\_WI\_READ\_CONTAINERS\_OF\_OBJ** will also assist in programmatically finding IDoc links.

##### 6.3.4.6.8.4 Checklist for Client Recovery

The following items need to be checked from a technical ALE point of view, after a system has been restored from a soft error:

- The ALE distribution model needs to be checked (all message flows and filters in the model needs to be checked as well as the partner profiles). The model on the recovered system may be out of synchronisation with the source system of the model. If the recovered system is the source system of an ALE model, the target systems of the model may have a later version of the model. The ALE documentation can be used for this.
- ALE conversion rules have to be checked to ensure that these rules are the latest version. The ALE documentation can be used for this.
- ALE classes and the linking of the classes to logical systems. The ALE documentation can be used for this.
- ALE jobs need to be checked to see that all jobs are running after the system has been recovered. The ALE documentation can be used for this.

### 6.3.5 ALE Foreground Processing - By the user

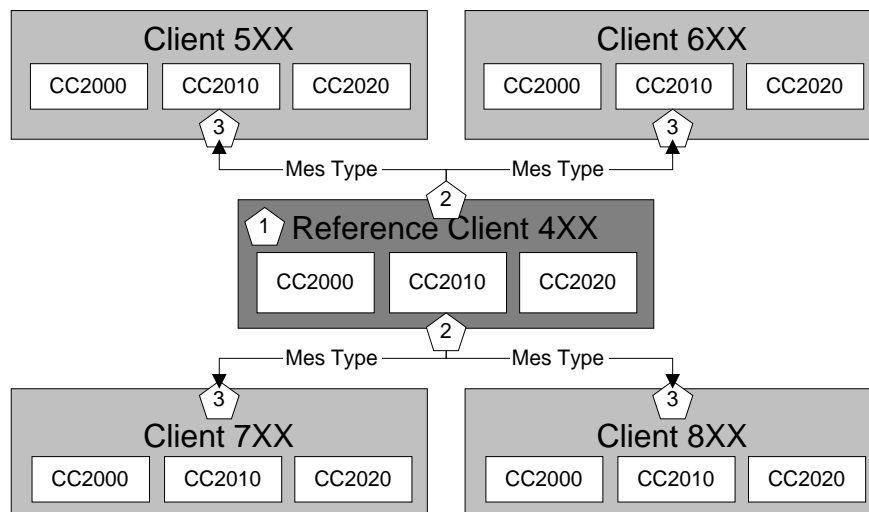
#### 6.3.5.1 Introduction

##### 6.3.5.1.1 Purpose

The purpose of this document is to give the ALE users (I.e. People who distribute master data records via ALE) the correct procedure to follow in order to minimise frustration through the understanding of the process.

##### 6.3.5.1.2 ALE Described

The following picture will be used to describe how the ALE process works:



*Figure 19: Describing the ALE Process*

With reference to Figure 19 the following process is applicable for ALE master data distribution:

1

- ALE related master data is added and changed in the 4XX range of reference clients
- Any additions or changes to the following master data records (I.e. Change pointers are created for each master record that is changed) are automatically distributed for the user (no user intervention is required):
  - Cost element & Hierarchy



- Cost Centre & Hierarchy
- Material Master (Create & Change) (Not internal charges materials)
- Debtor Master (Not internal charges customers)
- Vendor Master (Not internal charges vendors)
- Any additions or changes to the following master data records are not automatically distributed for the user and need to be manually sent:
  - Profit centers & Hierarchies
  - Material Master (Internal Charge Material \ Services)
  - Debtor Master (Internal charges customers)
  - Vendor Master (Internal charges vendors)

2

- Once the master data has been changed/added or manually sent, an IDoc will be created for each master data record. In the case of the automatic processing, this will only happen after hours when a batch job creates the IDocs from the change pointers.
- If the user requires a particular master data record urgently then he may process that IDoc manually from the reference client, which will then send the IDoc to the receiving system \ s.

3

- Once the master data has been received by the receiving system \ s, it is queued until it is automatically processed after hours.
- If the user requires a particular master data record urgently then he may process that IDoc manually from the receiving group client \ s, which will then process the IDoc on the receiving system \ s in order to create an application IDoc.

**NB: Do NOT process more than 20 records manually, due to the fact that it will affect the performance of all other users logged into the system during the time it takes to process all the IDocs.**

#### 6.3.5.2 SENDiNG master data

Follow this procedure to distribute your master data. Note that if you make a change to most of the master data records, it will automatically be processed overnight. In other words, a user can make a change on the ALE reference client and the following morning, without performing any further tasks, the master data records will be reflected in the respective group clients.

This document details how you would process master data immediately in such an environment.

6.3.5.2.1 Log on to the reference client 4XX

6.3.5.2.2 Create or change master data

6.3.5.2.3 Distribute master data

The master data scenario will be **automatically distributed (with the use of change pointers)**, overnight, for the following master data categories:

- Cost element & Hierarchy
- Cost Centre & Hierarchy
- Material Master (Create & Change)
- Debtor Master (Not internal charges customers)
- Vendor Master (Not internal charges vendors)

If you need to distribute the master data urgently, and it is of low volumes (<20) then you can send it manually. To do so, follow this procedure once your changes have been made.

#### 6.3.5.2.4 Manual Distribution for change pointers

If you want to specify what master data records you want to send and where then use the transaction **BALE** -> **Master Data** and **Send** the respective Master Data, as section: 6.3.5.2.5.

To distribute master data that you have changed or added (I.E. Via the use of change pointers, follow this procedure.)

##### 6.3.5.2.4.1 Generate IDoc Type from Change Pointers (On Reference Client 4XX)

- Start transaction **BD21** (Generate IDoc Type From Change Pointers)
- Select the required **message type** for required scenario. (See Appendix 1)
- **Execute** the function. A popup window will display how many master IDocs were created and how many communication IDocs were created.

##### 6.3.5.2.4.2 Dispatch IDocs Immediately (On Reference Client 4XX)

- Start transaction **BD88** (Processing Outgoing Intermediate Docs)
- **Check** the "Dispatch: **30**: IDoc ready for dispatch (ALE service)" radio button.
- **Execute** the function.
- Specify the required **message type** for your master data. A popup window will display how many IDocs were dispatched.

##### 6.3.5.2.4.3 Process IDocs Immediately (On Receiving Clients)

- Log in to the respective receiving clients.
- Start transaction **BD87** (Processing of incoming intermediate docs)
- **Check** the "Posting: **64**: IDoc ready to be passed to application" radio button.
- **Execute** the function.
- Specify the required **message type** for your master data. A message at the bottom of the screen will display how many IDocs are being processed.

#### 6.3.5.2.5 Manual Distribution when not using change pointers

If you want to specify what master data records you want to send and where then follow this procedure.

##### 6.3.5.2.5.1 Send the master data (On Reference Client 4XX)

- Start transaction **BALE**.
- Choose menu option **Master Data** and then the **relevant master data** entry
- Choose the **send** option
- Fill in the **required fields** for that particular master data scenario. Also fill in a receiving system if possible (See Appendix B)
- **Execute** the function. A popup window will display how many master IDocs were created and how many communication IDocs were created.

##### 6.3.5.2.5.2 Dispatch IDocs Immediately (On Reference Client 4XX)

- Start transaction **BD88** (Processing Outgoing Intermediate Docs)

- **Check** the “Dispatch: **30**: IDoc ready for dispatch (ALE service)” radio button.
- **Execute** the function.
- Specify the required **message type** for your master data. A popup window will display how many IDocs were dispatched.

#### 6.3.5.2.5.3 *Process IDocs Immediately (On Receiving Clients)*

- Log in to the respective receiving clients.
- Start transaction **BD87** (Processing of incoming intermediate docs)
- **Check** the “Posting: 64: IDoc ready to be passed to application” radio button.
- **Execute** the function.
- Specify the required **message type** for your master data. A message at the bottom of the screen will display how many IDocs are being processed.

### 6.3.5.3 *Viewing your sent data*

#### 6.3.5.3.1 Sending system (Client 4XX)

- Log in to the respective ALE reference client.
- Start transaction **WE05**.
- Specify an applicable **date and time range** when you want to view IDocs. Also specify your particular **Message Type**. Additionally you can narrow the selection by choosing the receiver or sender system as well.
- **Execute** the function.
- Your IDocs will be listed on the top half of the report (Total Direction Outbound) If they are displayed in green then they should have the status:
  - 03: Data Passed To Port OK, OR
  - 30: IDoc ready for dispatch (ALE service) - these will then be automatically dispatched over night. If you want to dispatch them now, and they are less than 20 in number, then proceed to Section: 6.3.5.2.4.2.
- If you want to see some detail of what happened to your IDoc. **Click on the row** of your message type and select the **Status List** button.
- A list of your IDocs will be displayed with a more detail status. To see further detail, double-click on the specific IDoc.

#### 6.3.5.3.2 Receiving system

- Log in to the respective group client.
- Start transaction **WE05**.
- Specify an applicable **date and time range** when you want to view IDocs. Also specify your particular **Message Type**. Additionally you can narrow the selection by choosing the receiver or sender system as well.
- **Execute** the function.
- Your IDocs will be listed on the bottom half of the report (Total Direction Inbound) If they are displayed in green then they should have the status:
  - 53: Application Document Posted, OR
  - 64: IDoc ready to be passed to application. These will then be automatically processed over night. If you want to process them now, and they are less than 20 in number, then proceed to Section: 6.3.5.2.4.3.

- If you want to see some detail of what happened to your IDoc. **Click on the row** of your message type and select the **Status List** button.
- A list of your IDocs will be displayed with a more detail status. To see further detail, double-click on the specific IDoc.

### 6.3.6 ALE Background Job Scheduling Procedure

Example of how to configure ALE for use with background jobs as opposed to immediate processing:

#### 6.3.6.1 Purpose of this Document

The purpose of this section is to provide a generic explanation on how to configure the background jobs used in ALE.

#### 6.3.6.2 Assumptions

The following assumptions are made when applying this procedure:

- Unless otherwise stated all IMG activities are performed at the SALE transaction level;
- The person applying this procedure is the ALE Administrator and has detailed knowledge of SAP job scheduling;
- The ALE basis environment is configured between the various clients. If this is not the case please follow the procedure documented in “*ALE Technical Configuration Section*”; and

#### 6.3.6.3 ALE background job description

Below is a pictorial representation of the flow of a complete ALE scenario from the sending system to the receiving system with emphasis on the jobs used to perform the distribution.

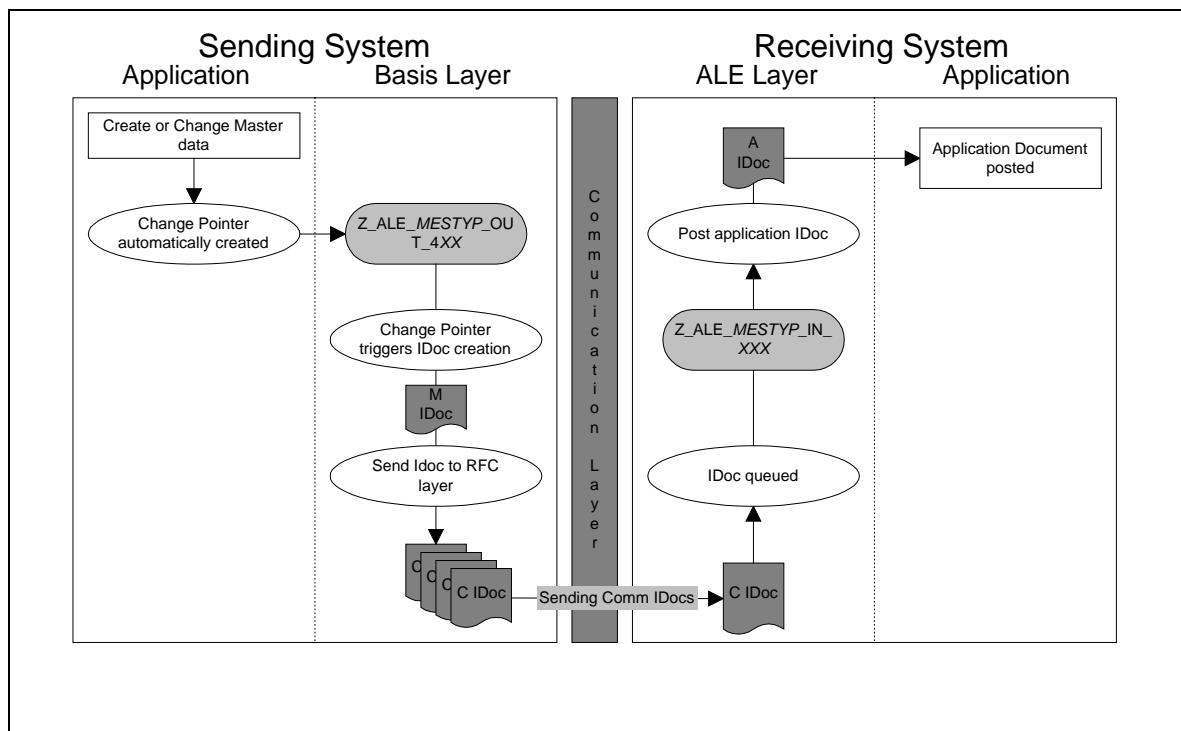


Figure 20: ALE background job description

#### 6.3.6.4 ALE Background job scheduling – Outbound on reference client (4XX)

This section of the document will describe the generic steps needed to configure the SAP system for ALE jobs on the sending systems. For the detail of the required data parameters please refer to the ALE Detailed Config Guide in conjunction with this document.

The outbound job is named as follows: Z\_ALE\_MESTYP\_OUT\_4XX where MESTYP is the applicable message type and 4XX is the applicable reference client on which you are scheduling the job.

For the purposes of this discussion we will use 1 job to process 1 message type.

The outbound job has the following 2 programs as steps, shown in Figure 20:

- RBDMIDOC – Change pointer leads to creation of IDocs
- RSEOUT00 – Some IDocs are not sent immediately, but are collected and then sent all together. The report RSEOUT00 must be scheduled in order that these IDocs can be sent.

These 2 steps will be described in more detail below.

#### 6.3.6.4.1 RBDMIDOC - Change pointers

##### 6.3.6.4.1.1 Define variant (SE38) Client Independent

1. Create a variant for the **RBDMIDOC** report for every message type (for example, for material master, customer master and so on).
2. To do this, execute the function and enter "**RBDMIDOC**", click on **variants** and choose **Change**.
3. Enter a **name** for the variant and choose **Create**. Use **Z\_MESTYP** as the variant name.
4. Enter the **message** type (for example, COSMAS).
5. **Save** the variant.
6. Repeat the process until variants have been created for all master data message types.

#### 6.3.6.4.2 RSEOUT00 - Schedule IDoc dispatch

##### 6.3.6.4.2.1 Define variant (SE38) Client Independent

1. You can maintain different values for sending:
  - port,
  - partner type, partner role
  - partner number,
  - Message type.
2. SAP recommends variants with partner number and message type.
3. Create corresponding variants for the report **RSEOUT00**.
4. To do this, execute the transaction **SE38** and enter "**RSEOUT00**", click on **Variants** and choose **Change**.
5. Enter a **name** for the variant and choose **Create**. Choose **Z\_MESTYP** as your variant name.
6. Enter the **message type** (for example, COSMAS).
7. **Save** the variant.
8. **Repeat the process** until variants have been created for all message types.

##### 6.3.6.4.2.2 Schedule Job (SM36) Client Independent

###### **Requirements**

Variants for each message type must be maintained.

1. Run transaction **SM36**
2. Type in the name of the job: **Z\_ALE\_MESTYP\_OUT\_4XX**
3. Enter it as job class **B**

4. **Goto STEPS** and enter 2 steps:
5. **RBDMIDOC** and a **variant** that you have maintained according to the **MESTYP** you are now defining a job for.
6. **RSEOUT00** and a **variant** that you have maintained according to the **MESTYP** you are now defining a job for. Username: **ALE-BATCH**.
7. Schedule the job as a **periodic** job, where the period must be adapted to your requirements (see Detailed Configuration Guide).
8. **Save** the job.
9. **Create a job for every variant** defined beforehand.

#### **6.3.6.5 ALE Background job scheduling – inbound on xxx**

This section of the document will describe the generic steps needed to configure the SAP system for ALE jobs on the receiving systems.

The inbound job is named as follows: **Z\_ALE\_MESTYP\_IN\_XXX** where **MESTYP** is the applicable message type and **XXX** is the applicable group client on which you are scheduling the job.

There is 1 job per message type.

The inbound job has the following program as a step, shown in Figure 20:

- **RBDAPP01**– Processing the IDocs that are not imported immediately on arrival, but periodically in the background (settings in partner profile)

This step will be described in more detail below.

##### **6.3.6.5.1 RBDAPP01 - Schedule IDoc processing**

###### **6.3.6.5.1.1 Define variant (SE38) Client Independent**

1. Create a variant for the **RBDAPP01** report for every message type (for example, for material master, customer master and so on).
2. To do this, execute the function and enter "**RBDAPP01**", click on variants and choose Change.
3. Enter a **name** for the variant and choose Create. Use **Z\_MESTYP** as the variant name.
4. Enter the **message type** (for example, **COSMAS**).
5. **Save** the variant.
6. **Repeat the process** until variants have been created for all master data message types.

###### **6.3.6.5.1.2 Schedule Job (SM36) Client Independent**

###### **Requirements**

Variants must be maintained for each message type.

1. Run transaction **SM36**
2. Type in the name of the job: **Z\_ALE\_MESTYP\_IN\_XXX**
3. Enter it as job class **B**
4. **Goto STEPS** and enter 1 step:
5. **RBDAPP01** and a **variant** that you have maintained according to the **MESTYP** you are now defining a job for. Username: **ALE-BATCH**.
6. The job needn't have a start date as it needs to be automatically started 60 seconds after its corresponding job outbound job, on **4XX**, has completed successfully.
7. **Save** the job.
8. **Create a job for every variant** defined beforehand.

### 6.3.6.6 Housekeeping jobs

#### 6.3.6.6.1 Outbound Housekeeping

##### 6.3.6.6.1.1 RSARFCEX – Resending IDocs unsuccessfully sent

###### 6.3.6.6.1.1.1 Define variant (SE38) Client Independent

In this step, you create a variant for report RSARFCEX.

1. Perform the transaction **SE38** and enter "**RSARFCEX**"; Click on **Variants** and choose **Change**
2. Enter a **name** for the variant and choose **Create**. Choose **Z\_ALL** as your name.
3. **Delete the date** in the field '**Date**' to and from
4. **Maintain** the attributes for the variant
5. **Check Communication error**
6. **Check not yet processed**
7. **Check Termination (dump, E/A msg.)**
8. **Save** the variant

##### 6.3.6.6.1.2 RBDMOIND - Schedule Status conversion after successful RFC

When outbound IDocs have been successfully passed to the communication layer, they are assigned the status "data passed to port OK".

This status does not, however, indicate that the communication via a transactional RFC was successful. Therefore a report should regularly be started that checks whether the communication was successfully completed and if so, changes the status of the IDoc.

###### 6.3.6.6.1.2.1 Define variant (SE38) Client Independent

In this step, you create a variant for report RBDMOIND.

1. Perform the transaction **SE38** and enter "**RBDMOIND**"; Click on **Variants** and choose **Change**
2. Enter a **name** for the variant and choose **Create**. Choose **Z\_ALL** as your name.
3. **Delete the date** in the field '**IDoc creation date (from)**'
4. Specify a value in the field '**IDocs per Commit Work**'
5. (SAP recommends the number 100)
6. **Maintain** the attributes for the variant
7. **Save** the variant

###### 6.3.6.6.1.2.2 Schedule Job (SM36) Client Independent

###### **Requirements**

Variants must be maintained for each message type.

1. Run transaction **SM36**
2. Type in the name of the job: **Z\_ALE\_HOUSEKEEPING\_OUT\_4XX**
3. Enter it as job class **B**
4. **Goto STEPS** and enter 4 steps:
5. **RBDMIDOC** and a **variant** that you have maintained for all message types. Username: **ALE-BATCH**.
6. **RSEOUT00** and a **variant** that you have maintained for all message types. Username: **ALE-BATCH**.

7. **RSARFCX** and a **variant** that you have maintained according to the parameters allowed you are now defining a job for. Username: **ALE-BATCH**.
8. **RBDMOIND** and a **variant** that you have maintained according to the parameters allowed you are now defining a job for. Username: **ALE-BATCH**.
9. Schedule the job **periodically** as per the detailed configuration guide.
10. **Save** the job.

#### 6.3.6.6.2 Inbound Housekeeping

##### 6.3.6.6.2.1 RBDMANIN – Reprocess IDocs in error

The report RBDMANIN allows intermediate documents which have already been cancelled once with errors (e.g. Status: 51 error in transfer to application) to be read into the application again. It is also possible to schedule a job with selected parameters, in order to collect up intermediate documents which could not be processed, e.g. because of a locking problem.

##### 6.3.6.6.2.1.1 Define variant (SE38) Client Independent

In this step, you create a variant for report RBDMANIN.

1. Perform the transaction **SE38** and enter "**RBDMANIN**"; Click on **Variants** and choose **Change**
2. Enter a **name** for the variant and choose **Create**. Choose **Z\_ALL** as your name.
3. **Leave attributes blank**
4. **Save** the variant

##### 6.3.6.6.2.1.2 Schedule Job (SM36) Client Independent

##### **Requirements**

Variants must be maintained for each message type.

1. Run transaction **SM36**
2. Type in the name of the job: **Z\_ALE\_HOUSEKEEPING\_IN\_XXX**
3. Enter it as job class **B**
4. **Goto STEPS** and enter 3 steps:
5. **RBDAPP01** and variant **Z\_ALL** to process all IDocs
6. **RBDMANIN** and a **variant** that you have maintained for all message types. Username: **ALE-BATCH**.
7. **RBDSTATE** and variant **Z\_ALL** to send ALE audit messages as described in the ALE configuration procedure
8. Schedule the job **periodically** as per the detailed configuration guide.
9. **Save** the job.

#### 6.3.7 ALE Performance Testing & Tuning Guide

Some tests done to measure performance, together with the results and recommendations. An example would be as follows:

##### 6.3.7.1 GENERAL

##### 6.3.7.1.1 Introduction

There are various ALE configuration settings that can be “tuned” to optimise ALE performance for high volumes of master data / transactions. These settings will be explained in this document and



some recommendations will be made regarding these settings and configuring them optimally for performance.

#### 6.3.7.1.2 Mass processing

The number of IDocs that are to be created influences the settings made in the partner profiles for the various message types. The users need to give a clear indication of the volumes, frequency and timings of these bulk creations.

Various jobs need to be scheduled to process IDocs collected on the outbound side as well as to process them in the background on the inbound side. These settings will be defined in a separate document detailing the ALE performance tuning options and settings.

#### 6.3.7.1.3 ALE performance

IDocs are fundamental to Application Link Enabling; optimal performance in ALE processing is synonymous with optimal performance of IDocs.

In ALE processing IDocs are:

- created in the R/3 sender system
- transferred to the communication layer in the R/3 sender system
- used when R/3 sender and receiver systems communicate.
- updated in the receiver system

You can optimise IDoc handling in one or more of the following ways:

- Controlling IDoc Events
- Processing IDocs in Parallel
- Packet Processing
- Managing IDoc Communication

### 6.3.7.2 *Procedure to change settings*

#### 6.3.7.2.1 Mass processing

- Mass processing refers to bundles of IDoc packets, which are dispatched and processed by the receiving R/3 System. Only one RFC call is needed to transfer several IDocs. Performance is considerably better when transferring optimal packet sizes.

To define a mass processing parameter, select Communication -> Manual maintenance of partner profile -> Maintain partner profile. For a message type the parameters packet size and output mode can be defined.

If the output mode is set to "Collect IDocs", outbound IDocs of the same message type and receiver are sent in a scheduled background job or in the BALE transaction in appropriately sized IDoc packets. The IDocs can be dispatched in batch or in the BALE transaction code.

Some distribution scenarios cannot support mass processing of inbound IDoc packets. This is especially true if the application sending the IDocs uses the ABAP/4 command CALL TRANSACTION USING. In this case the outbound parameter PACKETSZ must be set to "1".

To get a list of function modules that can be mass processed, select Enhancements -> Inbound -> specify inbound module in ALE Customising. INPUTTYP is "0".

- The option of performing mass processing is provided for reasons of performance. This involves sending packets consisting of several IDocs to an application workflow.

The IDoc packets are put together using collective processing in the output processing before dispatch takes place. Each packet contains IDocs of a single message type intended for a single recipient.

These IDocs are then all communicated in a single step.

There are some scenarios that cannot deal with mass processing of incoming documents. This is always the case if the application that imports the data has to execute a batch input.

In such a case the packet size has to be set to 1 in the output processing.

#### 6.3.7.2.2 Schedule IDoc dispatch

Some IDocs are not sent immediately, but are collected and then sent all together. The report RSEOUT00 must be scheduled in order that these IDocs can be sent.

The following steps are necessary for this:

- Define variants for the report.
- Schedule jobs, each with the report and a variant as the step.

#### Further notes

Numbers are assigned internally for IDocs which have been sent. Number ranges have to be maintained for this purpose. You do this under menu entry Basic settings -> Maintain number ranges for IDocs.

#### 6.3.7.2.3 Schedule IDoc processing

Jobs for inbound processing are scheduled in this section. These jobs must be scheduled for the following situations:

- Processing the IDocs that are not imported immediately on arrival, but periodically in the background (settings in partner profile). Use Report RBDAPP01.
- Importing IDocs that are stored in the system, but which were not transferred to the application due to exceptional situations. (Report RBDMANIN). With selected parameters you can also schedule a job to collect IDocs that have not yet been processed, for instance, because of a locking problem.

To do this, the following settings are necessary:

- Define variants for a report.
- Schedule jobs, with the report and a variant as the step.

#### Further notes

The report RBDMANIN allows intermediate documents which have already been cancelled once with errors (e.g. Status: 51 error in transfer to application) to be read into the application again. It is also possible to schedule a job with selected parameters, in order to collect up intermediate documents which could not be processed, e.g. because of a locking problem.

Numbers are assigned internally for intermediate documents which have been sent. Number ranges have to be maintained for this purpose. You do this under menu entry Number ranges -> Maintain number ranges for intermediate documents.

### 6.3.7.3 PERFORMANCE TESTING RESULTS

#### 6.3.7.3.1 Test cases

The following test cases were conducted using a GLMAST message type.

No	Message Type	Number of IDocs	Outbound			Inbound	
			Collect IDocs	Imm	Packet size	Imm	Bckgd
1	GLMAST	921	No	Yes	1	Yes	No
2	GLMAST	921	No	Yes	50	Yes	No

3	GLMAST	921	No	Yes	200	Yes	No
4	GLMAST	921	Yes	No	200	Yes	No
5	GLMAST	921	Yes	No	1	Yes	No
6	GLMAST	921	Yes	No	200	No	Yes
7	GLMAST	921	Yes	No	200	No	Yes X 5 jobs
8	GLMAST	921	Yes X change pointers	No	200	No	Yes

*Table 1 : ALE Performance Test Options*

#### 6.3.7.3.2 Test results

The following results were achieved against the test cases:

No	Total Transfer time				No of IDocs	Outbound	Inbound
	Ave.	Max	Min	TOTAL		Total Time	Total Time
1	25 secs	72 secs	6 secs	16m 31s	921	16m 16s	16m 13s
2	23 secs	48 secs	5 secs	15m 38s	921	15m 21s	15m 32s
3	22 secs	48 secs	4 secs	15m 13s	921	14m 58s	15m 09s
4	NA	NA	NA	+13m	921	Queue: 1m 3s Job: 51s	Queue: 1m 19s Proc: 10m 4s
5	NA	NA	NA	+17m	921	Queue: 58s Job: 15m 33s	Queue: NA Proc: 15m 36s
6	NA	NA	NA	+13m	921	Queue: 1m 7s Job: 56s	Queue: 1m 3s Job: 11m 08s
7	NA	NA	NA	+13m	921	As above	Queue: Above Job: 12m 40s
8	NA	NA	NA			Queue: 32s Job: 1m 30s	As above

*Table 2 : ALE Performance Test Results*

#### 6.3.7.3.3 Comments

No	Comments
1	Captured all available dialog processes on the receiving side to process the IDocs. Very limited number of dialog processes used on the sending system for very short periods of time to make RFC calls. CPU utilisation was unnoticeable.
2	Captured all available dialog processes on the receiving side to process the IDocs. Very limited number of dialog processes used on the sending system for very short periods of time to make RFC calls. CPU utilisation was unnoticeable.
3	Captured all available dialog processes on the receiving side to process the IDocs. Very limited number of dialog processes used on the sending system for very short periods of time to make RFC calls. CPU utilisation was unnoticeable.
4	1 dialog process used to collect the IDocs. 1 background process to send the IDocs. The bottleneck is on the inbound processing side. The IDocs collected up waiting to be processed on the inbound side.

5	1 dialog process used to collect the IDocs. 1 background process to send the IDocs. The bottleneck is on the outbound as well as the inbound processing side. As fast as the sending side could send the IDocs the receiving side could process them.
6	On outbound side 1 dialog process used to collect the IDocs. On outbound side only 1 batch process is used to process the IDocs and send. On inbound side a few (2-3) dialog processes were used to update the IDoc table. (Not a continuous process like when processing IDocs in the foreground). 1 batch process was used to process the IDocs.
7	On outbound side 1 dialog process used to collect the IDocs. On outbound side only 1 batch process is used to process the IDocs and send. On inbound side a few (2-3) dialog processes were used to update the IDoc table. (Not a continuous process like when processing IDocs in the foreground). 5 batch process was used to process the IDocs.
8	On outbound side 1 dialog process used to collect the IDocs. On outbound side only 1 batch process is used to process the IDocs and send. On inbound side a few (2-3) dialog processes were used to update the IDoc table. (Not a continuous process like when processing IDocs in the foreground). 1 batch process was used to process the IDocs.

### 6.3.7.4 Recommended settings

#### 6.3.7.4.1 Master data

Mess. Type	Outbound		Inbound	Job name	Period
	Packet size	Coll vs Imm	Bkg vs Imm		
GLMAST	200	Collect	Background	Z_GLMAST	Nightly

#### 6.3.7.4.2 Transaction scenarios

Mess. Type	Outbound		Inbound	Job name	Period
	Packet size	Coll vs Imm	Bkg vs Imm		
ORDERS	200	Collect	Background	Z_ORDERS	Hourly
FIDCMT	200	Imm	Immediate	N/A	Imm

#### 6.3.7.4.3 Other

Change pointer job needs to run before the job to dispatch the IDocs is run.

### 6.3.8 ALE Production client cleanup procedure

When a client is created, what steps do you need to take to clean it up? When a system is brought down what do you need to check? An example of what to look at follows:

#### 6.3.8.1 Introduction

The aim of this section is to describe the procedure for preparing a production client for ALE. This section will not describe the full ALE configuration process, but will only describe the general tasks that need to be performed after the client has been copied to remove all unwanted data.

This document will not provide detailed descriptions of all the procedures involved, but will mention the menu paths and transaction codes applicable for each area. The detailed procedures can be found in the ALE configuration procedure documentation.

#### 6.3.8.2 Copying Clients

Whenever an SAP client is copied, all client dependent configuration and data from the source client is also copied to the target client. Some of the ALE configuration data and general ALE information is

logical system specific, and therefore is not relevant to the target client. This information has to be removed, as part of the preparation of the client for ALE.

The information that has to be removed from the target client before the client is used in a production fashion is:

- ALE conversion rules (rule definitions, rule assignments to logical systems);
- ALE distribution models (message flows, filters, distribution models, partner profiles);
- Classes (link classes to logical systems);
- Change pointers;

### **6.3.8.3 ALE Conversion Rules**

#### **6.3.8.3.1 Rule Definitions**

ALE rule definitions are not logical system specific, so the rules that get copied from the source client do not have to be deleted. These rules will most likely also be used in the target system, but should be checked and any unwanted rule definitions should be deleted.

Rule definitions can be found under the ALE IMG in **Functions for IDoc processing->Conversion->Define Rules / Maintain Rules (BD62 / BD79)**.

#### **6.3.8.3.2 Rule Assignments to Logical Systems**

The assignment of rules to logical systems will have to be deleted, since this is logical system dependent. These assignments of rules to logical systems will have to be redone for the new logical systems as applicable in the target client.

Rule assignments can be found under the ALE IMG in **Functions for IDoc processing->Conversion->Define settings for IDocs conversion (BD55)**.

### **6.3.8.4 ALE Distribution Models**

#### **6.3.8.4.1 ALE Distribution Models (message flows, filters, models)**

Any unwanted distribution models have to be removed from the target client. This includes all message flows, filters and the actual models.

ALE distribution model maintenance can be found under the ALE IMG in **Distribution customer model->Maintain customer model directly (BD64)**.

#### **6.3.8.4.2 Partner Profiles**

The partner profiles that have been copied from the source client are logical system dependent and have to be removed.

ALE distribution model maintenance can be found under the ALE IMG in **Communication->Manual maintenance of partner profiles->Maintain partner profile (WE20)**.

### **6.3.8.5 Classes (linking to Logical Systems)**

The links of classes to logical systems that is copied from the source client must be deleted.

Allocation of listings to logical systems can be found under the ALE IMG in **Distribution scenarios->Master data distribution->Distribution via listings->Allocate listings to a logical system (BD68)**.

### **6.3.8.6 Change Pointers**

Change pointers in a copied client also need to be deleted. These change pointers get copied from the source client and are irrelevant to the target client. The procedure for deleting change pointers is documented in the ALE Basis Configuration Section.



*CHAPTER 7*  
**GENERAL ISSUES AROUND ALE**

## 7 GENERAL

This section contains all general documents and presentations related to ALE. The following documents belong in this section:

### 7.1 Glossary

#### **Application link enabling (ALE)**

Refers to the creation and operation of distributed SAP systems. The concept is to guarantee a distributed, yet integrated solution, through the use of business application messages and consistent data across loosely coupled systems.

#### **Basic IDoc type**

A basic IDoc type can be used on its own or in combination with an extension type. The combination of a basic IDoc type and an extension type forms an IDoc type.

#### **Control record**

One of the three record types of the IDoc. The control record contains data identifying the sender, the receiver, and the data structure.

#### **Communication idoc**

An IDoc contains the business data in the format required for ALE to operate. It consists of segment types, where each segment type holds the required fields for that segment. There are 2 types of IDocs we deal with:

- **Basic IDoc:** This IDoc is used internally by the function modules creating the IDoc to establish the data that is required for distribution. A basic IDoc is created per business object. E.g. Per material master record.
- **Communication IDoc:** For each basic IDoc that is created on the sending system the relevant number of communication IDocs are created based on the customer model and how many systems are to receive this IDoc. This is then the IDoc that is sent to the receiving systems and can be viewed by transaction WE05.

#### **Conversion rules**

These are used to convert the field contents of a sender field for use in a receiver field. This allows conversions of organisational units, of units of measurement and customer-specific conversions from one system to another to be performed.

#### **Customer distribution model (CDM)**

The customer distribution model specifies which applications can communicate with each other in your distributed systems. The customer distribution model contains all the cross-system message flows in your company.

#### **Filter object type**

A filter object type is a selection criterion in the distribution reference model that specifies the criteria that determine whether a function type can be used several times in a distributed system. *E.g. use it to specify the logical systems that are to receive a message (GLMAST) by stating the company code BUKRS as a filter object with 2000 as the value. This implies that only GL master records with 2000 as the company code will be distributed to the logical system with BUKRS = 2000 set as a filter object of GLMAST in the distribution model.*

#### **Function module**

Function modules are external subroutines that are stored centrally in the Function Library and can be called from any ABAP/4 program.

In contrast to normal subroutines, function modules have a uniquely defined interface.

#### **IDoc outbound processing**

Usage of programs and subprograms which control the processing of application data up to the point of transferral as an IDoc to the ALE communication layer.



## **IDoc type**

The IDoc type indicates the SAP format that is to be used to interpret the data of a business transaction.

An IDoc type consists of the following components:

- a control record

This is identical for each IDoc type.
- several data records

One data record consists of a fixed key part and a variable data part. The data part is interpreted using segments, which differ depending on the IDoc type selected.
- several status records

These are identical for each IDoc type and describe the statuses an IDoc has already passed through or the status an IDoc has attained.

## **Listings**

Listings are used in master data management when distribution rules cannot be modelled using organisational units only. Listings exist for material, customer and vendor master data.

For each object type a (customer-defined) class type can be specified as ALE-relevant. All classes belonging to this class type can be used in the listings.

Since listings are used as classifications, they are maintained in the master data transactions. A master data object can be added to a list by classifying it.

The listing groups together all the objects belonging to a single class, and this class belongs to the ALE class type for the object.

## **Logical system**

A system on which applications integrated on a common data basis run.

In SAP terms, this is a client in a database.

Messages are exchanged between the logical systems.

## **Message type**

The messages exchanged between systems are of various message types. The message type depends on the data contained and the process involved. It determines the technical structure of the message, the IDoc type. For example, the FIDCMT message type is used for journal messages.

## **Partner profile**

Definition of parameters for the message exchange of data with an ALE enabled partner via the ALE interface.

## **Port**

Logical description of the system that communicates with the SAP System during ALE message interchange.

In the case of outbound processing, this system receives the Intermediate Documents, processes them, and sends status records. In the case of inbound processing, it sends the Intermediate Documents to the SAP System, which then forwards them to the respective application.

## **Reference Client**

The reference client refers to the client that is used as the master data repository for that data that is to be distributed. I.E. If you are going to distribute the material master to 2 production systems then the reference client is that client which holds the material master data that was loaded directly by the user. The 2 production clients will receive the data via ALE and are referred to as destination clients. We suggest you separate the reference client from the production clients.

## **Remote function call (RFC)**

RFCs allow you to call and process predefined procedures (functions) in a remote SAP System. RFCs manage communication control, parameter transfer and error handling.

### SerialiSation

If several IDocs are sent within a short period of time then "overtaking" can occur, so that later changes arrive sooner than changes that were actually made earlier. In order to avoid incorrect updates being made, a serialisation mechanism must be used that can spot "overtaking" and report it.

## 7.2 ALE Diagrams of use

### 7.2.1 ALE Configuration setup diagram

See Appendix.

### 7.2.2 ALE Message Type Flow Diagram

See Appendix.

## 7.3 Setting up ALE to send messages from 1 client to itself

The purpose of this section is to describe the process of creating a dummy ALE client to permit the scenarios within a client. Typically ALE does not allow transfer of information from one client to itself. However, by creating a "dummy" client that appears to be logically separate, but is linked to the same physical system, the ALE system is able to transfer information (IDocs) within one system in the same way as it would between physically different systems.

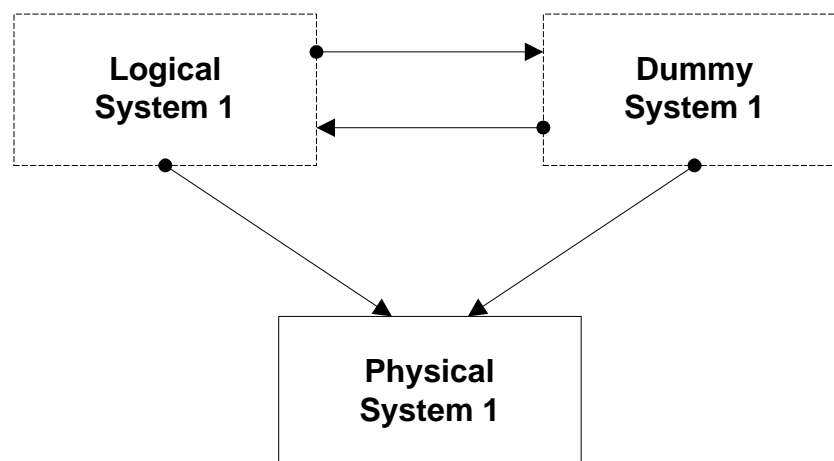


Figure 21 : Sending ALE messages from a client to itself

The procedure for configuring the dummy client and setting up the purchasing and selling scenario is outlined in the following sections. Please note that this is entirely a configuration change, so this document replaces any technical specification that would be available.

### 7.3.1 Setting up the dummy client logical system

The dummy logical system should have been set up both in the ALE reference client (where all model data is maintained) and the system in which the group client exists. **The BASIS configuration team normally does this.** However, the existence of logical systems can be checked using the following two steps.

### 7.3.2 Maintain Logical System

This can be accessed from the IMG ALE area, or from transaction SALE.

Choose Distribution (ALE) → Basic Configuration → Set up Logical System → Maintain logical systems

For each new dummy system, check that an entry has been added to the table. For example, if the client is yyy, add the following entry:

Name	Description
DUMxxxCyyy	Dummy Client

### 7.3.3 Define RFC Destination

This can be accessed from the IMG ALE area, or from transaction SALE. Choose Distribution (ALE) → Communication → Define RFC Destination

Select R/3 Connections menu option and check that a new RFC destination with the same name has been created as the logical system defined in the previous step.

For example, if the client is yyy, and a logical system DUMxxxCyyy has been created in the previous step, the following data should have been entered:

### 7.3.4 Define RFC Port

This should be done as part of the dummy client configuration on the reference client as well the group client being configured. **The BASIS administration team will not do this.**

Distribution → Communication → Manual Maintenance of Partner Profiles → define Port

Ports → Transactional RFC

Create a new port definition in this structure, using the fields Port, Description, Logical destination.

For example, for dummy system DUMxxxCyyy, enter the following:

Receiver Port	Description	Logical System
A000000032	Dummy Client yyy	DUMxxxCyyy

### 7.3.5 Setting up model messages

The standard messages for the scenario should be set up in the model, using transaction SALE. The procedure can be found in the ALE configuration section.

Unfortunately, due to the non-standard nature of this configuration, both automatic as well as manual generation of partner profiles is required.

### 7.3.6 Automatic Generation of Partner Profiles

- The following messages should be added to the 4xx-ORDR model for ALExxxCyyy to DUMxxxCyyy on the reference client:
  - ORDERS
  - ORDCHG
  - ZINVRV
  - ORDRSP
  - INVOIC
  - FIDCMT
- The partner profiles should first be generated using the standard auto-generation procedure on the reference client (for SYNCH messages to be created).
- This model should now be distributed to the both ALExxxCyyy and DUMxxxCyyy.
- The partner profiles should now be generated on the client being configured (i.e. yyy) using the auto-generation procedure defined in the standard ALE configuration procedures.

### 7.3.7 Creation of Filter Messages

Filter messages should be created for all the messages (with corresponding vendor and customer numbers) in the 4XX-ORDR model on the reference client, and distributed to the client being configured.

### 7.3.8 Create Partner Profiles

Although partner profiles are usually created automatically, because of the unusual inbound and outbound profiles needed on the same system, the manual generation of certain profiles is required.

The following subsections describe the partner profile settings for the standard client (assumed here to be ALExxxCyyy), as well as for the corresponding dummy client (assumed to be DUMxxxCyyy). The dummy client partner profiles should have been generated by the automatic generation process completed previously. Nevertheless, these should be checked to ensure that all profiles exist.

### 7.3.9 Standard ALE Client

#### Inbound

The partner profiles for inbound processing need to be set up manually on the standard client. The following table describes the inbound messages, and the message specific options that need to be set up for client ALEF01C700, partner type LS. All these messages should be set up for "Processing Immediately" with syntax checking.

Message	Proc. Code
ORDERS	ORDE
ORDCHG	ORDC
FIDCMT	ZALM
ORDRSP	ORDR
INVOIC	INVM
ZINVRV	ZINR

### 7.3.10 Dummy Client

The following partner profile parameters should be used to set up the dummy logical system, assumed to be DUMxxxCyyy. These should have been created by the automatic partner profile generation routine.

#### Message Control

The message control parameters for partner: DUMF01C700 and partner type LS are as follows:

App	Output	Message	Proc. Code
EF	NEU	ORDERS	ME10
EF	NEU	ORDCHG	ME11
V1	BA00	ORDRSP	SD10
V3	RD00	INVOIC	SD09

#### Outbound

The outbound parameters for partner: DUMF01C700 and partner type LS are detailed below. The receiver port should be selected as the dummy system. As mentioned previously, the default IDoc packet size should be determined from the official ALE configuration guide.

Message	IDoc
FIDCMT	ZFIDCM02
INVOIC	INVOIC01
ORDCHG	ORDERS02

Message	IDoc
ORDERS	ORDERS02
ORDRSP	ORDERS02
SYNCH	SYNCHRON
ZINVRV	ZINVRV01

#### Notes

Check Transaction **WE57** and ensure **Z\_IDOC\_INPUT\_ZINVRV** has an entry against object **BKPF**

#### 7.4 ALE User Guide

#### 7.5 Solution Pack

For each ALE scenario the following documentation should be produced:

- Business requirements document  
*How is distribution going to occur? Centrally vs decentrally. Immediate vs Delayed. What is going to be distributed? Where are these fields going to be maintained? Security considerations? Up time ito disaster recovery strategy.*
- Gap analysis document  
*What are the areas that differ between the actual process and the ALE process.*
- Configuration guide  
*This guide details how to configure the applicable scenario.*

If development was required then the following documents are needed in addition to the above-mentioned ones:

- Functional specification  
*Any changes to standard scenarios are specified by the business.*
- Technical specification  
*The functional specification is converted into technical specifications.*
- Quality assurance  
*Document ensuring that someone has assured the quality of the documentation, design and coding.*
- Testing results document  
*Document the test conditions as well as the test results for the implemented scenario.*
- Package sign-off  
*Form signing off the development work, once all the above documents have been completed.*

#### 7.6 ALE Related SAP programs

Name	Description
MSEIDOC2	IDoc Statistics
MSEIDOC3	IDoc Lists
MSESTA00	Status Maintenance (Tables TEDS!, TEDS2, TEDS3) via View
RBDAGAI2	Re-processing of intermediate docs after ALE input error
RBDAGAIE	Reprocessing of edited intermediate documents
RBDAGAIN	Re-processing incorrect intermediate docs. (outbound)

RBDAPP01	IDoc inbound processing ready for transfer
RBDAUD01	Statistical analyses for the ALE Audit
RBDAUD02	Reorganizing audit database - deleting table entries
RBDAUTHI	Include that contains all authorization checks
RBDAUTHO	Check authorizations
RBDCCMS1	IDocs for aRFC
RBDCHECK	Check IDOC Structure Against Database Tables
RBDCICO1	Monitoring Via Checkin/Checkout For The ALE Customer Model
RBDCNST2	Constants for Consistency Check
RBDPCCLR	Delete Change Pointers
RBDCUS01	Direct Customer Model Maintenance
RBDCUS02	Transport of ALE Tables for a Message Type
RBDCUS03	Cross-System Comparison of Number Range Objects
RBDCUS04	Sending Model to Logical Systems
RBDCUS15	Generating the partner profiles
RBDCUS16	Data: Generating The Partner Protocols
RBDCUS17	Routines: Generating The Partner Protocols
RBDCUS19	ALE Consistency Checking
RBDCUS23	ALE Consistency Check: Transfer Between Two Systems
RBDCUS24	Parameters for Report RBDCUS23
RBDCUS25	Interface control for RBDCUS23
RBDCUS26	ALE Consistency Check: Create Purchase Order from Sales Order
RBDCUS27	Data: Consistency Checking for SD Scenario
RBDCUS28	Routines: Consistency Checking for SD Scenario
RBDCUS31	Complete view of the message flow from view of one system.
RBDCUS33	Changing the Call Parameters for DA-OU
RBDCUS34	ALE Consistency Check
RBDCUS35	Modelling the control data distribution
RBDDATTR	Upload/Download Data from the Organizational Architect via Files
RBDEVACT	Activate Events for ALE/EDI
RBDFECLF	Fetch Classifications
RBDFECRE	Request Get Vendor Data
RBDIDOCA	Program for Object Type IDOC: EDI Intermediate Document
RBDIDOCR	IDOC Reduction
RBDINPUT	Processing of incoming intermediate docs.
RBDMANIN	Start Error Handling for Non-posted IDOCs
RBDMIDOC	Generate IDoc type from change pointers
RBDMMSD1	Consistency Check: Reassignment Between Two Systems

RBDMOIN1	IDoc list
RBDMOIN2	IDoc display
RBDMOIN3	IDoc overview
RBDMOIN4	Time Distribution of Intermediate Document Creation
RBDMOIN5	Statuses Reached
RBDMOIN6	Detailed Data on IDoc status
RBDMOIN7	Check: Customer model definitions <--> Partner profiles
RBDMOIN8	Cross-system IDOC Reporting
RBDMOIN9	Cross-system Analysis Whether IDOC Dispatch is Possible
RBDMOINB	Display: IDoc in sending system ==> IDoc in receiving system
RBDMOINC	Consistency check for input
RBDMOIND	Status Conversion for Successful RFC Execution
RBDMOINF	Consistency check for workflow for ALE/EDI inbound error processing
RBDMOINM	Model upload/download monitoring
RBDOUTPU	Processing outgoing intermediate documents
RBDPARPR	Conversion of process codes in existing partner protocols
RBDPRFTB	Value Tables for a Message Type
RBDPROSE	Converting ALE Settings for Productive Operation
RBDRCCLR	Reorganization of data in recovery objects
RBDRCLG	Display application log of recovery process
RBDRCOBJ	ALE: Process recovery objects
RBDREO01	Reorganize Long-Term Links
RBDSEALE	Message output, that SALE no longer exists
RBDSEALE1	Call of ALE IMG
RBDSDMM1	Consistency Check: Create Purchase Order From Sales Order
RBDSECRE	Send creditor data
RBDSEDEB	Customers - send
RBDSTATE	Sending Confirmations For The ALE Audit
RBDSYNEI	Continue processing IDoc despite syntax error (inbound)
RBDSYNEO	Continue processing IDoc despite syntax error (outbound)
RBDSYNER	Process IDoc despite syntax error
RBDTABCO	Table Analyses
RBDTBD22	Mapping between change document and IDoc
RBDTBD23	Required IDoc fields
RBDTBDA1	Activate Change Pointer
RBDTRAN1	Generates transport requests for ALE control data distribution
RBDTRAN2	Monitoring transport of control data
RBDTRANS	Program for object type TRANSID: Transaction ID

RBDTXT01	Text elements for customizing checks in distributed contracts
RSEAUTPA	INCLUDE RSEAUTPA: Central Routines for Auth. Check for 0EDS,0EDV,0EDY
RSEBASIS	EDI: INCLUDE for Alternative Structures and Access Routines
RSEBIDOC	Program for Object Type IDOCBASIS: Basic Intermediate Document
RSEBSCHK	EDI: Check System Type (Basis or Application System)
RSECEDI1	EDI Constants/Exceptions
RSECHK01	Test IDoc Type Segments for Consistency
RSECHK02	IDoc Types: Consistency Check
RSECHK03	IDoc Type Consistency Test
RSECHK04	Break Down of IDoc Types into Segments
RSECHK05	Evaluate Occurrences of IDoc Type Segments
RSECHK07	Consistency Test for Partner Profiles
RSECODIN	IDoc: Display Program for Inbound Process Codes
RSECODOU	IDoc: Display Program for Outbound Process Codes
RSECODST	IDoc: Display Program for Status Process Codes
RSECODSY	IDoc: Display Program for System Process Codes
RSEIDC00	Program for object type IDOCSYIDOC : Meta-IDoc
RSEIDOC1	WF-EDI: Documentation record types
RSEIDOC2	IDoc list
RSEIDOC3	Documentation IDoc Record Types and IDoc Types (Parser)
RSEIDOC4	WF-EDI: Documentation Segments
RSEIDOC5	WF-EDI: Documentation Intermediate Document Type
RSEIDOC6	Documentation of IDoc Types (Overview)
RSEIDOC7	WF-EDI: Information System
RSEIDOCA	Main Program for Subprograms of the Program Group RSEIDOCx
RSEIDOCB	EDI Statistics - Extended Selection
RSEIDOCC	EDI Statistics - Error History
RSEIDOCD	IDoc Lists
RSEIDOCE	Detail Data on IDoc Status
RSEIDOCF	Time Distribution of IDoc Creation
RSEIDOCG	EDI: Convert current status in Intermediate Documents
RSEIDOCI	INCLUDE for List Formatting
RSEIDOCL	Global form routines for IDoc display
RSEIDOCM	CA-EDI: Active monitoring for IDoc processing
RSEINB00	Inbound Processing of Intermediate Documents (EDI)
RSEINB10	SAP-IDoc: Receiving IDocs from an R/2 System via CPI-C
RSEINB11	IDoc inbound: Transfer of IDocs from an R/2 System to the application
RSEINB50	IDoc Inbox: Activate inbox and process IDocs with status '50'



RSEINBDT	EDI Inbound Processing (Global Data Declaration)
RSEINBEV	IDoc inbound: Customizing - Activate event-event receiver coupling
RSEINBF0	EDI Inbound Processing (FORM Routines)
RSEINBF1	EDI Inbound Processing (Other FORM Routines)
RSEINBT1	Include RSEINBT1
RSELINKD	Call display method for object
RSELOGAD	EDI: Selection of Maintenance Interface for EDILOGADR
RSEMASSA	EDI: Mass Activation of Partner Profiles
RSEOSTAT	Program for Object Type STATRECORD : Status Record
RSEOUT00	Process All Selected Intermediate Documents (EDI)
RSEPCON1	Print Control Record of an Intermediate Document
RSEPDAT1	Print a Data Record of an Intermediate Document
RSEPDATA	Print all Status Records of an Intermediate Document
RSEPORTA	IDoc: Tree display program for EDI and ALE ports
RSEPSEGA	Print List of all SAP Segments of an Intermediate Document
RSEPSTA1	Print a Status Record of an Intermediate Document
RSEPSTAA	Print all Status Records of an Intermediate Document
RSEREPSG	Repair IDoc Segments
RSESTA00	Inbound Processing of Status Records (EDI)
RSESTA02	Program for object type IDOCSTATUS : Status record
RSESTA11	IDoc: Sending status records for inbound IDocs to R/2 System
RSESTADT	EDI Inbound Processing (Global Data Declaration)
RSESTAF0	EDI Inbound Processing (FORM Routines)
RSESTAF1	EDI Inbound Processing (Other FORM Routines)
RSESTAT1	Print a Status Record of an Intermediate Document
RSESTATA	Print all Status Records of an Intermediate Document
RSETESTP	Report for Test for Existence of Master Data
RSETEXT0	Program for Object Type IDOCTXTRAW: Text Message
RSEXARCA	Archiving Program for IDocs
RSEXARCB	IDoc archiving program for periodic implementation in background
RSEXARCD	Delete Program for Archived IDocs
RSEXARCL	Reload Program IDoc Archive
RSEXARCR	IDoc Archive Read Program
RSIDOCAM	Program for object type IDOCSTATIS : IDoc statistics
RSIDOCWF	Program for Object Type IDOC: EDI Intermediate Document
RSNASTED	Analysis of NAST Record for Output Type 6 (EDI)
SAPMSED0	WF-EDI: Maintain Partner Profiles
SAPMSED2	WF-EDI: Interface for Test Environment

SAPMSED3	EDI: IDoc - Segment Editor
SAPMSED5	EDI: Development of IDoc Types
SAPMSED7	IDoc: Test tool for IDoc inbound processing
SAPMSED8	IDoc documentation tools menu

### 7.7 ALE Related Tables

Name	Description
BDAUDQUEUE	ALE Audit queue with not-processed Intermediate Documents
BDAUDSTATE	Statistical key figures for the ALE Audit
BDCP	Change pointer
BDCPS	Change pointer: Status
BDSER	Serialization: last date of object at sender
CIMHIS	EDI: Predecessors of Extension Types
CIMSYN	Table for Syntax Description of IDoc Type Extensions
EDADM	EDI client-specific system parameters
EDCIM	EDI: Value Table for Customer Extension Types
EDCIMT	EDI: Short Description of Extension Types
EDE1T	EDI: Text table for outbound process codes (TEDE1)
EDE2T	EDI: Text table for inbound process codes (TEDE2)
EDE5T	EDI: Text table for error processing process codes
EDE6T	IDOC: Text table for process codes for inbound statuses
EDFI2	EDI: Last processed document in file
EDFIL	EDI: Last processed document in file
EDI_MONIT	WF-EDI: Structure for Default Values in EDI Monitoring
EDICONFIG	CA-EDI: Parameter table for user-specific configuration
EDID2	Data segment (IDOC) from 3.0C
EDID3	Data segment table (EDI IDocs) from 3.0
EDIDC	Control record (EDI Intermediate Document)
EDIDD	Data segment (EDI Intermediate Document)
EDIDO	Value table for IDoc types
EDIDOT	Short description of IDoc types
EDIDS	Status record (EDI IDoc)
EDIE5	Assign logical address to physical destination in SAP
EDIFCT	IDoc: Assignment of FM to log. message and IDoc type
EDIFM	IDoc: Function modules for creating file names
EDIFMT	IDoc: Text table for EDIFM (funct. module for file names)
EDILOGADR	Assign logical address to physical destination (SAP)
EDIMAP	EDI: Assignment Table Between IDoc and Application Fields
EDIMSG	EDI: Message Types and Assignment to IDoc Types

EDIMSGT	Short description of SAP message types
EDIPHONE	CA-EDI: Telephony data for partner profiles
EDIPO	Table for description of ports for EDI
EDIPOI	Table for describing Internet ports
EDISEG	EDI: IDoc Type Segments with Assignment to Segment Type
EDISEGT	EDI: Short Description of Segments in EDSEA/EDISEG
EDISYN	Documentation Table for Generated Syntax of IDoc Types
EDIVAL	Reference From Check Table to Language-dependent Table
EDMAT	EDI: Texts for the 'Maturity Levels' in partner profile
EDMSG	Value table for SAP message types
EDP12	EDI PP: Outbound Table 2 (Find Process Code)
EDP13	EDI PP: Outbound Table 3 (Log. Message)
EDP21	EDI PP: Inbound Table 1 (Find Code f. Process Type)
EDPI1	EDI: Partner profiles (inbound)
EDPO1	EDI table for partner profiles (outbound), level 1
EDPP1	EDI PP: List of EDI Partners (Inbound and Outbound)
EDSEA	EDI: Table of all segments of current release
EDSEG	EDI: Convert segments to 3.0
EDSVRS	EDI: Table of all segments of current release
EDSYN	EDI: Convert IDoc type syntax to 3.0
IDOCHIS	EDI: Predecessor of basic IDoc types
IDOCSYN	CA-EDI: Syntax description for basis IDoc types
IDOCVRS	EDI: IDoc type versions created by customers
TBD00	customer models
TBD03	Communication relation between systems
TBD10	Def., out of which IDOC fields a filter object is to be read
TBD12	Mapping message type -> serialization and link type
TBD13	Organizational units to be converted in IDoc fields
TBD14	Message type -> object type
TBD20	ALE: data for IDOC projection filter
TBD21	ALE: IDOC reduction
TBD22	ALE: mapping IDOC-fields to change document fields
TBD23	ALE: definition of mandatory segments/fields
TBD30	ALE: Conversion rules for IDOC segments
TBD50	Message-dependent attributes for IDOC input methods
TBD51	Attributes of IDOC input function modules
TBD52	Function modules for inbound ALE-EDI
TBD62	Assignment of change document field to message type

TBDA1	ALE active
TBDA2	ALE message active
TBDLI	Listings for a logical system
TBDLS	Logical system
TBDLST	Text for logical system
TBDLT	Lists for a logical system for ALE distribution
TBDME	ALE supplement data for EDI message type
TBDMS	Assignment of message type to IDoc type for ALE
TBDRSTATE	Objects in Recovery environment (ALE)
TEDE1	EDI process types (outbound)
TEDE2	EDI process types (inbound)
TEDE3	Function module, which displays EDIDS record
TEDE5	EDI Process codes for error handling
TEDE6	IDOC process codes for inbound statuses
TEDEF	Assign proc. type to funct. module that processes proc. type
TEDS1	IDoc status values
TEDS2	Short description of IDoc status values
TEDS3	EDI status values: Importance/qualification
TEDST	SAP EDI Form routines to read master data
TEDTT	EDI: Partner Types - Language-dependent Names
TEDWL	Channel definition for transceiver
TEUDES	Configuration of upload files
TEUPL	Configuration of upload files
TEUPLT	Name of configuration IDoc for transceiver
TKANA	Entries for channel definitions
TKANAT	Descriptions for channel definitions

### 7.8 ALE Transaction Codes

Name	Description
BDM7	ALE Audit: statistical analyses
BDM8	ALE Audit: Sending the confirmations
BDM9	Reorganizing the audit database
BD76	Model upload/download monitoring
BD64	Maintain model for subsystems
BD71	Distribute customer model
BD89	General overview of message flow
BD95	Additional data for message type (Owner of CDM)
BD21	Select change pointer (Generate IDoc from CP)
BD22	Delete change pointer

BD50	Activ. change pointer for mess.type
BD52	Activ.change pointer per change document item
BD61	Activate change pointer (genrally)
BD85	Consistency check for transfer (Purch org -> Supplier -> Mat)
BD86	Consistency check for sales (Sales org -> Dist Chan -> Mat)
BDM5	ALE consistency checks
BD78	Monitoring control data distribution
BD77	Distribution of control data (Generates transport requests)
WE08	EDI: File Processing Status (EDFI2)
WE17	EDI: Inbound status report
WE45	EDI: Maint. Interface Sel. EDILOGADR (Forwarding inbound)
WE46	EDI: Display Administration Table
WE10	EDI: Generic Interface EDILOGADR
WE12	EDI: Inb.Processing of an Outb.File
WE16	EDI: Inbound File (Trigger inbound processing)
WE18	WF-EDI: Interface for Test Environm.
WE19	IDoc: Test tool inbound
WE54	EDI: Function modules for path names
WE55	IDoc: Funct. Modules for Path Names
WE32	EDI: Information Retrieval (Fields, segments in IDoc types)
WE33	Field values for IDoc documentation
BDM1	Monitoring: IDocs generated
BDM2	Monitoring: IDocs with recipient (Cross system IDoc reporting)
BDM4	Monitoring: Recipient reachability (Dispatch possible?)
WE02	EDI: Display Intermediate Documents
WE05	EDI: Monitoring of IDoc Lists
WE07	EDI Statistics
WE44	EDI: Maintain Partner Types
BD20	IDoc passed to application (RBDAPP01)
BD73	Reposting of IDocs (ALE) (RBDMANIN)
BD75	Convert IDoc status (after successful RFC)
BD83	Send IDocs after an ALE error - outbound (RBDAGAIN)
BD84	Register IDocs after ALE input error (RBDAGAI2)
BD87	Process inbound IDocs
BD88	Process outbound IDocs
BDA1	Call RSARFCEX
BDA2	Call RSARFCCP
WE14	EDI: Test Result from Intern. Doc. (RSEOUT00)

BD53	Reduction of IDoc types
WE43	EDI: Maintain Status Record Display
WE47	Status Maintenance
BD65	Maintain IDoc type required fields
BD66	IDoc type field -> change document field
WE30	EDI: Maintain IDoc Syntax Descript. (Create IDoc type)
WE31	EDI: Maintain IDoc Segments
WE80	EDI: Change EDIDO / IDoc Types (IDoc type header data)
WE83	EDI: Change EDCIM / Extension IDoc Types
WE60	CA-EDI: Documentation IDoc Types
WE61	CA-EDI: Doc. IDoc record types
WE62	EDI: Documentation Segments
WE63	EDI: Doc. IDoc Type for Parser
WE64	EDI: Doc. IDoc Types in Overview
WE57	IDoc: Messages and Applic. Objects
BD51	Maintain function modules (inbound)
BD59	Allocation object type -> IDoc type
BD60	Additional data for message type (Link to FM)
BD67	Maintain methods (inbound) - Link process code to FM
OE2C	Process Code Inbound
BD54	Maintain logical systems
BALE	Application Menu - Distribution
SBPT	Administration Process Technology
WEDI	EDI Basis
SALE	IMG Application Link Enabling
WE15	EDI: Outbound From Message Control
BD63	Transport ALE tables to message type
BD69	Assign message type to IDoc
WE81	EDI: Logical Message Types
WE82	EDI: Assignment IDoc<->Message Type
OYSM	Number Range Port Definition
OYSN	Number Range For IDocs
OYSO	Number Range IDoc Types
OYSQ	Number range for R/2-R/3 coupling
BD70	Adjust number ranges (Cross syst comparison of No ranges)
BD68	Maintain lists
BD57	Maintain link and serialization type
BD94	OA - file transfer (Organisational architect)

BDM3	Monitoring: Check against model
BD82	Generate partner profiles
OYEL	EDI: Display for EDP21 (Inbound partner profile)
OYEX	EDI PP: Create for EDP21 (Inbound partner profile overview)
OYEH	EDI: Display for EDP12 (Partner profile message control)
OYER	EDI PP: Create for EDP12 (Message control overview)
OYEI	EDI: Display for EDP13 (Outbound partner profile)
OYEU	EDI PP: Create for EDP13 (Outbound partner profile overview)
WE20	EDI PP: Maintain Partner Profiles
OYE0	EDI: Maintain Port Definition arfc
OYE3	Display EDI Ports
OYE4	Display ALE Ports
OYE5	EDI: Maintain Port Def. R/2 Systems
OYE6	EDI: Display Ports R/2 Systems
OYED	EDI: Maintain Port Definition
WE21	EDI: Maintain Port Definition
OE3C	System Process Code
WE40	EDI: Process Code System
WE42	EDI: Process Code Inbound
OE1C	Process Code (Out)
WE41	EDI: Process Code Outbound
WE50	EDI: Text Display Proc.Codes Control
WE51	EDI: Text Maint. Proc.Codes Control
WE56	EDI: Process Code Status
WE58	EDI:Text Displ. Process Codes Status
WE59	EDI:Text Maint. Process Codes Status
WE48	EDI: Texts for Process Codes (Inb.)
WE49	EDI: Texts for Process Codes (Inb.)
WE52	EDI: Texts for Process Codes (Outb.)
WE53	EDI: Texts for Process Codes (Outb.)
BD80	Conversion pre-production/production
BDR1	Display recovery application log
BDR2	Reorganization of recovery data
BDRC	ALE: Specify recovery objects
BDRL	ALE: Process recovery objects
BD55	Maintain IDoc conversion rules
BD62	Define segment conversion rule
BD79	Maintain IDoc conversion rules

BD56	Maintain IDoc segment filtering
BD14	Send vendor
BD15	Open vendor (Get vendor - CREFET)
BD30	Distribute material object list (Send BOM)
BD31	Distribute document (structure) object list
WE84	EDI: Assignment IDoc and Appl.Fields
BDM6	Monitor: Check input workflow
OYSP	EDI: Conversion of Error Proc.to 3.0 standard tasks
BD72	Activate events
BD58	Convert organizational units to IDocs
BD90	Start DA-OU (PC Tool)
BA10	Subsystem Configuration
BA11	Config. Transceiver / Upload Files



