ALE, IDOC

**ALE Technology**: Application Link Enabling (ALE) is a technology to create and run distributed applications. ALE Technology is proprietary of SAP.

**ALE Objectives** - ALE incorporates controlled exchange of data messages ensuring data consistency across loosely coupled applications. ALE comprises of three layers. Application Services, Distribution Services, and Communication Services. Basic principle of ALE is to provide a distributed and fully integrated R/3 system. Each application is self-sufficient. The use of self-sufficient system implies a certain measure of data redundancy. Hence data has to be both distributed and synchronized.

**Difference Between ALE and EDI**: Normally we refer to EDI technology when a non-SAP system is one of the communication channel. ALE communication occurs from the SAP side and EDI from the non-SAP side. IDOCs use ALE and EDI to deliver the data to the receiving system. If the data needs to be exchanged between two SAP systems, then IDOC uses ALE technology. For the exchange of data between a SAP and Non SAP system, IDOC uses EDI subsystem to convert and deliver the data.

**ALE Communication Diagram**

1. **What is the difference Between IDOC type and Message type? How to link the IDOC type with Message type?**
   
   Message type gives the meaning of the IDOC. IDOC type gives the structure of an IDOC. 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 and the IDOC type. The IDoc type indicates the SAP format that is to be used to interpret the data of a business transaction. In the OO (Object Oriented) approach, **Message Type** can be referred to a **Class** and **IDOC Type** as an instance of the class **Message Type**. The linkage will happen in transaction **WE82**.

2. **Where the IDOC information gets stored?**

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDIDC</td>
<td>Stores the Control Record information an IDOC</td>
</tr>
<tr>
<td>EDID4</td>
<td>Stores the Data Records (version 4.6)</td>
</tr>
<tr>
<td>EDIDD</td>
<td>Data Seg (EDI Intermediate doc)</td>
</tr>
<tr>
<td>EDIDS</td>
<td>Stores the Status of an IDOC</td>
</tr>
</tbody>
</table>
3. What is process code? What are type of process codes

Process code refers to a workflow or a function module which helps in reading or writing data from/to Idoc. Process Codes are used in both ALE and EDI framework to identify the function module or API (Application Programming Interface) to be invoked for subsequent processing. Inbound as well as outbound interfaces use process code but for different purposes. Outbound process codes are stored in table TEDE1, while inbound process codes are stored in TEDE2.

Following are types of process code

<table>
<thead>
<tr>
<th>Process Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outbound Process Code</td>
<td>This will read application data and place data in Idoc</td>
</tr>
<tr>
<td>Inbound Process Code</td>
<td>This will Idoc and create corresponding application data</td>
</tr>
<tr>
<td>System Process Code</td>
<td>This will create work item in case some error occurs in idoc / application document processing</td>
</tr>
<tr>
<td>Status Process Code</td>
<td>This will handle error that occurs when a idoc is sent to some other system</td>
</tr>
</tbody>
</table>

4. How to reprocess an IDOC?

You can use the below Programs for IDocs Reprocessing:

- RBDMANI2 : Reprocess Idocs manually
- RBDMANIN : Posting of IDocs with Status 51
- RBDMOIND : Outbound Idocs status 03->12
- RSEOUT00 : For Processing 30 Status IDocs
- RBDAPP01 : For Processing 64 Status IDocs
- RBDAGAIN : Reprocess Incorrect Outbound IDocs
- RBDAGAI2 : Reprocessing of IDocs after ALE Input Error

5. How to trace the IDocs of the Receiving system from the Sending system?

Steps:

- Execute the transaction BD87
- Give the Outbound IDocs in the IDoc Number field and execute or Just Note the Start time and End time when u execute the Master Data Transaction say for eg.BD10 and specify the same in the Time Created input field of BD87, also give the Message Type and the Receiving System Names and execute.
- Select the Repsective Message Type and Press the Trace IDoc Button.

6. How to trace IDocs based on the data.

Steps:

- Execute Transaction WE09
- Put in the message type name in the field labeled ‘Logical Message’ Ex - DEBMAS
- Now we will move to the section ‘Criteria for Search in Data Records ’
- In this section we will put the Segment name in which we want to search the existence of data in the field labeled ‘Search in Segment’ EX - E1KNA1M
- In the field labeled 'Search in Field' we will put the field name of the segment in which we want to search the existence of data. Ex - KUNNR for searching IDOCs for certain customer number.
- Now in the field labeled 'For Value' we will put the search value. Ex - Customer number as 100 .
- We will now execute the transaction.
- The result can be a single Idoc or a list of multiple IDOCS depending upon the Search Criteria entered.

7. How to Send and Receive an IDoc in the same system.

Steps:

- Create a Dummy Logical System.
  - Goto T-Code SALE-> sending and Receiving Systems -> Logical Systems -> New entries. Enter SYSID_CLNT, but this one is Dummy so use the first 2 characters of the SYSID and prefix ‘D’ then underscore and then the Client number.
  - E.g.: If ERP_100 is the logical system of the R/3 then create ERD_100 as the dummy system.
- Create Port for the Original System, (ERP_100)
  - Goto WE21 and select Transactional Port and press the Create button. Name the Port as "SAP" concatenated with the SYSID in our example it would be SAPERP Select the appropriate version and enter the RFC dest of the system that you are working on in this case it will be 'ERP'.
- Create Partner Profile in partner type LS:
  - Receiver Side (Outbound to) : In Partner type LS name ERD_100 create the Outbound Parameters, give the Message type,
Receiver Port same as the port we created in step 2. Enter the Basic type.

- **Sender Side (Inbound From):** In partner type LS name **ERP_100** create the Inbound Parameters, give the appropriate message type and the process code.

**Now create the stand alone program to send the IDoc:**
- The program will at some point call the `MASTER_IDOC_DISTRIBUTE` function module. When you pass the EDIDC structure it will be populated as follows:
  - `i_edidc-mestyp = message type`.
  - `i_edidc-idoctp = basic type`.
  - `i_edidc-rcvprt = 'LS'`.
  - Concatenate 'SAP' `sy-sysid` into `l_port`.
  - `i_edidc-RCVPOR = l_port`.
  - `i_edidc-rcvpnm = 'ERD_000'`.
  - `CONCATENATE sy-sysid '_' sy-mandt` INTO `l_sndprn`.
  - `i_edidc-SNDPRN = l_sndprn`.
  - `i_edidc-sndpnr = 'LS'`.
  - `i_edidc-sndpor = l_port`.
- Observe that the Sender port and the receiver port is the same, this does the trick. The outbound Idoc is sent on the port SAPERP with the Sender as ERP_100 and receiver as ERD_100 and then the Inbound Idoc is also sent to the same port SAPERP with the Sender as ERP_100 and receiver as ERD_100.

8. What is message control and when do we use it?

Message control is a mechanism by which documents are output based on a selection criteria and requirements. This concept is applicable not only to EDI and ALE, but also to other output mediums (for example: print, fax). Message control determines the type of document, its timing, number, and the medium. NAST table stores output records. The conditions (selection criteria and requirements) for creating an output message are stored in condition tables. Search mechanisms are used through access sequences, output processes, and requirements to determine whether an application document qualifies for output.

Some of the tables for Message Control are:

<table>
<thead>
<tr>
<th>Transaction Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NACE</td>
<td>Condition Record Maintenance</td>
</tr>
<tr>
<td>VOK2</td>
<td>SD Message Control Components</td>
</tr>
<tr>
<td>VOK3</td>
<td>Purchasing message Control Components</td>
</tr>
<tr>
<td>VOFM</td>
<td>View Message Requirements</td>
</tr>
<tr>
<td>V/86</td>
<td>Condition Table Field Catalog</td>
</tr>
<tr>
<td>WE15</td>
<td>IDOC test : outbound from Message Control</td>
</tr>
</tbody>
</table>

9. How do you reprocess the failed trfc entries?

You can reprocess the failed trfc entries using the program RSARFCEX.

10. What is the transaction used to find the outbound and inbound process codes at one place?

WE64