SAP IDOC: Definition, Structure, Types, Format & Tables

What is an IDOC?

IDOC is simply a data container used to exchange information between any two processes that can understand the syntax and semantics of the data.

In other words, an IDOC is like a data file with a specified format which is exchanged between 2 systems which know how to interpret that data.

IDOC stands for " Intermediate Document"

When we execute an outbound ALE or EDI Process, an IDOC is created.

In the SAP System, I DOCs are stored in database. Every IDOC has a unique number(within a client).

- The Outbound Process
- The Inbound Process

Key Features

- IDOCs are independent of the sending and receiving systems. (SAP-to-SAP as well as Non-SAP)
- IDOCs are based on EDI standards, ANSI ASC X12 and EDIFACT. In a case of any conflict in data size, it adopts one with greater length.
- IDOCs are independent of the direction of data exchange

e.g. ORDERS01: Purchasing module: Inbound and Outbound

- IDOCs can be viewed in a text editor. Data is stored in character format instead of binary format.

Structure of an IDOC
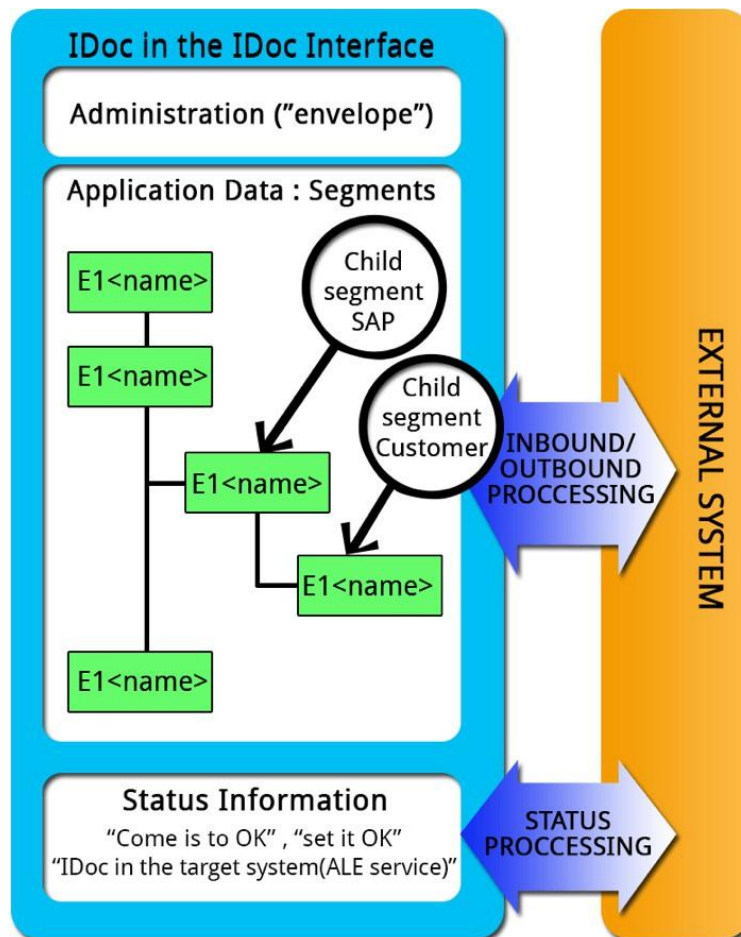
The I doc structure consists of 3 parts –

1. The administration part(Control Record)- which has the type of idoc, message type, the current status, the sender, receiver etc. This is referred to as the Control record.
2. The application data (Data Record) – Which contains the data. These are called the data records/*segments*.
3. The Status information (Status Record)- These give you information about the various stages the idoc has passed through.

You can view an I-DOC using transaction WEO2 or WEO5

Display basic type: ORDERS01
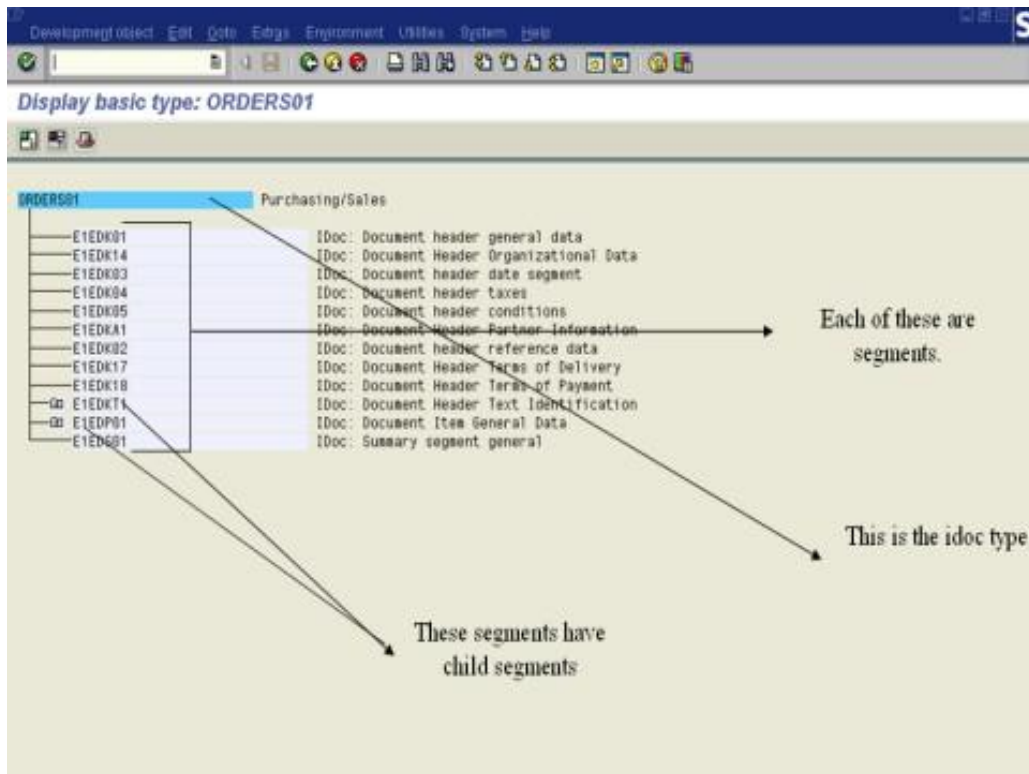
As seen the screenshot above IDOC record has three parts Control, Data and Status. Let's look into them in detail – Control Record

- All control record data is stored in EDIDC table. The key to this table is the IDOC Number
- It contains information like IDOC number, the direction(inbound/outbound), sender, recipient information, channel it is using, which port it is using etc.
- Direction '1' indicates outbound, '2' indicates inbound.

Data Record

- Data record contains application data like employee header info, weekly details, client details etc
- All data record data is stored in EDID2 to EDID4 tables and EDIDD is a structure where you can see its components.
- It contains data like the idoc number, name and number of the segment in the idoc, the hierarchy and the data
- The actual data is stored as a string in a field called SDATA, which is a 1000 char long field.

Status Record

- Status record is attached to an I-DOC at every milestone or when it encounter errors.
- All status record data is stored in EDIDS table.
- Statuses 1-42 are for outbound while 50-75 for inbound

IDOC Types

An I DOC Type, (Basic) defines the structure and format of the business document that is to be exchanged. An IDOC is an instance of an IDOC Type , just like the concept of variables and variables types in programming languages. You can define IDOC types using WE3O

What is a Segment?

A Segment defines the format and structure of a data record in I-DOC. Segments are reusable components.

For each segment SAP creates

- Segment Type (version independent)
- Segment Definition (version dependent)
- Segment Documentation

The last 3 characters is the version of the segment

Definitions keep changing as per the version but the segment type remains the same.

Transaction:WE31

Development segments: Display segment definition E2EDKT1002

Segment type attributes			
Segment type	E1EDKT1	☑ Qualified segment	
Short Description	IDoc: Document Header Text Identification		

Segm. definition	E2EDKT1002	☑ Released
Last changed by	SAP	

Po...	Field Name	Data element	ISO c...	Ex...	
1	TDID	EDI4451_A	☐	4	
2	TSSPRAS	EDI3453_A	☐	3	
3	TSSPRAS_ISO	LAISO	☑	2	
4	TDOBJECT	TDOBJECT	☐	10	
5	TDOBNAME	TDOBNAME	☐	70	

What is Extension IDOC type?

An IDOC is of 2 types:-

1. Basic
2. Extension

Develop IDoc Types: Initial Screen

Change Requests (Organizer)

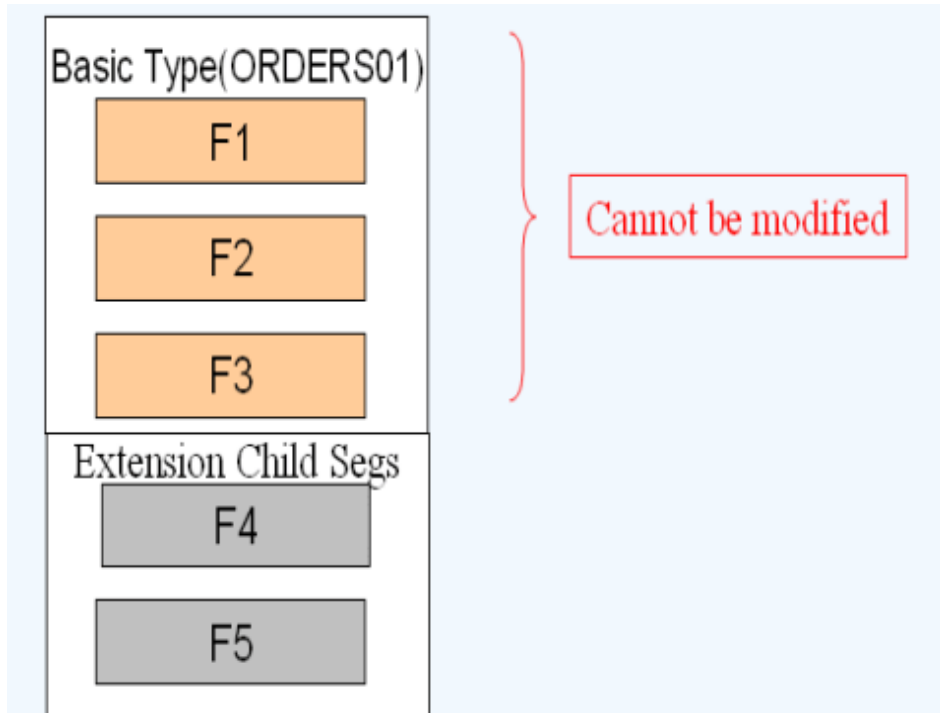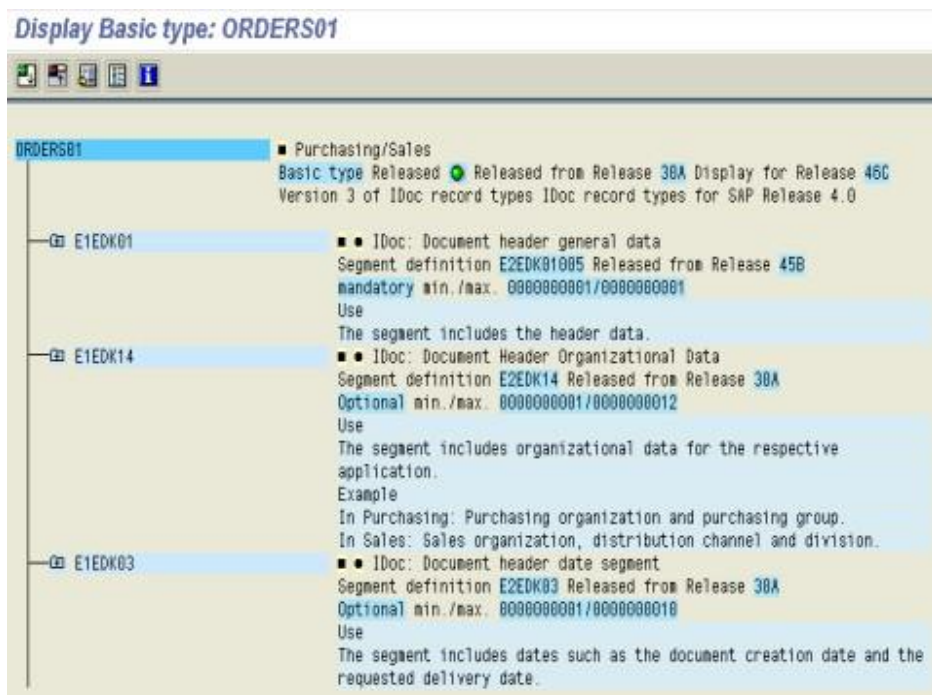Obj. Name

Development object

◉ Basic type
○ Extension

SAP provides many a pre-defined Basic IDOC Types which can not be modified. In case you want to add more data to these restricted basic type you may use an extension type. Most of the times you will NOT use extension.

Documentation

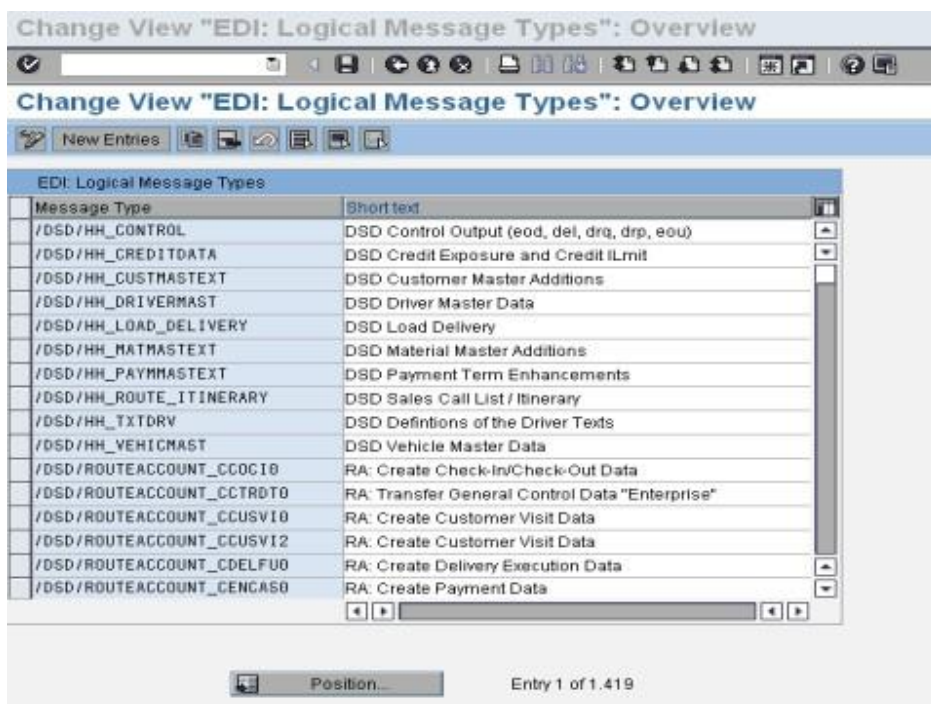Each IDOC are thoroughly documented in transaction WECO

Message Type

A message represents a specific type of document that is transmitted between two partners Ex. Orders, orders responses, invoices etc

An idoc type can be associated with many message types

Also, a message type can be associated with different idoc types. Transaction WE81

IDOC Views

An IDOC type can be used for more than one message type, which results in IDOCs containing more fields than required for a particular message type.
IDOC views are used to improve performance in generating IDOCs to ensure only the relevant segments are filled with data. IDOC Views are important only for Outbound Processing.
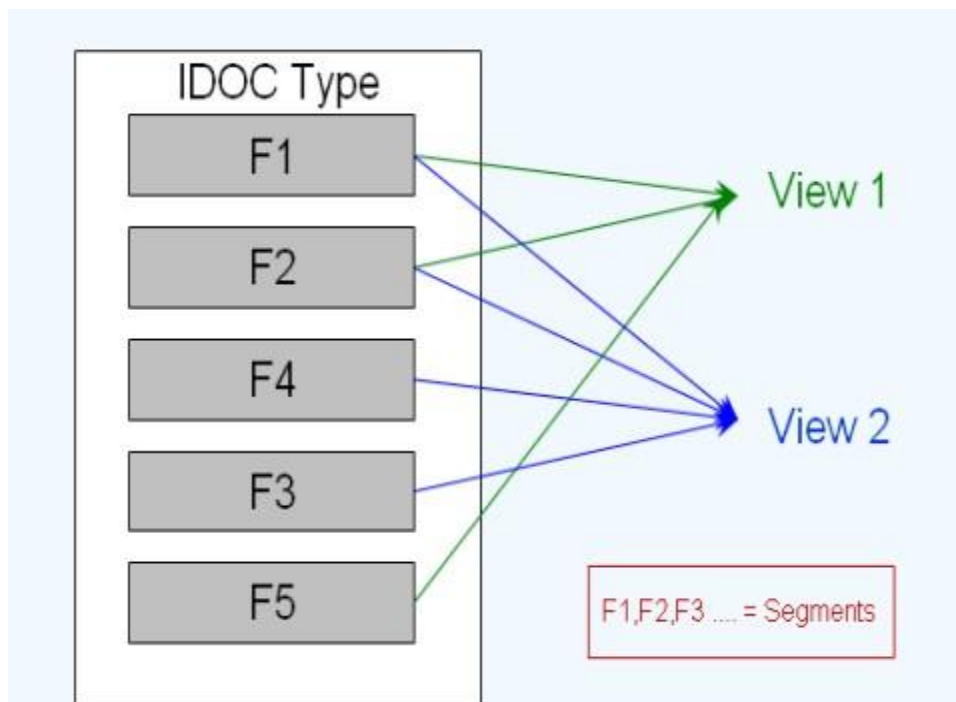
Partner Profiles

A partner is defined as a business partner with whom you conduct business and exchange documents

In the partner profile of a partner that we exchange Idocs with, we maintain the parameters that are necessary for exchanging the data. The transaction used is WE2O.

Port

The port defines the technical characteristics of the connection between your SAP system and the other system you want to transfer data with (subsystem). The port defines the medium in which data is exchanged between the 2 systems.

There are different types of ports. The 2 most commonly used are the TRFC ports used in ALE and File ports which EDI uses.

For TRFC ports we have to give the name of the logical destination created using SM59.

When using file port you can specify the directory where the IDOC file should be placed. The other system or the middleware will pick up the file from here. The Function module can be used to generate a file name for the idoc. While Testing you can use "Outbound file" to specify a constant file name. The tab "outbound trigger" can be used to supply information if we want to trigger some processing on the subsystem when an idoc is created at this location. We have to specify the command file name and the directory which has to be run.
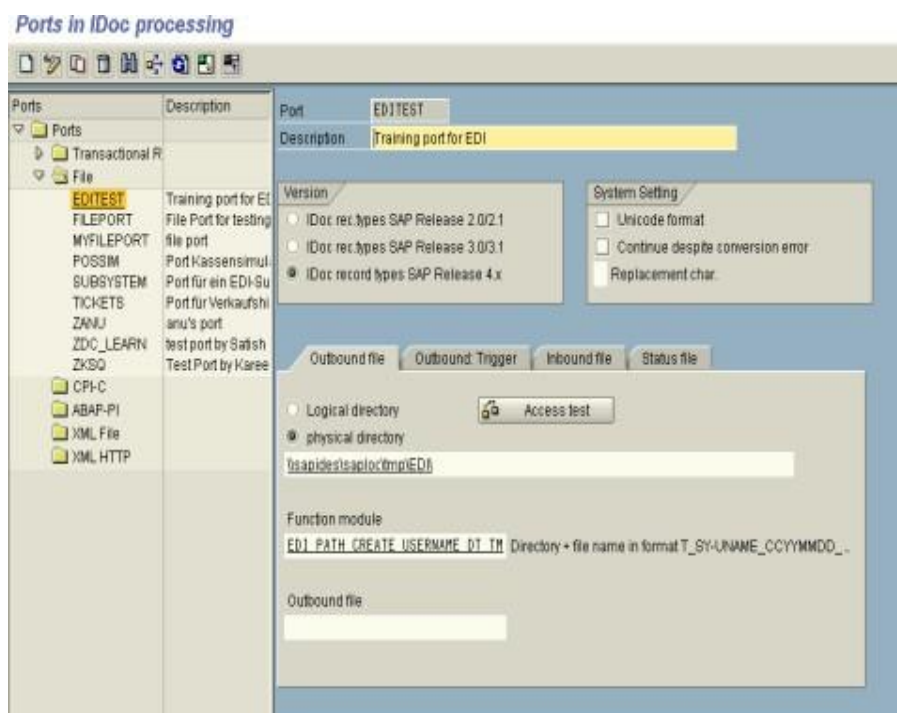
Let's understand the process of creating an IDOC with an example –

- Whenever a Purchase Order (PO) is created we want to send the IDOC to a vendor.
- The PO is sent in the form of an IDOC to the vendor (partner). That partner has to be EDI enabled in that system. SAP should realize that it could send doc to this vendor electronically.
- The PO sent as an outbound idoc by the customer will be inbound idoc for the vendor. The SAP system on the vendor's side can process this to create an application document (a sales order) on their system.
- Quotation, RFQ, PO, SO, Invoice, delivery note etc are some of the commonly exchanged documents through IDOC

The process of data transfer out of your SAP system is called the Outbound process, while that of data moving into you SAP system is called Inbound process. As a developer or a consultant who will be involved in setting up theses process for your organization. Here are the steps how to set them up-

The Outbound Process

Steps Involved –

1. Create segments(WE31)
2. Create an idoc type(WE30)
3. Create a message type (WE81)
4. Associate a message type to idoc type(WE82)
5. Create a port(WE21)
6. If you are going to use the message control method to trigger idocs then create the function module for creating the idoc and associate the function module to an outbound process code
7. Otherwise, create the function module or stand-alone program which will create the idoc

8. Create a partner profile(WE20) with the necessary information in the outbound parameters for the partner you want to exchange the idoc with. Trigger the idoc.

The Inbound Process

Steps Involved-

1. Creation of basic Idoc type (Transaction WE30)
2. Creating message type (Transaction WE81)
3. Associating the Message type to basic Idoc type (Transaction WE82)
4. Create the function module for processing the idoc
5. Define the function module characteristics (BD51)
6. Allocate the inbound function module to the message type(WE57)
7. Defining process code (Transaction WE42)
8. Creation of partner profile (Transaction WE2