



Jelena Perfiljeva

What on Earth is an SAP® IDoc?

- ▶ Fundamentals of IDoc interfaces and configuration
- ▶ Troubleshoot common post-implementation challenges
- ▶ Learn how to implement interfaces with ALE and EDI
- ▶ Quick reference guide to common IDoc transaction codes and reports

Table of Contents

Preface	7
1 Meet the IDocs	9
1.1 Definitions	9
1.2 IDoc anatomy	14
1.3 IDoc documentation	18
1.4 What is the deal with ALE and EDI?	21
1.5 When to use the IDoc interfaces and when you probably should not	22
2 Common IDoc interface elements	25
2.1 Port definition—WE21	25
2.2 Create new port—example	28
2.3 Logical system	29
2.4 Partner profile	30
2.5 Distribution model	32
2.6 Filters in the distribution model	33
2.7 Notes on IDoc display	36
3 Outbound interfaces	39
3.1 Outbound interface varieties	39
3.2 Example 1—send invoices to an EDI partner	41
3.3 Example 2—send purchase orders to external system	56
3.4 Example 3—send master data to an external system	73
3.5 Outbound interfaces—conclusion	87
4 Inbound interfaces	89
4.1 Example 1—receive EDI sales orders	89
4.2 Example 2—receive material master data from an external system	109
4.3 How does an IDoc get into an SAP system?	128

5 Proper care and feeding of the IDoc interfaces	131
5.1 IDoc interface ownership: IT or business?	131
5.2 IDoc interface monitoring options	132
5.3 Additional features in IDoc display transactions (WE02/WE05)	134
5.4 Wanted: IDoc	137
5.5 Searching for data within IDocs	139
5.6 IDoc processing and re-processing (BD87)	140
5.7 New IDoc Monitor transaction WLF_IDOC	143
5.8 Changing IDoc data	143
5.9 Output troubleshooting	147
5.10 Background jobs in the IDoc interfaces	152
5.11 DIY background job—example	155
5.12 Serialization	164
5.13 Archiving	164
6 When SAP Standard is not enough	169
6.1 IDoc extension	169
6.2 User exit example—outbound	180
6.3 User exit—inbound	190
6.4 Outbound file path—function module	197
Appendix A: How and where to search for additional information	201
Books	201
Online SAP Help and SCN	202
SCN blogs mentioned in the book	203
SAP Notes and KBAs	203
Appendix B: The sticky note version of the book	205
A The Author	208
B Index	209
C Disclaimer	211

2 Common IDoc interface elements

In this chapter, we will learn about the elements that are used in both inbound and outbound IDoc interfaces: port, partner profile, logical system, and distribution model. We will create the port and logical system that will be used in all further examples in this book. Depending on your preferences and experience level, you can choose to read the chapter entirely now or refer to the corresponding articles while reading about the specific interfaces in the next chapters.

2.1 Port definition—WE21

A communication *port* in an SAP system is quite similar to an actual sea port. Except instead of ships dropping off or picking up their cargo, it is information flowing in and out of an SAP system. And just like ships from many countries that arrive at the same sea port, one IDoc port can be used for multiple outbound and inbound interfaces.

Ports are maintained in transaction **WE21**. On the left-hand side (Figure 2.1) we can see a list of the port types.

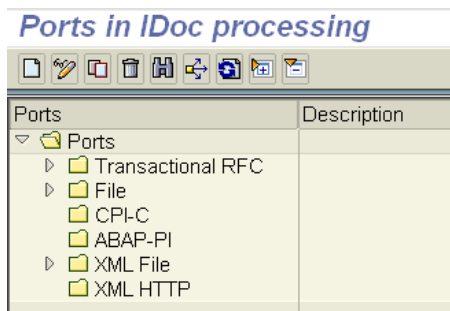


Figure 2.1: Port types

There is, of course, SAP documentation on the port definition but, in my opinion, it has too much information no one cares for yet lacks a clear and simple explanation of what all those port types are for. Although if SAP Help was perfect, then no one would need this book, so it is probably a good thing... Anyway, here is my “simplified” explanation of the available port types.

- ▶ **Transactional RFC**—This port type is used when two systems, or clients of the same SAP system, exchange data directly by establishing a remote connection and calling a special remote-enabled function (I briefly mentioned RFC in Chapter 1). The remote connection needs to be defined beforehand in transaction **SM59**. Note that the connection does not have to be between two SAP systems, it could be between an SAP system and a third-party application.
- ▶ **File**—This port creates a plain text file in the specified directory. The file will have the segment names at the beginning of each line followed by a stream of data (usually unreadable). This port type seems to be rarely used these days as more preference is given to RFC and XML types.
- ▶ **CPI-C**—This port is used for R/2 systems, which is an old version of SAP ERP.
- ▶ **ABAP-PI**—This port triggers a custom ABAP function. ABAP-PI is short for “ABAP Programming Interface” and has nothing to do with the SAP Process Integration solution (which has already been renamed twice and is now called SAP PRO). In the outbound interfaces, function module `OWN_FUNCTION` may be used as a template for the custom function. In the inbound interfaces, the custom function must call function module `IDOC_INBOUND_ASYNCHRONOUS` to create the IDoc.
- ▶ **XML file**—This port creates an XML file in the specified directory. In the XML file, you will see the IDoc segments as nodes. This port type is used as an example in this book because it is easy to deal with and works well for illustration purposes.
- ▶ **XML HTTP**—This port is sort of an XML / RFC hybrid. No file is created, but data is transmitted in XML format using an HTTP connection.

What about web services?



SAP does not provide a web service port for IDocs. One might argue that if a web service can be used, then why bother with IDocs at all? There is a great blog titled “IDoc as Web Service” on SCN by Amaresh Pani that describes a possible solution (without SAP PI/PO/PRO) and also mentions a valid business scenario in the comments (see Appendix A, page 203).

For the File and XML port types, there are two choices for the directory where the file will be created: *logical* and *physical*. What is the difference and when or why use one or the other?

Physical directory

If we choose physical directory name then we need to enter the actual directory pathname on the application server. SAP application server directories can be viewed in transaction **AL11**. Note that some shared network directories can also be added there.

It is a good idea to check with the system administrator which directory should be used in the port definition.

Most of the examples in this book use a physical directory for simplicity, but in real life it is better to use a logical directory.

Logical directory

If we choose the logical directory option, then we need to use a *logical file path*. Logical file paths (i.e., directory and file names) are maintained in the transaction **FILE**. In a nutshell, these paths allow the use of a neutral identifier instead of specific physical location.

Why would anyone need that? In some cases, it is convenient because we can maintain path definition once and then use the logical path name in many ports (or other places—logical paths are used all over SAP). So if the actual (i.e. “physical”) directory changes then it is only necessary to make a change in only one place—logical path.

But, most importantly, logical names can be dynamic. For example, `‘usr/DEV/temp’` is a *static* name, but `‘usr/<sysname>/temp’` is *dynamic*. In the last case, `<sysname>` is a variable that will be filled in with the system ID, so we can conveniently use the same logical path in DEV, QAS, and PRD systems and not worry about changing it.

Logical directories take some additional effort to set up, but they can be a life saver if the path names must be changed due to a server replacement, hosting provider switch, or any other technical reason.

Also, logical directories allow for some cool customization using a function module, see Section 6.4.


Beware of case sensitivity



Note that in some operating systems (OS), such as UNIX and Linux, file names are case sensitive. To find which OS is used in your SAP system, go to the menu SYSTEM • STATUS from any screen and check the OPERATING SYSTEM field.

2.2 Create new port—example

In this step, we will create a new port of the XML type to use in this book as an example. Naturally, any other port type can also be used in similar scenarios as needed. For example, in a real-life EDI implementation, you would most likely use an RFC port to connect to an EDI translator (see Section 3.2 for more information on the EDI interfaces).

In transaction **WE21**, select the port type XML from the list, then click the CREATE  icon. On the right-hand side, fill in the fields as shown in Figure 2.2:

- ▶ PORT (XMLFILE in this example) is the unique ID for the port
- ▶ DESCRIPTION
- ▶ PHYSICAL DIRECTORY or LOGICAL DIRECTORY (physical directory is used in this example)
- ▶ DIRECTORY—directory pathname (don't forget the slash at the end)
- ▶ OUTBOUND FILE—file name without extension ('.xml' extension will be added when file is created)

Technically we can leave the port name blank and have the system propose a name, but then we end up with a very “helpful” default name such as A000000002.

The last step is to click on the SAVE icon and marvel at the results of our work. Oh, wait, we almost forgot one important step. Make sure to click the ACCESS TEST button to confirm that the directory can actually be accessed. If the test is successful, you will get a message “Access to direc-

tory possible from current application server”. If the test is not successful, ask a system administrator for assistance.

Port	XMLFILE
Description	XML File
XML format	
<input type="radio"/>	SAP Release 46
<input checked="" type="radio"/>	Unicode
Outbound file	
Outbound: Trigger	
<input type="radio"/>	Logical directory
<input checked="" type="radio"/>	physical directory
Directory	D:\usr\sap\168\DV\EBMGS00\data\
Function module	
Description	
Outbound file	idoc_document

Figure 2.2: Create new XML file port

2.3 Logical system

Logical systems are defined in transaction **BD54**. Unlike the port or partner profile, this is a transportable configuration, which means the logical system must be created in a development environment and then imported into QA and production environments using a *transport request*. For more information regarding transports see SAP Help article “Change and Transport System”.

Creating a new logical system is super simple. Go to transaction **BD54** and hit the (Enter) key to dismiss the warning about cross-client configuration changes. Then click on the NEW ENTRIES button and enter a unique system name and description (see Figure 2.3). In this book, we will use the same logical system SKYNET for the examples in the next chapters. (Shout out to “The Terminator” movie fans!)

B Index

A

Application log 125
Archiving 164

B

Background job 152
 example 155

C

Change pointer 74
 activation 75
 process 81
Communication IDoc 82
Customer/material info record
 96

D

Distribution model 32, 66, 79,
 114

E

EDI 21
 customer / vendor cross-
 reference 94
 mapping to IDoc type 48
Enhancement *See* User Exit
Extend IDoc *See* IDoc:
 extension

F

File interfaces 129
Fine-Tuned Control in
 Purchasing 65

I

IDoc
 change data 143
 control record 14
 definition 10
 display 36
 documentation 18
 extension 169
 interface monitoring 132
 list specific segment 136
 Monitor 143
 processing 140
 search for content 139
 status list 136
 test tool 98
 type 12

L

Logical directory 27
Logical system 29

M

Master IDoc 82
Message Control 39
Message type 12
Model view 32, 67

O

Output 39
 condition records 51, 149
 processing log 53
 troubleshooting 147
 type 42, 57

P

- Partner function 44
- Partner profile 30, 91, 111
 - for customer 44
 - for logical system 60
- Physical directory 27
- Port
 - create 28
 - definition 25
 - File 26
 - Transactional RFC 26
 - types 26
 - XML file 26
- Process code 92

R

- RFC connection 11

S

- Segment type 16
- Serialization 164
- Services for object 108, 137
- Status record 15

T

- Test tool *See* IDoc test tool
- Trigger mechanism 129

U

- User exit 180, 190
 - debugging 189

W

- Workflow 132

X

- XML Editor 119
- XML-HTTP 130