

Axel Angeli
Robi Gonfalonieri, Ulrich Streit

<http://idocs.de>

The SAP R/3 Guide to EDI, IDocs and Interfaces

About The Authors

Axel Angeli,

is born in 1961. He is a Top Level SAP R/3 consultant and R/3 cross-application development coach. He specializes in coaching of large multi-national, multi-language development teams and troubleshooting development projects. His job description is also known as computer logistics, a delicate discipline that methodically wakes the synergetic effects in team to accelerate and mediate IT projects.

He is a learned Cybernetics scientist (also known as Artificial Intelligence) in the tradition of the Marvin Minsky [*The society of mind*] and Synergetics group of Herman Haken and Maria Krell. His competence in computer science is based on the works of Donald Knuth [*The Art of Computer Programming*], Niklas Wirth (the creator of the PASCAL language), the object oriented approach as described and developed during the XEROX PARC project (where the mouse and windows style GUIs have been invented in the early 1970ies) and Borland languages.

Before his life as SAP consultant, he made a living as a computer scientist for medical biometry and specialist for high precision industry robots. He concentrates now on big international projects. He speaks fluently several popular languages including German, English, French and Slavic. ☎ axela@logosworld.de

Robi Gonfalonieri,

born in 1964 is a senior ABAP IV developer and R/3 consultant for SD and MM. He is a learned economist turned ABAP IV developer. He specializes in international, multi-language projects both as developer and SD consultant. He speaks fluently several languages including German, French, English and Italian.

☎ robig@logosworld.de

Ulrich Streit,

born in 1974 is ABAP IV developer and interface specialist. He developed a serious of legacy system interfaces and interface monitors for several clients of the process industry. ☎ Ulis@logosworld.de

logosworld.com

is a group of loosely related freelance R/3 consultants and consulting companies. Current members of the logosworld.com bond are the following fine companies:

- Logos! Informatik GmbH, Brühl, Germany: R/3 technical troubleshooting
- OSCo GmbH, Mannheim, Germany: SAP R/3 implementation partner
- UNILAN Corp., Texas: ORACLE implementation competence

For true international R/3 competence and
enthusiastic consultants,

email us ☎ info@logosworld.de

or visit <http://idocs.de>

For Doris, Paul, Mini

Danke, Thank You, Graças, Tack så mycket, Merci, Bedankt, Grazie, Danjawad, Nandri, Se-Se

I due special thanks to a variety of people, clients, partners and friends. Their insistence in finding a solution and their way to ask the right questions made this book only possible.

I want especially honour *Francis Bettendorf*, who has been exactly that genre of knowledgeable and experienced IT professionals I had in mind, when writing this book. A man who understands an algorithm when he sees it and without being too proud to ask precise and well-prepared questions. He used to see me every day with the same phrase on the lips: "Every day one question." He heavily influenced my writing style, when I tried to write down the answers to his questions. He also often gave the pulse to write down the answers at all. At the age of 52, he joyfully left work the evening of Tuesday the 23rd March 1999 after I had another fruitful discussion with him. He entered immortality the following Wednesday morning. We will all keep his memory in our heart.

Thanks to *Detlef* and *Ingmar Streit* for doing the great cartoons.

Thanks also to Pete Kellogg of UNILAN Corp., Texas, Juergen Olbricht, Wolfgang Seehaus and his team of OSCo, Mannheim for continuously forming such perfect project teams. It is joy working with them.

Plans are fundamentally ineffective because the "circumstances of our actions are never fully anticipated and are continuously changing around us". Suchman does not deny the existence or use of plans but implies that deciding what to do next in the pursuit of some goal is a far more dynamic and context-dependent activity than the traditional notion of planning might suggest.

Wendy Suchman, Xerox PARC <http://innovate.bt.com/showcase/wearables/>

Who Would Read This Book?

This book was written for the experienced R/3 consultants, who wants to know more about interface programming and data migration. It is mainly a compilation of scripts and answers who arose during my daily work as an R/3 coach.

Quid – What is that book about? *The R/3 Guide* is a *Frequently Given Answers* book. It is a collection of answers, I have given to questions regarding EDI over and over again, both from developers, consultants and client's technical staff. It is focussed on the technical aspect of SAP R/3 IDoc technology. It is not a tutorial, but a supplement to the R/3 documentation and training courses.

Quis – Who should read the book? *The R/3 Guide* has been written with the experienced consultant or ABAP developer in mind. It does not expect any special knowledge about EDI, however, you should be familiar with ABAP IV and the R/3 repository.

Quo modo – how do you benefit from the book? Well, this book is a "How to" book, or a "Know-how"-book. *The R/3 Guide* has its value as a compendium. It is not a novel to read at a stretch but a book, where you search the answer when you have a question.

Quo (Ubi) – Where would you use the book? You would most likely use the book when being in a project involved in data interfaces, not necessarily a clean EDI project. IDocs are also helpful in data migration.

Quando – when should you read the book *The R/3 Guide* is not a tutorial. You should be familiar with the general concept of IDocs and it is meant to be used after you have attended an R/3 course on IDocs, ALE or similar. Instead of attending the course you may alternatively read one of the R/3 IDoc tutorial on the market.

Cur – Why should you read the book Because you always wanted to know the technical aspects of IDoc development, which you cannot find in any of the publicly accessible R/3 documentation.

ISBN 3528057297 <http://idocguide.de>

Table Of Contents

Where Has the Money Gone?	1
1.1 Communication	2
<i>More than 80% of the time of an EDI project is lost in waiting for answers, trying to understand proposals and retrieving data nobody actually needs.</i>	2
1.2 Psychology of Communication	3
<i>Bringing developers together accelerates every project. Especially when both parties are so much dependent on each other as in an EDI project, the partners need to communicate without pause.</i>	3
1.3 Phantom SAP Standards and a Calculation	4
<i>It is reported that SAP R/3 delivers standard EDI programs and that they should not be manipulated under no circumstances. Because this is not true, much project is lost in chasing the phantom.</i>	4
1.4 Strategy.....	5
<i>Do not loose your time in plans. Have prototypes developed and take them as a basis.</i>	5
1.5 Who Is on Duty?.....	5
<i>Writing interface programs is much like translating languages. The same rule apply.</i>	5
1.6 Marcus T. Cicero	6
<i>Some may have learned it in school: the basic rules of rhetoric according to Cicero. You will know the answers, when your program is at its end. Why don't you ask the questions in the beginning? Ask the right question, then you will know.</i>	6
What Are SAP R/3 IDocs?	7
2.1 What are IDocs?	8
<i>IDocs are structured ASCII files (or a virtual equivalent). They are the file format used by SAP R/3 to exchange data with foreign systems.</i>	8
2.2 Exploring a Typical Scenario.....	9
<i>The IDoc process is a straight forward communication scenario. A communication is requested, then data is retrieved, wrapped and sent to the destination in a predefined format and envelope.</i>	9
Get a Feeling for IDocs	11
3.1 Get A Feeling For IDocs	12
<i>For the beginning we want to give you a feeling of what IDocs are and how they may look like, when you receive it as a plain text file.</i>	12
3.2 The IDoc Control Record	14
<i>The very first record of an IDoc package is always a control record. The structure of this control record is the DDic structure <u>EDI DC</u> and describes the contents of the data contained in the package.</i>	14
3.3 The IDoc Data.....	15
<i>All records in the IDoc, which come after the control record are the IDoc data. They are all structured alike, with a segment information part and a data part which is 1000 characters in length, filling the rest of the line.</i>	15

3.4 Interpreting An IDoc Segment Info.....	16
<i>All IDoc data records are exchanged in a fixed format, regardless of the segment type. The segment's true structure is stored in R/3's repository as a DDic structure of the same name.</i>	16
3.5 IDoc Base - Database Tables Used to Store IDocs.....	17
<i>When R/3 processes an IDoc via the standard inbound or outbound mechanism, the IDoc is stored in the tables. The control record goes to table EDIDC and the data goes to table EDID4.</i>	17
Exercise: Setting Up IDocs	18
4.1 Quickly Setting up an Example.....	19
<i>If you have a naked system, you cannot send IDocs immediately. This chapter will guide you through the minimum steps to see how the IDoc engine works.</i>	19
4.2 Example: The IDoc Type MATMAS01.....	20
<i>To sharpen your understanding, we will show you an example of an IDoc of type MATMAS01, which contains material master data.</i>	20
4.3 Example: The IDoc Type ORDERS01.....	21
<i>To allow an interference, here is a sample of IDoc type ORDERS01 which is used for purchase orders and sales orders.</i>	21
Monitoring IDocs	23
Sample Processing Routines	25
6.1 Sample Processing Routines.....	26
<i>Creating and processing IDocs are a widely mechanical task, as it is true for all interface programming. We will show a short example that packs SAP R/3 SAPscript standard text elements into IDocs and stores them back.</i>	26
6.2 Sample Outbound Routines.....	27
<i>The most difficult work when creating outbound IDocs is the retrieval of the application data which needs sending. Once the data is well retrieved, the data needs to be converted to IDoc format, only.</i>	27
6.3 Sample Inbound Routines.....	29
<i>Inbound processing is widely the reverse process of an outbound.. The received IDoc has to be unpacked, interpreted and transferred to an application for further processing.</i>	29
IDocs Terminology	31
7.1 Basic Terms.....	32
<i>There are a couple of expressions and methods that you need to know, when dealing with IDoc.</i>	32
7.2 Terminology.....	33
7.2.1 Message Type – How to Know What the Data Means	33
<i>Data exchanged by an IDoc via EDI is known as message. Messages of the same kind belong to the same message type.</i>	33
7.2.2 Partner Profiles – How to Know the Format of the Partner	33
<i>Different partners may speak different languages. While the information remains the same, different receivers may require completely different file formats and communication protocols. This information is stored in a partner profile.</i>	33
7.2.3 IDoc Type – The Structure of The IDoc File	34

<i>The IDoc type is the name of the data structure used to describe the file format of a specific IDoc.</i>	34
7.2.4 Processing Codes	34
<i>The processing code is a pointer to an algorithm to process an IDoc. It is used to allow more flexibility in assigning the processing function to an IDoc message.</i>	34
IDocs Customizing	37
8.1 Basic Customizing Settings.....	38
<i>Segments define the structure of the records in an IDoc. They are defined with transaction WE31.</i>	38
8.2 Creating An IDoc Segment WE31.....	40
<i>The segment defines the structure of the records in an IDoc. They are defined with transaction WE31. We will define a structure to send a text from the text database.</i>	40
8.3 Defining The Message Type (EDMSG).....	43
<i>The message type defines the context under which an IDoc is transferred to its destination. It allows to use the same IDoc file format to use for several different applications.</i>	43
8.4 Define Valid Combination Of Message and IDoc Types.....	44
<i>The valid combinations of message type and IDoc type are stored in table EDIMSG.</i>	44
8.5 Assigning a processing function (Table EDIFCT).....	45
<i>The combination of message type and IDoc type determine the processing algorithm. This is usually a function module with a well defined interface or a SAP business object and is set up in table EDIFCT.</i>	45
8.6 Processing Codes.....	46
<i>R/3 uses the method of logical process codes to detach the IDoc processing and the processing function module. They assign a logical name to function instead of specifying the physical function name.</i>	46
8.7 Inbound Processing Code.....	48
<i>The inbound processing code is assigned analogously. The processing code is a pointer to a function module which can handle the inbound request for the specified IDoc and message type.</i>	48
IDoc Outbound Triggers	51
9.1 Individual ABAP.....	52
<i>The simplest way to create IDocs, is to write an ABAP which simply does it.</i>	52
9.2 NAST Messages Based Outbound IDocs.....	54
<i>You can use the R/3 message concept to trigger IDocs the same way as you trigger SapScript printing.</i>	54
9.3 The RSNAST00 ABAP.....	56
<i>The ABAP RSNAST00 is the standard ABAP, which is used to collect unprocessed NAST message and to execute the assigned action.</i>	56
9.4 Sending IDocs Via RSNASTED.....	57
<i>Standard R/3 provides you with powerful routines, to trigger, prepare and send out IDocs in a controlled way. There are only a few rare cases, where you do not want to send IDocs the standard way.</i>	57
9.5 Sending IDocs Via RSNAST00.....	58
<i>Here is the principle flow how RSNAST00 processes messages for IDocs.</i>	58

9.6 Workflow Based Outbound IDocs	59
<i>Unfortunately, there are application that do not create messages. This is especially true for master data applications. However, most applications fire a workflow event during update, which can easily be used to trigger the IDoc distribution.</i>	59
9.7 Workflow Event From Change Document	60
<i>Instead of waiting for a polling job to create IDocs, they can also be created immediately after a transaction finishes. This can be done by assigning an action to an workflow event.</i>	60
9.8 ALE Change Pointers	61
<i>Applications which write change documents will also try to write change pointers for ALE operations. These are log entries to remember all modified data records relevant for ALE.</i>	61
9.9 Activation of change pointer update	62
<i>Change pointers are log entries to table BDCP which are written every time a transaction modifies certain fields. The change pointers are designed for ALE distribution and written by the function CHANGE_DOCUMENT_CLOSE.</i>	62
9.10 Dispatching ALE IDocs for Change Pointers	63
<i>Change pointers must be processed by an ABAP, e.g. RBDMIDOC.</i>	63
<hr/> IDoc Recipes	65
10.1 How the IDoc Engine Works	66
<i>IDocs are usually created in a four step process. These steps are: retrieving the data, converting them to IDoc format, add a control record and delivering the IDoc to a port.</i>	66
10.2 How SAP Standard Processes Inbound IDocs	67
<i>When you receive an IDoc the standard way, the data is stored in the IDoc base and a function module is called, which decides how to process the received information.</i>	67
10.3 How To Create the IDoc Data	68
<i>R/3 provides a sophisticated IDoc processing framework. This framework determines a function module, which is responsible for creating or processing the IDoc.</i>	68
10.4 Interface Structure of IDoc Processing Functions	70
<i>To use the standard IDoc processing mechanism the processing function module must have certain interface parameters, because the function is called dynamically from a standard routine.</i>	70
10.5 Recipe To Develop An Outbound IDoc Function	71
<i>This is an individual coding part where you need to retrieve the information from the database and prepare it in the form the recipient of the IDoc will expect the data</i>	71
10.6 Converting Data Into IDoc Segment Format	72
<i>The physical format of the IDocs records is always the same. Therefore the application data must be converted into a 1000 character string.</i>	72
<hr/> Partner Profiles and Ports	73
11.1 IDoc Type and Message Type	74
<i>An IDoc file requires a minimum of accompanying information to give sense to it. These are the message type and the IDoc type. While the IDoc type tells you about the fields and segments of the IDoc file, the message type flags the context under which the IDoc was sent.</i>	74

11.2 Partner Profiles	75
<i>Partner profiles play an important role in EDI communications. They are parameter files which store the EDI partner dependent information.</i>	75
11.3 Defining the partner profile (WE20)	76
<i>The transaction WE20 is used to set up the partner profile.</i>	76
11.4 Data Ports (WE21)	77
<i>IDoc data can be sent and received through a multitude of different media. In order to decouple the definition of the media characteristics from the application using it, the media is accessed via ports.</i>	77
<hr/> RFC Remote Function Call	79
12.1 What Is Remote Function Call RFC?	80
<i>A Remote Function Call enables a computer to execute a program on another computer. The called program is executed locally on the remote computer using the remote computer's environment, CPU and data storage.</i>	80
12.2 RFC in R/3	81
<i>RFC provides interface shims for different operating systems and platforms, which provide the communication APIs for doing RFC from and to R/3.</i>	81
12.3 Teleport Text Documents With RFC	82
<i>This example demonstrates the use of RFC functions to send data from one SAP system to a remote destination. The example is a simple demonstration, how to efficiently and quickly use RFC in your installation.</i>	82
12.4 Calling A Command Line Via RFC ?	84
<i>R/3 RFC is not limited to communication between R/3 systems. Every computer providing supporting the RFC protocol can be called from R/3 via RFC. SAP provides necessary API libraries for all operating systems which support R/3 and many major programming languages e.g. C++, Visual Basic or Delphi.</i>	84
<hr/> Calling R/3 Via OLE/JavaScript	87
13.1 R/3 RFC from MS Office Via Visual Basic	88
<i>The Microsoft Office suite incorporates with Visual Basic for Applications (VBA) a fully object oriented language. JavaScript and JAVA are naturally object oriented. Therefore you can easily connect from JavaScript, JAVA, WORD, EXCEL and all the other VBA compliant software to R/3 via the CORBA compatible object library (in WINDOWS known also DLLs or ACTIVE-X (=OLE/2) components).</i>	88
13.2 Call Transaction From Visual Basic for WORD 97	89
<i>This is a little WORD 97 macro, that demonstrates how R/3 can be called with a mouse click directly from within WORD 97.</i>	89
13.3 R/3 RFC from JavaScript	91
<i>JavaScript is a fully object oriented language. Therefore you can easily connect from JavaScript to R/3 via the CORBA compatible object library (in WINDOWS known also DLLs or ACTIVE-X (=OLE/2) components).</i>	91
13.4 R/3 RFC/OLE Troubleshooting	93
<i>Problems connecting via RFC can usually be solved by reinstalling the full SAPGUI and/or checking your network connection with R/3.</i>	93

ALE - Application Link Enabling	95
14.1 A Distribution Scenario Based On IDocs	96
<i>ALE has become very famous in business circles. While it sounds mysterious and like a genial solution, it is simply a mean to automate data exchange between SAP systems. It is mainly meant to distribute data from one SAP system to the next. ALE is a mere enhancement of SAP-EDI and SAP-RFC technology.</i>	96
14.2 Example ALE Distribution Scenario	97
<i>To better understand let us model a small example ALE scenario for distribution of master data between several offices.</i>	97
14.3 ALE Distribution Scenario	98
<i>ALE is a simple add-on application propped upon the IDoc concept of SAP R/3. It consists on a couple of predefined ABAPs which rely on the customisable distribution scenario. These scenarios simple define the IDoc types and the pairs of partners which exchange data.</i>	98
14.4 Useful ALE Transaction Codes	99
<i>ALE is customized via three main transaction. These are SALE, WEDI and BALE.</i>	99
14.5 ALE Customizing SALE	101
<i>ALE customizing is relatively staright forward. The only mandatory task is the definition of the ALE distribution scenario. The other elements did not prove as being very helpful in practical applications.</i>	101
14.6 Basic Settings SALE	102
<i>Basic settings have do be adjusted before you can start working with ALE.</i>	102
14.7 Define The Distribution Model (The "Scenario") BD64	103
<i>The distribution model (also referred to as ALE-Scenario) is a more or less graphical approach to define the relationship between the participating senders and receivers.</i>	103
14.8 Generating Partner Profiles WE2Q	105
<i>A very useful utility is the automatic generation of partner profiles out of the ALE scenario. Even if you do not use ALE in your installation, it could be only helpful to define the EDI partners as ALE scenario partners and generate the partner profiles.</i>	105
14.9 Creating IDocs and ALE Interface From BAPI SDBG	109
<i>There is a very powerful utility which allows to generate most IDoc and ALE interface objects directly from a BAPI's method interface.</i>	109
14.10 Defining Filter Rules	113
<i>ALE allows to define simple filter and transformation rules. These are table entries, which are processed every time the IDoc is handed over to the port. Depending on the assigned path this happens either on inbound or outbound.</i>	113
Workflow Technology	115
15.1 Workflow in R/3 and Its Use For Development	116
<i>SAP R/3 provides a mechanism, called Workflow, that allows conditional and unconditional triggering of subsequent transactions from another transaction. This allows to build up automatic processing sequences without having the need to modify the SAP standard transactions.</i>	116

15.2 Event Coupling (Event Linkage)	117
<i>Contrary to what you mostly hear about R/3 workflow, it is relatively easy and mechanical to define a function module as a consecutive action after another routine raised a workflow event. This can e.g. be used to call the execution of a transaction after another one has finished.</i>	117
15.3 Workflow from Change Documents	118
<i>Every time a change document is written a workflow event for the change document object is triggered. This can be used to chain unconditionally an action from a transaction.</i>	118
15.4 Trigger a Workflow from Messaging	119
<i>The third common way to trigger a workflow is doing it from messaging.</i>	119
15.5 Example, How To Create A Sample Workflow Handler	120
<i>Let us show you a function module which is suitable to serve as a function module and define the linkage.</i>	120
Batch Input Recording	125
16.1 Recording a Transaction With SHDB	126
<i>The BTCI recorder lets you record the screen sequences and values entered during a transaction. It is one of the most precious tools in R/3 since release 3.1. It allows a fruitful cooperation between programmer and application consultant.</i>	126
16.2 How to Use the Recorder Efficiently	129
<i>This routine replaces BDCRECXX to allow executing the program generated by SHDB via a call transaction instead of generating a BTCI file.</i>	129
16.3 Include ZZBDCRECXX to Replace BDCRECXX	130
<i>This routine replaces BDCRECXX to allow executing the program generated by SHDB via a call transaction instead of generating a BTCI file.</i>	130
16.4 ZZBRCRECXX_FB_GEN: Generate a Function from Recording	132
<i>The shown routine ZZBDCRECXX_FB_GEN replaces BDCRECXX in a recorded ABAP. Upon executing, it will generate a function module from the recording with all variables as parameters.</i>	132
EDI and International Standards	137
17.1 EDI and International Standards	138
<i>Electronic Data Interchange (EDI) as a tool for paperless inter-company communication and basic instrument for e-commerce is heavily regulated by several international standards.</i>	138
17.2 Characteristics of the Standards	139
<i>The well-known standards EDIFACT, X.12 and XML have similar characteristics and are designed like a document description language. Other standards and R/3 IDocs are based on segmented files.</i>	139
17.3 XML	140
<i>This is an excerpt of an XML EDI message. The difference to all other EDI standards is, that the message information is tagged in a way, that it can be displayed in human readable form by a browser.</i>	140
17.4 ANSI X.12	142
<i>This is an example of how an ANSI X.12 EDI message for a sales order looks like. The examples do not show the control record (the "envelope"). EDIFACT looks very much the same.</i>	142

EDI Converter	143
18.1 Converter	144
<i>SAP R/3 has foregone to implement routines to convert IDocs into international EDI standard formats and forwards those requests to the numerous third party companies who specialize in commercial EDI and e-commerce solutions..</i>	144
18.2 A Converter from Germany	145
<i>In the forest of EDI converters there is only a very limited number of companies who have actual experience with R/3. We have chosen one very popular product for demonstration here.</i>	145
Appendix	147
19.1 Overview of Relevant Transactions	147
<i>There is a couple of transactions which you should know when working with IDocs in any form. I suggest to call each transaction at least once to see, what is really behind.</i>	147
19.2 Useful Routines for IDoc Handling	148
<i>These are some very useful routines, that can be used in IDoc processing.</i>	148
19.3 ALE Master Data Distribution	149
<i>The ALE functionality comes with a set of transaction which allow the distribution of important master data between systems. The busiest argument for installing ALE might be the distribution of the classification from development to production and back.</i>	149
19.4 WWW Links	150
<i>These is a random listing of interesting web sites dealing with the EDI topic. They are accurate as of November 1999.</i>	150
19.5 Questionnaire for Starting an IDoc Project	151
<i>This is a sample questionnaire with important questions that need to be cleared before any development can be started.</i>	151
Index	153

Table of Illustrations

<i>Illustration 1:</i>	<i>A typical EDI scenario from the viewpoint of R/3.....</i>	<i>9</i>
<i>Illustration 2:</i>	<i>Simplified Example of an IDoc control record for sales orders.....</i>	<i>12</i>
<i>Illustration 3:</i>	<i>Simplified Example of an IDoc data record for sales orders.....</i>	<i>12</i>
<i>Illustration 4:</i>	<i>Schematic example of an IDoc control record.....</i>	<i>14</i>
<i>Illustration 5:</i>	<i>Example of an IDoc with one segment per line, an info tag to the left of each segment and the IDoc data to the right.....</i>	<i>15</i>
<i>Illustration 6:</i>	<i>Tables used to store the IDoc within R/3.....</i>	<i>17</i>
<i>Illustration 7:</i>	<i>Step to customize outbound IDoc processing.....</i>	<i>38</i>
<i>Illustration 8:</i>	<i>Elements that influence IDoc processing.....</i>	<i>39</i>
<i>Illustration 9:</i>	<i>General Process logic of IDoc outbound.....</i>	<i>53</i>
<i>Illustration 10:</i>	<i>Communicating with message via table NAST.....</i>	<i>54</i>
<i>Illustration 11:</i>	<i>Process logic of RSNAST00 ABAP.....</i>	<i>58</i>
<i>Illustration 12:</i>	<i>Tables involved in change pointers processing.....</i>	<i>64</i>
<i>Illustration 13:</i>	<i>Sample content of view V_TBD62.....</i>	<i>64</i>
<i>Illustration 14:</i>	<i>Schematic of an IDoc Outbound Process.....</i>	<i>69</i>
<i>Illustration 15:</i>	<i>R/3 port types by release.....</i>	<i>77</i>
<i>Illustration 16:</i>	<i>WORD 97 text with MACROBUTTON field inserted.....</i>	<i>89</i>
<i>Illustration 17:</i>	<i>Visual Basic code with macros to call R/3 from WORD 97.....</i>	<i>90</i>
<i>Illustration 18:</i>	<i>ALE distribution scenario.....</i>	<i>97</i>
<i>Illustration 19:</i>	<i>Scenario in tabular form.....</i>	<i>97</i>
<i>Illustration 20:</i>	<i>Seeburger™ graphical EDI converter editor with R/3 linkage.....</i>	<i>146</i>

Directory of Programs

Program 1:	Sample IDoc outbound function module.....	27
Program 2:	Sample IDoc outbound function module.....	30
Program 3:	Interface structure of an NAST compatible function module.....	70
Program 4:	Interface structure of an IDoc inbound function	70
Program 5:	Routine to move the translate to IDoc data.....	72
Program 6:	Fill the essential information of an IDoc control record.....	72
Program 7:	Z_READ_TEXT, a copy of function READ_TEXT with RFC enabled.....	82
Program 8:	Program to copy text modules into a remote system via RFC	83
Program 9:	JavaScript example to call an R/3 function module via OLE/RFC.....	92
Program 10:	This is the call of the type coupled event in release 40B.....	117
Program 11:	This is the call of the change doc event in release 40B.....	118
Program 12:	This is the call of the type coupled event in release 40B.....	118
Program 13:	A workflow handler that sends an Sap Office mail.....	120
Program 14:	Send a SAPoffice mail triggered by a workflow event (full example).....	123
Program 15:	Program ZZBDCRECXX (find at http://www.idocs.de).....	131
Program 16:	Program ZZBDCRECXX_FBGEN found on http://www.idocs.de	136
Program 17:	XML Sales Order data.....	140

Preface

Proper Know-How Saves Costs

We always believed, what has been confirmed over and over again in manifold projects: The main source to cutting project costs, is a proper education of the team. Giving the team members the same book to read homogenizes the knowledge and sharpens a common sense within the group.

A Frequently Given Answers Book

This book is the result of thousands of hours of discussion and work with R/3 consultants, developer and clients about interface development from and to R/3. When we started a new big project in autumn 1998 at the Polar Circle, which involved a big number of interfaces, I observed curiously, that my developers were ordering numerous books, all being related to EDI.

Well, those books did not say any word about R/3 and it was obvious that they were not very helpful for our team. I consequently searched the directories for books on R/3 IDocs, but there was nothing. So I started to compile my material on IDocs and ALE with the intent to publish it in the WWW. Since I submit the site <http://idocs.de> to some search engines I got an astonishing amount of hits. Emails asked for a written version of the stuff on the web. So – here it is.

Mystery EDI Unveiled

EDI and e-commerce are miracle words in today's IT world. Like any other mystery it draws its magic from the ignorance of the potential users. It is true that there are many fortune making companies in the IT world who specialize on EDI. They sell software and know-how for giant sums of money. Looking behind the scenes reveals, that the whole EDI business can simply be reduced to writing some conversion programs. This is not too easy, but the secret of EDI companies is, that the so-called standards are sold for a lot of money. As soon as you get hold of the documentation, things turn out to be easy.

IDocs, A Universal Tool for Interface Programming

Although R/3 IDocs had been introduced as a tool to implement EDI solution for R/3, it is now accepted as a helpful tool for any kind of interface programming. While this is not taught clearly in SAP's learning courses, we put our focus on writing an interface quickly and easily.

<http://idocs.de>

We praise cutting edge technology. So this book takes advantage of the modern multimedia hype. Latest updates, corrections and more sophisticated and detailed examples are found on our web site.

Axel Angeli in December 1999

Logos! Informatik GmbH