Andreas Deutesfeld
Silvia Gruben
Dr. Wolfgang Weiss

SAPscript - Raw Data Interface (RDI)

Version 1.17

10. February 2000
last update: 16.February 2004

Please visit our web page at www.sap.com/csp/scenarios
for examples and the latest version of this document.

1 Raw Data Interface: Concept and Use

External text management systems can be connected via the SAPscript Raw Data Interface (RDI) for special purposes (e.g. postage optimization). This interface contains all R/3 forms data, but no layout information such as font or page size. Document data is formatted and managed by the external system outside R/3. Mail processing is an important application which uses this procedure: the external system sorts the interface document data and passes the documents on appropriately.

The raw data interface is a certified interface which provides a high-quality external system connection, but at the price of losing some R/3 System integration. It is, for example, not possible to find out from within an R/3 application whether external printing and sending of correspondence was successful. Standard form changes also require extra effort because both internal and external forms must be changed. The external tool is not integrated in the ABAP Development Workbench, so ABAP Dictionary field information (type, output length, etc.) is not available.

SAPscript knowledge is necessary to understand how an RDI data stream is created due to the interaction of a print program and a SAPscript form (e. g.: SAPscript course CA930 release 3.1 or BC460 release 4.0, online documentation).

2 Prerequisites and Conditions of Use of the Raw Data Interface

2.1 R/3 Customizing for Printing via RDI

An R/3 System form can be flagged for “external printing”. If a form is not explicitly flagged, application-dependent customizing settings apply (if available). For details see section 3.

2.2 External Printing Output Device

At least one “output device” must be setup for the external printing device type (PLAIN) delivered by SAP. For details please refer to section 9 or to the online documentation (spool administration).

2.3 Definition of External Forms and Assignment to R/3 Forms

The external text system must contain a form definition for each R/3 form which is to be printed externally. In particular, appropriate variables must be defined.

2.4 External Data Read Program

The external program reads the data according to the specification (see section 4) and formats and prints the documents.

2.5 Optional: RDI Data Status Control

The current print status of all documents which are to be printed externally should be able to be displayed. An overview of unsuccessful print requests is often required by customers.

2.6 Optional: Archiving

The external system should be able to send the printout to an archive if required.

3 Raw Data Interface Print Program Interface

Forms can be printed via the raw data interface using the function modules OPEN_FORM or a flag in the form (Management data: Pass data to an external program). The function module has an additional optional import parameter: RAW_DATA_INTERFACE.

Its possible values are 'X' (raw data interface, output mode spool), 'I' (raw data interface, output mode IDOC), space (SAPscript form printing) and '*' (default value). The default value means that the flag in the form (form maintenance SE71) decides whether to print via the raw data interface. This can be overruled with OPEN_FORM.

New functionality in rel. 4.6A: parameter RAW_DATA_INTERFACE value 'S' (simple-spool mode).

Only if at OPEN_FORM, the parameter RAW_DATA_INTERFACE contains the default valu e '*', the form settings apply.

All documents printed between OPEN_FORM and CLOSE_FORM, are placed in a spool request. You then specify the form with the function module START_FORM. The settings of this form then determine the print mode if with OPEN_FORM, the RAW_DATA_INTERFACE parameter was '*' and no form was specified there. You can not switch between the raw data interface and SAPscript form printing.

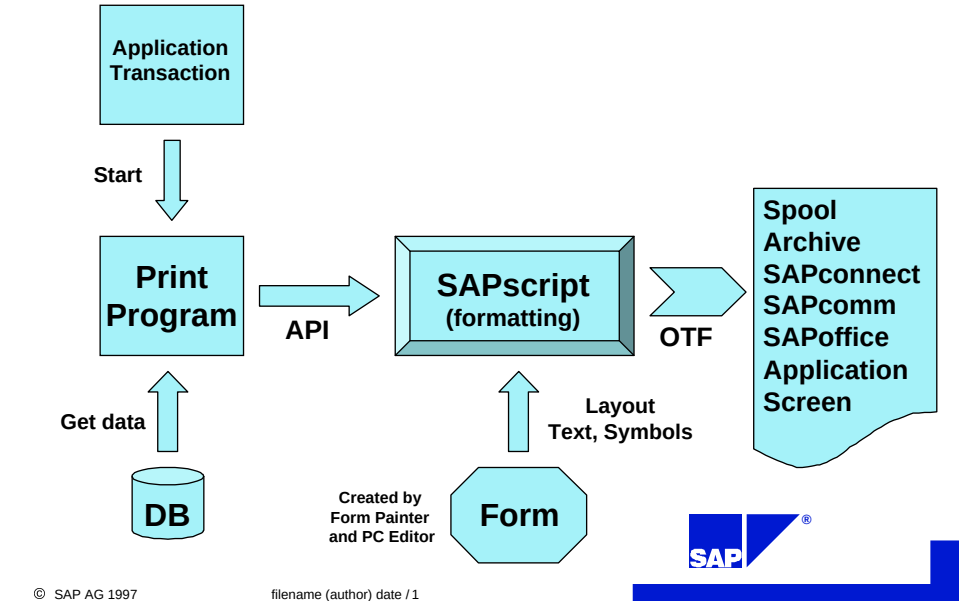
Backwards compatibility is guaranteed, as the function module OPEN_FORM default values let the form parameters "decide" whether data are to be printed via the raw data interface. If this parameter is not set in the form, "normal" SAPscript form printing applies.

New functionality in rel. 4.6A: You may set a specific output device for which the spool job wi ll be created. This may be done by setting a form attribute or by using parameter OPTIONS -TDRDIDEV in function module OPEN_FORM. The RDI data stream will then be send to this output device, whereas the printer that has been selected by the apllication is t ransmitted in the RDI header (TDDEST). In rel. 4.0 and 4.5 OSS note 111095 should be implemented.

The output for the spool mode can be displayed in the spool transaction SP01. Each document starts with a header record, followed by any number of data records. The structure of these records is described below.

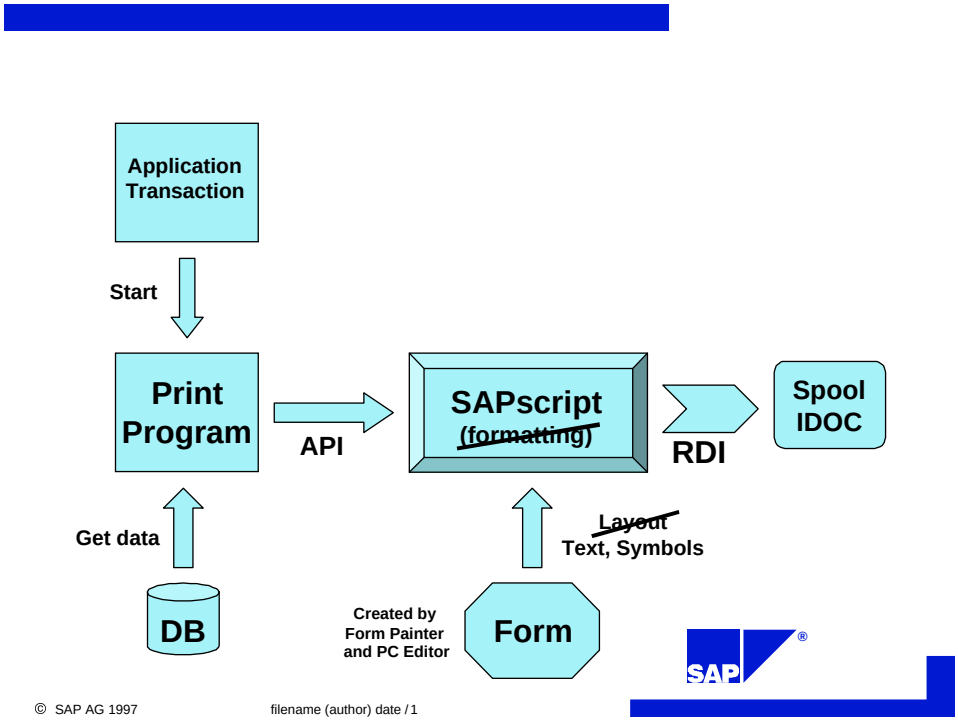
See section 5 for information on the IDOC mode.

3.1 Interaction Print Program – SAPscript



 filename (author) date / 1

3.2 Interaction Print Program – SAPscript (RDI)



 filename (author) date / 1

4 SAPscript Raw Data Interface : Output Stream (Output Mode Spool)

The output stream comprises header records, data records, and control records.

Releases 4.0A, 4.0B, 4.5A, 4.5B: The first character of each record (1 byte) indicates whether it is a header [H], sort[S], data [D] or control record [C].

Release 4.6A: The fi rst character of each record (1 byte) indicates whether it is a header [H], sort[S], archive index [I], archive parameter [P], data [D] or control record [C].

4.1 Header Record Structure

Each document header begins with the ID 'H' followed by the structure STXRDIH.

Field	Length	Info
RDI version	6	RDI version number (see section 4.1.1)
Client	3	
Document number	10	Return CLOSE_FORM: RESULT-TDSPOOLID
Language	1	
Form name	16	
Device type	8	PRINTER, SCREEN, ...
Name of terminal	64	
Batch mode	1	‚X' or SPACE

Please note that the name of the terminal from which the job has been started cannot be available if the apllication runs in background task (batch). The kind in which the terminal name appears is network dependent. Furthermore in release 4.5A the appe arance may change due to a change in task handler funcionality.

Then follows the relevant data of structure ITCPO:

Field name	Description	Length
TDPAGESLCT	Pages selected for printing	60
TDCOPIES	Number of copies	3
TDDEST	Output destination	4
TDPRINTER	Device type	8
TDPREVIEW	Print preview	1
TDNOPREV	Suppress print preview	1

TDNOPRINT	No printing from within display	1
TDNEWID	New spool request	1
TDDATASET	Name of spool request	6
TDSUFFIX1	Spool request suffix 1	4
TDSUFFIX2	Spool request suffix 2	12
TDIMMED	Print immediately	1
TDDELETE	Delete after printing	1
TDLIFETIME	Spool retention period (in days)	1
TDSCHEDULE	Request schedule	3
TDSENDDATE	Desired send date	8
TDSENDTIME	Desired send time	6
TDTELELAND	Country key	3
TDTELENUM	Telecommunications partner	30
TDTITLE	Text title	50
TDTEST	Test form	1
TDPROGRAM	Program name	40
TDSCRNPOS	Screen display position for OTF	15
TDCOVER	SAP cover page	1
TDCOVTITLE	Title for cover page	68
TDRECEIVER	Recipient	12
TDDIVISION	Division on cover page	12
TDAUTORITY	Authorization	12
TDARMOD	Archiving mode	1
TDIEXIT	Exit after printing from within display	1
TDGETOTF	Return only OTF table	1
TDFAXUSER	Name of the SAPOffice user	12

In release 4.6D a new field has been added, i. e. the „long name“ of the printer (character 30).

4.1.1 RDI Version Numbers

In R/3 releases 4.0A, 4.0B, 4.5A and 4.5B the RDI version is 040A01.

In release 4.6A the RDI version changes to 046A01. The data stream contains additional information about archiving and include texts. Furt hermore there is the possibility to add self -defined control information to the output stream.

4.2 Archive Information [Rel. 4.6A]

Please note that the archive information is only written if archiving is desired (see header record field TDARMOD [1=print only, 2=archive only, 3=archive and print]).

4.2.1 Archive Index

The structure of the archive index record which begins with the character ‚I' is STXRDII:

Field name	Description	Length
FUNCTION	ArchiveLink name of function	4
MANDANT	Client	3
DEL_DATE	Expiry date	8
SAP_OBJECT	Object type of business object	10
AR_OBJECT	Object type	10
OBJECT_ID	Object Identifier	50
FORM_ID	Document ID	40
FORMARCHIV	Target storage system	2
RESERVE	Reserved for future application	27
NOTIZ	SAP ArchiveLink domain character	256

4.2.2 Archive Parameters

The structure of the archive parameter record which begins with the character ‚P' is STXRDIP:

Field name	Description	Length
SAP_OBJECT	Object type of business object	10
AR_OBJECT	Document type of stored object	10
ARCHIV_ID	Target storage system	2

DOC_TYPE	Document class	20
RPC_HOST	RPC host	32
RPC_SERVICE	RPC service / RPC destination	32
INTERFACE	Name of communication connection	14
MANDANT	Client	3
REPORT	ArchiveLink report name	40
INFO	Info field	3
ARCTEXT	Text information field	40
DATUM	Archiving date	8
ARCUSER	User	12
PRINTER	Target printer	4
FORMULAR	Output format	16
ARCHIVPATH	Standard archive path	70
PROTOKOLL	Storage connection path	8
VERSION	Version number	4
ACHECK	Check sum for print and archiving parameters	10

4.3 Sort Record Structure

Then follow the sort fields (structures ITCSNDSORT, ITCRCVSORT), introduced by an 'S'. There are 10 internal and 5 external sort fields of length 32. How to fill the internal sort fields is described in section 7.

4.4 Mail Information [Rel. 4.6A]

Please note that the mail information is only written if device mail is used. There are three records for every document written between header and sort record: mail sender ('M'), mail recipient ('R') and mail application object ('A'). These r ecords do not contain an email address. They contain R/3 internal information (SAPscript/SAPconnect) that might be helpful for addressing. Each record has the same structure (SWOTOBJID):

Field name	Description	Length
LOGSYS	Logical system	10
OBJTYPE	Object type	10

OBJKEY	Object key	70
DESCRIBE	Describe flag	20

4.5 Data Record Structure

The structure of the data records begins with the character 'D'.

Description	Length	Info
Name of the form window	8	
Indicator for the start of a new main window	1	‚X‘ or SPACE
Indicator for the start of a text element	1	‚X‘ or SPACE
Name of the text elements	30	
Name of the symbol	130	SPACE for Plain-Text
Continuation flag	1	‚X‘ or SPACE
Ocupied langth	3	
Value of the symbol	Max. 255	Poss. Continuation flag

Note on Data records:

The beginning of each text element is flagged. As it is normal for a text element to contain several data records, this flag is important for unambiguous identification if several text elements are printed consecutively by WRITE_FORM. The start of a new main window is similarly flagged. This is particularly important when using the SAPscript command NEW-WINDOW.

Note that page feed must be triggered explicitly, either by the form (NEW -PAGE command) or by the print program (function module CON TROL_FORM) when printing via the raw data interface. The external tool can recognize a page feed from a control record (see next section).

The symbol occurs in the text element. It can be one of the symbols allowed in SAPscript. If it is "normal" text, the symbol name is empty (Space). The continuation flag follows the symbol name. If it is set (X), the following data record still belongs to this symbol. The occupied length gives the number of characters that belong to the symbol. Trailing spaces that belon g to the symbol are included. If a symbol occupies several data records and the continuation flag is set, the length is always 255 (maximum length per data record). Each data record ends with the symbol value in the length specified in occupied length. Thi s prevents that always the same fixed number of characters is printed, although only a small number are occupied.

Note that the symbol value never exceeds 255 characters. Actually, the continuation flag is only used for plain text if necessary.

The value o f a symbol is printed according to the specified formatting options. Any leading or trailing text is included in the symbol value. Symbols in the leading or trailing text are replaced; their values are assigned to the "main symbol".

If data records with em pty symbols (plain text) are not needed, OSS note 137277 shows a possible modification (it’s a modification, not a correction).

4.6 Control Record Structure

The data records can be interrupted by a control record containing data interpretation information. It begins with the ID 'C', followed by control information in the form of KEYWORD VALUE [VALUE2, ...].

The control records are currently used for the following information:

4.6.1 Codepage and Language

There is always a control record of this type after header and sort record and before the first data record. If a text in another language is included with the SAPscript statement INCLUDE, the control record with the appropriate codepage and language is written before the data of this include text. It may be necessary to switch back after the included text. The control record also contains this information.

The keywords are CODEPAGE and LANGUAGE.

Example: a control record for a German text with codepage 1100:

CCODEPAGE 1100 LANGUAGE DE

The first 'C' indicates the control record as described above.

4.6.2 Page Name

At the beginning of a new page, its name is sent. Pages are named using the keyword PAGENAME.

Example: a control record for the page FIRST:

CPAGENAME FIRST

The first 'C' again indicates the control record.

4.6.3 RDI-Control [rel. 4.6A]

The SAPscript command RDI -CONTROL allows to add specific information in the RDI output stream. Syntax in SAPscript is: `RDI-CONTROL string`, where 'string' represents the information you would like to add to the data stream.

Example: special information is the number 123:

CRDI-CONTROL 123

Note: Control string must never begin with ,%%'. This key is reserved for internal use only.

4.6.4 Include-Text Information [rel. 4.6A]

To identify an included text, the begin and end is marked with its full text key. The text key (name, object, id and language) is separated by space.

Example: begin and end of text ZABC, object TEXT, id ST and language DE:

CINC-BEGIN ZABC TEXT ST DE

...

CINC-END ZABC TEXT ST DE

Important: Texts written using function module WRIT E_FORM_LINES are NOT marked as include texts. See next section.

4.6.5 Additional text information [rel. 4.6A]

As mentioned in the previous section, texts written using function module WRITE_FORM_LINES are treated differently. The begin and end of such a text are marked by RDI controls. The key is given as name, object, id and language separated by space.

Example: begin and end of text ZABC, object TEXT, id ST and language DE:

```
CRDI-CONTROL %%LINES-BEGIN ZABC TEXT ST DE

...

CRDI-CONTROL %%LINES-END ZABC TEXT ST DE
```

5 SAPscript Raw Data Interface : Output Stream (Output Mode IDOC) [rel. 4.5A]

The data can also be printed as Intermediate Document (IDOC), as described in section 3.

Releases 4.0A, 4.0B, 4.5A and 4.5B: The name of the associated type is SAPRDI01 which comprises the segments E1RDIH (header), E1RDI_BODY (dummy segment), E1RDIS (sort record), E1RDIC (control record), and E1RDID (data record). The logical message type is SAPRDI.

Release 4.6A: The name of the associated type is SAPRDI02 which comprises the segments E1RDIH (header), E1RDI_BODY (dummy segment), E1RDII (archive index record), E1RDIP (archive parameter record), E1RDIMS (mail sender), E1RDIMR (mail recipient), E1RDIMA (mail application object), E1RDIS (sort record), E1RDIC (control record), and E1RDID (data record). The logical message type is SAPRDI.

The different records already mentioned have the structures described in “Spool”, but the IDs ‘H’, ‘C’, 'S', and ‘D’ are omitted. The segment E1RDI_BODY encapsules control, archive index, archive parameter, sort, and data segments for internal reasons. It can be ignored when interpreting the data.

In output mode IDOC, the header (E1RDIH) does not contain the document number. This number is returned in the return structure RDI_RESULT of the function module CLOSE_FORM (RDI_RESULT-DOCNUM). If START_FORM / END_FORM are used to print several documents in one request, only one IDOC is created. A header is then written for each document (as in Spool output mode).

To enable automatic creation of an IDOC of message type SAPRDI in the R/3 system, you must enter Customizing settings in the area ALE (Application Link Enabling).

ALE knows distribution models that describe the dependencies of messages between logical systems. The figure below described the scenario used here:

Using Transaction SALE in R/3, you can directly access the ALE Customizing menu (or use IMG maintenance, 'Cross-application components', 'Distribution (ALE)').

Below you find a short description of the special steps required for this IDoc application. As a prerequisite, basic Customizing for using IDocs in R/3 should be completed. For detailed information, refer to the ALE documentation and the IDoc documentation (CA - Business Framework Architecture - Application Link Enabling and CA - Intermediate Document - Interface /EDI).

Steps in IMG for ALE

1. Basic Settings → Set up logical system

Choose 'Assign logical system to the client' to check whether a logical system has already been assigned to the R/3 client. Write down the name of the logical system for the modelling procedure later.
If no logical system has been assigned to the client yet, choose 'Maintain logical systems' to enter a name for your R/3 client. Then choose 'Assign logical system to the client' to specify this name for the client (example of logical system: C11CLNT001).

For your external system, you can choose any name that is not yet in use to specify the logical system. Choose 'Maintain logical systems' to enter this name into R/3 (example of logical system: PRINTER).

2. Maintain distribution model

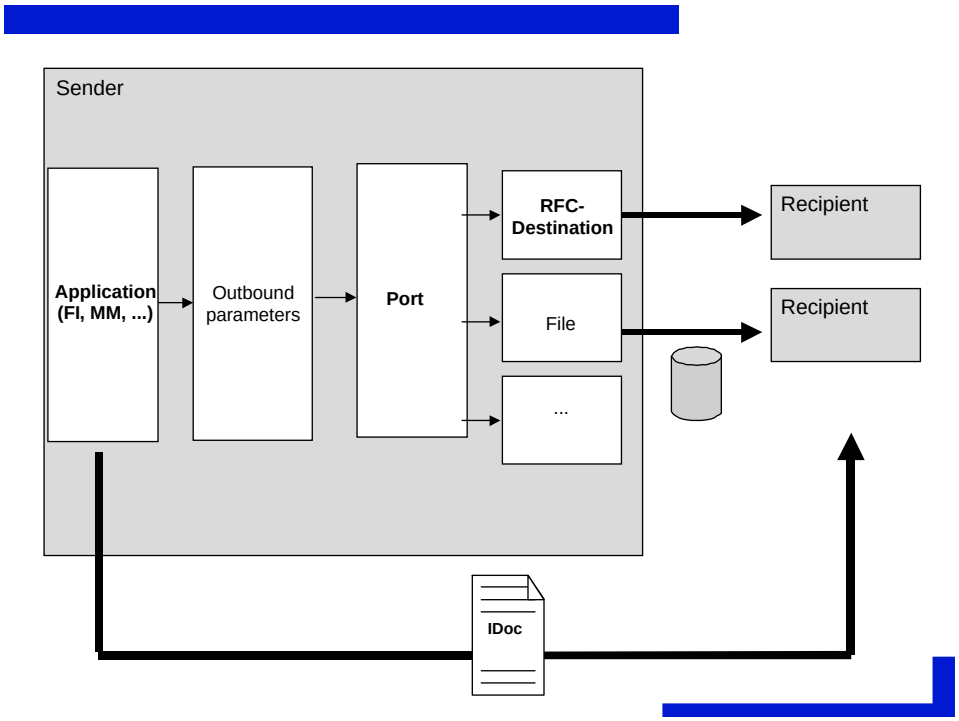
In this step, choose 'Maintain distribution model' to define the type of message sent between the logical systems. You can include the message type SAPRDI into an existing model (by defaul t, SAP delivers the model SUBSYSTEMS) or create an entirely new model. In any case, for the above example, you need an entry containing the following information:

Model view	(any, for example SUBSYSTEMS)
Sender/Client	C11CLNT00
Recipient/Server	PRINTER
Message type	SAPRDI

Remember to save before leaving model maintenance.

3. Communication

This menu includes any actions required to make the Customizing settings for IDoc processing. In this special case, you only need the settings for the IDoc outbound p rocessing. Below, you find a short representation of the terms defined thereafter.



A port definition in R/3 can refer to either a file interface or a tRFC connection. For more information on the file port, refer to the EDI documentation (CA → Intermediate Document - Interface / EDI → CA - The IDoc Interface → Instructions → Describing the File Port) . For more information on the tRFC port, see the Customizing online help 'Communication → Manual Maintenance of Partner Profiles → Define port'. We recommend to use the same name for the port as for the logical system (for example, PRINTER).

After setting up the port, you can create the partner profile for the logical system and the outbound parameters that go with it (choose 'Communication → Manual Maintenance of Partner Profiles → Maintain partner profile'). For your external logical system (PRINTER in the example) and the partner type LS, you must create the general partner profile.

For the outbound parameters, create an entry with the following information (all other data fields ae optional):

Message type	:	SAPRDI
Receiving port	:	Name of your ports
Basic type	:	SAPRDI01

For the output mode, you can either select immediate output or collect IDocs. When collecting IDocs, you must use job maintenance to sch edule a report RSEOUT00 in the R/3 client for later output.

6 SAPscript Raw Data Interface : Output Stream (Output Mode Simple-Spool) [rel. 4.6A]

In the simple -spool mode header, sort, archive, control, mail sender, mail recipient and mail apllication object records are written as in the normal spool mode. The data record is reduced to symbol name and value, separated by space. Usually, spool mode should be used to receive full logical information (window, element, ...). But if all this information is not ne eded, simple-spool mode should be used.

7 SAPscript Statement ADDRESS

The use of the ADDRESS statement is a special case. General information (Dictionary structure STXADDRESS), which is helpful in formatting addresses, is printed first:

Name	Description
STXADDRESS-TDPARGRAPH	Parameter PARAGRAPH of the ADDRESS statement
STXADDRESS-TYPE	Parameter TYPE of the ADDRESS statement (1,2,3)
STXADDRESS-NUMBER	Number of the address
STXADDRESS-FROM_COUNT	Country key of sender country
STXADDRESS-RECEIVER_L	Language key of recipient
STXADDRESS-PRIORITY	Priority rule, parameter PRIORITY
STXADDRESS-DELIVERY	Parameter DELIVERY
STXADDRESS-ANZZL	Parameter LINES
STXADDRESS-PERSONNUMB	Person number
STXADDRESS-COUNT_IRL	Flag, whether country is in recipient language
STXADDRESS-LANG_COUNT	Language for country
STXADDRESS-NO_UPPER	Flag, whether not to use uppercase for city

The structure of the actual address, depending on its type, follows immediately.

Type 1	Type 2	Type 3
ADRS1-TITLE_TEXT	ADRS2-TITLE_PERS	ADRS3-TITLE_COMP
ADRS1-NAME1	ADRS2-NAME_PERS	ADRS3-NAME1
ADRS1-NAME2		ADRS3-NAME2
ADRS1-NAME3		ADRS3-NAME3
ADRS1-NAME4		ADRS3-NAME4
		ADRS3-TITLE_PERS
		ADRS3-NAME_PERS
		ADRS3-DEPARTMENT

Followed by [x=1, 2 or 3]:

ADRSx-NAME_CO
ADRSx-STREET
ADRSx-HOUSE_NUM1
ADRSx-STR_SUPPL1
ADRSx-STR_SUPPL2
ADRSx-CITY1
ADRSx-CITY2
ADRSx-POST_CODE1
ADRSx-POST_CODE2
ADRSx-POST_CODE3
ADRSx-PO_BOX
ADRSx-PO_BOX_LOC
ADRSx-LOCATION
ADRSx-REGION
ADRSx-COUNTRY

If symbols are used in the ADDRESS command, they are replaced by your values. In the corresponding data record, the symbol name is the data field (e.g. ADRS1-NAME1).

8 Filling Internal Sort Fields

The SAPscript statement PERFORM is used to fill the 10 internal sort fields in the header record. It is important that it is placed in the main window default text element before any text is printed. Only statements such as DEFINE are allowed before this is done.

The up to 10 sort fields must be passed as USING parameters to the RSTXSORT program subroutine RDI_FILL_INTERNAL_SORTFIELDS. Example:

```
/: DEFINE &SORT01& = 'SAP'
/: DEFINE &SORT02& = 'FI'
/: PERFORM RDI_FILL_INTERNAL_SORTFIELDS IN PROGRAM RSTXSORT
/: USING &SORT01&
/: USING &SORT02&
/: ENDPERFORM
```

This enters the values 'SAP' and 'FI' in the sort record. Sort fields must not be longer than 32 characters each.

Note [rel. 4.6A]: The internal sort fields may also be set by the print program calling function module RDI_FILL_SORTFIELDS. This has to be done after calling function modules OPEN_FORM and START_FORM and before the first call of WRITE_FORM or WRITE_FORM_LINES.

9 Setup Output Device for RDI (Spool Mode)

To pass the RDI data to an external system, a special output device has to be created.

```
Output Device: Every outgoing document in R/3 requires a destination.
Since output may be done not only to printers but also to fax or
archiving devices, the general term in R/3 for these destinations is
output device.
```

The device type should be PLAIN.

```
Device Type: In order to format a document, R/3 needs to know some
information on the capabilities of the output device (e.g. list of
available fonts). This kind of information is contained in a device
type definition. For every output device that exists in R/3, a device
type must be specified that will be used when processing the output
document. One physical printer may have several output devices
assigned to it in R/3 with different device types.
```

PLAIN is a special device type for RDI (no fonts are used, no conversion due to different codepages are done).

The easiest way to route the RDI data to the external system is to choose access method L. Please note that this is only one possibility. For details please refer to the online documentation (spool administration).

```
Access Method: The access method (plus associated parameters like
printer queue name or name of network host) tells the R/3 spooler how
to route the print data to ist destination. In general, access methods
can be subdivided into local and remote access methods.
```

Access method L allows you to define a command set.

```
Command Set: It is possible to specify the command line executed by
R/3 when passing an output request to the UNIX/NT spooler on per-
device basis. For every output device using access method L, a command
set can be specified that defines the contents of the command line
used to transfer the spooler file to the operating system spooler (a
second command line must be specified for querying the job status).
For the meaning of the different command line parameters, please refer
to the online documentation (spool administration).
```

Example of a command set for RDI (UNIX):

command to transfer print data: cp &F /usr/sap/BIN/SYS/&f

command to demand the job status: echo ok

&F is the path- and filename of the R/3 print file, &f is the filename.

10 Helpful OSS Notes

10.1 Rel. 4.0A/B

92609 Spool error: RSPO_CLOSE_SPOOLREQUEST SUBRC 1
94355 TD860: Switch to RDI - SAPscript formatting
100215 RDI: Window name not filled in data stream
108263 Incorrect symbol or text element (RDI data stream)
111095 Choose special output device for RDI data stream
119515 Symbol name truncated in RDI data stream
130299 Sort fields not initialized in RDI data stream
136594 Internal error RD 001 LSTXCRDI (3b)
137277 Empty lines in RDI data stream
137771 RDI data stream contains only one page
141047 Terminal name not in RDI data stream

10.2 Rel. 4.5A

108263 Incorrect symbol or text element (RDI data stream)
111095 Choose special output device for RDI data stream
119515 Symbol name truncated in RDI data stream
124647 TD862: IDOC could not be created via RDI (invalid parameters)
137277 Empty lines in RDI data stream
137771 RDI data stream contains only one page
141047 Terminal name not in RDI data stream

10.3 Rel. 4.5B

108263 Incorrect symbol or text element (RDI data stream)
111095 Choose special output device for RDI data stream
124647 TD862: IDOC could not be created via RDI (invalid parameters)
137277 Empty lines in RDI data stream
137771 RDI data stream contains only one page
141047 Terminal name not in RDI data stream

10.4 Rel. 4.6A

137277 Empty lines in RDI data stream
137771 RDI data stream contains only one page
141047 Terminal name not in RDI data stream