



StreamServe Persuasion SP5 **StreamServe Connect *for SAP-*** **E-docs**

User Guide

Rev A

StreamServe Persuasion SP5StreamServe Connect for SAP - E-docs User Guide
Rev A

SAP, mySAP.com, and all other names of SAP solutions, products, and services are trademarks of SAP AG.

© 2001-2010 STREAMSERVE, INC.
ALL RIGHTS RESERVED
United States patent #7,127,520

No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of StreamServe, Inc. Information in this document is subject to change without notice. StreamServe Inc. assumes no responsibility or liability for any errors or inaccuracies that may appear in this book. All registered names, product names and trademarks of other companies mentioned in this documentation are used for identification purposes only and are acknowledged as property of the respective company. Companies, names and data used in examples in this document are fictitious unless otherwise noted.

StreamServe, Inc. offers no guarantees and assumes no responsibility or liability of any type with respect to third party products and services, including any liability resulting from incompatibility between the third party products and services and the products and services offered by StreamServe, Inc. By using StreamServe and the third party products mentioned in this document, you agree that you will not hold StreamServe, Inc. responsible or liable with respect to the third party products and services or seek to do so.

The trademarks, logos, and service marks in this document are the property of StreamServe, Inc. or other third parties. You are not permitted to use the marks without the prior written consent of StreamServe, Inc. or the third party that owns the marks.

Use of the StreamServe product with third party products not mentioned in this document is entirely at your own risk, also as regards the StreamServe products.

StreamServe Web Site
<http://www.streamserve.com>

Contents

E-docs	7
Introduction	7
SAP Output Architecture	7
Interface between SAP and StreamServe	8
Example of setting up SAP connectivity	10
XFP (XML for PDF based print forms)	12
XFP certification.....	12
XML for Smart Forms (XSF) data	12
XSF certification and StreamServe.....	12
ABAP List data.....	12
Configuring SAP for XFP data	13
Configuring SAP to send XFP data	14
Creating an XFP output device	14
Activating XFP data output for a PDF based print form	15
Sending XFP data to StreamServe	16
Creating a local printer definition.....	16
Creating an XFP output file	17
Creating a printer definition for sample XFP data	17
Printing sample XFP data to a file	18
Configuring SAP for XSF data	19
Configuring SAP to send XSF data	20
Creating a plain output device.....	20
Activating XSF data output for a Smart Form	21
Sending XSF data to StreamServe	23
Creating a local printer definition.....	23
Sending sample XSF data from SAP to StreamServe	24
Creating an XSF output file	25
Creating a printer definition for sample XSF data	26
Printing sample XSF data to a file	26
Configuring SAP for RDI data	29
Configuring SAP to send RDI data	30
Creating a plain output device.....	30
Activating RDI data output for a SAPscript Form	31
Assigning customized SAPscript Forms to print programs	33
Exporting customized SAPscript Forms.....	34
Sending RDI data from SAP to StreamServe	35
Creating a local printer definition.....	35
Sending sample RDI data from SAP to StreamServe.....	36
Creating an RDI output file.....	38
Creating a printer definition for sample RDI data.....	38

Printing sample RDI data to a file	39
Generating SAPscript Form documentation.....	40
Configuring StreamServe for XFP data.....	41
Creating a Message for XFP data.....	41
Running the XFP filter stand-alone.....	42
Configuring StreamServe for XSF data.....	43
Converting XSF to StreamServe XML	43
Creating a Message for XSF data.....	45
Configuring an XMLIN Event for XSF data	45
Specifying data collection method for XSF data	46
Importing sample XSF data into the XMLIN Event	46
Declaring namespaces for XSF data	48
Generating match criteria for fields and patterns	48
Configuring how field names are generated for XSF data	49
Defining a pattern for XSF data	51
Defining fields for XSF data	52
Defining match criteria for fields for XSF data	53
Defining blocks for XSF data	56
Running the XSF filter stand-alone.....	57
Configuring StreamServe for RDI data	59
Creating a Message for RDI data.....	60
Naming a StreamIN Event for a SAPscript Form.....	60
Configuring a StreamIN Event for RDI data	62
Importing SAP RDI data into the SAP Form Reader	62
Importing a SAPscript Form into the SAP Form Reader.....	63
Importing an RDI file into the SAP Form Reader.....	64
Applying sample data in the StreamIN Event	65
Configuring the StreamIN Event for RDI data.....	66
Creating a Page Layout Template	67
Changing the SAPscript Form language	67
Mapping SAP fonts to StreamServe fonts	68
Comparing Messages.....	68
StreamServe and the XFP format	71
StreamServe and the RDI format	73
RDI data	74
Header records	75
Control records	77
Data records	77
SAP 4.6/4.7 RDI functions	78
Archive records.....	78
Archive index - 'I' records.....	78
Archive parameters - 'P' records.....	79
Include-texts elements.....	80
Include-Texts additional element	81
RDI-Control command.....	81

Keeping spaces.....	81
Specifying input analyzer parameters for RDI data.....	81
Generating INCLUDE-BEGIN field names	83
StreamServe and the XSF format.....	85
XSF data elements	86
Root element.....	86
Control elements.....	86
Data elements.....	88
Useful SAP transaction codes.....	91



E-docs

This guide describes how to configure StreamServe Connect *for SAP* - E-docs with your SAP system. The E-docs Connect solution is an add-on module to StreamServe.

Note: This guide only contains StreamServe information specific to the E-docs Connect solution. For general information on StreamServe, see the standard StreamServe documentation.

E-docs is one of four StreamServe Connect solutions available for use with SAP. For information on the other solutions, see the following documentation:

- StreamServe Connect *for SAP* - Output+
- StreamServe Connect *for SAP* - Business Processes
- StreamServe Connect *for SAP* - Delivery Manager

Installation

For information on how to install the StreamServe Connect solutions, see the *StreamServe Connect for SAP - Installation Guide*.

Introduction

The E-docs Connect solution enables you to receive data from your SAP system, and process and distribute the data using StreamServe.

Using the E-docs Connect solution, you can configure StreamServe to receive four types of data:

- **XFP data** — XML for PDF based print forms.
- **RDI data** — Raw Data Interface data created from Forms generated in the SAPscript Composer
- **XSF data** — XML for Smart Forms data generated from Smart Forms in the SAP system
- **ABAP List data** — Data created from Reports generated by the ABAP language.

SAP Output Architecture

In a SAP system, you can print a document and thereby generate a spool request from various sources, including:

- an ABAP program (list output)
- a program editor

- the SAPscript word processing system.

The SAP spool system manages all ‘printed’ documents as spool and output requests. The spool system holds the output generated by one of the document sources as a spool request. If the output is actually sent to a printer or fax machine, the spool system generates an output request and formats the data for output.

Interface between SAP and StreamServe

The interface for business documents between SAP and StreamServe comprises a virtual printer. This means that to send SAP data to StreamServe, you can simply print documents in the SAP system in the usual manner, where StreamServe is set up as a printer device in the SAP spool system.

Output device

To process RDI or XSF data generated from the SAP system using StreamServe, you use an output device of plain device type. For ABAP List data, you use an output device of SAPGOFU or SAPGOF type. For XFP data, use device type XFP.

Transferring method

The way data is transferred from SAP to StreamServe depends on the host spool access method that you specify for the output device.

Host spool access method	Use for:
L	<ul style="list-style-type: none"> • File copy using a common file share between the SAP spool server and the StreamServer host. This is recommended for high-volume printing. • Local LPD/CUPS printing on UNIX.
U	Remote LPD/CUPS printing on UNIX.
C	Local spooler service printing on Windows.
S	Remote Windows printing using the LPD server of the SAPSprint service.
F	Local printing on front-end computer. Use for testing.
E	The XOM interface. See the <i>StreamServe Connect for SAP - Delivery Manager</i> documentation.

For more information on the host spool access methods, see <http://help.sap.com>

Layout of output

You can use the original layout configured for the Form or the Report in the SAP system, or you can enhance the layout using the tools within StreamServe. You use the following StreamServe tools to configure SAP output data:

- StreamIN to configure RDI data.
- XMLIN to configure XSF and XFP data.
- PreformatIN to configure ABAP List data.

You can output the data from StreamServe using any format.

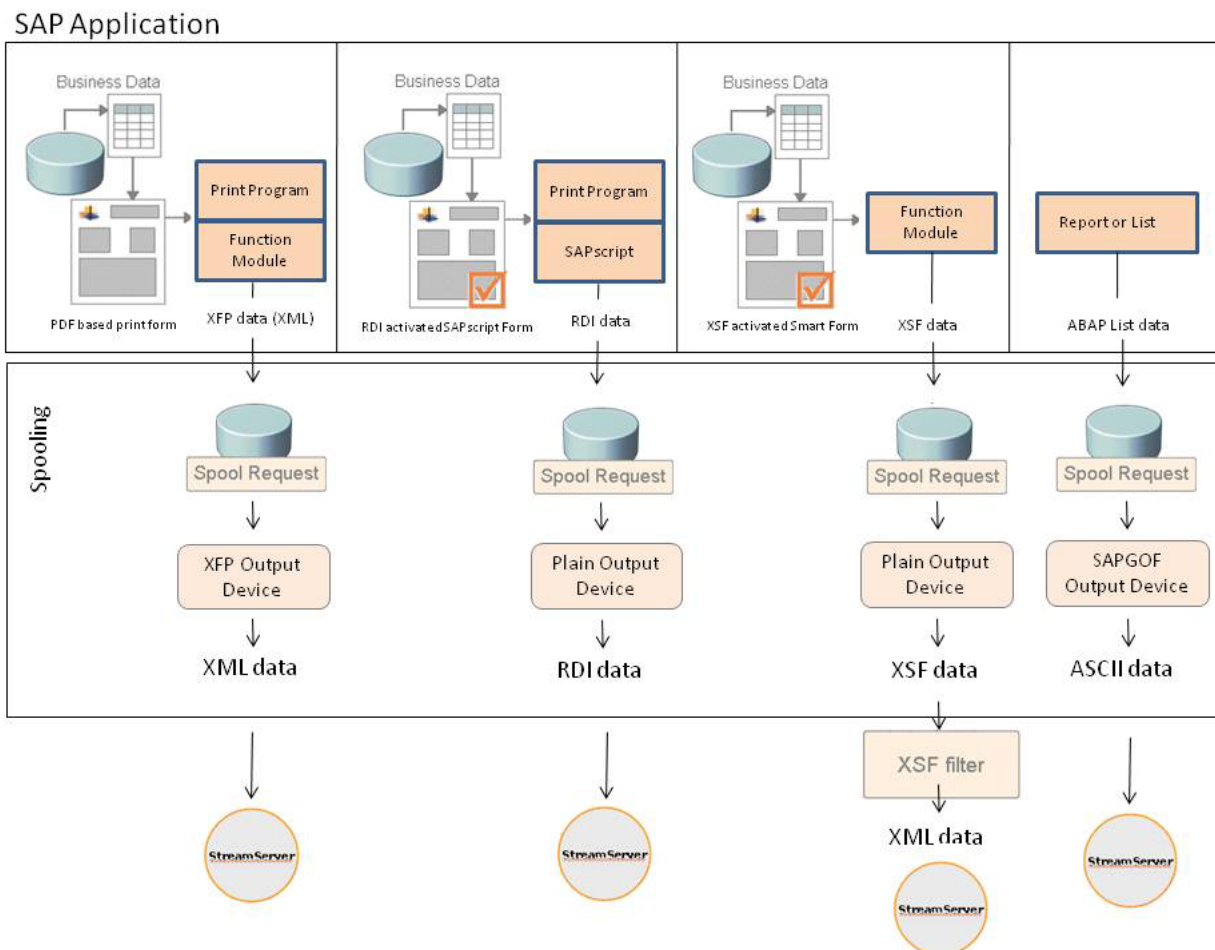


Figure 1 Sending data in different formats from SAP to StreamServe

Example of setting up SAP connectivity

This example shows one way of setting up StreamServe as a virtual printer in Windows to use remotely with SAP. In the example, ABAB List data is to be printed through the SAPSprint LPD service. The following is required:

- Install Streamserve port monitor.
See the *StreamServe Installation* documentation.
- Create a local spool folder. In this example, C:\strs_spool.
- Create and configure a local StreamServe printer.
- Install the SAPSprint service, see <http://help.sap.com>
- Check that the SAPSprint service is running on the StreamServe host
(e.g. Control Panel > System and Security > Administrative tools > Services)
- Create and configure an output device in SAP.

Creating a local StreamServe printer

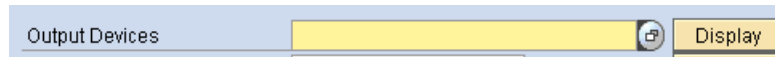
Add a new local printer by clicking **Add a printer** in the Devices and Printers dialog in Windows, and specify the following:



Note: In this example, Windows 7 is the OS.

Setting	Value
What type of printer do you want to install?	Add a local printer
Choose a printer port	Create a new port
Type of port	StreamServe
Port name	STRS
Destination: path/named pipe/address	C:\strs_spool
Manufacturer	Generic
Printers	Generic / Text Only
Printer name	STRS
Printer Sharing	Share this printer so that others on your network can find and use it
Share name	STRS
Set as default printer	Uncheck this box, it should not be the default printer, and do not print a test page.

Creating an output device

- 1 Log on to the SAP client.
- 2 Enter `/nspad` in the transaction field. The Spool Administration: Initial Screen is displayed.
- 3 On the Output Devices row, click Display.



- 4 Click Change.

- 5 Click Create.

- 6 Enter a name for the Output Device.
- 7 From the Device Type drop-down list, select **SAPGOFU :Generic outp.fmt Unicode 1.0**.
- 8 Select in the Spool Server field, and double-click the SAP server to use.
- 9 Select the **Access Method** tab.
- 10 From the Host Spool Access Method drop-down list, select the following:
S: Print Using SAP Protocol
- 11 In the Host printer field, enter the StreamServe port monitor printer you have created, STRS.
- 12 In the Destination host field, enter the host name or IP address of the StreamServe port monitor printer.



For better performance, select **Do Not Query Host Spooler for Output Status**.

- 13 Save the device.

To test the device

- 1 Go back to `/nspad` and select to display output devices.
- 2 Click Print this list to output the list on your new device.



- 3 Select your output device and click Continue.



XFP (XML for PDF based print forms)

The XFP interface lets you receive business data represented as XML generated from PDF-based print forms. The PDF-based forms currently supported in the E-docs solution are LiveCycle Designer forms.

By setting up a generic XMLIN message in StreamServe, a PDF-based Live Cycle Designer print form from SAP can be processed and rendered as designed in. A pattern is only set on the root node of the Message and no other field selection or mapping is required.

You can use any Process tool to create the output, but to re-use the design you must use Adobe LiveCycle Designer ES Process, where you also can specify a HTTP URI to dynamically select the print form from SAP in runtime.

XFP certification

StreamServe's XFP certification enables StreamServer to receive XFP data directly from SAP NetWeaver 2004 and onwards.

XML for Smart Forms (XSF) data

Using the E-docs solution, you can receive XSF data from your SAP system, and process and distribute the data using StreamServe. XSF data (XML for Smart Forms) is generated from Smart Forms in the SAP system.

XSF certification and StreamServe

StreamServe's XSF certification enables StreamServer to receive XSF data directly from SAP 4.6c and newer.

XSF is the XML output data interface for SAP Smart Forms. StreamServe connects via the SAP Smart Forms XML interface to intercept the XSF data from SAP.

ABAP List data

Using the E-docs Connect solution, you can receive ABAP List data from your SAP system, and process and distribute the data using StreamServe. ABAP List data is generated by the ABAP/4 language directly from the SAP system.

From SAP 4.0 onwards, StreamServe supports processing and distribution of ABAP List reports through the SAPGOF (SAP Generic Output Format) interface.

For information on how to configure SAP and StreamServe for ABAP List data, see the *StreamServe Connect for SAP - Output+* documentation.

Configuring SAP for XFP data

This section describes how to configure the E-docs Connect solution with your SAP system, to receive XFP data from your SAP system. XFP data is generated from PDF based print forms in the SAP system.

Included activities

To configure SAP to process XFP data for StreamServe, complete the following steps:

- [Configuring SAP to send XFP data](#) on page 14
- [Sending XFP data to StreamServe](#) on page 16

To process the XFP data automatically with StreamServe, a Design Center project must be deployed to a StreamServer application on the host receiving the data from SAP. See [Configuring StreamServe for XFP data](#) on page 41.

XFP supported SAP releases

SAP systems from NetWeaver 2004 and onwards include XFP as the external data interface. The E-docs Connect solution supports XFP from NetWeaver 2004 and onwards.

Configuring SAP to send XFP data

To configure the SAP system to send XSF data for processing in StreamServe, carry out the following steps:

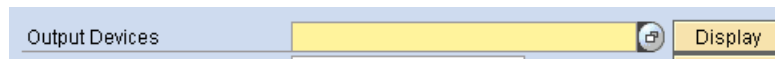
- [Creating an XFP output device](#) on page 14
- [Activating XFP data output for a PDF based print form](#) on page 15

Creating an XFP output device

To send XFP data from your SAP system for output via StreamServe without using the Delivery Manager, you must create an XFP output device in the SAP system.

To create an XFP output device

- 1 Log on to your SAP system as a user with administrative permissions.
- 2 In the transaction box, enter `/nspad`. The initial Spool Administration window opens.
- 3 For the Output Devices option, click **Display**.



The List of Output Devices window opens.

- 4 Click the **Change** toolbar button.



- 5 Click the **Create** toolbar button.



The Create Output Device window opens.

- 6 Specify the output device settings.

Output device settings	
Output Device	The name of the StreamServe output device. Use only XFP.
Short name	The short name for the StreamServe output device, or leave blank for SAP to generate a short name.

- 7 Click the **DeviceAttributes** tab.
- 8 Specify the device settings.

Device settings	
Device type	Select XFP .

Device settings	
Device class	Select Standard printer .

- 9 Click the **HostSpoolAccMethod** tab.
- 10 Specify the host spool settings.

Host spool settings	
Host spool access method	<p>The access method you want to use for the host spool. For example, L to print locally using a LP/LPR printer, or E if you use the Delivery Manager.</p> <p>Note: If you select an access method that requires a spool server, you must specify the spool server on the DeviceAttributes tab before you save the device definition.</p>
Host printer	<p>The name of the StreamServe printer queue, for example <code>strs</code>. For information on configuring the StreamServe printer queue, see</p> <ul style="list-style-type: none"> • Creating a local printer definition on page 16. • Example of setting up SAP connectivity on page 10. <p>Note: The name of the host printer is case sensitive and cannot contain any spaces.</p>

- 11 Click **Enter**.
- 12 Save the new device definition.

Activating XFP data output for a PDF based print form

Unlike SAPscript and Smart Forms, XFP output can not be activated on the form itself. Instead, XFP output is activated via a parameter in the print program.

See the documentation for PDF-based print forms on e.g. <http://help.sap.com>.

Sending XFP data to StreamServe

When you have enabled XFP in the print program to send XFP data from your SAP system and created a plain output device, you must create a local printer definition (if you do not use Delivery Manager).

If you only want to generate an XFP file, without sending it to StreamServe, see [Creating an XFP output file](#) on page 17.

Creating a local printer definition

To capture data on your SAP system, you need to create a printer definition on your local machine (Windows) that you can use to receive data as the StreamServe port.

To create a printer definition

- 1 On your local machine, add a printer definition with the following settings:

Printer settings	
Local or Network Printer	Select Local printer . Note: Ensure Automatically detect and install my Plug and Play printer is <i>not</i> selected.
Port	A new port of StreamServe type. Note: The StreamServe port type is only available when you have installed StreamServe port monitor. See the <i>StreamServe Installation</i> documentation
Port Definition	A name for the port, such as <code>strs</code> . Note: The port name must match the name of the host printer name specified for the output device. Enter a destination path/pipe to the StreamServe spool from where StreamServer will receive data, for example: <code>C:\StreamServe\spool</code>
Manufacturer	Select Generic .
Model	Select Generic/Text only .
Printer Name	A name for the printer, For example, <code>strs</code> . Note: Do not specify the printer as the default printer.

Printer settings	
Shared As	A share name for the printer, for example <code>strs</code> . Note: The share name must match the name of the host printer name specified for the output device.

- 2 Right-click the new printer and select **Properties > Advanced > Print Processor**
- 3 Select **passthru** and **RAW**.

Creating an XFP output file

Required activities

- 1 [Creating an XFP output device](#) on page 14
Note: When configuring an output device for sending sample XFP data, ensure you specify the access method as **F** to print locally to a front end printer.
- 2 [Creating a printer definition for sample XFP data on page 17](#)
- 3 [Printing sample XFP data to a file](#) on page 18

Creating a printer definition for sample XFP data

To create an XFP output file, you need to create a printer definition on your local machine (Windows) that you can use to send the sample data as a file to your local machine.

To create a printer definition for sample XFP data

- 1 On your local machine, add a printer definition with the following settings:

Printer settings	
Local or Network Printer	Select Local printer . Note: Ensure Automatically detect and install my Plug and Play printer is <i>not</i> selected.
Port	Select FILE (Print to File) .
Manufacturer	Select Generic .
Model	Select Generic/Text only .
Printer Name	A name for the printer, For example, <code>strs</code> . Note: Do not specify the printer as the default printer.

Printer settings	
Shared As	A share name for the printer, for example <code>strs</code> . Note: The share name must match the name of the host printer name specified for the output device.

- 2 Right-click the new printer and select **Properties > Advanced > Print Processor**
- 3 Select **passthru** and **RAW**.

Printing sample XFP data to a file

You print sample XFP data from your SAP system using the output device configured to use the local printer definition. The sample data is sent from your SAP system to your local machine as a file.

When you have received the file, you can use the sample data as a spool file to run through StreamServer and test your StreamServe configuration.

Note: This section uses an invoice from the SAP system as an example.

To print sample XFP data to a file

- 1 Log on to your SAP system as a user with administrative permissions.
- 2 In the transaction box, enter `/nVF03`. The Display Billing Document window.
- 3 In the Billing document box, select the billing document you want to print. The billing document must be an XFP print form.
- 4 Select **Billing document > Issue output to, Printer**. The Issue Output dialog box opens.
- 5 Ensure the correct output type is used. (The standard output type for an invoice is usually RD00.) Click **Execute**. The second Issue Output dialog box opens.
- 6 In the Logical destination box, enter the name of the output device you created for sending sample XFP data to StreamServe. See [Creating a printer definition for sample XFP data on page 17](#).
- 7 Ensure **Print immediately** is selected.
- 8 Click **Execute**. If printing is successful, your SAP system will launch SAPLPD on your local machine showing the print job in the log file. See [Configuring SAP to send XFP data on page 14](#). The output is sent to the printer and the Print to File dialog box opens.
- 9 Enter the full path and name of the XFP data file, for example the spool directory for the input connector:
`StreamServe\spool\invoice.xfp`
- 10 Click **OK**.

Configuring SAP for XSF data

This section describes how to configure the E-docs Connect solution with your SAP system, to receive XSF data from your SAP system. XSF data (XML for Smart Forms) is generated from Smart Forms in the SAP system.

Included activities

To configure SAP to process XSF data for StreamServe, complete the following steps:

- See [Configuring SAP to send XSF data](#) on page 20
- See [Sending XSF data to StreamServe](#) on page 23

XSF supported SAP releases

SAP systems from release 4.6c and onwards include Smart Forms with XSF as the external data interface. The E-docs Connect solution supports XSF from release 4.6c onwards.

Configuring SAP to send XSF data

To configure the SAP system to send XSF data for processing in StreamServe, carry out the following steps:

- [Creating a plain output device](#) on page 20
- [Activating XSF data output for a Smart Form](#) on page 21

Creating a plain output device

To send XSF data from your SAP system to StreamServe, you need to create a plain output device in SAP to output data.

To create a plain output device

- 1 Log on to your SAP system as a user with administrative permissions.
- 2 In the transaction box, enter `/nspad`. The initial Spool Administration window opens.
- 3 For the Output Devices option, click **Display**. The List of Output Devices window opens.
- 4 Click the **Change** toolbar button.



- 5 Click the **Create** toolbar button.



The Create Output Device window opens.

- 6 Specify the output device settings.

Output device settings	
Output device	The name of the StreamServe output device, such as ZSTRS.
Short name	The short name for the StreamServe output device, such as ZSTR, or leave blank for SAP to generate a short name.

- 7 Click the **DeviceAttributes** tab.
- 8 Specify the device settings.

Device settings	
Device type	Select XSF :SAP Smart Forms: XSF Ausgabe
Device class	Select Standard printer .

- 9 Click the **Access Method** tab.
- 10 Specify the host spool settings.

Access Method settings	
Host Spool Access Method	<p>The access method you want to use for the host spool. For example, L to print locally using a LP/LPR printer, or E when using the Delivery Manager.</p> <p>Note: If you select an access method that requires a spool server, you must specify the spool server on the DeviceAttributes tab before you save the device definition.</p>
Host printer	<p>The name of the StreamServe printer queue, for example <code>strs</code>. For information on configuring the StreamServe printer queue, see:</p> <ul style="list-style-type: none"> • Creating a local printer definition on page 23 • Example of setting up SAP connectivity on page 10 <p>Note: The name of the host printer is case sensitive and cannot contain any spaces.</p>

- 11 Click **Enter**.
- 12 Save the new device definition.

Activating XSF data output for a Smart Form

You can either activate XSF output for the Smart Form or specify XSF output for the print program.

We recommend you make these changes to a customized Smart Form (copied from an original Smart Form) to ensure the original Form is still available in the system, and that the customized Form is not overwritten in system upgrades.

Note: The customized Smart Form is given a name to differentiate the Form from the original Smart Form, usually starting with a Y or Z, for example `Z_F150_Dunn_SF`.

This section describes how to activate XSF data output by setting the **XSF Output Active** flag for a Smart Form. If you want to activate XSF output in the print program, see your SAP system documentation.

To activate XSF data output for a Smart Form

- 1 Create a copy of the original Smart Form from which you want to output XSF data. This section uses a copy of the `F150_Dunn_SF` invoice in the Smart Forms module as an example.

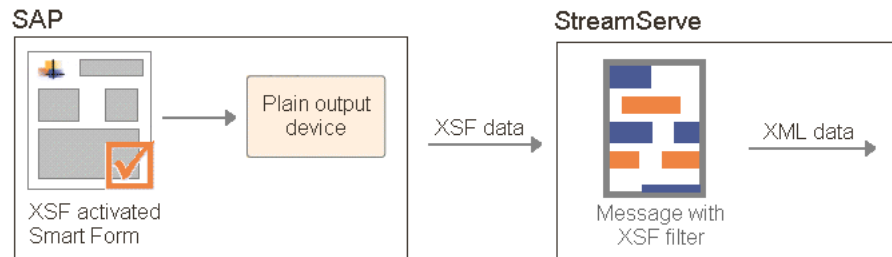
Note: Give the customized Smart Form a unique name to differentiate the Form from the original Smart Form, for example Z_F150_Dunn_SF.

- 2** In the transaction box, enter /nSmartForms. The initial Smart Forms window opens.
- 3** In the Form box, select the form which you want to activate XSF output for. This form should be a copy of an original Smart Form.
Note: Ensure the copied Smart Form has a unique name to differentiate it from the original Form, for example Z_F150_Dunn_SF.
- 4** Click **Enter**. The initial window reopens showing the form name.
- 5** Click **Change**. The Form Builder window opens.
- 6** Select the **Output options** tab.
- 7** In the XSF Output area, select **XSF output active**.
- 8** In the Output device box, enter the name of the plain output device you created in *Creating a plain output device* on page 20, for example ZSTRS.
- 9** Click **Enter**.
- 10** Save the Form.

Sending XSF data to StreamServe

To send XSF data from your SAP system, you need to activate XSF output for the Smart Form you want to use, then output the data using a plain output device.

The Message for the XSF data uses an XSF filter to convert the XSF data to well-formed XML data, which StreamServer can process.



To output XSF data from your SAP system to StreamServe, you need to complete the configuration describe in this section. To assist you, this section contains an example of how to send XSF data to StreamServe. You should ensure that this example works correctly before you activate XSF data output for a Smart Form. This example is configured for a Windows environment.

If you only want to generate an XSF file, without sending it to StreamServe (in case StreamServe is not installed), see [Creating an XSF output file](#) on page 25.

Required activities

- 1 [Creating a plain output device](#) on page 20
- 2 [Creating a local printer definition on page 23](#)
- 3 [Sending sample XSF data from SAP to StreamServe](#) on page 24

Creating a local printer definition

To capture data on your SAP system, you need to create a printer definition on your local machine (Windows) that you can use to receive data as the StreamServe port.

To create a printer definition

- 1 On your local machine, add a printer definition with the following settings:

Printer settings	
Local or Network Printer	Select Local printer . Note: Ensure Automatically detect and install my Plug and Play printer is <i>not</i> selected.

Printer settings	
Port	A new port of StreamServe type. Note: The StreamServe port type is only available when you have installed StreamServe port monitor. See the <i>StreamServe installation</i> documentation.
Port Definition	A name for the port, such as <i>strs</i> . Note: The port name must match the name of the host printer name specified for the output device. Enter a destination path/pipe to the StreamServe spool from where StreamServer will receive data, for example: <code>C:\StreamServe\spool</code>
Manufacturer	Select Generic .
Model	Select Generic/Text only .
Printer Name	A name for the printer, For example, <i>strs</i> . Note: Do not specify the printer as the default printer.
Shared As	A share name for the printer, for example <i>strs</i> . Note: The share name must match the name of the host printer name specified for the output device.

- 2 Right-click the new printer and select **Properties > Advanced > Print Processor**
- 3 Select **passthru** and **RAW**.

Sending sample XSF data from SAP to StreamServe

To print sample XSF data from your SAP system you simply print the XSF-activated Smart Form in the usual manner using the output device you configured for StreamServe.

The output device sends the XSF data to the host printer, which in this case is the StreamServe printer queue. The Message for the XSF data uses an XSF filter to convert the XSF data to well-formed XML data, which StreamServer can process.

After receiving the sample file, you can import the sample XSF data into the XMLIN tool to build the Event for the Smart Form, see [Creating a Message for XSF data](#) on page 45. Upon completion of the design phase, you can send the file to Communication Serve to test the design.

Note: This section uses an invoice from the SAP system as an example. It is assumed that an XSF-activated Smart Form has been assigned to process the output type RD00. For this scenario, it is done using transaction NACE.

To send sample XSF data to StreamServe

- 1 Log on to your SAP system as a user with administrative permissions.
- 2 In the transaction box, enter /nvf03. The Display Billing Document window opens.
- 3 In the Billing document box, select the billing document you want to print. The billing document must be an XSF-activated Smart Form.
- 4 Select **Billing document > Issue output to, Printer**. The Issue Output dialog box opens.
- 5 Ensure the correct output type is used. (The standard output type for an invoice is usually RD00.) Click **Execute**. The second Issue Output dialog box opens.
- 6 In the Logical destination box, enter the name of the output device you created for sending XSF data to StreamServe. See [Creating a plain output device](#) on page 20.
- 7 Ensure **Print immediately** is selected.
- 8 Click **Execute**. If printing is successful, your SAP system will launch SAPLPD on your local machine showing the print job in the log file. The output is sent to the device.

The output file is sent to the directory specified for the port definition of the local printer definition, for example C:\StreamServe\spool, see [Creating a local printer definition on page 23](#).

The Message for the XSF data uses an XSF filter to convert the XSF data to well-formed XML data, which StreamServer can process. You can import this data into the XMLIN tool to design the Event for the Form. See [Creating a Message for XSF data](#) on page 45.

Creating an XSF output file

Required activities

- 1 [Creating a plain output device](#) on page 20
Note: When configuring an output device for sending sample XSF data, ensure you specify the access method as **F** to print locally to a front end printer.
- 2 [Creating a printer definition for sample XSF data on page 26](#)
- 3 [Printing sample XSF data to a file](#) on page 26

Creating a printer definition for sample XSF data

To create an XSF output file, you need to create a printer definition on your local machine (Windows) that you can use to send the sample data as a file to your local machine.

To create a printer definition for sample XSF data

Note: You must have StreamServe port monitor installed locally. See the *StreamServe installation* documentation.

Printer settings	
Local or Network Printer	Select Local printer . Note: Ensure Automatically detect and install my Plug and Play printer is <i>not</i> selected.
Port	Select FILE (Print to File) .
Manufacturer	Select Generic .
Model	Select Generic/Text only .
Printer Name	A name for the printer, For example, <code>strs</code> . Note: Do not specify the printer as the default printer.
Shared As	A share name for the printer, for example <code>strs</code> . Note: The share name must match the name of the host printer name specified for the output device.

- 4 Right-click the new printer and select **Properties > Advanced > Print Processor**
- 5 Select **passthru** and **RAW**.

Printing sample XSF data to a file

You print sample XSF data from your SAP system using the output device configured to use the local printer definition. The sample data is sent from your SAP system to your local machine as a file.

Note: The output device must be configured with a Host Spool Access Method **F**, i.e. printing to a front-end computer.

The Message for the XSF data uses an XSF filter to convert the XSF data to well-formed XML data, which StreamServer can process. When you have received the file, you can use the sample data as a spool file to run through StreamServer and test your StreamServe configuration.

Note: This section uses an invoice from the SAP system as an example.

To print sample XSF data to a file

- 1 Log on to your SAP system as a user with administrative permissions.
- 2 In the transaction box, enter /nvf03. The Display Billing Document window.
- 3 In the Billing document box, select the billing document you want to print. The billing document must be an XSF-activated Smart Form.
- 4 Select **Billing document > Issue output to, Printer**. The Issue Output dialog box opens.
- 5 Ensure the correct output type is used. (The standard output type for an invoice is usually RD00.) Click **Execute**. The second Issue Output dialog box opens.
- 6 In the Logical destination box, enter the name of the output device you created for sending sample XSF data to StreamServe. See [Creating a printer definition for sample XSF data on page 26](#).
- 7 Ensure **Print immediately** is selected.
- 8 Click **Execute**. If printing is successful, your SAP system will launch SAPLPD on your local machine (assuming you have SAPGUI installed) showing the print job in the log file. The output is sent to the local StreamServe printer.
- 9 Click **OK**. You can import this file into the XMLIN tool to build the Event for the Smart Form. See [Importing sample XSF data into the XMLIN Event on page 46](#).

28 | Sending XSF data to StreamServe
Configuring SAP for XSF data

Configuring SAP for RDI data

This section describes how to configure the E-docs Connect solution with your SAP system to receive RDI data from your SAP system. RDI data (SAPscript Raw Data Interface) is created from Forms generated in the SAPscript Composer.

Requirements

To use this section, your SAP system should already be configured to use IDocs and you should have assistance from a person with SAP knowledge during the configuration phase.

Included activities

To configure SAP to process RDI data for StreamServe, complete the following steps:

- **Configuring your SAP system for RDI data**
See [Configuring SAP to send RDI data](#) on page 30.
- **Getting sample RDI data**
See [Sending RDI data from SAP to StreamServe](#) on page 35.

To process the RDI data automatically with StreamServe, a Design Center project must be deployed to a StreamServer application on the host receiving the data from SAP. See [Configuring StreamServe for RDI data](#) on page 59.

RDI supported SAP releases

The E-docs Connect solution supports RDI in all releases of SAP from 4.0.

Configuring SAP to send RDI data

Required activities

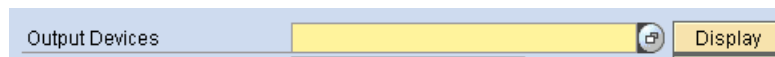
- [Creating a plain output device](#) on page 30
- [Activating RDI data output for a SAPscript Form](#) on page 31
- [Assigning customized SAPscript Forms to print programs](#) on page 33
- [Exporting customized SAPscript Forms](#) on page 34

Creating a plain output device

To send RDI data from your SAP system to StreamServe, you need to create a plain output device in SAP to output data.

To create a plain output device

- 1 Log on to your SAP system as a user with administrative permissions.
- 2 In the transaction box, enter `/nspad`. The initial Spool Administration window opens.
- 3 For the Output Devices option, click **Display**.



The List of Output Devices window opens.

- 4 Click the **Change** toolbar button.



- 5 Click the **Create** toolbar button.



The Create Output Device window opens.

- 6 Specify the output device settings.

Output device settings	
Output device	The name of the StreamServe output device, such as ZSTRS.
Short name	The short name for the StreamServe output device, such as ZSTR, or leave blank for SAP to generate a short name.

- 7 Click the **DeviceAttributes** tab.

- 8 Specify the device attribute settings.

Device settings	
Device type	Select one of the following depending on the code page used: PLAIN :SAPscript RDI(ASCII) UCPLAIN :SAPscriptRDI(Unicode)
Device class	Select Standard printer .

- 9 Click the **Access Method** tab.

- 10 Specify the settings.

Access Method settings	
Host Spool Access Method	Select the access method you want to use for the host spool. For example, L to print locally using a LP/LPR printer. Note: If you select an access method that requires a spool server, you must specify the spool server on the DeviceAttributes tab before you save the device definition.
Host printer	The name of the StreamServe printer queue, for example <code>strs</code> . For information on the StreamServe printer queue, see: <ul style="list-style-type: none"> • <i>Creating a local printer definition on page 35</i> • <i>Example of setting up SAP connectivity on page 10.</i> Note: The name of the host printer is case sensitive and cannot contain any spaces.

- 11 Click **Enter**.

- 12 Save the new device definition.

Activating RDI data output for a SAPscript Form

To activate RDI data output for a SAPscript Form, you need to either set a flag in the SAPscript Form or specify RDI output for the print program.

We recommend you make these changes to a customized SAPscript Form (copied from an original SAPscript Form) to ensure the original Form is still available in the system, and that the customized Form is not overwritten in system upgrades.

Note: The customized SAPscript Form is given a name to differentiate the Form from the original SAPscript Form, usually starting with a Y or Z, for example Z_RVINVOICE01.

This section describes how to activate RDI data output by setting a flag in a customized SAPscript Form. If you want to activate RDI output in the print program, see your SAP system documentation.

To activate RDI data output from a SAPscript Form

- 1 Create a copy of the original SAPscript Form from which you want to output RDI data. This section uses a copy of the RVINVOICE01 invoice in the Sales & Distribution module as an example.

Note: Give the customized SAPscript Form a unique name to differentiate the Form from the original SAPscript Form, for example Z_RVINVOICE01.

- 2 In the transaction box, enter /nSE71. The Form Painter: Request window opens.
- 3 Specify the SAPscript Form settings.

SAPscript Form settings	
Form	The name of the customized SAPscript Form you copied. Note: Ensure the customized SAPscript Form has a unique name to differentiate it from the original Form, for example Z_RVINVOICE01.
Language	The language for the <i>original</i> Form, such as DE (German). Note: The language for all Forms in a standard SAP system distribution is German.

- 4 In the Sub-objects area, select **Header**.
- 5 Click **Change**. The Form: Change Header window opens.
- 6 Select **Attributes > Miscellaneous**. The RDI Output Parameter dialog box opens.
- 7 In the RDI output mode box, select **RDI Output Mode: 'Spool' (X)**.



In SAP, **RDI output mode** is equivalent to **Enable Raw Data Interface** – any application using this SAPscript Form will generate raw data instead of formatted data, given that the value of RAW_DATA_INTERFACE in the application's print program is set to '*' (default value).

- 8 If you want to use a default output device for RDI data for this SAPscript Form, in the Output device box, enter the name of the default output device.
 If you specify a default output device, when a user prints data with this SAPscript Form, the system will override the printer selection and use the specified default output device. Thereby making RDI processing transparent to end-users, enabling you to select the same printer for standard SAP output and output to be processed by StreamServe.
 You must configure the default output device as described in *Creating a plain output device* on page 30.
- 9 Click **Enter**.

Assigning customized SAPscript Forms to print programs

When you have created a customized SAPscript Form that is RDI-activated in the SAP system, you need to assign the customized Form to the appropriate print program in the SAP system. SAP will then use the converted Form instead of the original SAPscript Form when generating output.

To assign a SAPscript Form to a SAP print program

- 1 Log on to your SAP system as a user with administrative permissions.
- 2 In the transaction box, enter `/nspro`.
- 3 Press F5 to access the *SAP Reference Implementation Guide*.
Note: The following steps describe how to assign a SAPscript Form to the print program for invoices in the Sales and Distribution module. Consult your SAP system documentation for the correct procedure for other documents.
- 4 Navigate to the folder containing **Maintain output types**.
- 5 Select **Maintain output types** and click the corresponding check mark. The output types are displayed in a list.
- 6 Double-click the **RD00 – Invoice** line. The Display View window opens.
- 7 Click **Processing program**. The Change View window opens.
- 8 In the Output Processing Programs table, change the Form value for the line with a value of 1 (Printer) in the Med (Output Medium) column. In the Form column, enter the name of the converted SAPscript Form, for example, `Z_RVINVOICE01`.
- 9 Save the program.
- 10 If the system requires you to enter a Change Request identifier, the Enter Change Request dialog box opens. Enter a Change Request identifier.
Note: If you have no existing requests, or if you are not authorized to create a new request, consult your SAP administrator.

- 11 Click **Enter**.

Exporting customized SAPscript Forms

In StreamServe, you can import the customized SAPscript Form for the RDI output into the SAP Form Reader, and use the fields contained within the Form to configure the Message for the raw data. The SAP Form Reader is a component of the StreamIN tool.

In order for you to import the SAPscript Form into the SAP Form Reader, you must first export the Form from your SAP system.

To export a customized SAPscript Form from SAP

- 1 In the transaction box, enter `/nse38`. The initial ABAP Editor windows opens.
- 2 In the Program box, enter `RSTXSCR`.
- 3 Click **Execute**. The SAPscript Export/Import window opens.
- 4 Specify the SAPscript export settings.

SAPscript export settings	
Object selection and session ctrl	Select Form . In the Object name box, enter the name of the customized SAPscript Form you want to export, for example <code>Z_RVINVOICE01</code> .
Ctrl parameters for file operation	Select From/on frontend .

- 5 Click **Execute**. The Transfer to a Local File dialog box opens.
- 6 In the File name box, specify a file in an appropriate directory, using the `*.for` extension. For example:

`C:\temp\Z_INVOICE01.for`

- 7 Click **Transfer** to copy the SAPscript Form to the file.

You can import this file into the SAP Form Reader to enable you to build a StreamServe Message for the SAPscript Form. See [Importing SAP RDI data into the SAP Form Reader](#) on page 62.

Sending RDI data from SAP to StreamServe

To send RDI data from your SAP system, you need to activate RDI output for the SAPscript Form you want to use, then output the data using a plain output device. When you activate RDI data output for a SAPscript Form, you can print raw data from the Form without a page layout.

The Message for the RDI data is configured with an E-docs agent, which enables StreamServer to recognize and process the RDI data correctly.

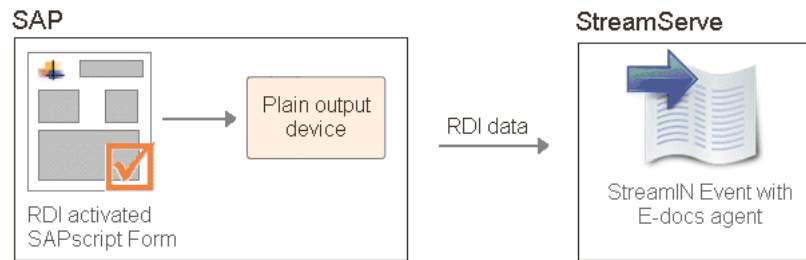


Figure 2 Sending RDI data from SAP to StreamServe

To output RDI data from your SAP system to StreamServe, you need to complete the configuration describe in this section. To assist you, this section contains an example of how to send RDI data to StreamServe. You should ensure that this example works correctly before you activate RDI data output for a SAPscript Form. This example is configured for a Windows environment.

If you only want to generate an RDI file, without sending it to StreamServe (in case StreamServe is not installed), see [Creating an RDI output file](#) on page 38.

Required activities

- 1 Create an output device for sending RDI data to StreamServe.
 See [Creating a plain output device](#) on page 30.
- 2 [Creating a local printer definition on page 35](#)
- 3 [Sending sample RDI data from SAP to StreamServe](#) on page 36

Creating a local printer definition

To capture data on your SAP system, you need to create a printer definition on your local machine (Windows) that you can use to receive data as the StreamServe port.

To create a printer definition

- 1 On your local machine, add a printer definition with the following settings.

Printer settings	
Local or Network Printer	Select Local printer . Note: Ensure Automatically detect and install my Plug and Play printer is <i>not</i> selected.
Port	A new port of StreamServe type. Note: The StreamServe port type is only available when you have installed StreamServe port monitor.
Port Definition	A name for the port, such as <i>strs</i> . Note: The port name must match the name of the host printer name specified for the output device. Enter a destination path/pipe to the StreamServe spool from where StreamServer will receive data, for example: C:\StreamServe\spool
Manufacturer	Select Generic .
Model	Select Generic/Text only .
Printer Name	A name for the printer, For example, <i>strs</i> . Note: Do not specify the printer as the default printer.
Shared As	A share name for the printer, for example <i>strs</i> . Note: The share name must match the name of the host printer name specified for the output device.

- 2 Right-click the new printer and select **Properties > Advanced > Print Processor**
- 3 Select **passthru** and **RAW**.

Sending sample RDI data from SAP to StreamServe

To print sample RDI data from your SAP system, you simply print the RDI-activated SAPscript Forms in the usual manner using the output device you configured for StreamServe.

After receiving the sample file, you can import the sample RDI data into the StreamIN tool using the SAP Form Reader to build the Event for the SAPscript Form, see [Creating a Message for RDI data](#) on page 60. Upon completion of the design phase, you can send the file to Communication Serve to test the design.

Note: This section uses an invoice from the SAP system as an example.

To send sample RDI data to StreamServe

- 1** Log on to your SAP system as a user with administrative permissions.
- 2** In the transaction box, enter `/nvf03`. The Display Billing Document window opens.
- 3** In the Billing document box, select the billing document you want to print. The billing document must be an RDI-activated SAPscript Form.
- 4** Select **Billing document > Issue output to, Printer**. The Issue Output dialog box opens.
- 5** Ensure the correct output type is used. (The standard output type for an invoice is usually RD00.) Click **Execute**. The second Issue Output dialog box opens.
- 6** In the Logical destination box, enter the name of the output device you created for sending RDI data to StreamServe:
 See [Creating a plain output device](#) on page 30.
- 7** Ensure **Print immediately** is selected.
- 8** Click **Execute**. If printing is successful, your SAP system will launch SAPLPD on your local machine showing the print job in the log file. See [Configuring SAP to send RDI data](#) on page 30. The output is sent to the device.

The output file is sent to the directory specified for the port definition of the local printer definition, for example `C:\StreamServe\spool`, see [Creating a local printer definition on page 35](#).

You can import this file into the StreamIN tool using the SAP Form Reader to design the Message for the Form. See [Importing SAP RDI data into the SAP Form Reader](#) on page 62.

Creating an RDI output file

Required activities

- 1 Create an output device for sending sample RDI data to StreamServe.
See [Creating a plain output device](#) on page 30
Note: When configuring an output device for sending sample RDI data, ensure you specify the access method as **F** to print locally to a front end printer.
- 2 [Creating a printer definition for sample RDI data](#) on page 38
- 3 [Printing sample RDI data to a file](#) on page 39

Creating a printer definition for sample RDI data

To create an RDI output file, you need to create a printer definition on your local machine (Windows) that you can use to send the sample data as a file to your local machine.

To create a printer definition for sample RDI data

- 1 On your local machine, add a printer definition with the following settings.

Printer settings	
Local or Network Printer	Select Local printer . Note: Ensure Automatically detect and install my Plug and Play printer is <i>not</i> selected.
Port	Select FILE (Print to File) .
Manufacturer	Select Generic .
Model	Select Generic/Text only .
Printer Name	A name for the printer, For example, <code>strs</code> . Note: Do not specify the printer as the default printer.
Shared As	A share name for the printer, for example <code>strs</code> . Note: The share name must match the name of the host printer name specified for the output device.

- 2 Right-click the new printer and select **Properties > Advanced > Print Processor**
- 3 Select **passthru** and **RAW**.

Printing sample RDI data to a file

You print sample RDI data from your SAP system using the output device configured to use the local printer definition. The sample data is sent from your SAP system to your local machine as a file.

When you have received the file, you can use the sample data as a spool file to run through StreamServer and test your StreamServe configuration.

Note: This section uses an invoice from the SAP system as an example. It is assumed that an RDI enabled SAPScript has been assigned to process the output type RD00. For this scenario, it is done using transaction NACE.

To print sample RDI data to a file

- 1 Log on to your SAP system as a user with administrative permissions.
- 2 In the transaction box, enter `/nVF03`. The Display Billing Document window.
- 3 In the Billing document box, select the billing document you want to print. The billing document must be an RDI-activated SAPscript Form.
- 4 Select **Billing document > Issue output to, Printer**. The Issue Output dialog box opens.
- 5 Ensure the correct output type is used. (The standard output type for an invoice is usually RD00.) Click **Execute**. The second Issue Output dialog box opens.
- 6 In the Logical destination box, enter the name of the output device you created for sending sample RDI data to StreamServe. See [Creating a printer definition for sample RDI data on page 38](#).
- 7 Ensure **Print immediately** is selected.
- 8 Click **Execute**. If printing is successful, your SAP system will launch SAPLPD on your local machine showing the print job in the log file. See [Configuring SAP to send RDI data on page 30](#). The output is sent to the printer and the Print to File dialog box opens.
- 9 Enter the full path and name of the SAP output file, for example:
`C:\SAP\invoice.rdi`
- 10 Click **OK**. You can import this file into the SAP Form Reader component of the StreamIN tool to build the Form Message. See [Importing SAP RDI data into the SAP Form Reader on page 62](#).

Generating SAPscript Form documentation

If you are unfamiliar with the fields and elements in the SAPscript Form or RDI file, you can generate HTML documentation of the SAPscript from your SAP system that contains a description of all fields.

Note: The following instructions and screen example are taken from SAP system 4.6c and newer - the transaction code is the same for all releases.

To generate documentation for a SAPscript Form

- 1 Log on to your SAP system as a user with administrative permissions.
- 2 In the transaction box, enter `/nse71`. The Form Painter: Request screen opens.
- 3 In the Form box, enter the name of the SAPscript Form, for example `Z_RVINVOICE01`.
- 4 Select **Documentation**.
- 5 Click **Display**. The Display Documentation screen opens showing a description of all elements and fields in the SAPscript Form.
- 6 When you have finished viewing the description, click **Exit** twice to return to the main screen.

Configuring StreamServe for XFP data

This section describes how to configure StreamServe to receive and process XFP data. XFP data is generated from PDF based print forms in the SAP system.

Note: This guide only contains instructions specific to configuring a StreamServe Platform for the E-docs Connect solution. For general information on configuring Platforms, see the *Design Center* documentation.

Prerequisites

The following section assumes you have already installed StreamServer and Design Center. You should know how to configure a StreamServe Platform and Message for XML data before you start configuring StreamServe for use with XFP data. See the *XMLIN* documentation.

Creating a Message for XFP data

The XFP interface lets you receive business data represented as XML generated from PDF-based print forms. The PDF-based forms currently supported in the E-docs solution are LiveCycle Designer forms.

By setting up a generic XMLIN message in StreamServe, a PDF-based Live Cycle Designer print form from SAP can be processed and rendered as designed in. A pattern is only set on the root node of the Message and no other field selection or mapping is required.

You can use any Process tool to create the output, but to re-use the design you must use Adobe LiveCycle Designer ES Process, where you also can specify a HTTP URI to dynamically select the print form from SAP in runtime.

Data from PDF based print forms is output in XFP (XML for PDF based print forms) format which is already well-formed XML and does not need filter conversion for use with StreamServe (as opposed to for example XSF).

However, to enable node collection mode on the XFP data when pattern matching, you can add the XFP Filter to the filter chain and specify headers to extract.

If the XFP input to StreamServe contains SmartForms text blocks, you have also the option to replace tags, for example `<new-line>` must be replaced with `\n`.

For more information on how to create a Message including using the filter chain, see [Creating a Message for XSF data](#) on page 45.

Running the XFP filter stand-alone

For test purposes you can run the XFP filter as a stand-alone application.

Windows

Browse to the E-docs \bin directory, by default:

```
<StreamServe installation>\Applications\SAP connect  
\<version>\edocs\bin
```

Enter the following in a command line:

```
xfpfilter <FILENAMEIN> <FILENAMEOUT>
```

Optionally, you can extract headers to enable node collection mode while pattern matching on the form name, e.g:

```
xfpfilter <FILENAMEIN> <FILENAMEOUT> --extract-headers=form,device
```

Optionally, you can replace tags if the XFP data contains SmartForm text blocks, e.g:

```
xfpfilter <FILENAMEIN> <FILENAMEOUT> --replace-  
tags=tab=\t,new_line=\n
```

UNIX

Browse to the XFP filter directory, e.g:

```
.../applications/sapconnect/sapedocs/xfpfilter-5.5.0.GA.952
```

Enter the following in a command line:

```
./start <FILENAMEIN> <FILENAMEOUT>
```

Optionally, you can extract headers to enable node collection mode while pattern matching on the form name, e.g:

```
./start <FILENAMEIN> <FILENAMEOUT> --extract-headers=form,device
```

Optionally, you can replace tags if the XFP data contains SmartForm text blocks, e.g:

```
./start <FILENAMEIN> <FILENAMEOUT> --replace-  
tags=tab=\t,new_line=\n
```

Configuring StreamServe for XSF data

This section describes how to configure StreamServe to receive and process XSF data. XSF data (XML for Smart Forms) is generated from Smart Forms in the SAP system.

Note: This guide only contains instructions specific to configuring a StreamServe Platform for the E-docs Connect solution. For general information on configuring Platforms, see the *Design Center* documentation.

Prerequisites

The following section assumes you have already installed StreamServer and Tools with XML Exchange functionality. You should know how to configure a StreamServe Platform and Message for XML data before you start configuring StreamServe for use with XSF data. See the *XMLIN* documentation.

Required activities

- [Creating a Message for XSF data](#) on page 45
- [To create an XML Event for XSF data](#) on page 45

Converting XSF to StreamServe XML

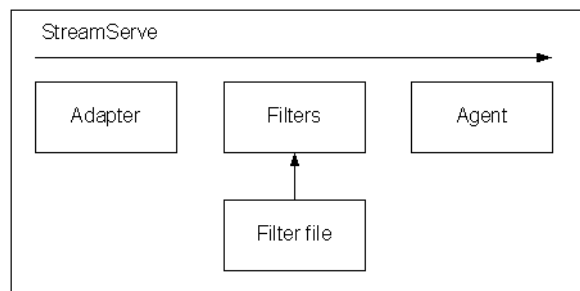
You can convert XSF to an XML format suitable XMLIN, either by configuring a Message or, for test purposes, run the XSF filter in stand-alone mode. For stand-alone mode, see [Running the XSF filter stand-alone](#) on page 57.

Using the XSF filter in a Message

For StreamServe to receive XSF data from SAP, you need to configure a Message to recognize the XSF data and transform the data using filter files.

When StreamServer detects a new XML file, the XSF file is processed through the `xsffilter` filter. This filter transforms the XSF data into well-formed XML data.

The XSF filter replaces all occurrences of <new-line/> and <tab/> with a line-break character (Hex <0D><0A>) and a tab character (Hex <09>), respectively.



See [Creating a Message for XSF data](#) on page 45.

Creating a Message for XSF data

To process XSF data from your SAP system, you use a StreamServe Message containing an XMLIN Event and one or more Processes. The Processes can be of any type. The Message uses an XSF filter as a loadable filter to convert the XSF data to well-formed XML data.

To enable node collection mode when processing the input XML, and also use pattern matching on the forms name, you can specify to extract header elements. The specified elements are added as attributes to the `<smartxsf>` tag.

If the XFP data contains SmartForm text blocks, you can also replace tags. For example, `<new-line>` must be replaced with `\n`.

To create a Message for XSF data

- 1 In Design Center, create a new Message definition.
- 2 Select or create the Resource Set for the Message.
- 3 Add a Filter Chain to the Resource Set.
- 4 Start the Filter Chain editor for the new Filter Chain.
- 5 Add an XSF Filter to the Filter Chain. (Right-click in the Filter Chain editor, and select **Add Filter > XSF Filter**).
- 6 Optionally, in the **Extract headers** property, enter a comma-separated list of header elements to extract.
- 7 Optionally, in the **Replace tags** property, edit the list of comma-separated tags to replace in the incoming XML.
- 8 Save and close the Filter Chain editor.

To create an XML Event for XSF data

- 1 Add an XMLIN Event to the Message.
- 2 Right-click the XMLIN Event and select **Settings**. The Event Settings dialog box opens.
- 3 Select **Collect entire documents**.
- 4 Click **OK**.

You can now configure the XMLIN Event, see [Configuring an XMLIN Event for XSF data](#) on page 45.

Configuring an XMLIN Event for XSF data

To configure the XMLIN Event, we recommend you define the input data manually using an XML sample document.

Using a sample document means you can define the fields to be extracted by selecting them directly in the XML document.



Examples provided in this chapter are adapted to the XSF data displayed. Recommendations are not included to serve as best practice, but rather as examples.

Specifying data collection method for XSF data

The collection method determines how the incoming data is collected during runtime, that is how much of the XML document is parsed and processed at a time. The more data parsed, the more data available to StreamServer, but the less performance achieved.

With XSF data, we recommend you use the Message data collection method for the XMLIN Event, which means StreamServer collects data from one Event (node-set selected by a pattern match) at a time. All data not included in an Event for which the Message collection method is configured, is collected using the node collection method.

Note: If you have specified to collect the entire documents on the Event settings, the value you specify on the Message data collection method option in the XMLIN tool has no effect.

To specify the data collection method for XSF data

- 1 In the XMLIN tool, select **Tools > Event Options**. The Options dialog box opens.
- 2 Select the **Message** tab.
- 3 In the Data collection level area, select **Message**.
- 4 Click **OK**.

Importing sample XSF data into the XMLIN Event

You can use a sample of the XML document to configure the Message structure automatically. Or you can create the fields, blocks, and patterns manually by selecting them from the sample. In both cases, the XMLIN tool suggests properties, such as:

- Names for the fields, blocks, and patterns.
- XSLT Patterns for the fields and patterns.

For information about generating the sample XSF data from you SAP system, see [Sending sample XSF data from SAP to StreamServe](#) on page 24.

If the sample document does not contain all data items that the incoming data can include, you can add the missing fields, blocks, or patterns to the Message model without using the sample. You can do this either by adding them manually or by selecting them from another sample document.

If the XSF document has a complicated structure and you have to configure field values using data from several nodes in the XML document, you cannot generate the XPath expressions and XSLT Patterns automatically.

Before you can import a sample XML document, you must add the sample XSF file to a Resource Set in StreamServe.

You can filter the sample file automatically before it is opened in the XMLIN tool, for example to make valid XML out of the the XSF content. You can use an original XSF input file as a sample.

To automatically use the XSF filter if sample is an .xsf file

- 1 In the XMLIN tool, select **Tools > Options**.
- 2 On the Samples tab, select **Filter input before opening sample**.

Select to filter the sample file before opening the sample in the XMLIN tool.

If the sample is a *.xsf file (SAP connectivity pack only), an XSF filter is used as filter. The XSF filter GUI is launched when you open the sample file, and in this GUI you must specify the appropriate filter parameters.

To import a sample XSF file into an XMLIN Event

- 1 In the XMLIN tool, select **File > Open Sample**. The Select Resource dialog box opens.
- 2 Locate and select the resource set containing the samples folder.
- 3 Right-click the samples folder, and select **Import**. The Select file to import dialog box opens.
- 4 Select the sample file you want to add to the Resource Set, and click **Open**. You created this sample file in [Printing sample XSF data to a file](#) on page 26. The sample file is added to the Resource Set.
- 5 Double-click on the sample file to select the file for the XMLIN tool, then click **OK**.

The sample file is loaded into the XMLIN tool. The resource is now available in the Resource view.

Note: If the sample is a *.xsf file, the XSF filter is used as filter. The XSF filter GUI is launched when you open the sample file, and in this GUI you must specify the appropriate filter parameters.

You can now define the blocks and fields in the Message. See [Defining blocks for XSF data](#) on page 56 and [Defining fields for XSF data](#) on page 52.

Declaring namespaces for XSF data

If namespaces are declared in the incoming data, the namespaces are used in the XSLT Patterns and XPath expressions. You must make sure that any namespaces declared in the incoming data are also declared in the Message definition. If not, the XSLT Patterns and XPath expressions that you configure may result in a failed pattern match or loss of data. The namespaces must be declared before you create the patterns or define the fields and blocks.

For example:

```
<ns_1:sf xmlns:ns="urn:sap-com:SmartForms:2000:xsf">
```

For information on how to declare namespaces, see the *XMLIN* documentation.

Generating match criteria for fields and patterns

When you define input data in the XMLIN tool, you use XSLT Patterns to specify the paths (match criteria) to the nodes of the patterns and fields. You can configure how the XMLIN tool suggests these XSLT Patterns. You can edit the suggested **Match** properties at any time. See [Defining match criteria for fields for XSF data](#) on page 53.

For XSF data, we recommend you use one level of Ancestor axis when generating match criteria for fields and patterns, but do not use path indexes.

To configure how match criteria are generated

- 1 Select **Tools > Event Options**. The Options dialog box opens.
- 2 Select the **Message** tab.
- 3 For XSF data, we recommend the following XSLT pattern settings.

XSLT pattern settings	
Ancestor axis level	Clear Use absolute path . Enter 1 in the field provided to specify the Ancestor axis level. Note: A long relative path (or an absolute path) decreases performance. Select the minimum level that still ensures an exact match.
Use path index	Disable. We recommend you do not use paths in XSLT patterns for XSF data.
Set sample data	Assigns the value of each field as sample data.
Create field variable	Assigns a variable to each field. The variable name is taken from the field name and the variable value from the field value. Note: Automatic assignment of variables decreases performance.

- 4 Click **OK**.

Configuring how field names are generated for XSF data

You can configure the XMLIN tool to generate field names based on XPath expressions. You can edit the generated field properties at any time. See [Defining fields for XSF data](#) on page 52.

For XSF data, we recommend you generate field names from the SAP field name in the XML document.

For information about the XSLT and XPath functions that you can use in the XPath expressions, see the *XMLIN* documentation.

To configure how field names are generated

- 1 In XMLIN, select **Tools > Event Options**. The Options dialog box opens.
- 2 Select the **Custom Field Names** tab.
- 3 Specify the **Custom field name for elements** as one of the following:
 - `name (.)`
 - `@name`

See [Custom field names for XSF data](#) on page 49.
- 4 Click **OK**.

Custom field names for XSF data

We recommend you use one of the following custom field names when generating field names for XSF data:

name (.)

Use this XPath expression to generate the same name for the field and the element. You should use this expression if the element does not have a name attribute.

For example, for the following XSF data, if you enter `name (.)` as the custom field name setting, then click on `<device>` attribute, the field name shown in XMLIN is `device`.

```
- <general>
  <version>1.11</version>
  <form>ZLB_BIL_INVOICE</form>
  <language>EN</language>
  <device>PRINTER</device>
  <output-device>RD11</output-device>
  <device-type>PLAIN</device-type>
  <tddataset>24684</tddataset>
  <tdsuffix1>13</tdsuffix1>
  <tdsuffix2>INVOICE</tdsuffix2>
  <tdlifetime>8</tdlifetime>
  <tdreceiver>STUMPE</tdreceiver>
  <tdcopies>001</tdcopies>
</general>
```

@name

Use this XPath expression to generate a field name different to the element name. You should use this expression if the element has a name attribute.

For example, for the following XSF data, if you enter @name as the custom field name settings, then click on the <sym> element to extract the Rechnung value, the field name shown in XMLIN is TITLE.

```
- <p name="L">
  - <chr name="TI">
    <sym name="TITLE">Rechnung</sym>
  </chr>
</p>
```

If you want to name the field with another attribute in that element, you can use @ and the attribute name. For example, in the following XSF data, @objname generates the field name ADRS_HEADER

```
<include-text name="HEADERTEXT" objname="ADRS_HEADER" language="EN">
```

Example 1 *Generating names and match values for fields in XSF data*

```
<text name="REF_DELNOTE" lang="DE" style="LO_STYLE">
  <p name="L">
    <sym name="IS_DLV_DELNOTE-HD_GEN-DELIV_NUMB">80007485</sym>
    <sym name="IS_DLV_DELNOTE-HD_GEN-CREA_DATE">/ 12.12.2000</sym>
  </p>
</text>
```

For this XSF data, the recommended settings would generate the following when you click on the first <sym> element:

- Field name:
IS_DLV_DELNOTE-HD_GEN-DELIV_NUMB
- Sample data for the Delivery Number:
80007485
- And XSLT pattern for field match criteria:
sym
You would need to manually add
[@name=' IS_DLV_DELNOTE-HD_GEN-DELIV_NUMB ']

This XSLT pattern for the field matches the <sym> element with the attribute name and a value of IS_DLV_DELNOTE-HD_GEN-DELIV_NUMB.

Defining a pattern for XSF data

You must create and configure at least one pattern in an XMLIN Event to identify the Message definition.

You can use all node types in an XML document as patterns, for example a text node or an element node. You specify the path to the node using an XSLT Pattern. You create the pattern as the top node of the node-set that you want to extract data from. When the pattern is found, StreamServer includes the corresponding node-set in the Event and data from the node-set can be extracted.

Note: Before you create patterns, you must make sure that any namespaces in the incoming data are declared in the Message definition.

Naming a pattern for XSF data

For XSF data the pattern in the Event is given the same name as the Smart Form from the SAP system. This is so StreamServer can recognize different types of incoming files and match them with the appropriate Event configuration. The name of the form is located in the XSF file <general> element.

For example, if you were defining an Event for the following XSF data, you would specify the name ZLB_BIL_INVOICE in the pattern

```
- <general>
  <version>1.11</version>
  <form>ZLB_BIL_INVOICE</form>
  <language>EN</language>
  <device>PRINTER</device>
  <output-device>RD11</output-device>
  <device-type>PLAIN</device-type>
  <tddataset>24681</tddataset>
  <tdsuffix1>RD11</tdsuffix1>
  <tdsuffix2>D028472</tdsuffix2>
  <tdlifetime>8</tdlifetime>
  <tdreceiver>STUMPE</tdreceiver>
  <tdcopies>001</tdcopies>
</general>
```

To configure a pattern for XSF data

- 1 In the XMLIN Message Browser, right-click the Message object and select **New > Pattern**. A pattern is created.
- 2 Select the pattern to show the properties in the Properties view.
- 3 Specify the pattern settings.

Pattern settings	
ID	A name for the Pattern, such as Pattern_Form.

Pattern settings	
Match	<p>The XSLT Pattern for the XSF data that specifies which Smart Form the data originates from. Use the following format:</p> <pre>path[header/general/form = 'form_name']</pre> <p>Where:</p> <ul style="list-style-type: none"> • <i>path</i> specifies the path to the node • <i>form_name</i> specifies the name of the Smart Form. <p>For example:</p> <pre>/sf/smartsf[header/general/form = 'ZLB_BIL_INVOICE']</pre>
Enabled	Ensure the pattern is enabled.

Defining fields for XSF data

You can extract data from all node types in an XML document.

To define fields for XSF data, you specify the path to the node using an XSLT Pattern, and the value of the node using an XPath expression.

You can also specify any content conditions that the node must satisfy. For example, the XPath expression allows you to create a field value by using data from different parts of the XML document.

Note: Do not define fields that are not required by the corresponding Process, as this decreases the performance of the Communications Server.



This section contains instructions on how to define fields for an XMLIN Event for XSF data. These instructions are only one example how you can do this, as the process for building field values in an XSF structure can vary from one SAP implementation to another.

Note: We recommend you first add fields at the root level (Message object), then create any required blocks, and create fields to be added to blocks. See [Defining blocks for XSF data](#) on page 56.

Prerequisites

Before you define fields in the XMLIN Event, you need to:

- Configure how the XMLIN tool generates field names and match criteria from the XSF data, see [Generating match criteria for fields and patterns](#) on page 48 and [Configuring how field names are generated for XSF data](#) on page 49.

- Make sure that any namespaces in the incoming data are declared in the Message definition. See [Declaring namespaces for XSF data](#) on page 48.

To define fields using an XML document as a template

- 1 In the Message Browser, select the object you want to add the field to, for example, the Message object or a block.
- 2 Select the **Field Tool** toolbar button.
- 3 In the XML document, select the node you want to use for the field match. A field is created and the field properties are suggested by the XMLIN tool. The XMLIN tool suggests a name and a match value based on the XML custom field name settings you have configured.
- 4 Specify the path to the node that contains the information you want to extract and the value of the node that contains the field in the **Match** Property. See [Defining match criteria for fields for XSF data](#) on page 53.

Defining match criteria for fields for XSF data

When you create a field for XSF data, you must specify the path to the node that contains the information you want to extract. You specify the path as an XSLT Pattern in the **Match** property in the Properties view.

If you create the field using a resource, XMLIN suggests a path according to the custom field name settings you have defined, see [Generating match criteria for fields and patterns](#) on page 48.

Example match value							
Format	<i>path/element_name/text</i> [@name = ' value']						
Accepts	<table> <tr> <td><i>path</i></td> <td>The path to the element in the XSF data.</td> </tr> <tr> <td><i>element_name</i></td> <td>The name of the element in the XSF data</td> </tr> <tr> <td><i>value</i></td> <td>The match criteria, which for XSF data specifies the XSF element attribute</td> </tr> </table>	<i>path</i>	The path to the element in the XSF data.	<i>element_name</i>	The name of the element in the XSF data	<i>value</i>	The match criteria, which for XSF data specifies the XSF element attribute
<i>path</i>	The path to the element in the XSF data.						
<i>element_name</i>	The name of the element in the XSF data						
<i>value</i>	The match criteria, which for XSF data specifies the XSF element attribute						
Returns	<table> <tr> <td>field ID</td> <td>WT</td> </tr> <tr> <td>match value</td> <td>chr [@name= ' WT']</td> </tr> </table>	field ID	WT	match value	chr [@name= ' WT']		
field ID	WT						
match value	chr [@name= ' WT']						



Using a long or short path in a match value

When defining a match value for a field for XSF data, you should use the shortest path possible, while still ensuring an exact match in the SAP data. Using a long path (relative or absolute) significantly decreases performance. For example:

- Recommended

```
sym[@name='GS_HD_ADR-PARTN_NUMB']
```

- Not recommended

```
smartxsf/data/window/table/tbody/tr/tc/text/p/  
sym[@name='GS_HD_ADR-PARTN_NUMB']
```

Match values for fields

When defining match values for fields, you need to specify an XSLT pattern that enables StreamServer to locate the correct record in the recurrent data. For XSF data, you use the name of the text attribute as the match value for the element.

Example 2 *Extracting a text attribute using a match value*

The item number `GS_IT_GEN-ITM_NUMBER` is an example of a field belonging to a block. This example shows how the item number `000010` is extracted using the specified match value.

For example:

```
<tc cell="1">  
  <text name="IT_NUM" lang="EN" style="LO_STYLE"  
    appmode="new-line">  
    <p name="L">  
      <sym name="GS_IT_GEN-ITM_NUMBER">000010</sym>  
    </p>  
  </text>  
</tc>
```

Would have the following field name

```
GS_IT_GEN-ITM_NUMBER
```

and the following match value

```
text[@name='IT_NUM']
```

Match values for recurrent fields with no unique identifiers

When defining match values for recurrent fields that do not have unique identifiers, you need to specify for StreamServer to search for an attribute within an element, and if found, will read the data within the attribute.

Note: Ensure you create the recurrent fields at the correct level, so that all records in the data is extracted.

Example 3 *Extracting data from recurrent fields with no unique identifiers*

To read all the data for the following address element:

```
- <window name="HEADER" page="FIRST" page-id="001">
- <include-text name="HEADERTEXT" style="SAPADRS" objname="ADRS_HEADER" id
  <p name="AS">SAP AG</p>
  <p name="AS">Integration and Certification Center</p>
  <p name="AS">Phone: (+49)-6227 474 74</p>
  <p name="AS">Fax: (+49)-6227 575 75</p>
  <p name="AS">Address: D-69189 Walldorf</p>
  <p name="AS">http://www.sap.com/softwarepartner</p>
  <p name="AS" />
</include-text>
</window>
```

You could have the following:

- field name
ADRS_HEADER
- match value
window/include-text [@objname='ADRS_HEADER']

StreamServer would locate the address SAP element, and then search for the ADRS_HEADER text attribute. If a match was found, the Server would read all the data contained within the ADRS_HEADER attribute.

Error handling with multiple instances

Errors can occur when StreamServer processes a document with a field matching multiple elements. To solve this problem, you need to extend the match path with another parent element, and if necessary, include its name attribute.

Example 4 *Error handling multiple instances*

For example, if you create a root field with the following match criteria:

```
chr [@name='WT']
```

When the server processes a document, this field matches several different elements, and the following error message is displayed in the server log:

```
"Multiple instances at root level"
```

To solve this problem, you need to extend the match path with another parent element, such as:

```
p [@name='L'] /chr [@name='WT']
```

If multiple root instances still occurred, you need to add another parent element, and so on, until there was only one match per element. For example:

```
text [@name= ' INFO ' ] /p [@name= ' L ' ] /chr [@name= ' WT ' ]
```

Defining blocks for XSF data

A block is a container for recurring fields. You must create separate blocks for every type of recurring data. A block can contain fields and sub-blocks.

Note: The block is identified by the fields within the block. This means that you can define the block from any node, not only from the node that actually includes the recurring fields. You can even create a block from another template XML document than the one that includes the fields.

To define a block using an XML document as a template

- 1 In the Message Browser, right-click the object you want to add the block to, and select **New > Block**. A block is created.
- 2 Give the block a descriptive name for the recurrent data items.
- 3 Create the fields for the block, see [Defining fields for XSF data](#) on page 52.

Running the XSF filter stand-alone

For test purposes you can run the XSF filter as a stand-alone application.

Windows

Browse to the E-docs \bin directory, by default:

```
<StreamServe installation>\Applications\SAP connect  
\<version>\edocs\bin
```

Enter the following in a command line:

```
xsffilter <FILENAMEIN> <FILENAMEOUT>
```

Optionally, you can extract headers to enable node collection mode while pattern matching on the form name, e.g:

```
xsffilter <FILENAMEIN> <FILENAMEOUT> --extract-headers=form,device
```

Optionally, you can replace tags in the SmartForm data, e.g:

```
xfpfilter <FILENAMEIN> <FILENAMEOUT> --replace-  
tags=tab=\t,new_line=\n
```

UNIX

Browse to the XSF filter directory, e.g.:

```
.../applications/sapconnect/sapedocs/xsffilter-5.5.0
```

and enter the following in a command line:

```
./start <FILENAMEIN> <FILENAMEOUT>
```

Optionally, you can extract headers to enable node collection mode while pattern matching on the form name, e.g:

```
./start <FILENAMEIN> <FILENAMEOUT> --extract-headers=form,device
```

Optionally, you can replace tags in the SmartForm data, e.g:

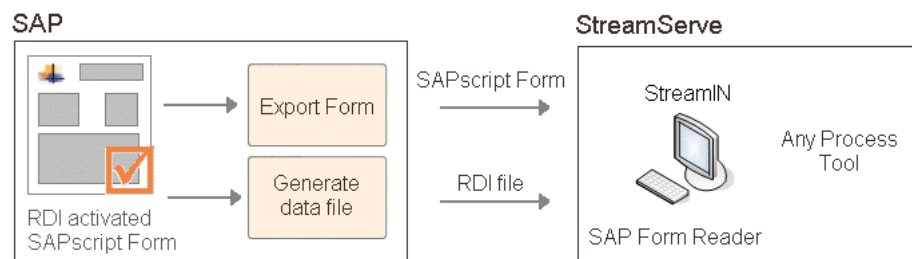
```
./start <FILENAMEIN> <FILENAMEOUT> --replace-  
tags=tab=\t,new_line=\n
```


Configuring StreamServe for RDI data

Note: This guide only contains instructions specific for configuring the E-docs Connect solution. For general information, see the standard StreamServe documentation.

This section describes how to configure StreamServe to receive and process RDI data. To process RDI data from your SAP system, you use a Message containing a StreamIN Event and one or more Processes.

To build the StreamIN Event, you use the SAP Form Reader to import RDI data for each SAPscript Form you want to use. The SAP Form Reader is an add-on component installed with the E-docs connect solution.



StreamServe can receive two types of RDI data.

RDI data - standard output mode

RDI data output in standard mode contains full logical information of the SAPscript Form, including window and element information. If you want to create unique field names from the RDI file, you should use this option.

RDI data - simple output mode

RDI data output in simple mode contains the same header information as for standard RDI data, however data records are reduced to symbol name and value, separated by spaces.

Required activities

- [Creating a Message for RDI data](#) on page 60
- [To create a StreamIN Event for RDI data](#) on page 60
- [Configuring a StreamIN Event for RDI data](#) on page 62

Creating a Message for RDI data

To process RDI data from your SAP system, you use a StreamServe Message containing a StreamIN Event and one or more Processes. The Processes can be of any type, for example StoryTeller.

To create a Message for RDI data

- In Design Center, create a new Message definition.

To create a StreamIN Event for RDI data

- 1 Add a StreamIN Event to the Message.
- 2 Give the StreamIN Event the same name as the SAPscript Form that you have exported from your SAP system. See [Naming a StreamIN Event for a SAPscript Form](#) on page 60.
- 3 Right-click the StreamIN Event and select **Settings**. The Event Settings dialog box opens.
- 4 Select the Agent Settings tab.
- 5 From the Input Type list select one of the following agents to specify which type of RDI data you want to receive for the Event:

StreamServe Connect for SAP - E-docs

RDI data output in standard mode contains full logical information of the SAPscript Form, including window and element information.

StreamServe Connect for SAP - E-docs/Simple Mode

RDI data output in simple mode contains the same header information as for standard RDI data, however data records are reduced to symbol name and value, separated by spaces.

- 6 Click **OK**.

You can now use the SAP Form Reader to build the StreamIN Event. See [Configuring a StreamIN Event for RDI data](#) on page 62.

Naming a StreamIN Event for a SAPscript Form

StreamServer must be able to recognize different types of incoming files and match them with the appropriate StreamServe Event. For RDI data, the Event name must correspond to the name of the SAPscript Form used to generate the data in your SAP system.

The name of the SAPscript Form is included on the first line of the RDI file. In SAP 4.x the name of the SAPscript Form is located in position 22.

Example 5 *Naming an Event for a SAPscript Form*

The following example is from a SAP 4.x system and the name of the SAPscript Form is ZS_RVINVOICE01.

```
HO40A018500000021153E ZS_RVINVOICE01 | PRINTER  
S  
CCODEPAGE 1100 LANGUAGE EN  
CPAGENAME FIRST
```

You can ask your SAP administrator for the name of the SAPscript Form, or you can open the RDI file you created earlier to determine the name for the Event.

Configuring a StreamIN Event for RDI data

You use the SAP Form Reader to configure a StreamIN Event for RDI data. The SAP Form Reader is an add-on component to the StreamIN tool included in the E-docs Connect installation.

Note: To import RDI sample files or SAPScript form files they must be stored in the resource set.

Included activities

Use the SAP Form Reader to perform the following functions:

- **Import RDI data into the Event**
Import RDI data as a dictionary file or SAPscript Form into the Event. The data is used to build the Message structure.
See *Importing SAP RDI data into the SAP Form Reader* on page 62.
- **Import an RDI overlay file**
Import an RDI file containing sample data from SAP.
See *Applying sample data in the StreamIN Event* on page 65.
- **Create a Page Layout Template file**
Create a Page Layout Template file for each page of the imported SAPscript Form.
See *Creating a Page Layout Template* on page 67.
- **Map SAP fonts to StreamServe fonts**
Map internal SAP fonts to a font which will appear in StreamServe. You can specify the font name, style and size for the font.
See *Mapping SAP fonts to StreamServe fonts* on page 68.
- **Change Form language**
The default language for SAPscript Forms is German (**D**). You can change the language for the Form in the SAP Form Reader.
See *Changing the SAPscript Form language* on page 67.

Importing SAP RDI data into the SAP Form Reader

Using the SAP Form Reader you can either:

- import RDI data from your SAP system as a SAPscript Form; or
- import as an RDI file generated from a SAPscript Form.

Importing a SAPscript Form

When importing RDI data generated from SAPscript Form, all fields are included in the SAPscript Form.

See *Importing a SAPscript Form into the SAP Form Reader* on page 63.

Importing an RDI file from a SAPscript Form

When importing an RDI file generated from a SAPscript Form, only fields that were contained in the printed data are included. If you import an RDI file, you need to ensure the file contains all the fields you require.

You can import an RDI file in either standard output mode or simple spool output mode.

See *Importing an RDI file into the SAP Form Reader* on page 64.



Deleting unrequired tabs

Every time you import RDI data into the SAP Form Reader, the data is allocated a new tab in the Integration Tool browser. If you import several files, your StreamServe configuration will become quite large. Therefore, when you no longer require RDI data for an imported file, you should delete the tab.

Importing a SAPscript Form into the SAP Form Reader

To use the SAP Form Reader to create a Message from a SAPscript Form, you can import the SAPscript Form into the Reader.

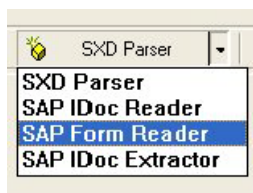
Requirements

Before you import a SAPscript Form into the SAP Form Reader, you need to first export the SAPscript Form from the SAP system:

- SAP 4.6/4.7 - *Exporting customized SAPscript Forms* on page 34

To import a SAPscript Form into the SAP Form Reader

- 1 In Design Center, open the Event you created in *Creating a Message for RDI data* on page 60.
- 2 In StreamIN, from the active connectors list in the Integration Tool browser, select **SAP Form Reader**.



- 3 Click **SAP Form Reader**. The Import dialog box opens.
- 4 Select **SAPscript Format**.
- 5 Click **OK**. The Select FOR file dialog opens.

- 6 Browse to the SAPscript Form you want to import. The RDI data for the SAPscript Form is imported into the SAP Form Reader, with the browser showing a new tab for the Form.

Importing an RDI file into the SAP Form Reader

You can import an RDI file containing sample data into the SAP Form Reader, and use the data to configure the Message layout. For information on how to create an RDI file, see [Sending RDI data from SAP to StreamServe](#) on page 35.

When you import an RDI file, only those fields included in the printed data from the SAP system are imported into the SAP Form Reader. For example, if you import an RDI file that does not include a Discount field as it was not used when the data was printed, you will need to manually add that field to the Message.

If you want to import all fields from the SAPscript Form, see [Importing a SAPscript Form into the SAP Form Reader](#) on page 63. If you are handling printed data, you can only use the RDI format.

Note: You should import an RDI file rather than a SAPscript Form when using Print Workbench in the SAP IS-U/CCS module where text elements are not defined within SAPscript Forms.

RDI file formats

You can import two types of RDI files into the SAP Form Reader:

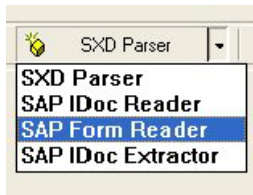
- RDI file - Output mode spool
RDI files output in standard mode contains full logical information of the SAPscript Form, including window and element information. If you want to create unique field names from the RDI file, you should use this option.
- RDI file - Output mode simple spool
RDI files output in simple spool mode contain the same header information as for standard RDI files, however data records are reduced to symbol name and value, separated by spaces.

Note: A simple spool RDI file does not include full logical information, such as window and element information. If you need this information to create unique field names, use the **RDI - Output mode spool** option instead.

Note: In order to import simple spool RDI files, you must have specified the **StreamServe Connect for SAP - E-docs/Simple** agent when defining the Event settings, see [Creating a Message for RDI data](#) on page 60.

To import an RDI file into the SAP Form Reader

- 1 In Design Center, open the Event you created in [Creating a Message for RDI data](#) on page 60.
- 2 In StreamIN, from the active connectors list in the Integration Tool browser, select **SAP Form Reader**.



- 3 Click **SAP Form Reader**. The Import dialog box opens.
- 4 To import a standard RDI file, select **RDI - Output mode spool**.
 To import a simple spool RDI file, select **RDI - Output mode simple spool**.
- 5 Click **OK**. The Open dialog box opens.
- 6 Locate the directory containing the RDI file you want to import. In the File Name box, enter the name of the file and click **OK**.
Note: You can only load an RDI file for a SAPscript Form whose name matches the Message. See *Naming a StreamIN Event for a SAPscript Form* on page 60.
- 7 Click **Open**.
 The RDI file is imported into the SAP Form Reader, with the browser showing a new tab for the data. Click the tab to access the data.

Applying sample data in the StreamIN Event

Note: This function is not available for SAP 3.0/3.1.

In StoryTeller and PageOUT, you can select to show data values instead of field names. As SAP often uses long field names, showing data values can make the fields easier to view and work with.

To view sample data in an Event, you import an RDI file containing sample data from SAP. For example, for the Invoice_Date field, the value 01/03/2001 will be displayed.

You can only load an RDI file containing sample data from a SAPscript Form whose name matches the Event. See *Naming a StreamIN Event for a SAPscript Form* on page 60.



Apply RDI data before exiting StreamIN

If you exit StreamIN after having imported the SAPscript Form, you will no longer be able to apply sample RDI data. It is therefore very important to apply the RDI data before you start building the basic structure of the Message.

For information on how to create an RDI file, see *Sending RDI data from SAP to StreamServe* on page 35.

To apply an RDI overlay file

- 1 In StreamIN, click **Options** in the Integration Tool browser, then select **Apply RDI data**. You are prompted to import an RDI overlay file for the default language.
- 2 Click **Yes** to confirm that you want to import an RDI overlay file. The Open dialog box opens.
- 3 In the File name box, enter the path and name of the *.rdi file containing sample data to import into StreamIN.
- 4 Click **Open**.
- 5 Select **View > Properties Window** to display the sample data added to the field, for example, sample data from an invoice.

Configuring the StreamIN Event for RDI data

When you have imported the SAPscript Form or RDI file, you can use the SAP Form Reader to configure the Event. When you have imported your SAPscript Form or RDI into the SAP Form Reader, all fields from the imported data are shown in a block structure in the Integration Tool Browser.

If you are unfamiliar with the fields and elements in the SAPscript Form or RDI file, you can generate documentation describing the fields in the SAPscript Form from your SAP system. See [Generating SAPscript Form documentation](#) on page 40.

**You should not change field labels**

Field labels in the Event must match the corresponding field names in the SAPscript Form in the SAP system. If you change the field labels, you will need to change the field names in the SAP system.

You should analyze the data and determine whether to place data in a block or as fields directly under the Message (root level). Generally, you should place fields that could have multiple occurrences in the data (RDI) in a block. You should place all fields that appear only once, for example, fields that contain an invoice date or number, at the root level, ie. directly under the Message.

For more information on configuring StreamIN Events, see the *StreamIN* documentation.

To configure a StreamIN Event for RDI data

- 1 In Design Center, open the Event you created in [Creating a Message for RDI data](#) on page 60.

StreamIN opens showing the Event containing the SAPscript Form or RDI file you imported in [Importing SAP RDI data into the SAP Form Reader](#) on page 62.

- 2 In the Integration Tool browser, open the structure of the imported data to show all blocks and fields. If you want to view sample data in the Event, see [Applying sample data in the StreamIN Event](#) on page 65.
- 3 From the Integration Tool browser, drag and drop the blocks or fields you want to include from the SAPscript Form or RDI file to the appropriate location in the Message browser.

Creating a Page Layout Template

A Page Layout Template file (*.plt) is an XML file that specifies the coordinates for the placement of all objects in an Event based on the layout from the data source.

In the SAP Form Reader, you can generate a Page Layout Template file for each page of a SAPscript Form imported in the StreamIN Event. You can then apply the Page Layout Template file to a PageOUT Process and auto-generate the Process based on the template layout.

For more information, see the *Page Layout Template* documentation.

To create a Page Layout Template file for a SAPscript Form page

- 1 In StreamIN, import the SAPscript Form you want to create a Page Layout Template file from. See [Importing a SAPscript Form into the SAP Form Reader](#) on page 63.
- 2 In the Integration Tool browser, click **Options**, and select **Create PLT**.
- 3 You are prompted to confirm whether you want to create a PLT file of the SAPscript Form for the current language. Click **Yes** to continue. The Save As dialog box opens.
- 4 Enter a name for the file. The default is the same name as the Event.
- 5 Click **Save**.
- 6 If the SAPscript Form contains more than one page, the Select page dialog box opens.
- 7 Select the page you want to create the Page Layout Template file from those pages listed, and click **OK**. The file is saved with a *.plt file extension.

Changing the SAPscript Form language

When you import a copy of an original SAPscript Form, the default language for the Form is German (**D**). You can change the language for the Form in the SAP Form Reader.

To change the SAPscript Form language

- 1 In StreamIN, import the SAPscript Form into the SAP Form Reader. See [Importing a SAPscript Form into the SAP Form Reader](#) on page 63.

- 2 In the Integration Tool browser, click the Change Language button (shows the title of the SAPscript Form).
- 3 Select a language from those listed, for example **E** for English.
- 4 The name of the SAPscript Form shown in the Integration Tool browser is shown in the selected language.

Mapping SAP fonts to StreamServe fonts

In SAP data, some fonts are referenced with internal SAP names. In the SAP Form Reader, you can map these SAP fonts to a font used in StreamServe. You can specify the font name, style and size for the font in StreamServe.

If you auto-generate a PageOUT Process based on a Page Layout Template (*.plt) of a SAPscript Form, the Process will use fonts mapped in the SAP Form Reader. If you do not map any fonts, the Process will use the SAP fonts.

To map internal SAP font names to StreamServe font names

- 1 In StreamIN, import the SAPscript Form you want to map SAP font names to StreamServe font names. See *Importing a SAPscript Form into the SAP Form Reader* on page 63.
- 2 In the Integration Tool browser, click **Options** and select **Edit FontMap**. The FontMap dialog box opens showing the names of the internal SAP fonts which are used in the current SAPscript Form.
- 3 In the R3 Font column, double-click the SAP font name you want to map. The Font dialog box opens.
- 4 Select a name, style and size for the font as it will appear in StreamServe.
- 5 Click **OK**. The name of the mapped font is shown in the Strs Font column.
- 6 Repeat steps 3 to 5 for all the internal SAP fonts you want to map.
- 7 Click **OK**.

Comparing Messages

In the SAP Form Reader you can use **Difference** mode to compare the current Event (Message) with an imported Message, and display the differences.

Comparing Messages can be useful, for example, if you have built your own Invoice template Message, you could import a new Invoice SAPscript Form into the SAP Form Reader and compare the new Form with your existing Message. If you have added three new fields in the new SAPscript Form, those three fields are indicated as different (diff) in the SAP Form Reader.

To set Difference mode

- 1 In Design Center, open the StreamIN Event you created in *Creating a Message for RDI data* on page 60.

- 2** In the StreamIN Integration Tool browser, click **Mode** and select **Diff**. (Normal is the default.)
- 3** Select **File > Load Message Definition** to load a new Message. The Load Message Definition dialog box opens.
- 4** Click **Browse**. The Select Resource dialog box opens.
- 5** From the Resource Sets, select the dictionary file you want to use to create the Message structure and click **OK**.

If there are items in the dictionary that StreamServe cannot process, or that already exist in the Message structure, they will be removed before the Message structure is created.

The SAP Form Reader indicates objects that differ from those already in the current Event with **(diff)** following the name of the object.

70 | **Configuring a StreamIN Event for RDI data** **Configuring StreamServe for RDI data**

StreamServe and the XFP format

The E-docs Connect solution enables StreamServe to receive data generated from PDF based print forms in your SAP system. Data from PDF based print forms is output in XFP (XML for PDF based print forms) format which is already well-formed XML and does not need filter conversion for use with StreamServe (as opposed to for example XSF).

StreamServe and the RDI format

The E-docs Connect solution enables StreamServe to receive raw data generated from SAPscript Forms in your SAP system. You output data from SAPscript Forms as raw data using the RDI (Raw Data Interface) format.

RDI data types

StreamServe can receive two types of RDI data:

- **RDI data - standard output mode**
RDI data output in standard mode contains full logical information of the SAPscript Form, including window and element information. If you want to create unique field names from the RDI file, you should use this option.
- **RDI data - simple output mode**
RDI data output in simple mode contains the same header information as for standard RDI data, however data records are reduced to symbol name and value, separated by spaces.

RDI data

RDI data is divided into different sections or ‘records’. These records are:

- Header records
- Control records
- Data records
- Archive records.

Example 6 Samples of RDI data

RDI data - standard output mode

```

Header records — [ H040A018500000013644ERVDELNOTE      PRINTER gvb01
                   S
Control records — [ CCODEPAGE 1100 LANGUAGE EN
                   CPAGENAME FIRST
Data records —   [ DMAIN      XXHEADER_DATA
                   DMAIN      HEADER_DATA
                   DMAIN      HEADER_DATA
                   DMAIN      HEADER_DATA
                   DMAIN      HEADER_DATA
                   DMAIN      HEADER_DATA      VBDKL-VSBED_BEZ
                   DMAIN      HEADER_DATA      VBDKL-INCO1
                   DMAIN      HEADER_DATA      VBDKL-INCO2
                   DMAIN      HEADER_DATA
                   DMAIN      HEADER_DATA
                   DMAIN      HEADER_DATA
                   DMAIN      HEADER_DATA
                   DMAIN      HEADER_DATA
                   DMAIN      HEADER_DATA
                   DMAIN      HEADER_DATA      VBDKL-BRGEW
                   DMAIN      HEADER_DATA      VBDKL-NTGEW
                   DMAIN      HEADER_DATA
                   DMAIN      XITEM_HEADER      ULINE

```

RDI data - simple output mode

```

Header records — [ H046A113000000016204DZ_BCA_BAST      PRINTER
                   S
Control records — [ CCODEPAGE 1100 LANGUAGE DE
                   CPAGENAME FIRST
                   CRDI-CONTROL %%LINES-BEGIN ZUSAMMENFASSUNG_BAST TEXT ST DE
Data records —   [ DWA_BASTHEAD-BKSTATLDAT 20.02.0020
                   DWA_BASTHEAD-BKSTATLTIM 13:00:01
                   DWA_BASTHEAD-BKSTATDATE 16.11.2001
                   DWA_BASTHEAD-BKSTATTIME 19:08:04
                   DWA_BASTHEAD-BASTSERIES NR 00001
                   DWA_BASTHEAD-BKSTATNO 001
                   DWA_BASTHEAD-ACNUM_EXT 1000048085
                   DWA_BASTHEAD-BANKKEY 20010020
                   DWA_SENDER-ORGNOME Standort Hamburg
                   DSTXADDRESS-TDPARGRAPH AD
                   DSTXADDRESS-TYPE 2
                   DSTXADDRESS-NUMBER
                   DSTXADDRESS-FROM_COUNT
                   DSTXADDRESS-RECEIVER_L
                   DSTXADDRESS-PRIORITY
                   DSTXADDRESS-DELIVERY

```

Note: These graphics have been cropped and do not display all records.

Header records

A header record in raw data identifies the data and acts as a trigger for StreamServe to determine that an Event should be executed.

StreamServer uses header records to recognize different types of incoming files and match them with an appropriate Event configuration. The name of the Event must correspond to the name of the SAPscript Form used to generate the raw data.

The name of the SAPscript Form is included on the first line in an RDI file. In SAP 4.x the name of the SAPscript Form is located in position 22.

Example 7 Header record

The following example is from SAP system 4.x using the ZS_RVINVOICE01 SAPscript Form.

```

H040A018500000021153E2S_RVINVOICE01 PRINTER
S
CCODEPAGE 1100 LANGUAGE EN
CPAGENAME FIRST
    
```

Information contained in the header record is automatically inserted into the following header variables:

Variable Name	Position	Length (chars)
\$r3_rdiversion	2	6
\$r3_client	8	3
\$r3_docnumber	11	10
\$r3_lslanguage	21	1
\$r3_formname	22	16
\$r3_devicetype	38	8
\$r3_terminal	46	64
\$r3_batchmode	110	1
\$r3_tdpageslct	111	60
\$r3_tdcopies	171	3
\$r3_tddest	174	4
\$r3_tdprinter	178	8
\$r3_tdpreview	186	1
\$r3_tdnoprev	187	1
\$r3_tdnoprint	188	1

Variable Name	Position	Length (chars)
\$r3_tdnwid	189	1
\$r3_tddataset	190	6
\$r3_tdsuffix1	196	4
\$r3_tdsuffix2	200	12
\$r3_tdimmed	212	1
\$r3_tdddelete	213	1
\$r3_lifetime	214	1
\$r3_schedule	215	3
\$r3_senddate	218	8
\$r3_sendtime	226	6
\$r3_teleland	232	3
\$r3_telenium	235	30
\$r3_tdttitle	265	50
\$r3_tdttest	315	1
\$r3_tdprogram	316	40
\$r3_tdsrnpos	356	15
\$r3_tdcover	371	1
\$r3_tdcovtitle	372	68
\$r3_tdreceiver	440	12
\$r3_tddivision	452	12
\$r3_tdauthority	464	12
\$r3_tdarmod	476	1
\$r3_tdiexit	477	1
\$r3_tdgetotf	478	1
\$r3_tdfaxuser	479	12
\$r3_ldest	491	30

Control records

Control records contain (in addition to other information) the language and code page to be used. They also contain the page name to differentiate pages within a document.

This information is shown in the following control variables:

- \$r3_language
- \$r3_codepage
- \$r3_pagename

Data records

Data records are used to produce unique field identifiers, by combining window, text element and field information. This key together with the data value (position 176 and onwards) is sent to StreamServe to generate the Message.

SAP 4.6/4.7 RDI functions

Supported RDI functions in SAP 4.6/4.7

- [Archive records](#) on page 78
- [Include-texts elements](#) on page 80
- [Include-Texts additional element](#) on page 81
- [RDI-Control command](#) on page 81
- [Specifying input analyzer parameters for RDI data](#) on page 81

Archive records

Archive information is only written if archiving has been selected in the SAP system. If selected, the archiving information is shown in the TDARMOD header record field, where:

- 1 = print only
- 2 = archive
- 3 = archive and print

Archive record types

Archive records can contain two types of records.

- **Index records ('I' records)**
Contain information specific to the document to be archived, that is the attributes needed to retrieve a given document.
See [Archive index - 'I' records](#) on page 78.
- **Parameter records ('P' records)**
Contain information about the archiving system and the link to the archiving system.
See [Archive parameters - 'P' records](#) on page 79.

Archive index - 'I' records

StreamServer handles archive index records ('I' records) as special control records. The following variables are available for index records. (The corresponding SAP structure is STXRDII.)

Variable Name	Position	Length (chars)
\$r3i_function	2	4
\$r3i_mandant	6	3
\$r3i_del_date	9	8

Variable Name	Position	Length (chars)
\$r3i_sap_object	17	10
\$r3i_ar_object	27	10
\$r3i_object_id	37	50
\$r3i_form_id	87	40
\$r3i_formarchiv	127	2
\$r3i_reserve	129	27
\$r3i_notiz	156	256

Archive parameters - 'P' records

StreamServer handles archive parameter records ('P' records) as special control records. The following variables are available for parameter records. (The corresponding SAP structure is STXRDIIP.)

Variable Name	Position	Length (chars)
\$r3p_sap_object	2	10
\$r3p_ar_object	12	10
\$r3p_archiv_id	22	2
\$r3p_doc_type	24	20
\$r3p_rpc_host	44	32
\$r3p_rpc_service	76	32
\$r3p_interface	108	14
\$r3p_mandant	122	3
\$r3p_report	125	40
\$r3p_info	165	3
\$r3p_arctext	168	40
\$r3p_datum	208	8
\$r3p_arcuser	216	12
\$r3p_printer	228	4
\$r3p_formular	232	16
\$r3p_archivpath	248	70
\$r3p_protokoll	318	8

Variable Name	Position	Length (chars)
\$r3p_version	326	4
\$r3p_acheck	330	10

Include-texts elements

In SAP 4.6/4.7, include-texts elements are identified in the `incparam` file with an `INC-BEGIN` and `INC-END` tag starting and ending the text element.

Note: You can configure an alias table to use for include handling.

To import the `incparam` file to the resource set

- 1 In the resource set, right-click the top node and select **Import**. The Select file to import dialog opens.
- 2 Browse to the `incparam` text file in the `<Streamserve installation>\Applications\StreamServer\<version>\Tools\System` folder.

Identification of include texts

StreamServer identifies include texts found in RDI data by combining the names of the window, the text element and text ID. The field ID can contain alphanumeric characters up to 32 characters.



You can set the log level to its maximum value which will write errors to the log when fields are not found. This can help to understand the required syntax.

Example 8

Printed Purchase Order

The following example shows a printed purchase order using RDI mode:

DMAIN	XITEM_TEXT	001			
CINC-BEGIN		450000482300010	EKPO	F03	DE
DMAIN	ITEM_TEXT	029	This is the Material PO text		
CINC-END		450000482300010	EKPO	F03	DE
DMAIN	XITEM_TEXT	001			
CINC-BEGIN		450000482300010	EKPO	F02	DE
DMAIN	ITEM_TEXT	035	Here comes the Info record PO text		
CINC-END		450000482300010	EKPO	F02	DE

In this example, the unique identifier for the first and second purchase order standard item text are:

MAIN_ITEM_TEXT_EKPO_F03


```
MAIN_ITEM_TEXT_EKPO_F02
```

StreamServer handles all data that occurs between `INC-BEGIN` and `INC-END` as one field. If the wrap function is used in `PageOUT`, this output could vary.

Note: Because this field is not imported when using the SAP Form Reader, you need to add this type of field manually to your StreamIN Event.

Include-Texts additional element

If a function module named `WRITE_FORM_LINES` was used when writing texts in the SAP system, these lines are enclosed in the following RDI records:

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

When a field does not have a field name associated with it, StreamServe identifies this field by combining the window name and the text object to create a unique field identifier. The field identifier can contain up to 32 alphanumeric characters.

RDI-Control command

In SAP, you use the SAPscript Form command `RDI_CONTROL` to add specific information to the RDI data. If you use this command, StreamServe creates a variable named `$r3_rdi_control` containing the value passed in the `RDI_CONTROL` line from the SAP system.

For example:

```
CRDI_CONTROL 123
```

where the added information is the number 123.

Keeping spaces

By default, space characters at end of line are preserved in included fields.

You can switch this off by setting the `INCPARAM_KEEP_SPACES` to any value except for the plus sign.

Specifying input analyzer parameters for RDI data

When configuring runtime for input connectors in StreamServe, you can pass input analyzer parameters to the in E-docs agent to alter how StreamServer will treat the RDI data. The parameters are sent to the agent when StreamServer starts.

Note: The input analyzer parameter is only available when using Design Center. See the *Design Center* documentation.

For RDI data, you can use three input analyzer parameters to alter how StreamServer generates RDI data.

Note: When using input analyzer parameters, you must specify a value for *each* of the three parameters.

Input analyzer parameters

Enter the three parameters in the following order, without spaces:

ignore_level,ignore_blankfieldvalue,INCLUDE-BEGIN_names

Input analyzer parameter	Possible values	Description
<i>ignore_level</i>	TRUE FALSE	Ignores levels in RDI data.
<i>ignore_blankfieldvalue</i>	TRUE FALSE	Ignores blank field values in RDI data.
<i>INCLUDE-BEGIN_names</i>	<i>n</i>	Where <i>n</i> is a number indicating what part of the INCLUDE-BEGIN field name is to be generated from the RDI data, see Generating INCLUDE-BEGIN field names on page 83.

Example 9 Specifying input analyzer parameters for RDI data

In this example, the E-docs agent will ignore levels in the RDI data, will process blank field values in the RDI data, and will generate INCLUDE-BEGIN field names which include the second value only as the unique identifier.

TRUE, FALSE, 2

To specify values for input analyzer parameters

- 1 In your Runtime configuration, make sure you have defined the link between the input connectors and the StreamIN Event for RDI data.
- 2 Right-click the input connector, and select **Settings**. The Runtime Input Connector Settings dialog box opens.
- 3 Select the E-docs agent shown, and click the Edit button. The Edit Input Analyzer dialog box opens.

Note: If the E-docs agent is not shown, make sure you have defined the agent for the Message, see [Creating a Message for RDI data](#) on page 60, and linked the input connector with the Event in the Runtime configuration.

- 4 Enter the input analyzer parameters as described above.
- 5 Click **OK**.
- 6 Save the Runtime configuration.

Generating INCLUDE-BEGIN field names

When using the input analyzer parameter for RDI data, you use the third parameter to specify how the agent will generate INCLUDE-BEGIN field names. Valid values are as follows:

Value	Description	Example
0	Default	MAIN_ITEM_TEXT_EKPO_F03
-	No first and second value	MAIN_ITEM_TEXT
1	First value only	MAIN_ITEM_TEXT_EKPO
2	Second value only	MAIN_ITEM_TEXT_F03

Example 10 Printed Purchase Order

The following example shows a printed purchase order using RDI mode:

DMAIN	XITEM_TEXT	001		
CINC-BEGIN	450000482300010		EKPO F03	DE
DMAIN	ITEM_TEXT	029	This is the Material PO text	
CINC-END	450000482300010		EKPO F03	DE
DMAIN	XITEM_TEXT	001		
CINC-BEGIN	450000482300010		EKPO F02	DE
DMAIN	ITEM_TEXT	035	Here comes the Info record PO text	
CINC-END	450000482300010		EKPO F02	DE

In this example, the unique identifier for the first and second purchase order standard item text are:

```
MAIN_ITEM_TEXT_EKPO_F03
MAIN_ITEM_TEXT_EKPO_F02
```


StreamServe and the XSF format

The E-docs Connect solution enables StreamServe to receive data generated from Smart Forms in your SAP system. Data from Smart Forms is output in XSF (XML for Smart Forms) format.

StreamServe can only process 'well-formed' XML data, and as XSF data from SAP is not well-formed XML data, the E-docs Connect solution includes filters that you can use to convert XSF data from your SAP system to well-formed XML data.

You specify the filter in the StreamServe Message for the XSF data.

XSF data elements

This section describes elements in XSF data.

The elements included in this section are relevant to the StreamServe – SAP XSF integration. For a complete description of XSF specifications, see the SAP Smart Forms XSF- XML Output for Smart Forms at:

<http://www.sap.com/partners/software/integration-opportunities/interface-certification/technology/bc-xsf.asp>

Root element

XSF data begins with the root element `smartxsf`, which contains control and data elements (`header` and `data` elements).

Elements	SAP Description	Use	Purpose
<code>smartxsf</code>	Root element	Yes	Creates job

Control elements

The `<header>` element contains control information for the form, which includes general information about the form, archive information, and distribution attributes, such as mail and telefax. This information is represented as child elements to the `<header>` element. The child element `<general>` consists of useful information for StreamServe.

StreamServe can use this header information to determine the distribution method and to trigger StreamServe language sets to translate the output to a different language.

Elements	SAP Description	Use	Purpose
<code>header</code>	Contains control information that is not supposed to be printed.	No	Container
<code>general</code>	Comprises information about XSF version, form name and language, etc.	No	Container
<code>version</code>	XSF Version	Yes	Determines transformation method in the future
<code>device</code>	Output device, for example PRINTER, MAIL, FAX.	Yes	Creates field

Elements	SAP Description	Use	Purpose
form	Form name	Yes	Used to set event name
language	Form language	Yes	Creates field
device-type	Device type, for example, HPLJ5.	Yes	Creates field
output-device	Printer name	Yes	Creates field
tdtitle	Title in dialog box	Yes	Creates field
tdnoprint	No printing from print preview	Yes	Creates field
tdnoprev	No print preview	Yes	Creates field
tdnoarch	No changes by user in archiving mode	Yes	Creates field
tdiexit	Immediately exit after printing/faxing from print preview	Yes	Creates field
tddataset	Spool request: Name	Yes	Creates field
tdsuffix1	Spool request: Suffix 1	Yes	Creates field
tdsuffix2	Spool request: Suffix 2	Yes	Creates field
tdcovtitle	Print parameters: text for cover page	Yes	Creates field
tdnewid	Print parameters: new spool request	Yes	Creates field
tdimmed	Print parameters: print immediately	Yes	Creates field
tddelete	Print parameters: delete after printing	Yes	Creates field
tdlifetime	Print parameters: spool retention period	Yes	Creates field
tdauthority	Print: authorization	Yes	Creates field
tdcover	Print: SAP cover page	Yes	Creates field
tdreceiver	Print parameters: recipient	Yes	Creates field
tddivision	Print parameters: department on cover page	Yes	Creates field
tdcopies	Number of identical copies	Yes	Creates field
tdpageselect	Pages selected for printing	Yes	Creates field

Elements	SAP Description	Use	Purpose
page	Identifies pages in the XSF data.	No	

Data elements

The data element contains all data from the application program.

Elements	SAP Description	Use	Purpose
data	Contains data from the application program. The data is structured at runtime by embedding the data into specific Smart Forms elements.	No	
window	Describes a logical unit of structured information. Each window refers to a certain page via the page and page-id attributes.	Yes	Window name is prefixed to fields
graphics	Can occur both as a separate window and as part of a regular window (main/auxiliary). If this element represents a separate window, the attributes page and page-id refer to the appropriate page.	No	
address	Can occur both as a separate window and as part of a regular window (main/auxiliary). If this element represents a separate window the attributes page and page-id refer to the appropriate page.	No	Treated as text
section	Used to group elements <thead>, <tbody> and <tfoot>.	No	
thead	Belongs to <section>. Used for grouping	No	
tbody	Belongs to <section>. Used for grouping	No	
tfoot	Belongs to <section>. Used for grouping	No	

Elements	SAP Description	Use	Purpose
new-page	Used for explicit page breaks within a Smart Form.	No	
print-control	Contains printer escape sequences.	No	
include-text	Contains attributes, including text node name (attribute name), current style (attribute style) and style variant (attribute variant).	Yes	Name attribute used
text	Contains attributes, including text node name (attribute name), current style (attribute style) and style variant (attribute variant).	Yes	Name attribute used
p	Describes a logical text paragraph.	Yes	Creates field
url	Used for Universal Resource Locator (URL) specification.	Yes	Creates field
chr	Describes a logical unit of text within a paragraph format (element <p>).	Yes	Creates field
sym	Describes the data fields of Smart Forms.	Yes	Creates field
new-line	Converted to a real line break character (Hex <0D><0A>) by XSF filter.	Yes	Needs to be treated as a real line-break
tab	Converted to a real tab character (Hex <09>) by XSF filter.	Yes	
table	Defines a table structure using elements <thead>, <tbody> and <tfoot>.	No	
thead	Defines the table header.	No	
tbody	Defines the table body.	No	
tfoot	Defines the table footer.	No	
tr	Defines a table row.	Yes	Sets virtual block
tc	Defines a table cell.	No	
sort	Unknown	No	

Elements	SAP Description	Use	Purpose
sort-field	Unknown	No	
archive	Contains information (as attributes to the element) on archiving.	Yes	
mail	Mail information is only written if the mail device is used. Contains mail sender and recipient information.	Yes	
sender	Specifies the mail sender.	Yes	
recipient	Specifies the mail recipient.	Yes	
telefax	The telefax information is only written if the telefax device is used. It provides, amongst others, the fax number and country.	Yes	

Useful SAP transaction codes

This section lists SAP transaction codes which are commonly used in the SAP system to activate transactions.

Note: To enter a transaction code from any screen within the SAP system other than the initial screen, prefix the code with /n. For example, the /nVF03 transaction code would display the Display Billing Document screen from any screen in the SAP system.

Configuration

SPRO	Customizing
OMFE	Processing Program/Layout Set for Purchase Order (MM)
V/30	Processing Program/Layout Set for Order Confirmation (SD)
V/34	Processing Program/Layout Set for Delivery Note (SD)
V/40	Processing Program/Layout Set for Invoice (SD)

Spool functions

SPAD	Spool Administration
SP01	Spool Requests
SP02	Own Spool Requests

Form processing

SE71	SAPscript
SFP	PDF based print forms
SE73	Font Maintenance
SE78	Graphics Management
SMARTFORMS	Smart Forms
SO10	Standard Texts

Programs and reports

SE38	ABAP Editor
RSTXSCR P	Import/Export SAPscript objects
RSTXSYMB	List SAP symbols
RSTXICON	List SAP icons
RSPO0049	Activate Access Method Z (Spool Exit)

Generating application output

ME22	Change Purchase Order (MM)
ME90	Print Purchase Order (MM)
VA02	Change Sales Order (Order Confirmation, SD)
VA03	Display Sales Order (Order Confirmation, SD)
VF02	Change Billing Document (Invoice, SD)
VF03	Display Billing Document (Invoice, SD)
VL02	Change Outbound Delivery (Delivery Note, SD)
VL03	Display Outbound Delivery (Delivery Note, SD)
SM69	List of external commands (for box drawing characters)
SM04	List of users currently logged on (short list)

Data and metadata

SE11	Data Dictionary
SE16	Data Browser
WE63	IDoc Types