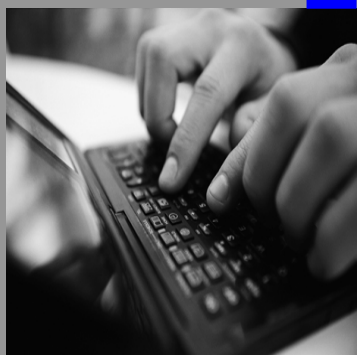


How-to Guide
SAP NetWeaver 2004s

How To... Sample IDoc-XI Scenarios

Version 1.00 – Sept 2006

Applicable Releases:
SAP NetWeaver 2004s
End-to-End Process Integration
Enabling Application-to-Application Processes

THE BEST-RUN BUSINESSES RUN SAP

© Copyright 2006 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, and Informix are trademarks or registered trademarks of IBM Corporation in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

These materials are provided “as is” without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. SAP shall not be liable for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials.
SAP does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third party web pages nor provide any warranty whatsoever relating to third party web pages.
SAP NetWeaver “How-to” Guides are intended to simplify the product implementation. While specific product features and procedures typically are explained in a practical business context, it is not implied that those features and procedures are the only approach in solving a specific business problem using SAP NetWeaver. Should you wish to receive additional information, clarification or support, please refer to SAP Consulting.
Any software coding and/or code lines / strings (“Code”) included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.

1 Scenario

This guide gives an overview of how to configure different IDoc-XI-IDoc scenarios using alternative identifiers and header mapping. However, the contents are also relevant to other interfaces, for instance, scenarios involving systems that use the RNIF adapter. The table below shows the scenarios that will be addressed, and the different approaches that will be applied.

Case	Outbound Party Type Sender / Recv.	Inbound Party Type Sender / Recv.	Alternative Identifiers	Header Mapping	Identifiers in Communication Channel	Chapter
1	LS / LS	LS / LS	—	—	—	3
2	≠LS / ≠LS	≠LS / ≠LS	✓	—	✓	4
3*	≠LS / ≠LS	≠LS / ≠LS				5
4	≠LS / ≠LS	LS / LS	✓	✓	—	6
5	≠LS / ≠LS	≠LS / LS	✓	✓	✓	7
6	LS / LS	≠LS / ≠LS	✓	✓	✓	8
7	LS / ≠LS	≠LS / ≠LS	✓	✓	✓	9

* Master data consolidated

2 Introduction

An IDoc that is sent to the inbound channel of the Integration Server is converted from native IDoc format to IDoc XML format by the IDoc adapter. The IDoc adapter calls the Integration Engine pipeline for routing and mapping purposes. The XML message that is sent to the Integration Server consists of a header and a body with the IDoc XML as the payload.

The sender service within the message header is identified by the adapter-specific identifiers. For business systems that are maintained in the System Landscape Directory (SLD), the relevant properties are replicated into the Integration Directory. For business services, you can enter the adapter-specific identifiers directly in the Integration Directory. For SAP systems, the sender service is determined by the system ID of the sender port and by the client. For non-SAP systems, it is identified by the logical system name.

The sender and receiver party within the message header are specified depending on the IDoc partner type. For logical systems (partner type *LS*), the service remains party-less. For any other, an alternative party is generated as follows:

Party name	IDoc partner number
Identification scheme	ALE#<partner-type>#<partner-role>
Agency	Service (identified as described above)

In order to normalize the alternative party, that is, to replace it with an internal XI party, you have to specify alternative identifiers. The conversion can be done in both the inbound channel and the outbound channel of the Integration Server.

The IDoc adapter at the Integration Server outbound channel converts the IDoc XML to native IDoc format, and sends the IDoc to the receiver system. The IDoc partner is

identified by the sender and receiver service in the message header. For party-less services, the IDoc partner is a logical system. Otherwise, the IDoc partner is identified using the alternative party. In the case of a normalized party, you have to define alternative identifiers in the communication channel. Furthermore, you can modify the party and service by a header mapping. The header mapping is required within the receiver agreement, hence you only define it at the outbound channel of the Integration Server. There, it is executed before the internal XI party is converted to the alternative party.

The control record of the IDoc is discarded and renewed by the IDoc adapter. If you want to add values to the control record, you have to select the *Apply Control Record Values from Payload* checkbox in the communication channel. When a new communication channel of type IDoc is created, the flag is set by default. If the flag is not set, the settings comply with XI 2.0. For more information, see SAP Note 728792.

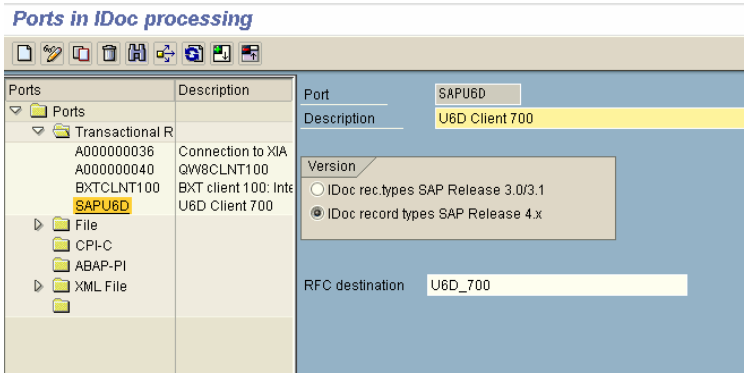
3 Case 1: IDoc of Type Logical System (LS)

	IDoc Party Name / Type	XI Party Agency / Scheme / Name / Service	IDoc Party Name / Type
Sender:	**B6MCLNT000 / LS**	**- / - / - / B6M_000**	**B6MCLNT000 / LS**
Receiver:	**XIDCLNT113 / LS**	**- / - / - / XID_113**	**XIDCLNT113 / LS**

3.1 Maintain Partner Profiles in Sender and Receiver Systems

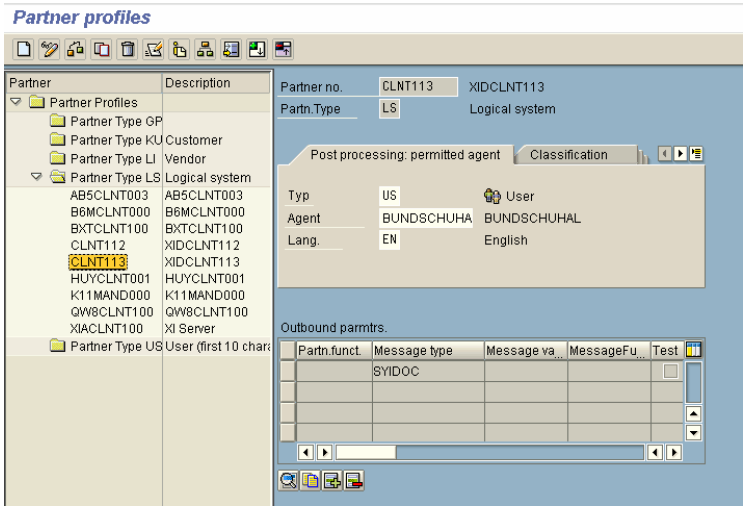
1. In the sender system, call transaction **WE21** to create the tRFC port `SAPU6D`.

   Maintain the RFC destination `U6D_700` to address the Integration Server.



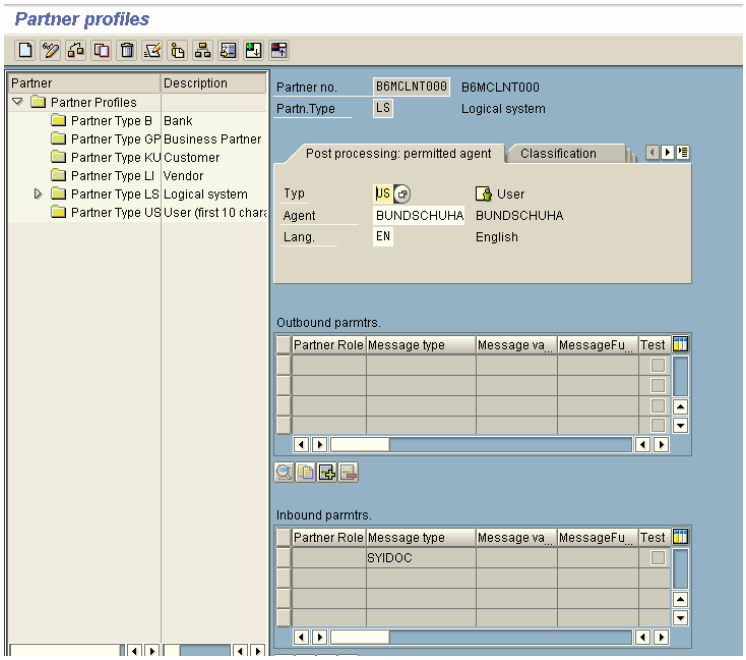
2. In the sender system, call transaction **WE20** to maintain the partner profile for the receiver system `CLNT113`.

   Select message type `SYIDOC`, the corresponding basic type, and receiver port `SAPU6D` as outbound parameters.



3. In the receiver system, call transaction **WE20** to maintain the partner profile for the sender system `B6MCLNT000`.

   Select message type `SYIDOC` as an inbound parameter.

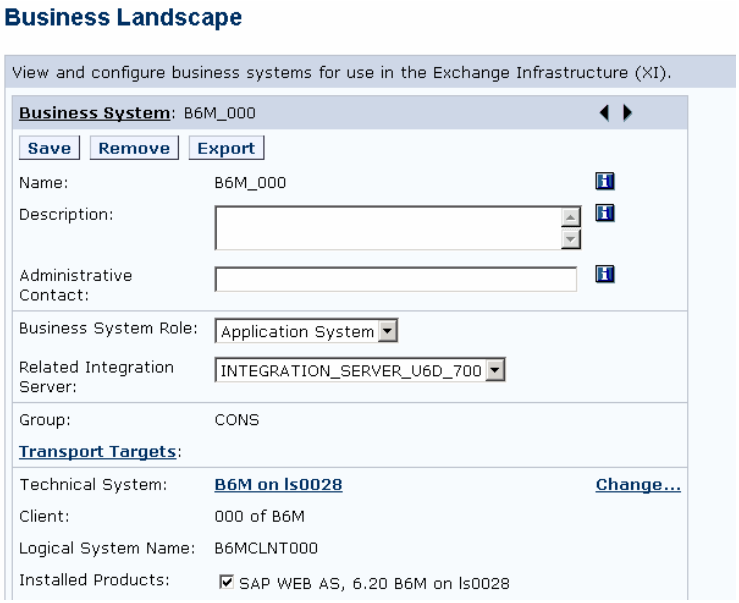


3.2 Maintain the System Landscape Directory (SLD)

The following steps are valid for all other business cases as well.

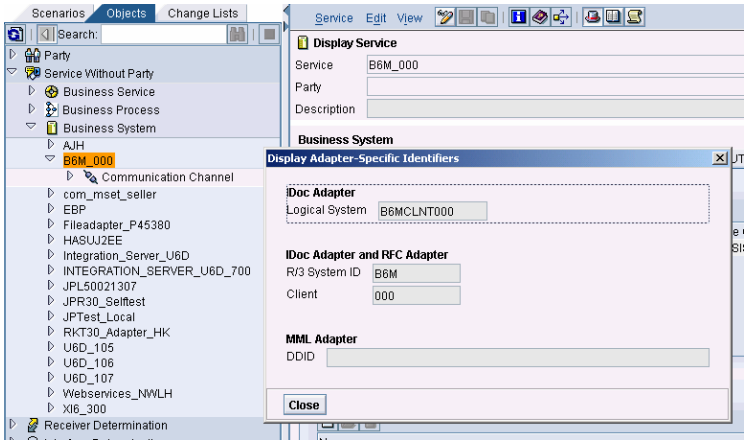
1. In the SLD, maintain the business system `B6M_000`.

   Select the *technical system*, the *system ID*, the *client*, and the *logical system name*.



2. In the Integration Directory, the adapter-specific identifiers of business system `B6M_000` are retrieved from the SLD.

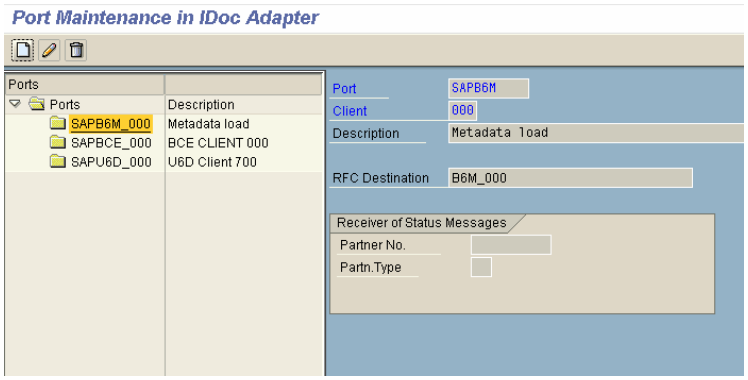
   The IDoc adapter uses *system ID*, and *client* to determine the corresponding service.



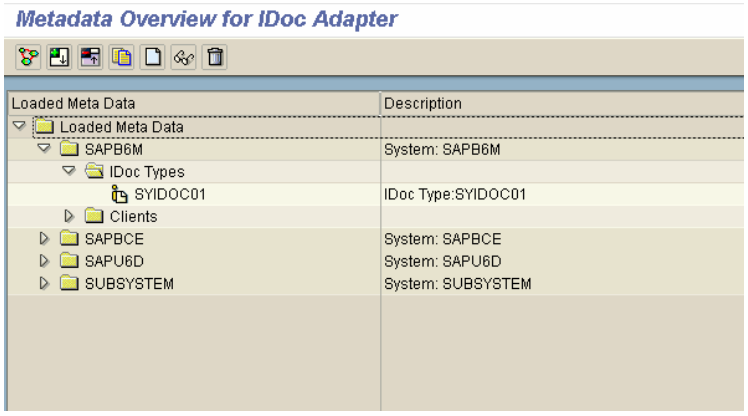
3.3 Maintain the Integration Server

The following steps are valid for all other business cases as well.

1. Call transaction **IDX1** to maintain port `SAPB6M_000` to load the IDoc metadata.



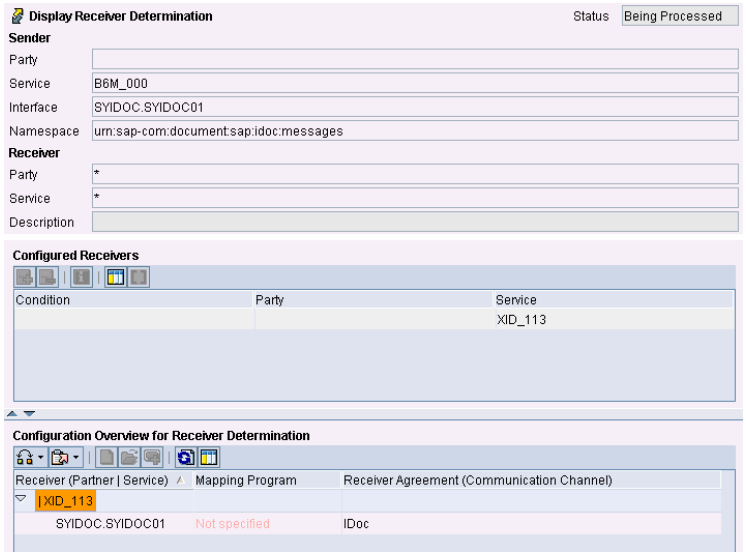
2. Call transaction **IDX2** to obtain an overview of the metadata that is already loaded to the Integration Server.



3.4 Integration Builder: Configuration

1. In the Integration Directory, create a receiver determination, an interface determination, and a receiver agreement.

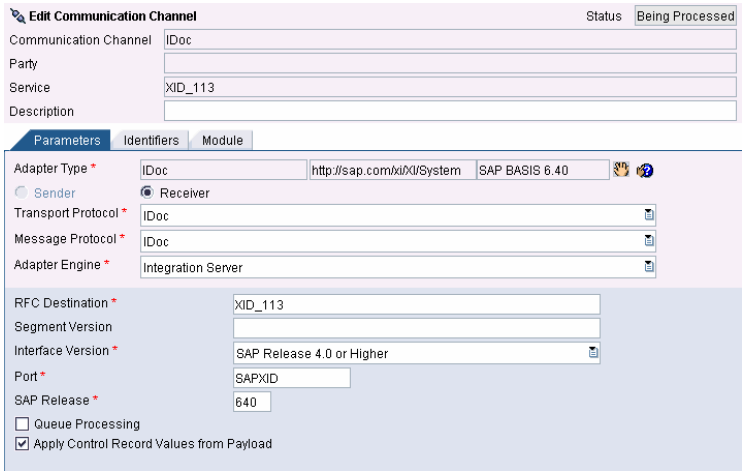
   Select the party-less sender service `B6M_000`, the sender interface `SYIDOC.SYIDOC01`, the party-less receiver service `XID_113`, the inbound interface `SYIDOC.SYIDOC01`, and the communication channel `IDoc`.



2. Create a communication channel of adapter type *IDoc*.

   Select adapter type *IDoc*, RFC destination `XID_113`, port `SAPXID`, and the appropriate SAP release.

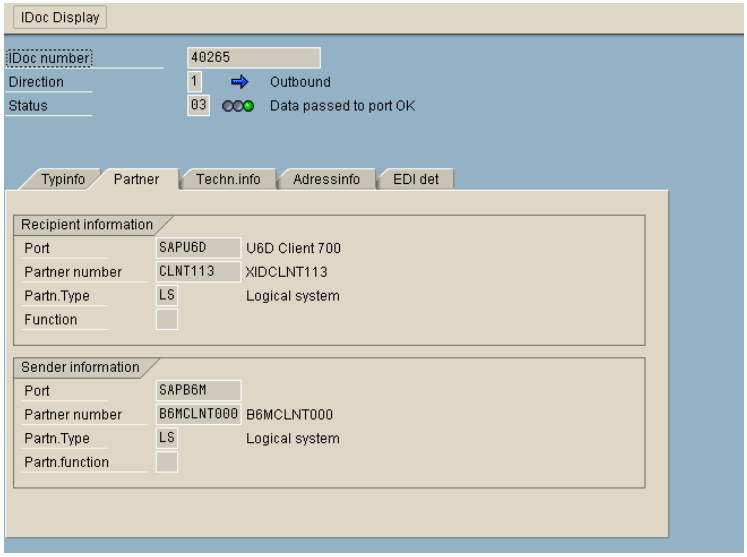
   The *Apply Control Record Values from Payload* checkbox is selected by default.



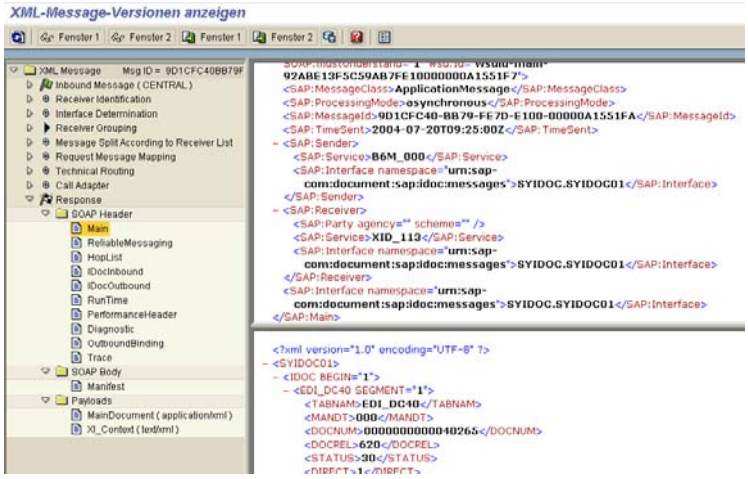
3.5 Execute the Scenario

1. Send an IDoc of partner type *LS* (*logical system*).

   In the sender system, call transaction **WE05** to display the outbound IDoc. The IDoc partner type of both receiver and sender is *LS*.

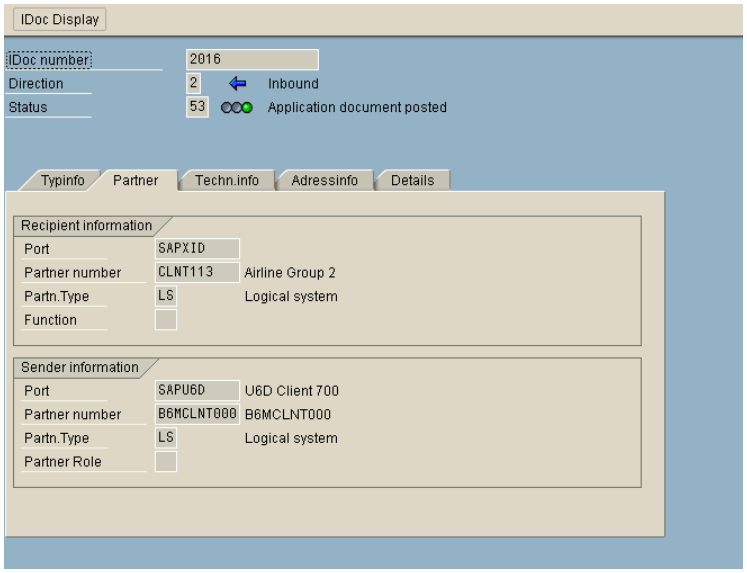


2. On the Integration Server, call transaction **SXMB_MONI** (*Integration Engine Monitoring → Monitor for Processed XML Messages*) to display the message.



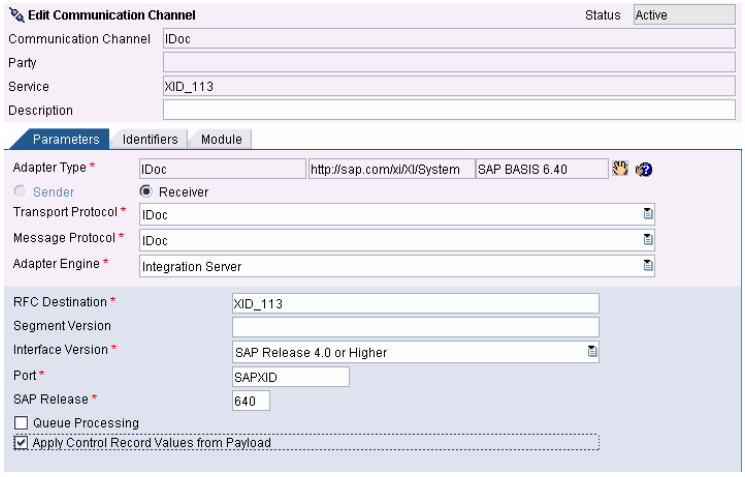
3. In the receiver system, call transaction **WE05** to display the inbound IDoc.

   IDoc partner name and type remain unchanged.

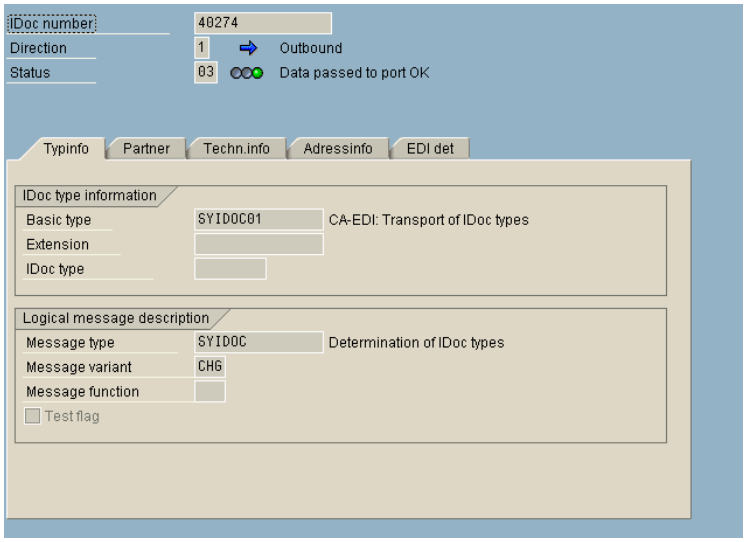


3.6 XI 2.0 Compliance

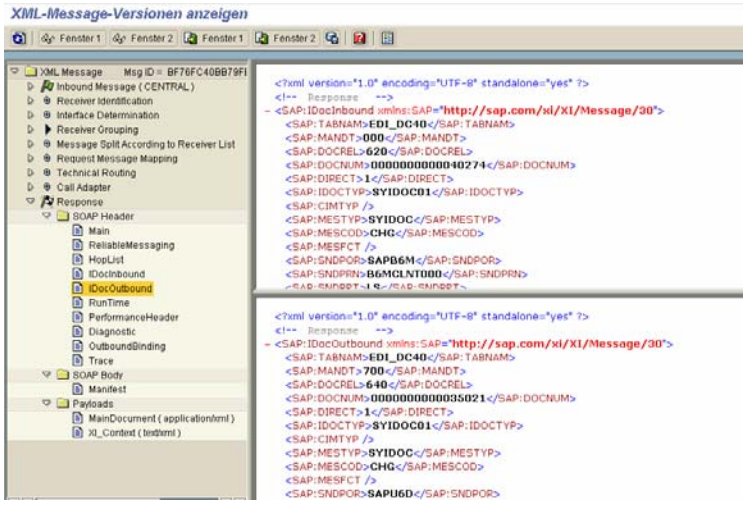
1. Option 1: The *Apply Control Record Values from Payload* checkbox is selected in the communication channel.



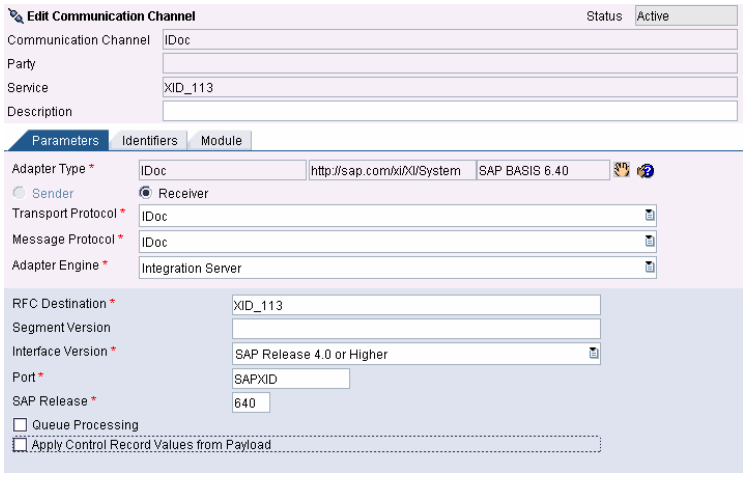
2. Send an IDoc of message type `SYIDOC` and message variant (technical name: *MESCOD*) `CHG`.



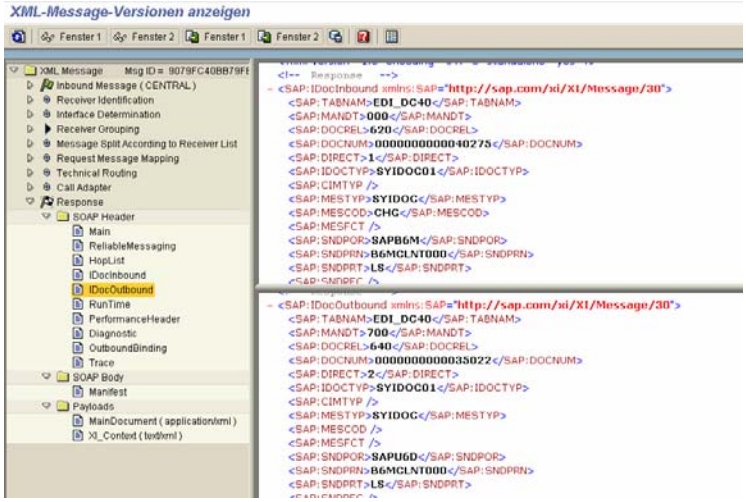
3. Comparing *IDoc Inbound* and *IDoc Outbound* header segments indicates that the *MESCOD* field is filled by the corresponding value of the IDoc XML.



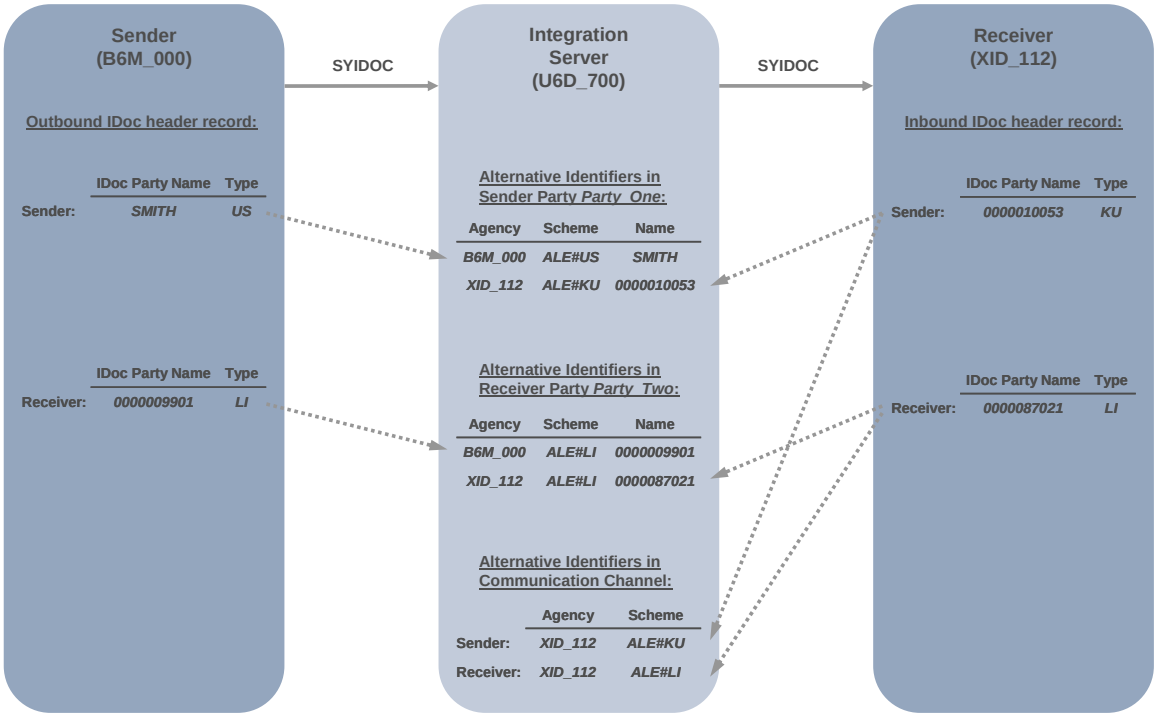
4. Option 2: The *Apply Control Record Values from Payload* checkbox is not selected (XI 2.0 compliant).



5. Comparing *IDoc Inbound* and *IDoc Outbound* header segments indicates that the *MESCOD* field is not set.



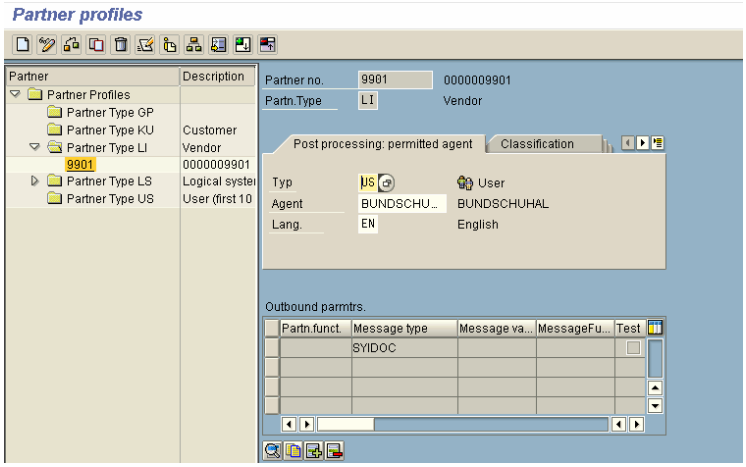
4 Case 2: IDoc of Type ≠LS



4.1 Maintain Partner Profiles in Sender and Receiver Systems

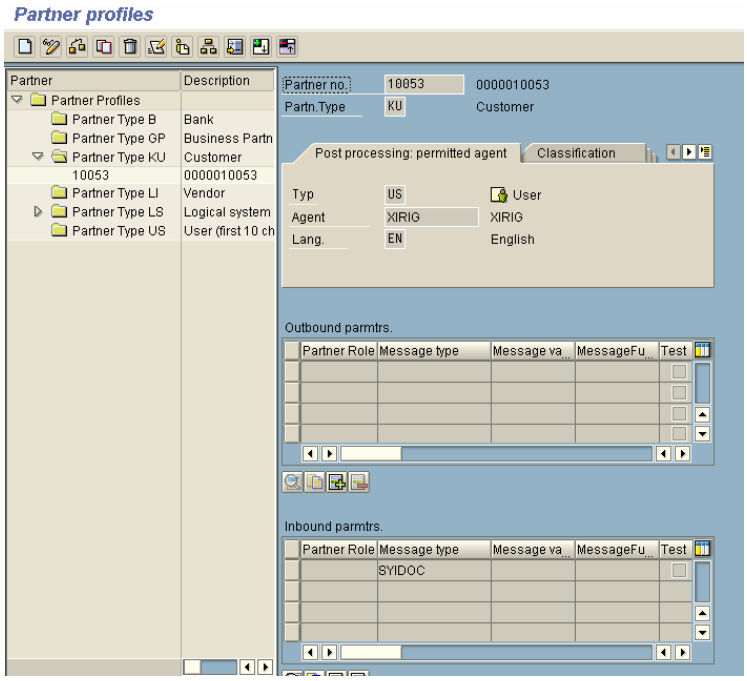
1. In the sender system, call transaction **WE20** to maintain the partner profile for partner `9901` of type *Vendor* (*LI*).

   Select message type `SYIDOC`, and receiver port `SAPU6D` as outbound parameters.



2. In the receiver system, call transaction **WE20** to maintain the partner profile for partner `10053` of type *Customer* (*KU*).

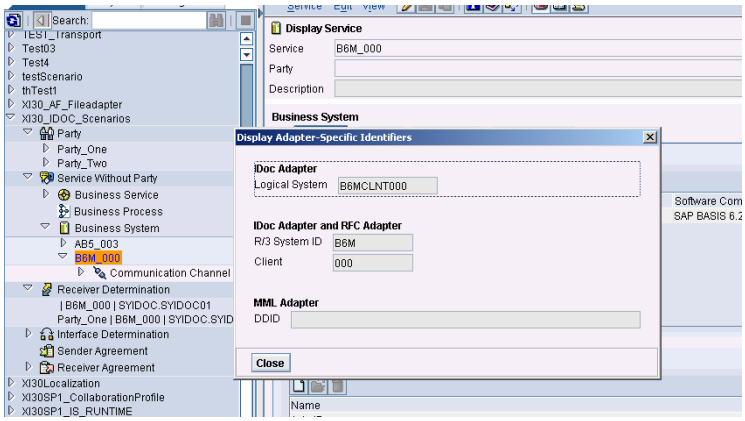
   Select message type `SYIDOC` as inbound parameter.



4.2 Integration Builder: Configuration

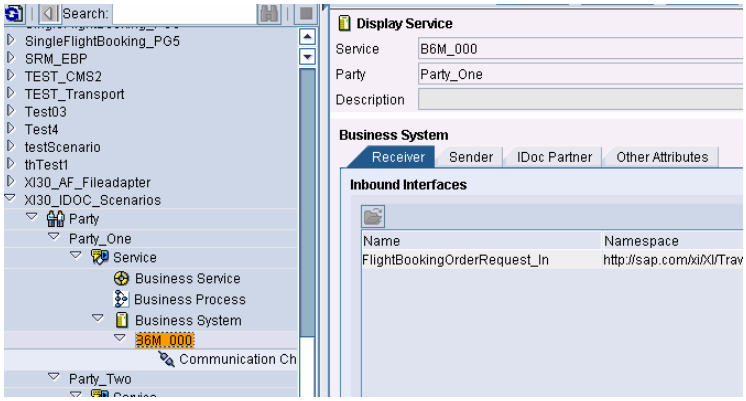
1. In the Integration Directory, assign business system `B6M_000` to the configuration scenario.

   The party-less service is required for the IDoc adapter to determine the service by using *system ID* and *client*.

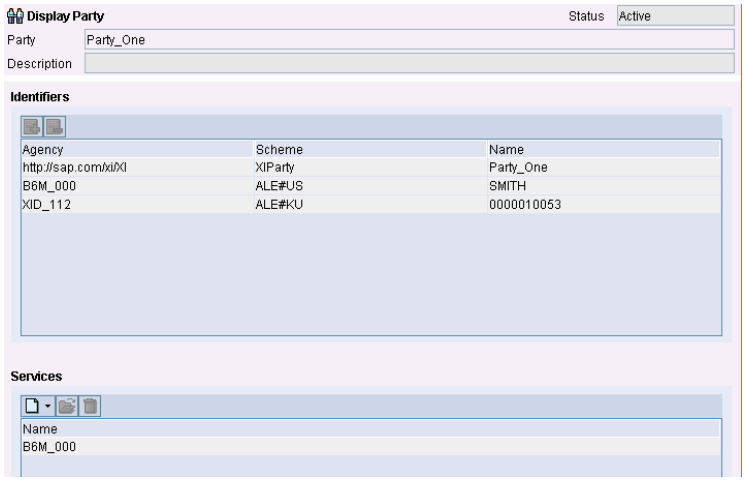


2. Create party `Party_One`, and assign business system `B6M_000` to it.

   The XI party `Party_One` acts as the sender party.

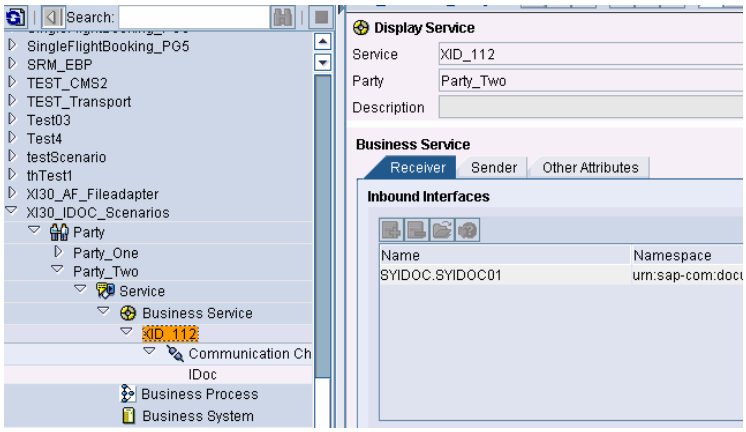


3. Maintain the alternative identifiers for `Party_One` according to the figure on the right.

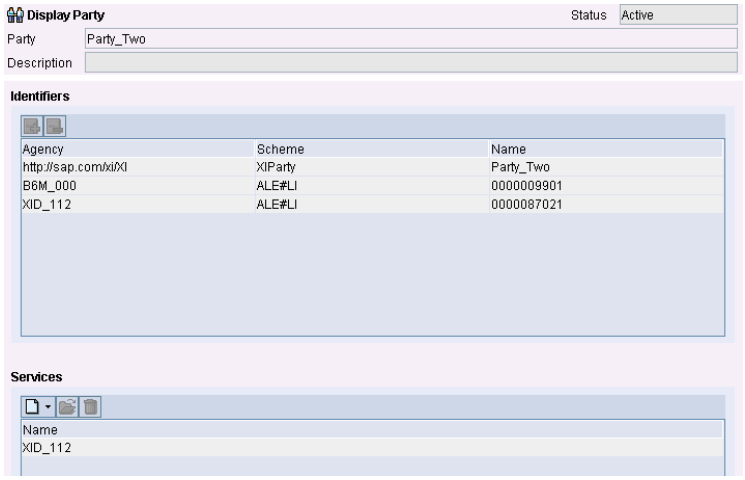


4. Create party `Party_Two`, and assign business system `XID_112` to it.

   The XI party `Party_Two` acts as the receiver party.

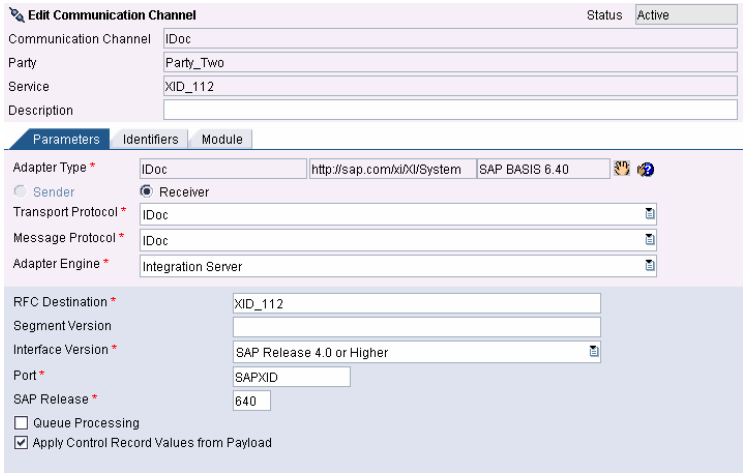


5. Maintain the alternative identifiers for `Party_Two` according to the figure on the right.



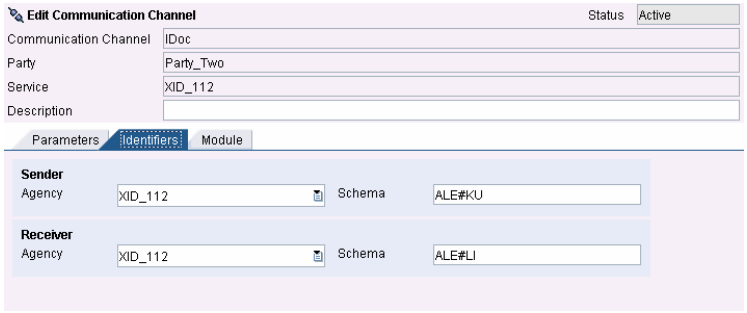
6. Create a communication channel of adapter type *IDoc*.

   Select adapter type *IDoc*, RFC destination `XID_112`, port `SAPXID`, and the appropriate SAP release.



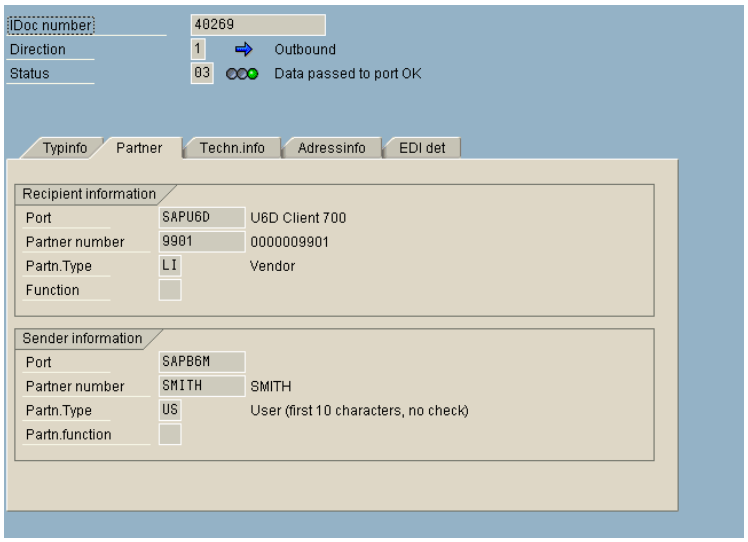
7. Switch to the *Identifiers* tab page, and maintain the alternative identifiers for the sender and the receiver according to the figure on the right.

   Party, *agency*, and *scheme* are used to identify the IDoc partner at the Integration Server outbound side.



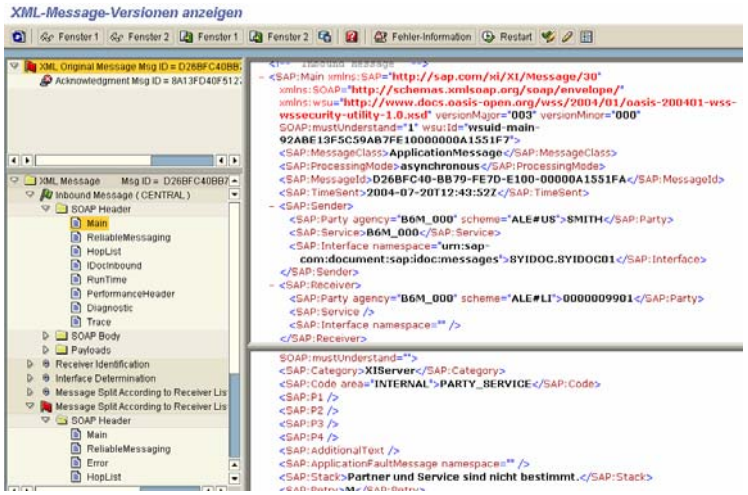
4.3 Execute the Scenario

1. Send an IDoc with sender partner `SMITH` of type *User* (*US*), and with receiver partner `9901` of type *Vendor* (*LI*).

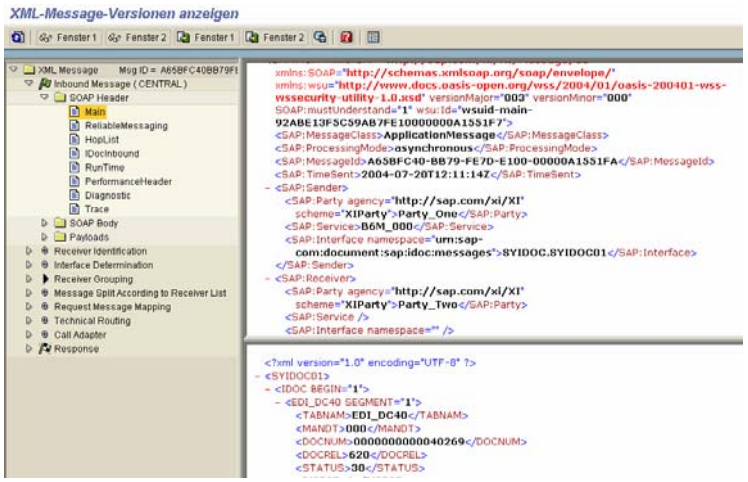


2. At the inbound channel of the Integration Server, an alternative party is generated in the message header consisting of *party name*, *identification scheme*, and *agency*.

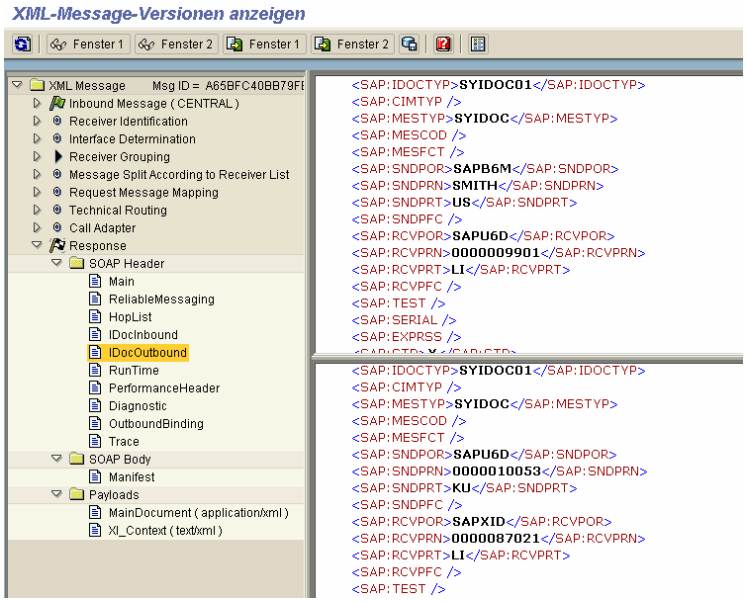
Note: If no alternative identifiers are maintained, the alternative party is displayed in the main header of the XML message as shown in the figure on the right.



3. If alternative identifiers are maintained, the alternative parties are normalized, that is, they are replaced by the XI parties `Party_One` and `Party_Two`, respectively.

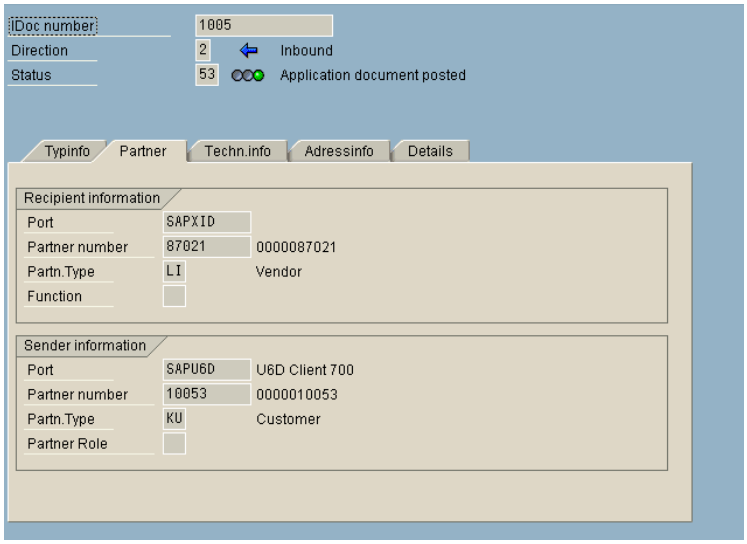


4. At the outbound channel of the Integration Server, the IDoc partner is identified using the alternative party.



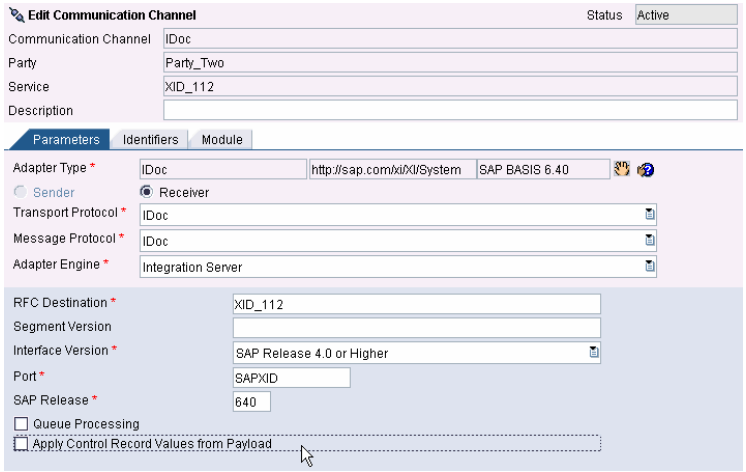
5. In the receiver system, call transaction **WE05** to display the inbound IDoc.

   The partner names and types are converted.

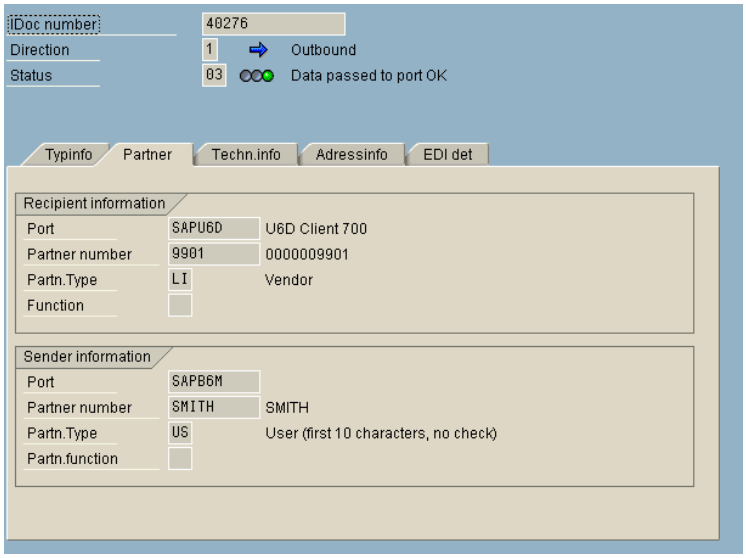


4.4 XI 2.0 Compliance

1. In the communication channel, the *Apply Control Record Values from Payload* checkbox is not selected (XI 2.0 compliant).

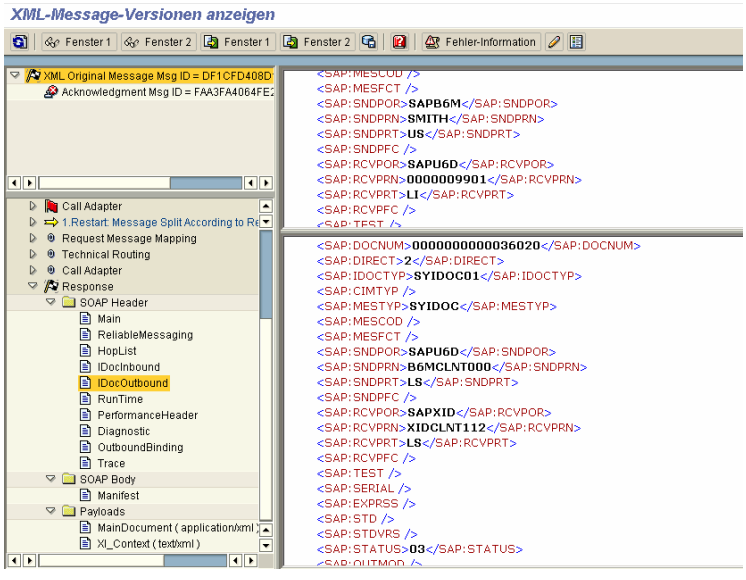


2. Send an IDoc with sender partner `SMITH` of type *User* (*US*) and receiver partner `9901` of type *Vendor* (*LI*).



3. The *IDoc Outbound* header section indicates that the fields are filled as follows:

Partner Number:	<logical system>
Sender Type:	LS (constant)
Partner Function:	blank
Direction:	2 (constant)
Status:	03 (constant)



5 Case 3: IDoc of Type ≠LS – Master Data Consolidated

This scenario is relevant for system landscapes where master data is already consolidated, that is, master data for business partners, material, products etc., is globally identical.

	IDoc Party Name / Type	XI Party Agency / Scheme / Name / Service	IDoc Party Name / Type
Sender:	MILLER / US	B6M_000 / ALE#US / MILLER / B6M_000	MILLER / US
Receiver:	0000029603 / LI	B6M_000 / ALE#LI / 0000029603 / XID_113	0000029603 / LI

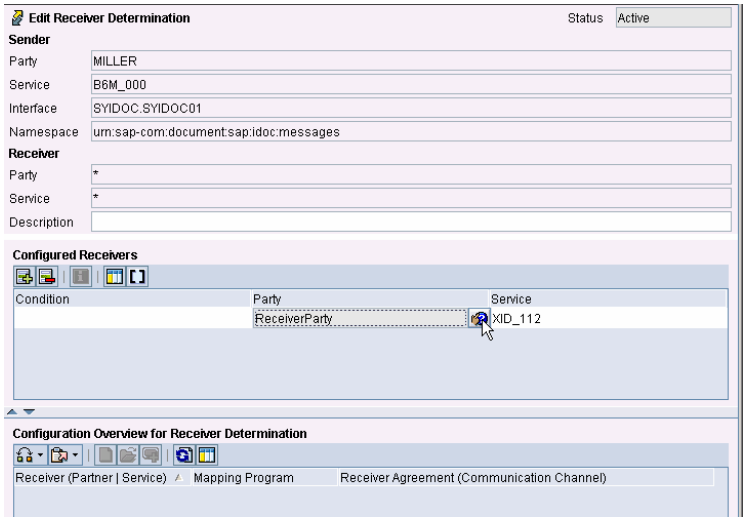
The alternative parties do not necessarily have to be normalized, because the partner names and types are identical in both systems involved, although for monitoring reasons it is recommended to do so. So, you do not have to maintain any alternative identifiers within XI. This can be a significant advantage when you have a very large amount of master data.
One approach for configuring this scenario would be to leave the alternative parties unchanged, without any need for conversion. This works for the sender party but causes problems for the receiver party: The party name is restricted to letters only, and if the name of the party contains a number, it is not possible to enter the party in the receiver agreement. Another problem occurs if two partners exist with identical partner numbers but different partner types. Party agency and party scheme are ignored during interface determination and receiver agreement. Therefore, the two parties cannot be distinguished from each other. The next section illustrates this approach up to the point where the problem occurs.
To solve this problem, use the following workaround: Enter a party-less receiver service in the receiver determination. The alternative party is reconstructed using header mapping (see section 5.2).

5.1 Approach: Configuring Using Context Objects Within the Receiver Determination

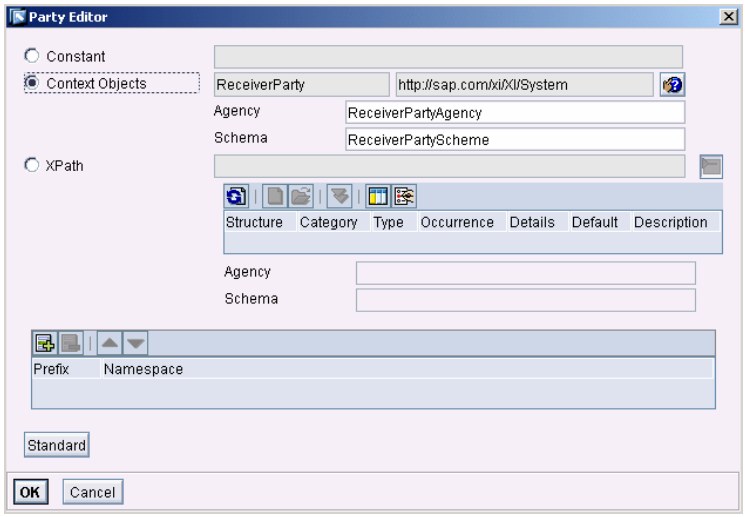
1. Create a receiver determination.

   Select sender party `MILLER`, and sender service `B6M_000`.

   Note: This is only possible because the sender party in the current example does not contain any numbers (see below).



2. For the receiver party, select context objects *ReceiverPartyAgency*, and *ReceiverPartyScheme*, respectively.



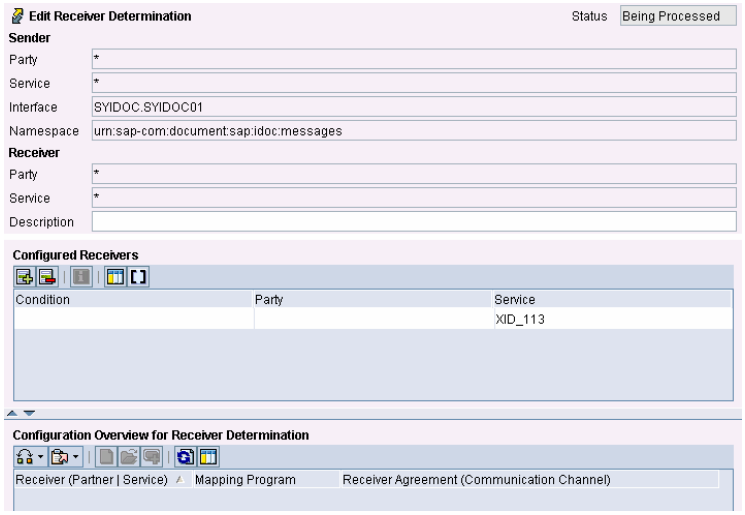
3. **Problem**: For the receiver agreement, you have to enter the receiver party. However, this is not possible if the receiver party contains a number, as is the case in the current example. As mentioned above, only letters are permitted.

5.2 Workaround: Configuring Using Header Mapping

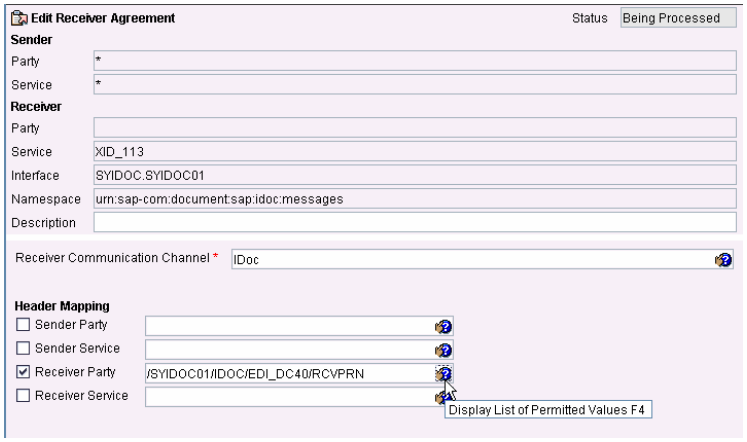
1. Create a receiver determination, and an interface determination.

   Enter asterisks for *sender party* and *sender service*, respectively.

   Select receiver service `XID_113`. This causes party, agency, and scheme to disappear from the message header.

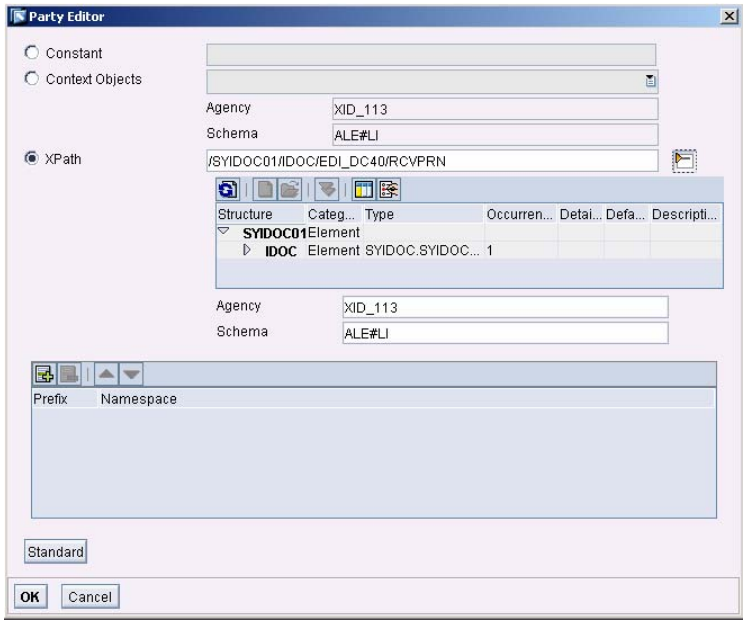


2. In the receiver agreement, a header mapping is required for the receiver party.

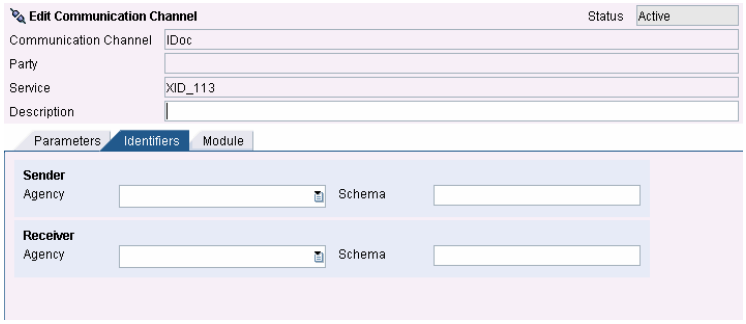


3. In the header mapping, you have to replace the receiver party with the IDoc receiver partner number that is provided in the payload of the XML message. Hence, select the appropriate XPath expression.

   For the current example, enter agency `XID_113`, and scheme `ALE#LI`. Alternatively, enter an XPath expression.

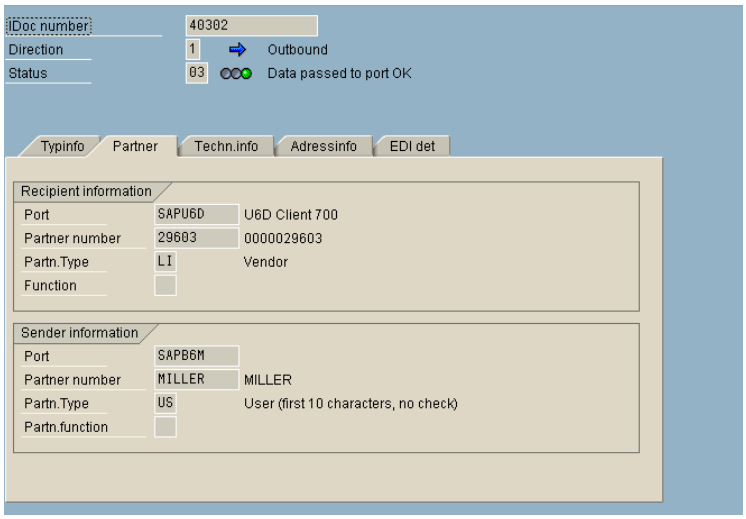


4. In the communication channel, no alternative identifiers are required.

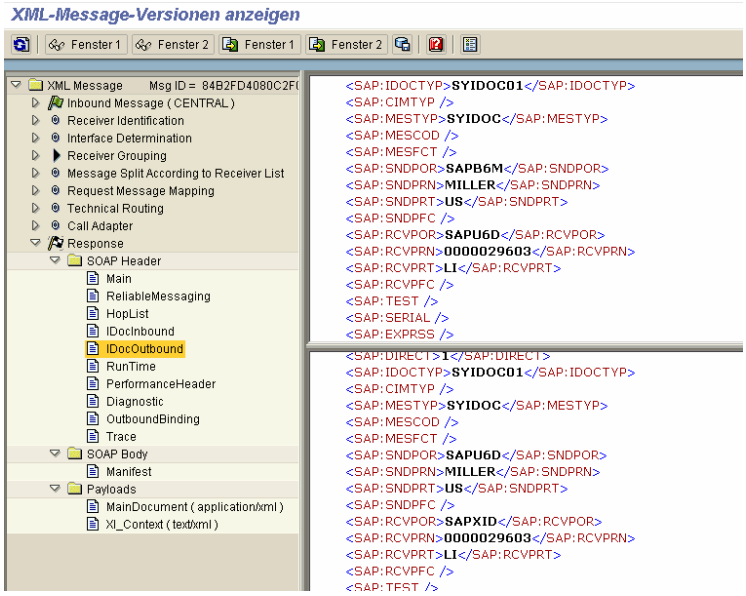


5.3 Execute the Scenario (Workaround)

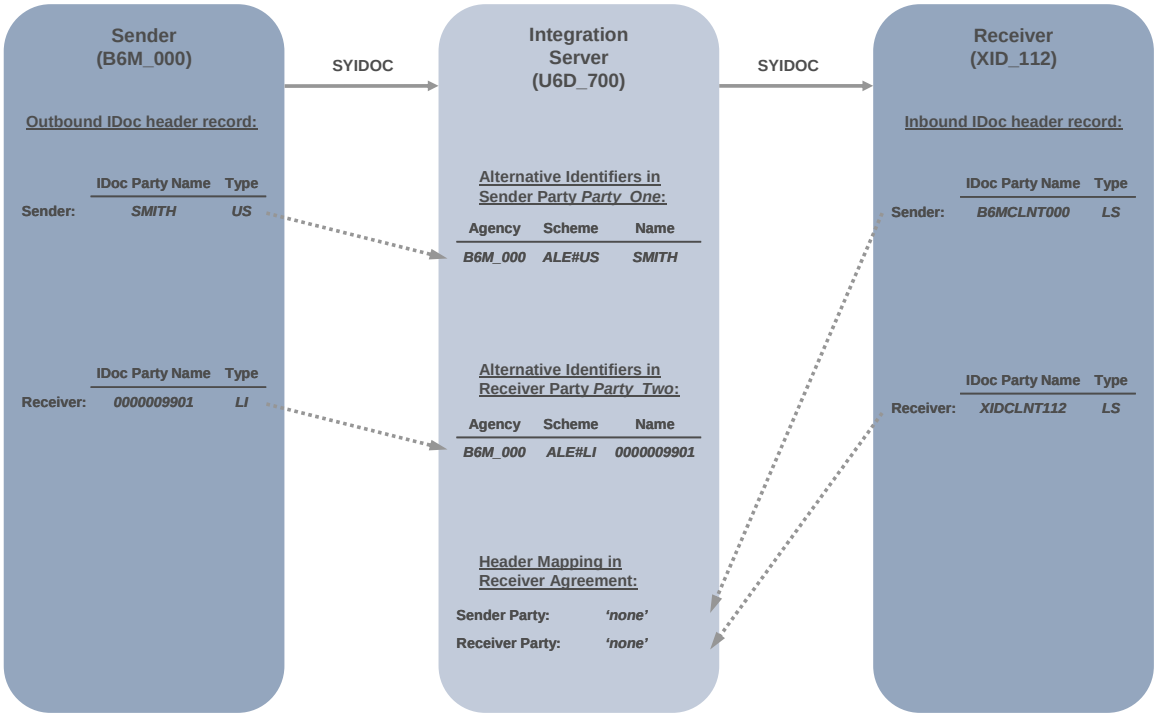
1. Send an IDoc with sender partner `MILLER` of type *User* (*US*), and receiver partner `29603` of type *Vendor* (*LI*).



2. As you can see in the message monitoring, IDoc partner type and number remain unchanged.

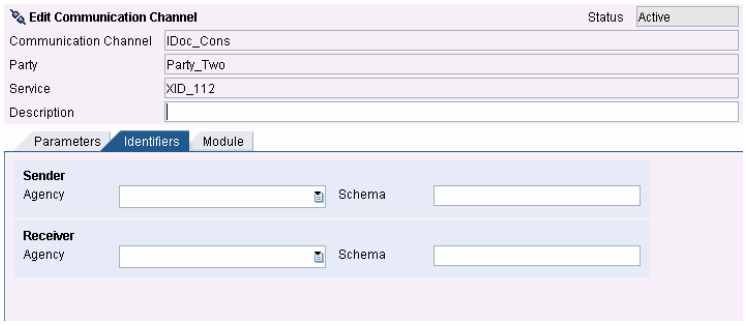


6 Case 4: Party (≠LS) → Partyless (LS)



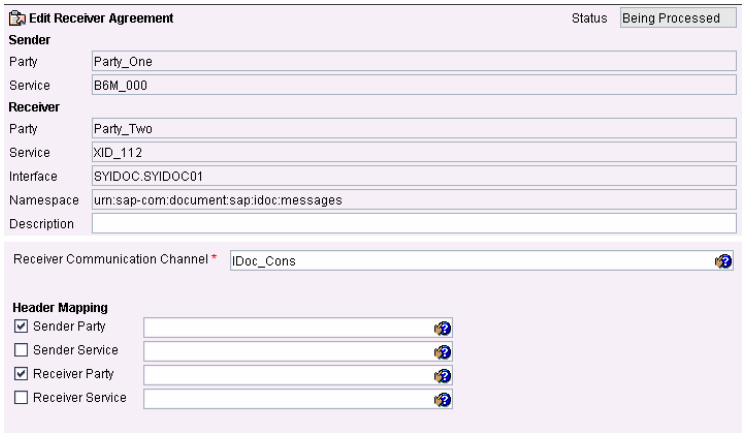
6.1 Integration Builder: Configuration

1. In the communication channel, no identifiers are required.



2. In the receiver agreement, header mappings for *sender party*, and *receiver party* are required.

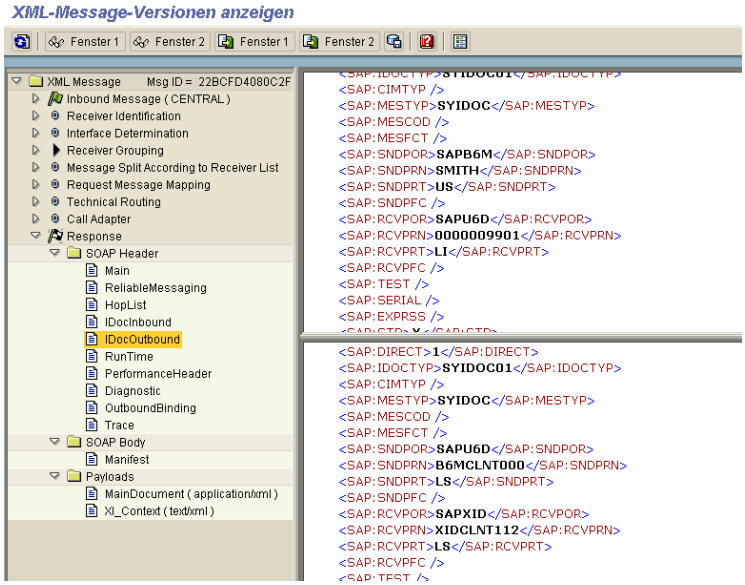
   Leave *sender party* and *receiver party* empty.



6.2 Execute the Scenario

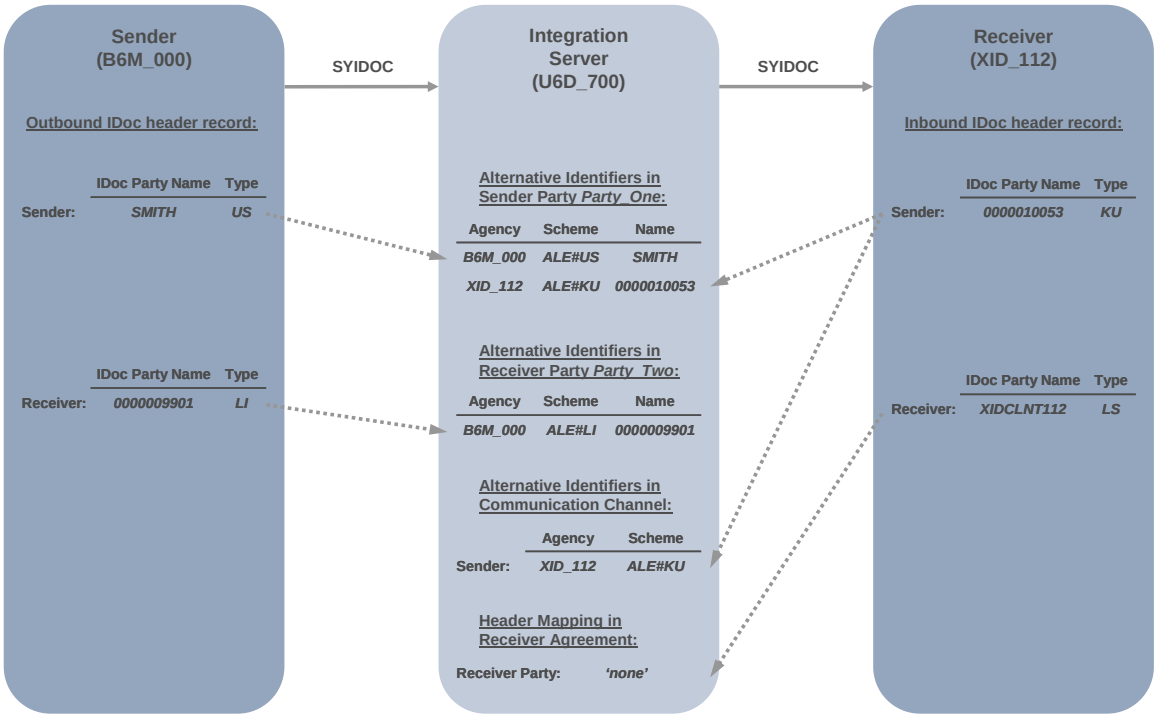
1. Send an IDoc with sender partner `SMITH` of type *User* (*US*), and receiver partner `9901` of type *Vendor* (*LI*).

   The IDoc Outbound header section indicates that the parties are replaced by the corresponding logical systems.



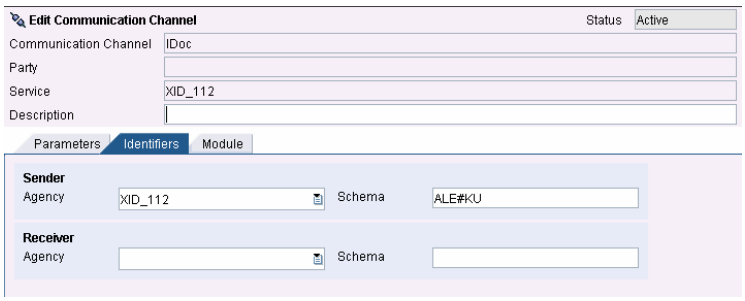
7 Case 5: Party (≠LS) → Partyless Receiver

This case is also relevant to RNIF, since RNIF requires parties (RNIF → IDoc).



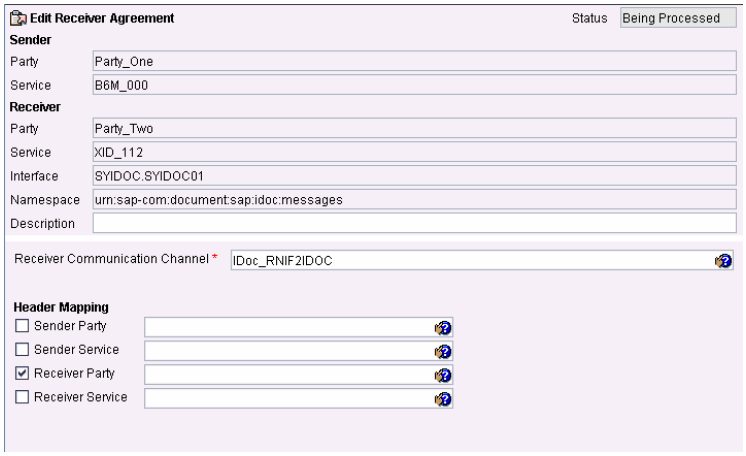
7.1 Integration Builder: Configuration

1. In the communication channel, only identifiers for the sender are required.



2. In the receiver agreement, a header mapping for the *receiver party* is required.

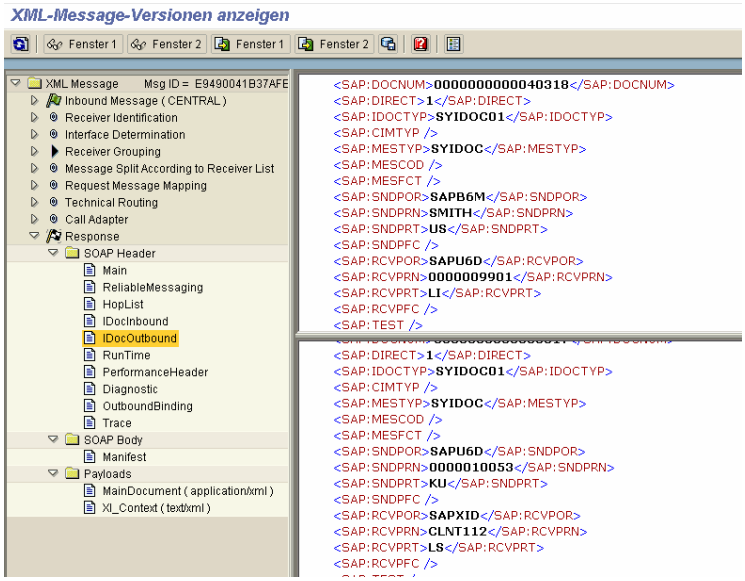
   Leave the *receiver party* empty.



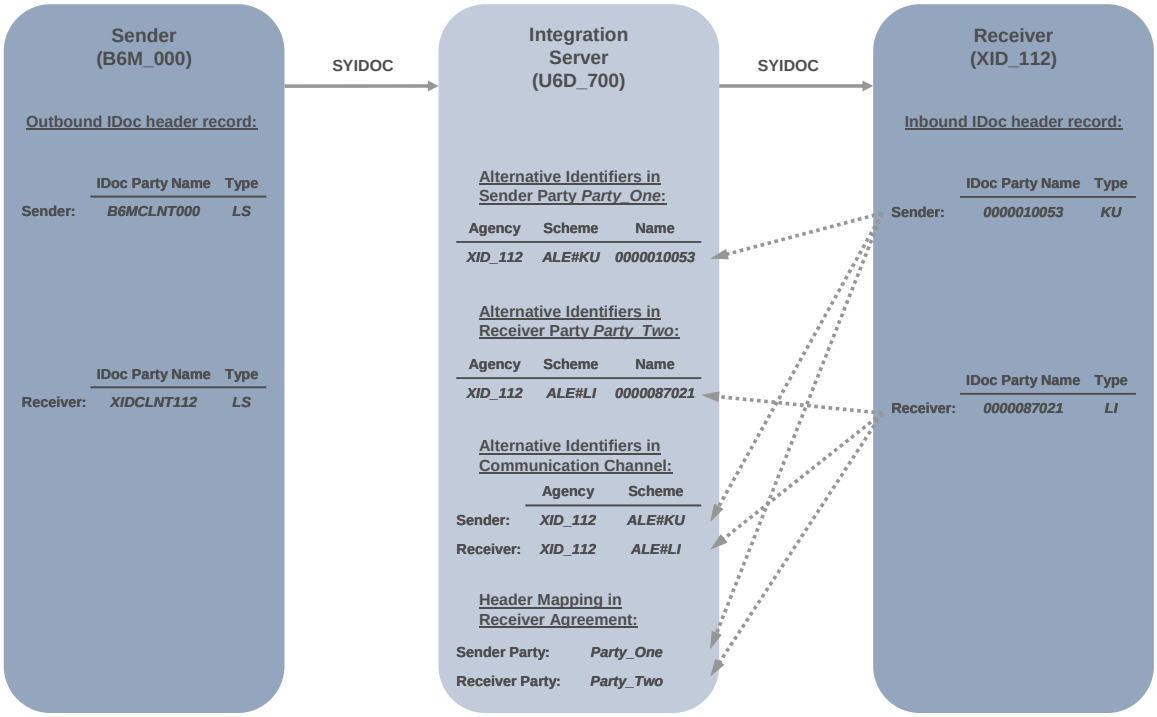
7.2 Execute the Scenario

1. Send an IDoc with sender partner `SMITH` of type *User* (*US*), and receiver partner `9901` of type *Vendor* (*LI*).

   The IDoc Outbound header section indicates that the receiver party is replaced by the corresponding logical systems.

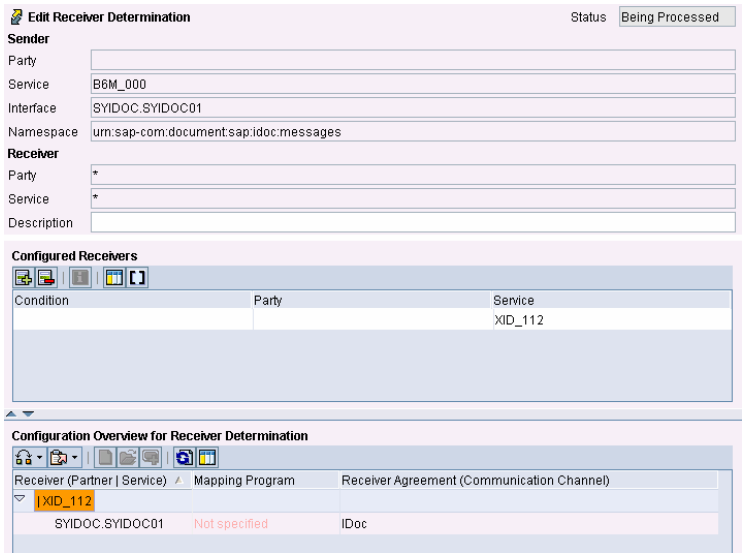


8 Case 6: Partyless (LS) → Party (≠LS)



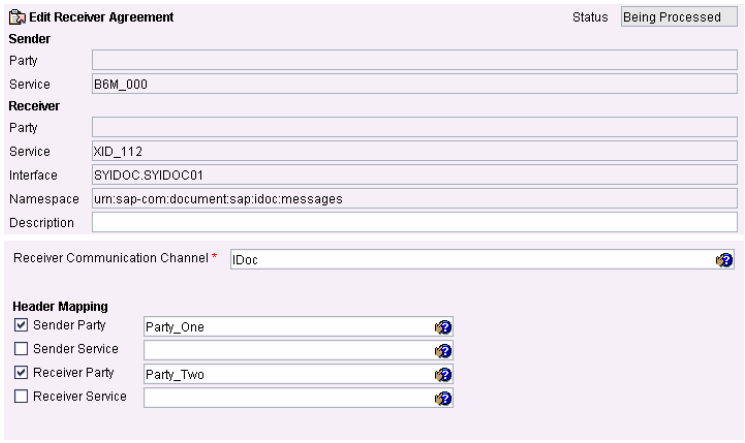
8.1 Integration Builder: Configuration

1. In the receiver determination, select the party-less sender service `B6M_000`, the sender interface `SYIDOC.SYIDOC01`, and the party-less receiver service `XID_112`.

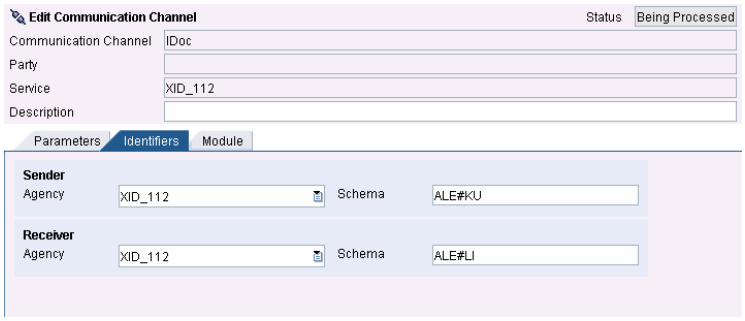


2. In the receiver agreement, header mappings for *sender party* and *receiver party* are required.

   Replace the *sender party* with `Party_One`, and the *receiver party* with `Party_Two`.



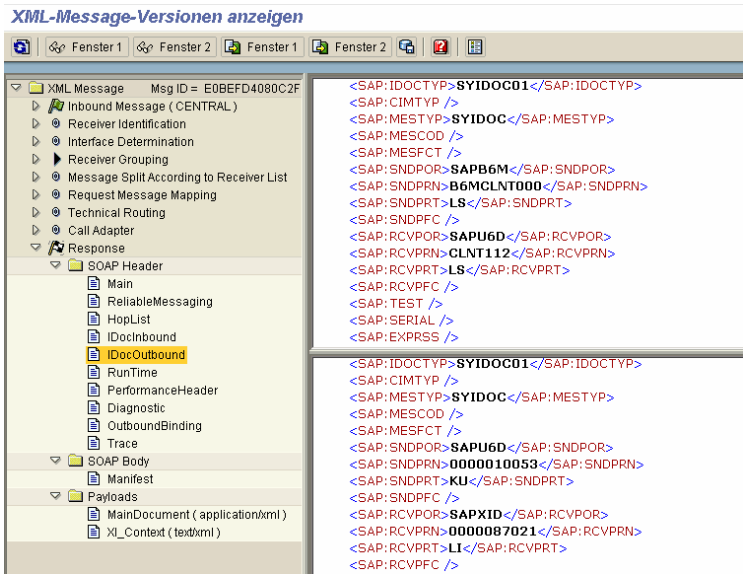
3. In the communication channel, you have to define identifiers for the sender and receiver.



8.2 Execute the Scenario

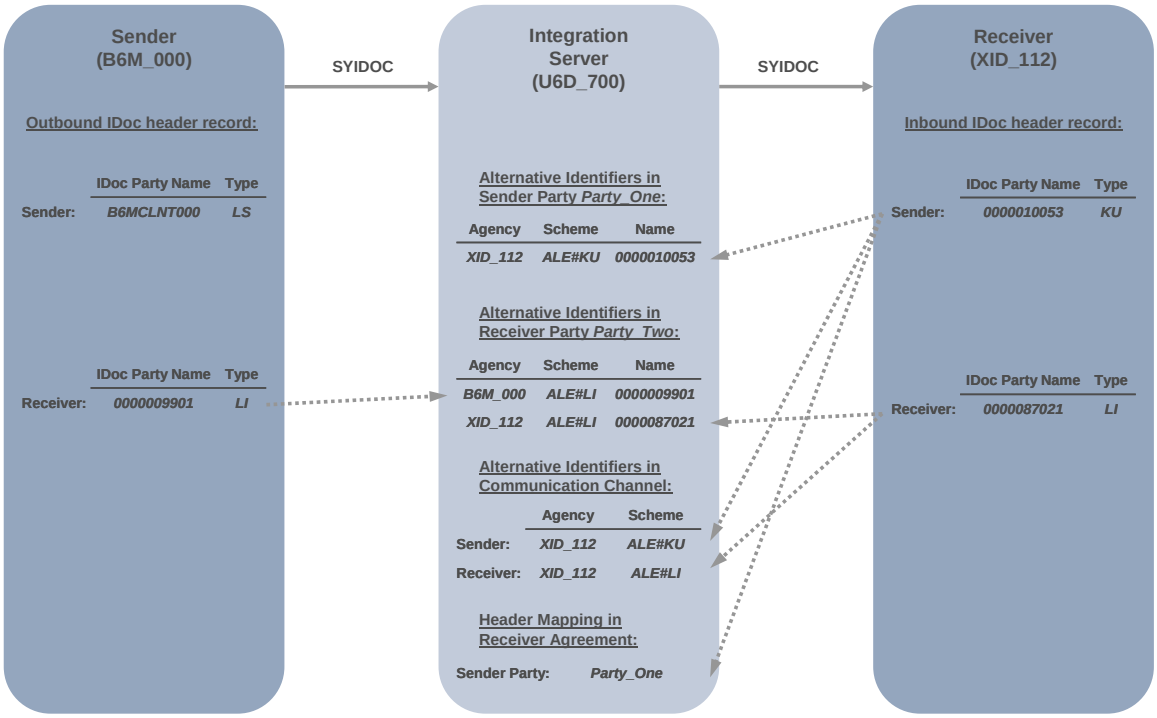
1. Send an IDoc of partner type *LS* (*logical system*).

   The IDoc Outbound header section indicates that the partner names and types are replaced according to the alternative identifiers.



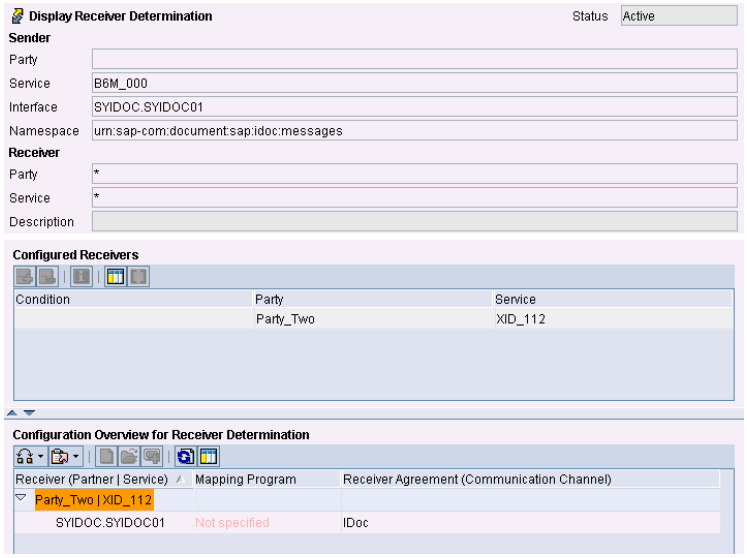
9 Case 7: Partyless Sender → Party (≠LS)

This case is also relevant to RNIF, since RNIF requires parties (IDoc → RNIF).



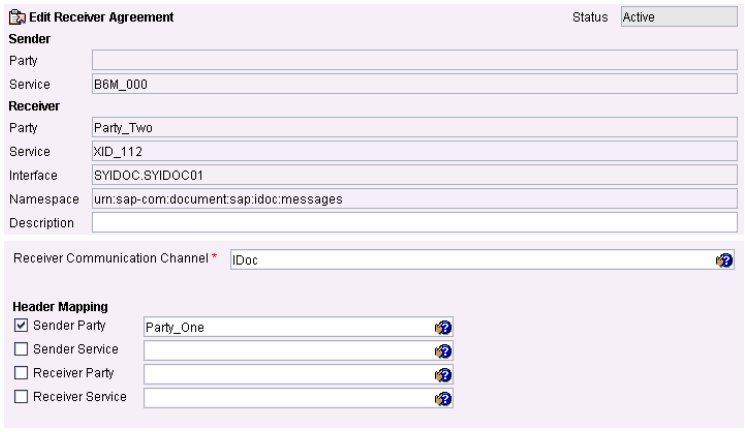
9.1 Integration Builder: Configuration

1. In the receiver determination, select the party-less sender service `B6M_000`, the sender interface `SYIDOC.SYIDOC01`, and the receiver service `XID_112` with party `Party_Two`.

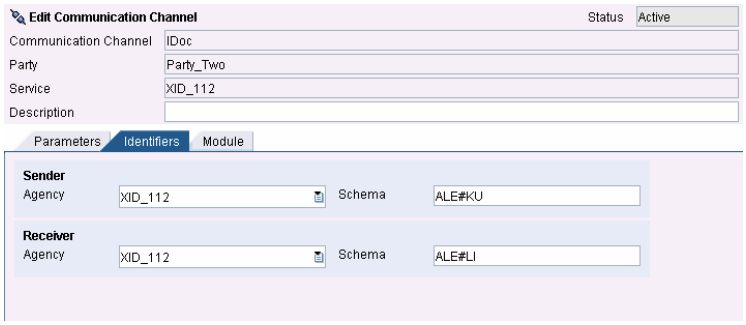


2. In the receiver agreement, a header mapping for the *sender party* is required.

   Replace the *sender party* with `Party_One`.



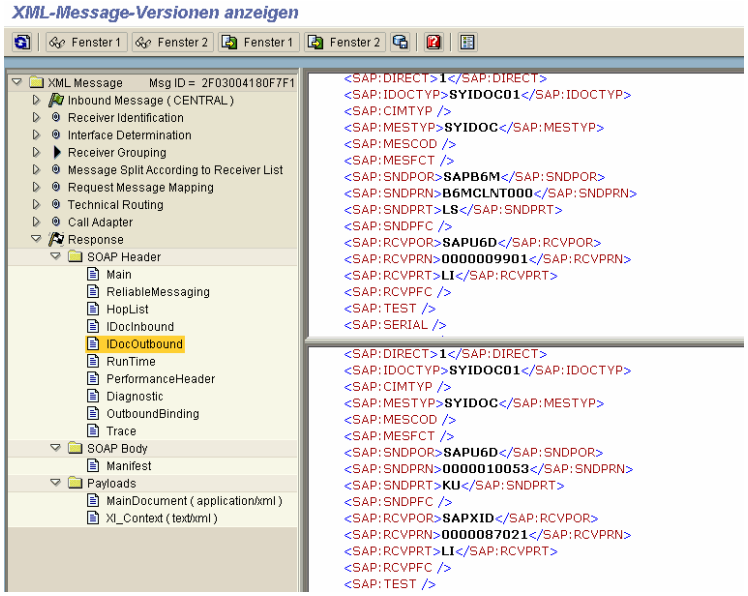
3. In the communication channel, you have to define identifiers for the sender and receiver.



9.2 Execute the Scenario

1. Send an IDoc with sender partner type *LS*, and receiver partner type *LI*.

   The IDoc Outbound header section indicates that the partner names and types are replaced according to the alternative identifiers.



www.sdn.sap.com/irj/sdn/howtoguides