

Training BIC 6 Advanced - Handout



Version: 6.5.2

September 28, 2018

SEEBURGER
BUSINESS INTEGRATION

Table of Contents

| | |
|-----------------------|----|
| 1 Seminar Opening | 1 |
| 2 CSV Adapter | 7 |
| 3 XML Adapter | 12 |
| 4 SQL Adapter | 23 |
| 5 Procedures | 38 |
| 6 Arrays and Hashmaps | 48 |
| 7 Error Child Mapping | 64 |
| 8 BIS:MT Mappings | 68 |
| 9 BAPI Adapter | 78 |
| 10 Seminar Closing | 85 |

Copyright (c) 2018 SEEBURGER AG (<http://www.seeburger.de>). All rights reserved.

If (registered or pending) trademarks are named in this document, the rights of the respective proprietors apply.

Note: False configuration and/or improper use of communication components may result in significant charges from your telecommunication provider. Also consider configuration changes initiated by your telecommunication provider. SEEBURGER is not liable for related additional costs.

Note: Please refer to the Knowledgebase article 13379 for an important notice concerning possible incurred charges from your telecommunication provider.

Note: The information in this document is proprietary to SEEBURGER. No part of this document may be reproduced, copied, or transmitted in any form or purpose without the express prior written permission of SEEBURGER AG.

Note: Export-/Import Regulations

The SEEBURGER AG is a German corporate with subsidiaries in countries worldwide. In daily trading with our customers and partners, each SEEBURGER corporate deals in compliance with the local export and import regulations of the specific country.

SEEBURGER standard software is partly based on US technology (tomcat, Red Hat JBoss, Oracle Java etc.). To be compliant with the US Export Administration Regulation and the Treasury Department regulations SEEBURGER software and/or technical data may NOT be exported /reexported, either directly or indirectly, to any destination subject to U.S. embargos or trade sanctions unless formally authorized by the U.S. Government. Note that said embargoed destinations are subject to change and the scope of what technology is included in the embargo is specific to each embargoed country. For the most current information on U.S. embargoed and sanctioned countries, please contact the U.S. Export Administration Regulations and Treasury Department regulations.


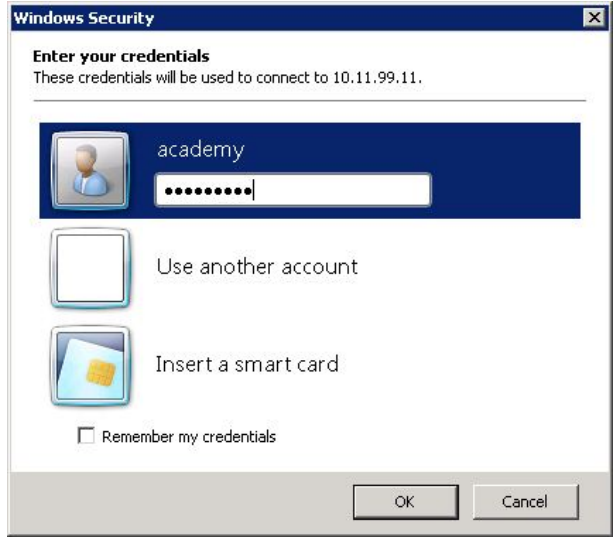
Note: We expressly declare that the document "**SEEBURGER Legal Information**" (delivered also with your BIS installation media) is part of this documentation.

1 Seminar Opening

1.1 Notes

- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____


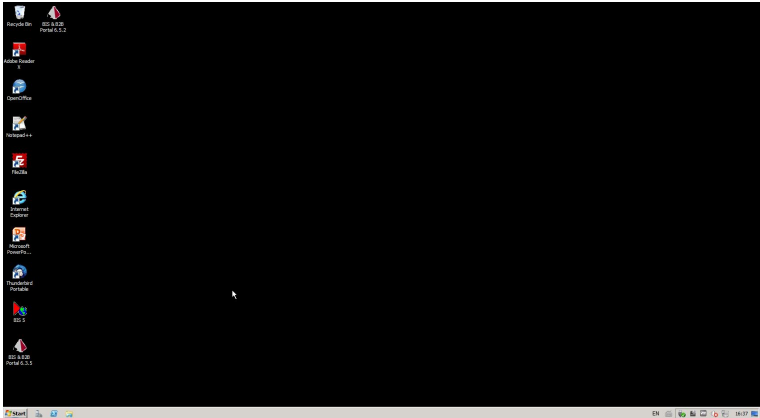
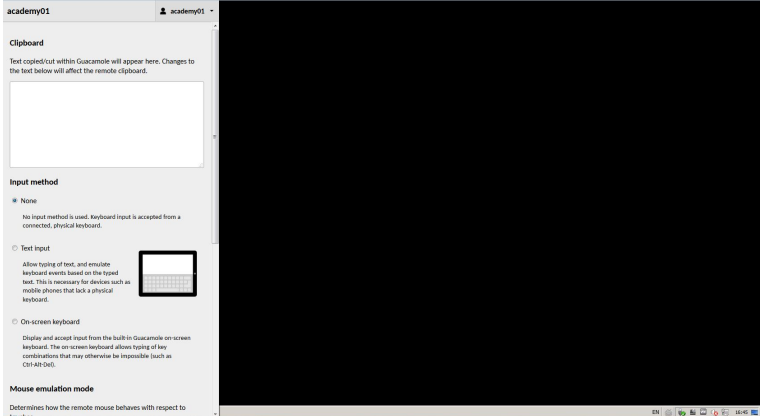
1.2 Academy Room - login to your virtual machine

| Instruction | Illustration |
|---|---|
| 1. Start the computer. You will be logged in automatically. If you have to type in a password use <i>seeburger</i> . | |
| 2. Connect to the VMware image double-clicking the icon on your desktop. |  |
| 3. To login to the VMware image please use the below credentials: <ul style="list-style-type: none"> • User: academy • Password: <i>seeburger</i> |  |

1.3 Academy Cloud - login to your virtual machine

To connect to the Academy Cloud at customers office please follow the steps below. The login details and images which should be used will be provided by your trainer.

| Instruction | Illustration |
|---|--------------|
| 1. Start your web browser and type in https://training.seeburger.com/ | |

| Instruction | Illustration |
|--|--|
| <p>2. Logon to the Academy Cloud</p> <p>User und password will be provided by the trainer. Each attendee will receive his own credentials. These credentials will route the attendee to the correct virtual image.</p> <p>I Please keep in mind that user and password is case sensitive.</p> |  <p>The illustration shows the Seeburger Business Integration Training login page. It features the Seeburger logo at the top, followed by 'SEEBURGER TRAINING'. Below this are two input fields labeled 'Username' and 'Password', and a dark 'Login' button. At the bottom, a red banner reads 'WELCOME TO SEEBURGER REMOTE TRAINING PAGE!' and a red instruction says 'To log in, enter your user name and password and click Login.'</p> <p style="text-align: center;">Logon Academy Cloud</p> |
| <p>3. After the login you will be automatically redirected to the to your assigned virtual image.</p> |  <p>The illustration shows a virtual desktop environment. On the left side, there is a vertical dock with various application icons including 'Academy Cloud', 'PowerPoint', 'Excel', 'Word', 'Outlook', 'Internet Explorer', 'Firefox', 'Google Chrome', 'VLC media player', 'iTunes', 'Skype', 'Zoom', 'Teams', 'OneDrive', 'SharePoint', 'PowerBI', 'Excel', 'Word', 'Outlook', 'Internet Explorer', 'Firefox', 'Google Chrome', 'VLC media player', 'iTunes', 'Skype', 'Zoom', 'Teams', 'OneDrive', 'SharePoint', 'PowerBI'. The desktop background is black. The taskbar at the bottom shows the Start button and several open applications.</p> <p style="text-align: center;">Academy Cloud Image</p> |
| <p>4. With CTRL+ALT+SHIFT you can access the settings menu.</p> <p>5. With this menu you can logout or disconnect. You also have access to some options like display scaling, mouse options and clipboard.</p> <p>6. With CTRL+ALT+SHIFT again this menu will be hidden.</p> |  <p>The illustration shows the 'academy01' settings menu. It has a sidebar with sections: 'Clipboard', 'Input method', and 'Mouse emulation mode'. The 'Clipboard' section is active, showing options for 'Text copied/cut within Guacamole' and 'Changes to the text below will affect the remote clipboard'. The 'Input method' section has radio buttons for 'None', 'Text input', and 'On-screen keyboard'. The 'Mouse emulation mode' section has a dropdown for 'Determines how the remote mouse behaves with respect to _mousemove_'. The main area of the menu is a large black rectangle.</p> <p style="text-align: center;">settings menu</p> |

1.4 Software installed on your VMware Image

BIS 6.5.2

| Name | Path | Root Variables |
|-------------------------------------|---------------------------------------|----------------|
| Business Integration Server (BIS 6) | C:\SEEBURGER\BIS652 | <BIS-ROOT> |
| Mapping Designer (BIC MD 6) | C:\SEEBURGER\MappingDesigner652 | <BICMD-ROOT> |
| Developer Studio (DEV Studio) | C:\SEEBURGER\BIS Developer Studio 652 | <PD-ROOT> |
| B2B Portal (BIP) | C:\SEEBURGER\BIP652 | <PORTAL-ROOT> |
| BIS File Exchange (BIS FX) | C:\SEEBURGER\BISFX-6.5.2 | <BISFX.ROOT> |

BIS 6.3.5

| Name | Path | Root Variables |
|-------------------------------------|-----------------------------------|----------------|
| Business Integration Server (BIS 6) | C:\SEEBURGER\BIS6 | <BIS-ROOT> |
| Mapping Designer (BIC MD 6) | C:\SEEBURGER\MappingDesigner | <BICMD-ROOT> |
| Developer Studio (DEV Studio) | C:\SEEBURGER\BIS Developer Studio | <PD-ROOT> |
| B2B Portal (BIP) | C:\SEEBURGER\BIP | <PORTAL-ROOT> |

BIS 5

| Name | Path | Root Variables |
|-------------------------------------|-------------------|----------------|
| Business Integration Server (BIS 5) | C:\SEEBURGER\BIS5 | <BIS-ROOT> |
| Mapping Designer (BIC MD 5) | C:\SEEBURGER\BIS5 | <BIS-ROOT> |
| Workflow Designer (WFD 5) | C:\SEEBURGER\BIS5 | <BIS-ROOT> |

Additional software

| Name | Path |
|----------------------|---|
| 7-Zip | C:\Program Files (x86)\7-Zip |
| Filezilla FTP Client | C:\Program Files (x86)\FileZilla FTP Client |
| Mozilla Thunderbird | C:\SEEBURGER\PortableApps\ThunderbirdPortable |
| Notepad++ | C:\Program Files (x86)\Notepad++ |
| OpenOffice | C:\SEEBURGER\PortableApps\6.3.5\OpenOfficePortable\App\openoffice |

1.5 Logon details additional servers

Email Server, FTP Server, BIS FX and Database Servers

Below you will find some additional logon details to examine the exercises during the training session.

| Logon Details FTP Server | | Logon Details E-mail Server | |
|--------------------------|--------------------------|-----------------------------|--|
| Hostname | acadcentral | Hostname | acadcentral |
| IP Address | 10.11.99.10 | IP Address (smtp/pop3) | 10.11.99.10 |
| Port | 21 | Ports (smtp/pop3) | 25/110 |
| User | academy<NN> ¹ | E-mail Address | academy<NN>@academy.local ¹ |
| Password | seeburger | Password | seeburger |



E-mail address and password is only for pop3 authentication. For smtp no E-mail address and password must be used for authentication.

| Logon Details BIS FX Server | |
|-----------------------------|---|
| Hostname | acadcentral |
| IP Address | 10.11.99.10 |
| Port | 8080 |
| URL (Web Application) | http://acadcentral:8080/portal-seefx |
| URL (Identity Management) | http://acadcentral:8080/portal-seefx/identity |
| User | academy<NN> ¹ |
| Password | seeburger |

| Logon Details MySQL Server | | Logon MSSQL Server | |
|----------------------------|--------------------------|--------------------|---|
| Hostname | acadcentral | Hostname | localhost |
| IP Address | 10.11.99.10 | IP Address | 127.0.0.1 |
| Port | 3306 | Port | 1433 |
| User | academy<NN> ¹ | Users | sa SeeASDB0 SeePRTDB SeeASB652 SeePRTDB652 |
| Password | seeburger | Passwords | seeburger secret secret secret secret |

¹ <NN> stands for the number of the used VMware image. The user name is identical to the hostname of the used VMware image. Your trainer will help you to use the right user name.

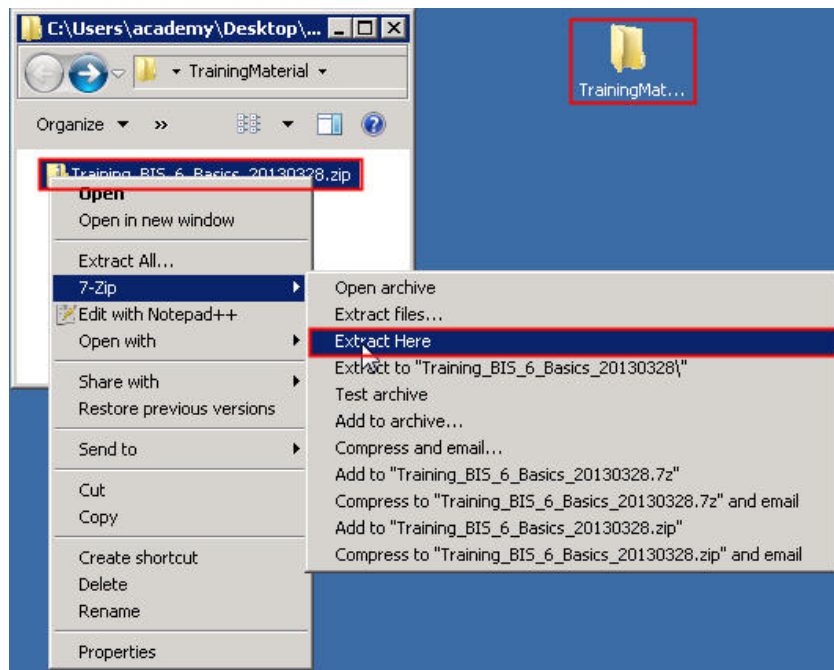
1.6 Inside Academy - Logon Details

Please use our Inside Academy Portal for your feedback after the training. You will also find a shortcut on your desktop to reach the portal start page.

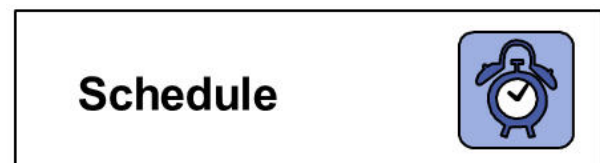
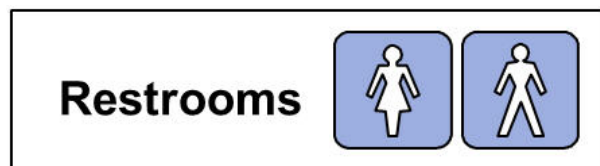
- Url: <http://www.seeburger-academy.de/index.php?id=15>
- User: academy
- Password: seeburger

1.7 Training Material

- You will receive a handout with exercises and tutorials in a printed version.
- After the training we will share the whole training material (slides, exercises, tutorials and classmaterial) as download link via e-mail.
- Your trainer will provide you with a zip-file of the actual training. Please have a look for the folder TrainingMaterial on your desktop.
- Please extract the zip-file and use the content during your training.



1.8 Training Format & Schedule



- Breaks will be accommodated during or after each section.
- SEEBURGER invites you for lunch.
- Training is divided into topics which follow an overall format (Introduction, Walkthrough, Exercise & Tutorial, Questions & Answers). [Table of contents](#).

2 CSV Adapter

2.1 Notes

- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____

2.2 Exercises

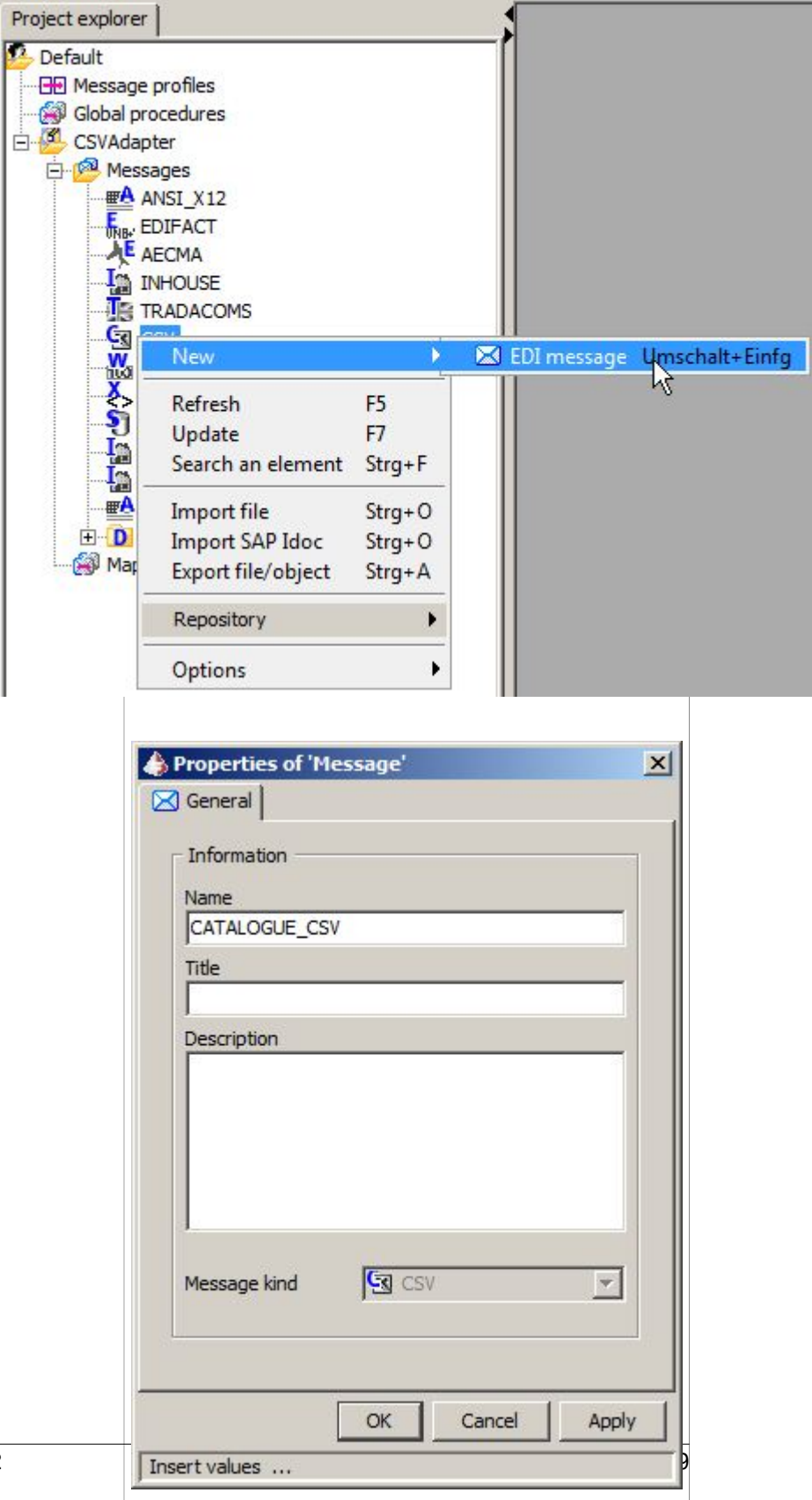
2.2.1 CSV - Exercise 1

Create a message description

- Create a new project **CSVAdapter** within the BIC MD.
- Create a new message structure **CATALOGUE_CSV**.
- Add the defined fields to the CSV message. Details can be found in the description below.

| Level | Record | Field | Min/Max | Type | Length | Option |
|-------|--------|-----------------------------------|---------|------|--------|--------|
| 1 | RECORD | | 1/999 | | | MAN |
| | | CAT_VERSION - Catalogue version | 0/1 | AN.. | 0-3 | OPT |
| | | CAT_NAME - Catalogue name | 0/1 | AN.. | 0-35 | OPT |
| | | CURRENCY – Currency | 0/1 | AN.. | 0-10 | OPT |
| | | POS - Pos number | 0/1 | AN.. | 0-5 | OPT |
| | | EAN - article code | 0/1 | AN.. | 0-13 | OPT |
| | | DESC_SHORT - short description | 0/1 | AN.. | 0-35 | OPT |
| | | DESC_LONG - long description | 0/1 | AN.. | 0-70 | OPT |
| | | ORDERUNIT - order unit | 0/1 | AN.. | 0-25 | OPT |
| | | CONTENTUNIT - content of one unit | 0/1 | AN.. | 0-20 | OPT |
| | | MEASURE - measure unit | 0/1 | AN.. | 0-30 | OPT |
| | | PRICE - price of one content unit | 0/1 | AN.. | 0-20 | OPT |
| | | AMOUNT - price of one order unit | 0/1 | AN.. | 0-5 | OPT |
| | | TOTALAMOUNT – total amount | 0/1 | AN.. | 0-10 | OPT |

2.2.2 CSV - Tutorial 1

| Instruction | Illustration |
|--|---|
| 1. Start BIC Mapping Designer and create a new project named CSVAdapter . | Check chapter "How to Operate BIC Mapping Designer" in Training BIC 6 Basics if you need further assistance. |
| 2. Create a new CSV message named CATALOGUE_CSV and open it with a click. |  <p>The illustration shows two screenshots from the BIC Mapping Designer. The top screenshot shows the 'Project explorer' on the left with the 'Messages' folder expanded under the 'CSVAdapter' project. A right-click context menu is open over the 'Messages' folder, and the 'New' option is selected, leading to a sub-menu where 'EDI message' is highlighted. The bottom screenshot shows the 'Properties of 'Message'' dialog box. The 'General' tab is active, and the 'Name' field contains 'CATALOGUE_CSV'. The 'Message kind' dropdown is set to 'CSV'. The 'Description' field is empty. The 'OK', 'Cancel', and 'Apply' buttons are at the bottom.</p> |
| Training BIC 6 Advanced - Handout - 6.5.2 | |

3. Create a new record and define the record details. Rightclick on message **CATALOGUE_CSV** and select **Re**



2.2.3 CSV - Exercise 2

Import of the destination structure

- Import the message description "CATALOGUE_INH" from the file [msg_CATALOGUE_INH.xml](#) into your current project. It would act as destination structure for the CSV to INHOUSE-mapping.

Create a new mapping

- New objects can be created on the project level by means of the context menu. In this way create a new mapping. Select as an input structure CATALOGUE_CSV from exercise before and as destination structure the before imported INHOUSE structure. Use the attached [testfile](#) to test your mapping.

Source Options (Global Settings for a mapping)

- setCsvSourceHeadline to "on"
- set SourceDelimiter for Element (+), DecimalPoint(.) and Masking (?)

Simple assignments

- Search in both message structures comparable fields and assign the source fields to the corresponding target field. Simple assignments can be made via Drag&Drop.
- Record-ID of the destination file should be unique.
- If the destination field is smaller than the source field, check out the string manipulation commands. It is important to test consistently the newly created mapping. Take the relevant configuration of the test environment and test regularly the mapping.
- Be aware of the delimiter settings and the headline of the source message

I Tip: Comment your work steps.

I The sourceoptions in this exercise are set in the code instead of in the sourceoptions

Use of traces

- Please trace the Record-ID and the Price for Controlling purposes.

2.2.4 CSV - Tutorial 2

| Instruction | Illustration |
|---|---|
| <ol style="list-style-type: none"> 1.Import the message structure msg_CATALOGUE_INH.xml from the classmaterial. 2.Create a new mapping (source = "CATALOGUE_CSV", | Check chapter "How to Operate BIC Mapping Designer" if you need further assistance. |

| Instruction | Illustration |
|--|--|
| destination = "CATALOGUE_INH"). | |
| 3. Go to record program RECORD. | <pre> // global setting setCsvSourceHeadLine("on"); // delimiter settings setSourceDelimiter("Element", "+"); setSourceDelimiter("DecimalPoint", "."); setSourceDelimiter("Masking", "?"); // use counter as unique Record-ID local counter%; copy getCounter("UniqueID") to counter%; copy counter% to RECORD:ID; copy THIS:CAT_NAME to RECORD:CAT_NAME; copy THIS:POS to RECORD:POS; copy THIS:DESC_LONG to RECORD:DESC_LONG; // copy right 10 characters copy right(THIS:ORDERUNIT,10) to RECORD:ORDERUNIT; copy THIS:MEASURE to RECORD:MEASURE; local price%; copy THIS:PRICE to price%; copy price% to RECORD:PRICE; copy THIS:AMOUNT to RECORD:AMOUNT; // trace Record-ID and PRICE traceln("Record-Id: " & counter% & ", Price: " & price%); </pre> |
| 4. You can compare your results with the attached trainer solution . | |

3 XML Adapter

3.1 Notes

- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____

3.2 Exercises

3.2.1 XML - Exercise

1. Create a XML to XML Mapping with [company.xsd](#) as source and also as destination

- Fill the root.customer in the destination with your own profile
 - customer id = 1
 - customer name = <your last name>
 - street = Edisonstraße. 1
 - volume = 5000
 - volume height = 5000
 - volume weight = 5000
 - bio height = 180
 - bio weight = 80
- All information from source root.customer must be mapped to destination root.supplier
- Convert the file [customer.xml](#) with the mapping. The file has namespaces and is UTF-8 encoded. As destination there should be no namespaces and it should also use UTF8 encoding.

I During the import of the file "comany.xsd" some warnings are thrown that can be ignored

2. Download [company_ns.xsd](#) and [bio.xsd](#) to one local folder. Import the company_ns.xsd into BIC MD. Here we have our namespaces. Make the same mapping as above but with Namespaces. Define the Namespace in the root tag of our output file with xmlns:bio="http://example.com/namespace/for/bio".

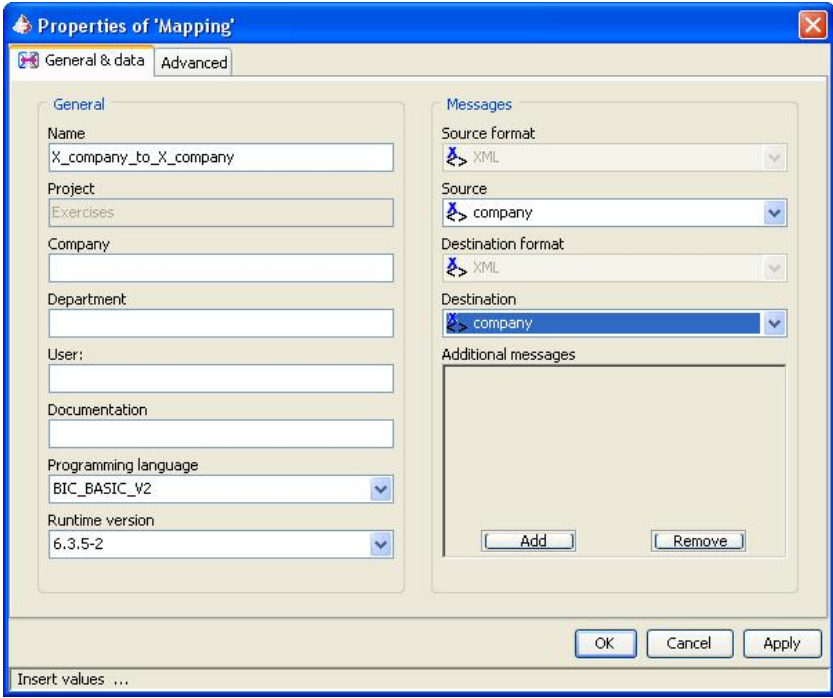
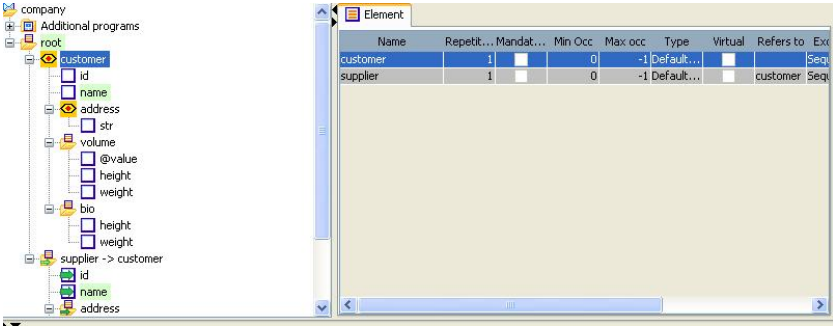
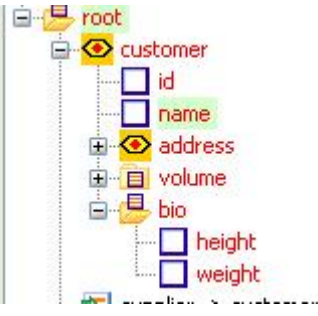
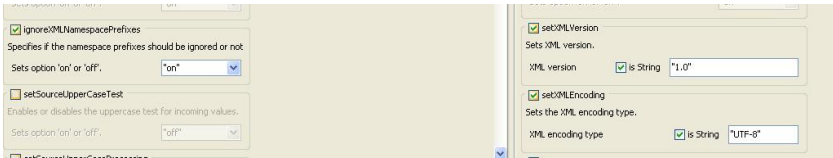
- What needs to be changed here?
- Convert the [customer.xml](#) again.

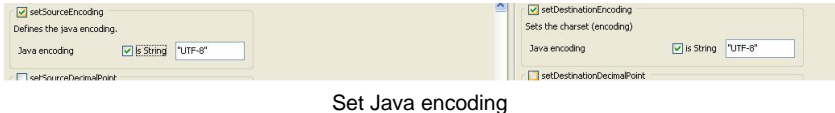
3. Change the existing Namespace from bio to person without changing the structure.

4. Try to convert the file [customer_big.xml](#) with your mapping and make the conversion more performant (SetSelectiveReadOn). Please unzip the file before you try to convert.

I To download the classmaterial files please right-click → save as

3.2.2 XML - Tutorial 1

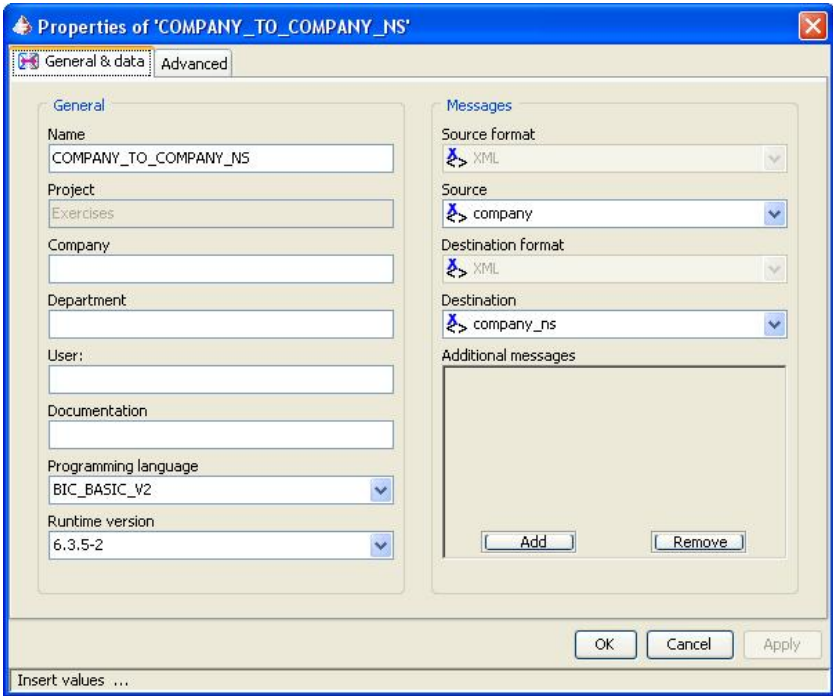
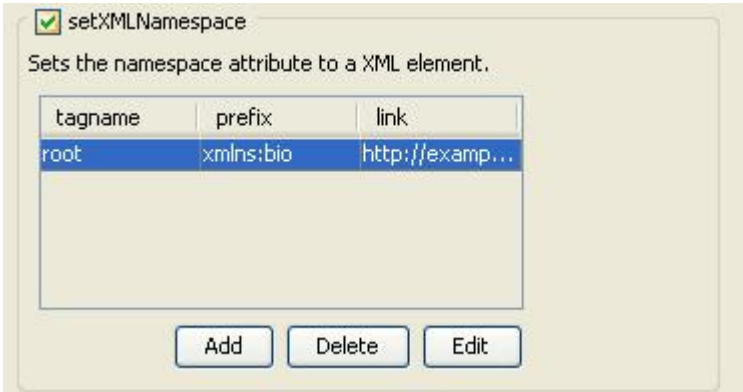
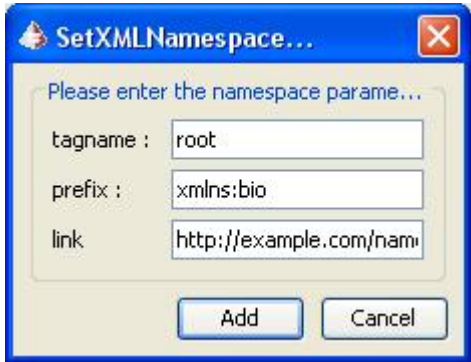
| Instruction | Illustration |
|---|--|
| <p>1. Create a new Mapping with company as source and destination structure</p> |  <p>Create a new mapping</p> |
| <p>2. Set the correct trigger in your xml. Look where you have 1:n relations and set the trigger there</p> <p>I Supplier don't need a trigger, because we don't map anything from there.</p> |  <p>Set trigger in source xml</p> |
| <p>3. Look into the file. You see there is a Namespace bio available. So check if you have this in your structure.</p> |  <p>Check structure for namespace</p> |
| <p>4. As we don't have any Namespace-Definition in our source we must ignore them.</p> <p>5. Use "ignoreXML</p> |  <p>Set ignoreXMLNamespacePrefixes</p> |

| Instruction | Illustration |
|--|--|
| <p>NamespacePrefixes" otherwise you get the following error</p> | <p>Error:</p> <p>otherwise you will get the following output</p> <pre>[Error:ID=2099;LEVEL=1] in Record "PROG_NEW_MAPPING" near Line 15 : copy "80" to root.customer.bio.weight:value; Error-Text: InhouseDocReader doSyntaxCheck(): offset[304]: the found segment 'bio:height' is not in the message description.</pre> |
| <p>6. Check also the encoding settings, before you start to map</p> | <div data-bbox="571 589 1409 701">  <p>Set Java encoding</p> </div> <p>Error:</p> <p>Otherwise you will get something like this output</p> <pre><?xml version="1.0" encoding="UTF-8"?> <root> <customer id="1"> <name>trainer</name> <address str="Edisonstraße 1"/> <volume value="5000"> <height>5000</height> <weight>5000</weight> </volume> <bio> <height>180</height> <weight>80</weight> </bio> </customer> <supplier id="2378"> <name>Customer One</name> <address str="rosenstraße 100"/> <address str="Bonnenweg 223"/> <volume value="2200"> <height>123</height> <weight>800</weight> </volume> <bio> <height>180</height> <weight>80</weight> </bio> </supplier> <supplier id="2379"> <name>Customer Four</name> <address str="Äbungsstrasse 100"/> <address str="Ä, -Weg 223"/> <volume value="2200"> <height>123</height> <weight>800</weight> </volume> <bio> <height>180</height> <weight>80</weight> </bio> </supplier> </root></pre> |
| <p>7. Copy your static information from "New Mapping" or tag root,</p> | <pre>copy "1" to root.customer:id; copy "trainer" to root.customer.name:value; copy "Edisonstraße 1" to root.customer.address:str;</pre> |

| Instruction | Illustration |
|---|---|
| because customer has a 1:n relation | <pre> copy "5000" to root.customer.volume:@value; copy "5000" to root.customer.volume.height:value; copy "5000" to root.customer.volume.weight:value; copy "180" to root.customer.bio.height:value; copy "80" to root.customer.bio.weight:value; </pre> |
| <p>8. Copy the information from customer to supplier.</p> <p>9. Take care that also address has a 1:n relation.</p> | <p>tag customer</p> <pre> copy THIS:id to root.supplier:id; copy THIS.name:value to root.supplier.name:value; copy THIS.volume:@value to root.supplier.volume:@value; copy THIS.volume.height:value to root.supplier.volume.height:value; copy THIS.volume.weight:value to root.supplier.volume.weight:value; copy THIS.bio.height:value to root.supplier.bio.height:value; copy THIS.bio.weight:value to root.supplier.bio.weight:value; </pre> <p>tag address</p> <pre> copy THIS:str to root.supplier.address:str; </pre> |
| <p>10. Check your result.</p> <p>I You can double check your mapping with the trainersolution.</p> | <pre> <?xml version="1.0" encoding="UTF-8"?> <root> <customer id="1"> <name>trainer</name> <address str="Edisonstraße 1"/> <volume value="5000"> <height>5000</height> <weight>5000</weight> </volume> <bio> <height>180</height> <weight>80</weight> </bio> </customer> <supplier id="2378"> <name>Customer One</name> <address str="rosenstraße 100"/> <address str="äonenweg 223"/> <volume value="2200"> <height>123</height> <weight>800</weight> </volume> <bio> <height>180</height> <weight>80</weight> </bio> </supplier> <supplier id="2379"> <name>Customer Four</name> <address str="Übungsstrasse 100"/> <address str="€-Weg 223"/> <volume value="2200"> <height>123</height> <weight>800</weight> </volume> <bio> <height>180</height> <weight>80</weight> </bio> </supplier> </pre> |

| Instruction | Illustration |
|-------------|--|
| | <pre></supplier> </root></pre> |

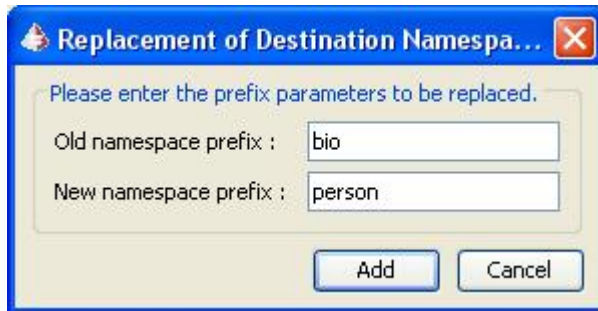
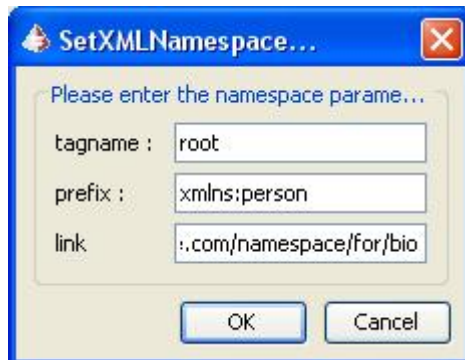
3.2.3 XML - Tutorial 2

| Instruction | Illustration |
|--|--|
| <p>1. Create a new Mapping with company as source and company_ns as destination structure</p> <p>I It is not necessary to define company_ns as source structure because we can use "ignoreXML NamespacePrefixes".</p> |  <p>Create new mapping</p> |
| <p>2. In the destination options add a new setxmlnamespace Definition</p> |  <p>Add new xml namespace definition</p> |
| <p>3. Add the following lines</p> <pre>tag: root prefix: xmlns:bio link: http://example.com/ namespace/for/bio</pre> |  |

| Instruction | Illustration |
|---|---|
| | <p style="text-align: center;">Set xml namespace definition</p> <pre> copy "1" to root.customer.id; copy "trainer" to root.customer.name:value; copy "Edisonstraße 1" to root.customer.address:str; copy "5000" to root.customer.volume:@value; copy "5000" to root.customer.volume.height:value; copy "5000" to root.customer.volume.weight:value; copy "180" to root.customer.bio.bio\:@height:value; copy "80" to root.customer.bio.bio\:@weight:value; </pre> |
| <p>4. Copy your information from "New Mapping" or tag root, because customer has a 1:n relation. Be careful with the new destination definition of bio.</p> | <p>5. Copy the information from customer to supplier. Take care that also address has a 1:n relation</p> <p>tag customer</p> <pre> copy THIS:id to root.supplier.id; copy THIS.name:value to root.supplier.name:value; copy THIS.volume:@value to root.supplier.volume:@value; copy THIS.volume.height:value to root.supplier.volume.height:value; copy THIS.volume.weight:value to root.supplier.volume.weight:value; copy THIS.bio.height:value to root.supplier.bio.bio\:@height:value; copy THIS.bio.weight:value to root.supplier.bio.bio\:@weight:value; </pre> <p>tag address</p> <pre> copy THIS:str to root.supplier.address:str; </pre> |
| <p>6. Check your result.</p> <p>I You can double check your mapping with the trainersolution.</p> | <pre> <?xml version="1.0" encoding="UTF-8"?> <root xmlns:bio="http://example.com/namespace/for/bio"> <customer id="1"> <name>trainer</name> <address str="Edisonstraße 1"/> <volume value="5000"> <height>5000</height> <weight>5000</weight> </volume> <bio> <bio:height>180</bio:height> <bio:weight>80</bio:weight> </bio> </customer> <supplier id="2378"> <name>Customer One</name> <address str="rosenstraße 100"/> <address str="äonenweg 223"/> <volume value="2200"> <height>123</height> <weight>800</weight> </volume> <bio> <bio:height>180</bio:height> <bio:weight>80</bio:weight> </bio> </supplier> <supplier id="2379"> <name>Customer Four</name> <address str="Übungsstrasse 100"/> <address str="€-Weg 223"/> <volume value="2200"> <height>123</height> </pre> |

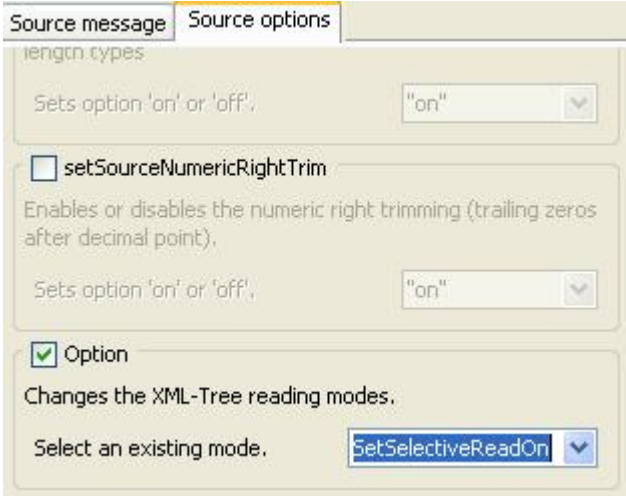
| Instruction | Illustration |
|-------------|---|
| | <pre> <weight>800</weight> </volume> <bio> <bio:height>180</bio:height> <bio:weight>80</bio:weight> </bio> </supplier> </root> </pre> |

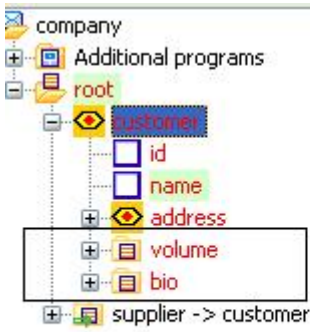
3.2.4 XML - Tutorial 3

| Instruction | Illustration |
|---|---|
| 1. In the destination options add a new "exchangeDestination NamespacePrefix" Definition |  <p>Add a new destination namespace prefix</p> |
| 2. Also change the prefix for the Namespace in setxmlnamespace tag: root prefix: xmlns:personallink: http://example.com/ namespace/for/bio |  |
| 3. Don't be afraid if you get an error when you look into this file with the MessageViewer. Use an Editor - the MessageViewer is based on the structure but we change this. | <p>Error:</p> <pre> [Error:ID=2099;LEVEL=1] InhouseDocReader doSyntaxCheck(): offset[257]: the found segment 'person:height' is not in the message description. </pre> |
| 4. Enjoy your result in an editor I You can double check your mapping with the trainer solution . | <pre> <?xml version="1.0" encoding="UTF-8"?> <root xmlns:person="http://example.com/namespace/for/bio"> <customer id="1"> <name>trainer</name> <address str="EdisonstraÙe 1"/> <volume value="5000"> <height>5000</height> <weight>5000</weight> </volume> <bio> <person:height>180</person:height> <person:weight>80</person:weight> </bio> </customer> </supplier id="2378"> </pre> |

| Instruction | Illustration |
|-------------|---|
| | <pre> <name>Customer One</name> <address str="rosenstraße 100"/> <address str="äonenweg 223"/> <volume value="2200"> <height>123</height> <weight>800</weight> </volume> <bio> <person:height>180</person:height> <person:weight>80</person:weight> </bio> </supplier> <supplier id="2379"> <name>Customer Four</name> <address str="Übungsstrasse 100"/> <address str="€-Weg 223"/> <volume value="2200"> <height>123</height> <weight>800</weight> </volume> <bio> <person:height>180</person:height> <person:weight>80</person:weight> </bio> </supplier> </root> </pre> |

3.2.5 XML - Tutorial 4

| Instruction | Illustration |
|---|--|
| <p>1. Set the option "SetSelectiveReadOn" in the Source Options</p> |  <p>Set selective readon</p> |

| Instruction | Illustration |
|--|--|
| <p>2. Be careful with the settings of the trigger in the mapping now. The following is not valid anymore if you want to read the volume and bio record</p> |  |
| <p>3. You will get something like the following in your output</p> | <p>Error:</p> <pre data-bbox="598 750 1093 1422"><?xml version="1.0" encoding="UTF-8"?> <root> <customer id="1"> <name>trainer</name> <address str="Edisonstraße 1"/> <volume value="5000"> <height>5000</height> <weight>5000</weight> </volume> <bio> <height>180</height> <weight>80</weight> </bio> </customer> <supplier id="2378"> <name>Customer One</name> <address str="rosenstraße 100"/> <address str="äonenweg 223"/> <volume value="0"> <height/> <weight/> </volume> <bio> <height/> <weight/> </bio> </supplier> </root></pre> |

| Instruction | Illustration |
|---|---|
| <p>4. The triggers must be set correctly</p> | |
| <p>5. The mapping code from the customer tag must be changed - volume and bio must be filled with their reading commands</p> <p>I You can double check your mapping with the trainer solution.</p> | <p>Record customer:</p> <pre> /* copy customer information to supplier information */ copy THIS:id to root.supplier:id; copy THIS.name:value to root.supplier.name:value; </pre> <p>Record volume:</p> <pre> copy THIS:@value to root.supplier.volume:@value; copy THIS.height:value to root.supplier.volume.height:value; copy THIS.weight:value to root.supplier.volume.weight:value; </pre> <p>Record bio:</p> <pre> copy THIS.height:value to root.supplier.bio.height:value; copy THIS.weight:value to root.supplier.bio.weight:value; </pre> |

4 SQL Adapter

4.1 Notes

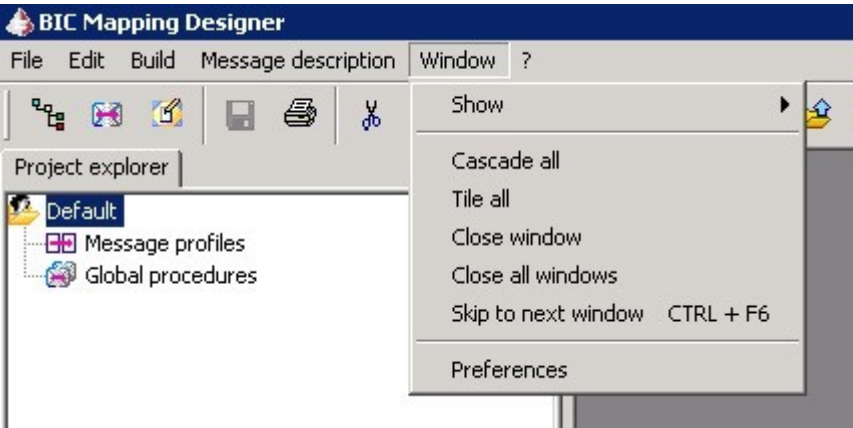
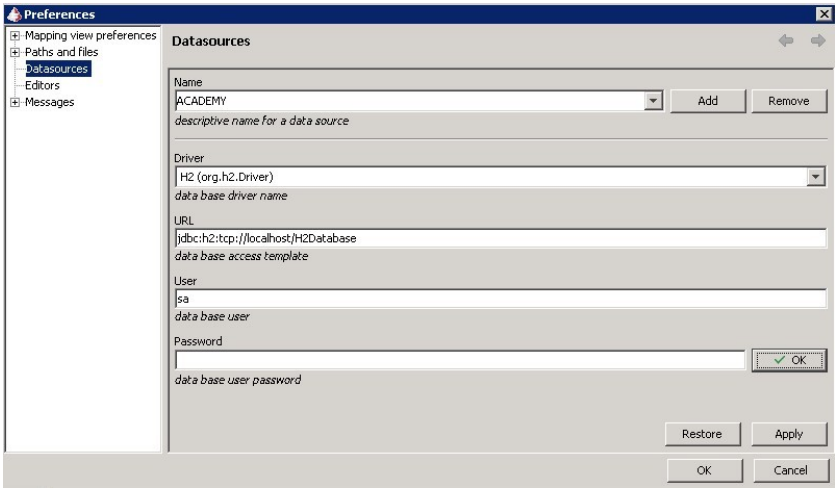
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____

4.2 Exercises

4.2.1 SQL Adapter - Exercise 1

1. Please create a data source to the H2 database which is part of the BIC Mapping Designer installation.

4.2.2 SQL Adapter - Tutorial 1

| Instruction | Illustration |
|---|---|
| <ol style="list-style-type: none"> 1. Please create a datasource to the H2 database which is part of the mapping designer installation. 2. Open the preferences |  <p>BICMD - create datasource</p> |
| <ol style="list-style-type: none"> 3. ADD a new datasource with name ACADEMY. <p>I Follow the details on the screenshot. The password must be <empty>.</p> |  <p>BICMD - set preferences for ACADEMY data source</p> |

4.2.3 SQL Adapter - Exercise 2

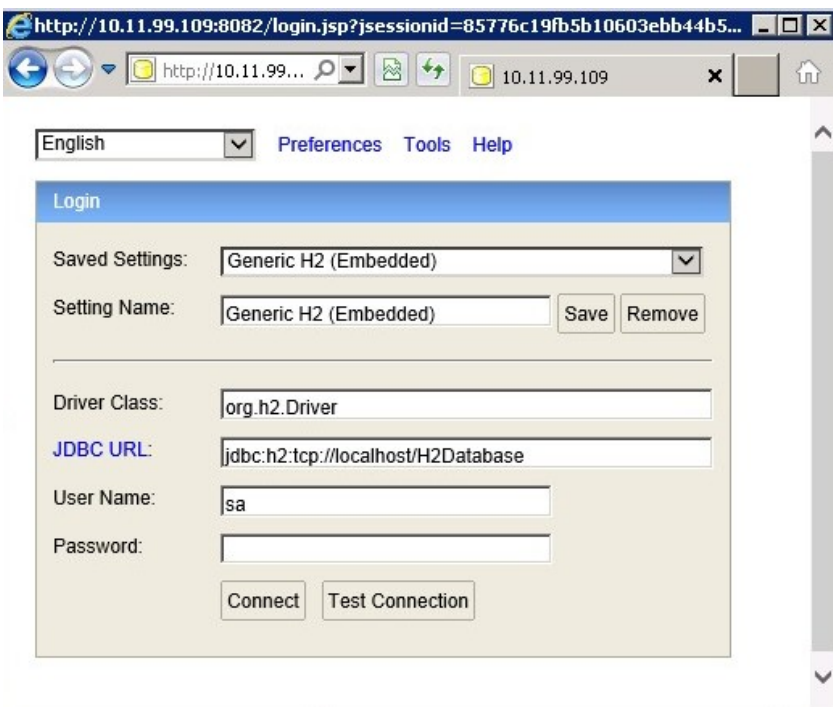
1. In the following part of the exercise you will learn how to create a SQL structure with the Mapping Designer.
2. Before we are able to create the SQL structure in BICMD we have to execute the following steps:
 - start H2 client from <BICMD>\bin\runH2.client.bat - an Internet Explorer Window will open.
 - create the DB tables at the H2 database using the below script.

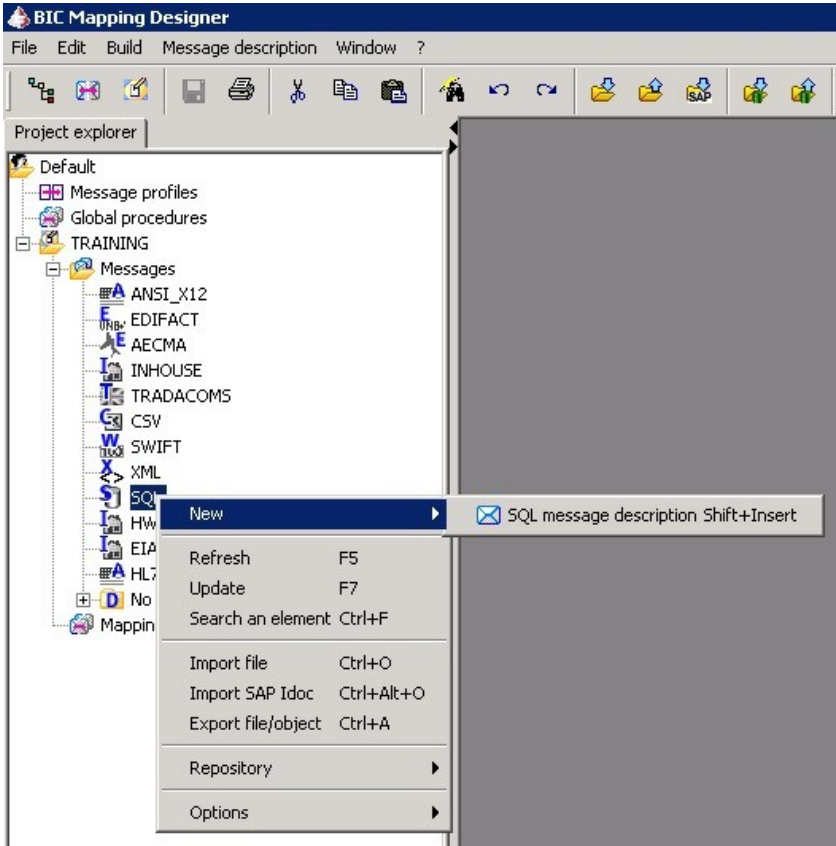
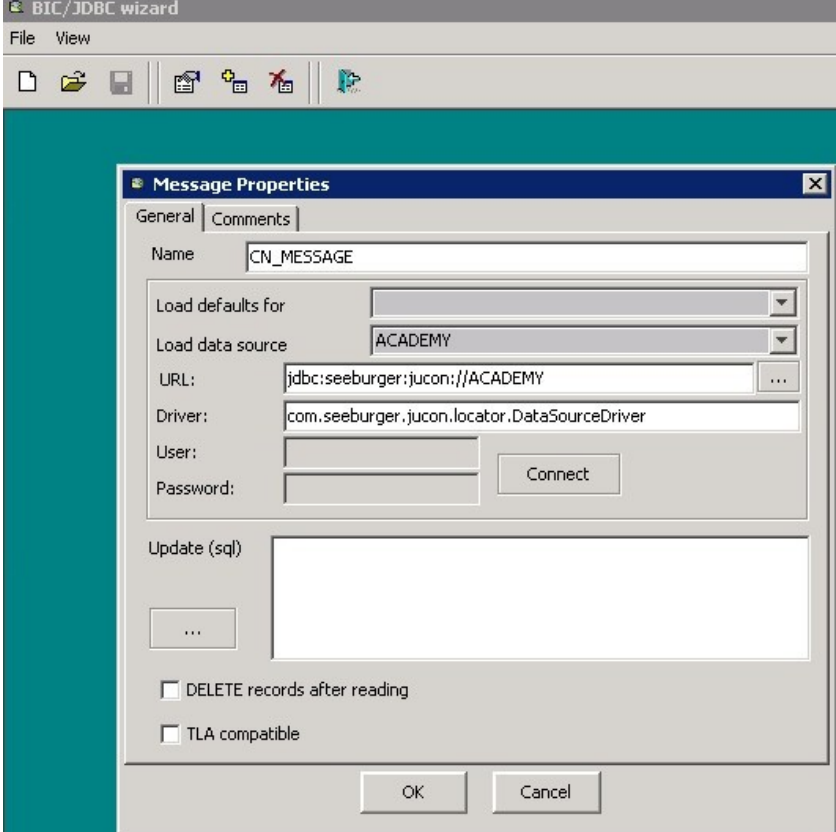
```
CREATE TABLE CN_LINE (EDI_SENDER VARCHAR(16), EDI_RECEIVER VARCHAR(16),
TRANS_NR NUMBER(5,0), CN_NR VARCHAR(20), NET_AMOUNT NUMBER(12,2),
GROSS_AMOUNT NUMBER(12,2), VAT_AMOUNT NUMBER(12,2), VAT_RATE NUMBER(4,2),
TEXT_1 VARCHAR(250), TEXT_2 VARCHAR(250), TEXT_3 VARCHAR(250));

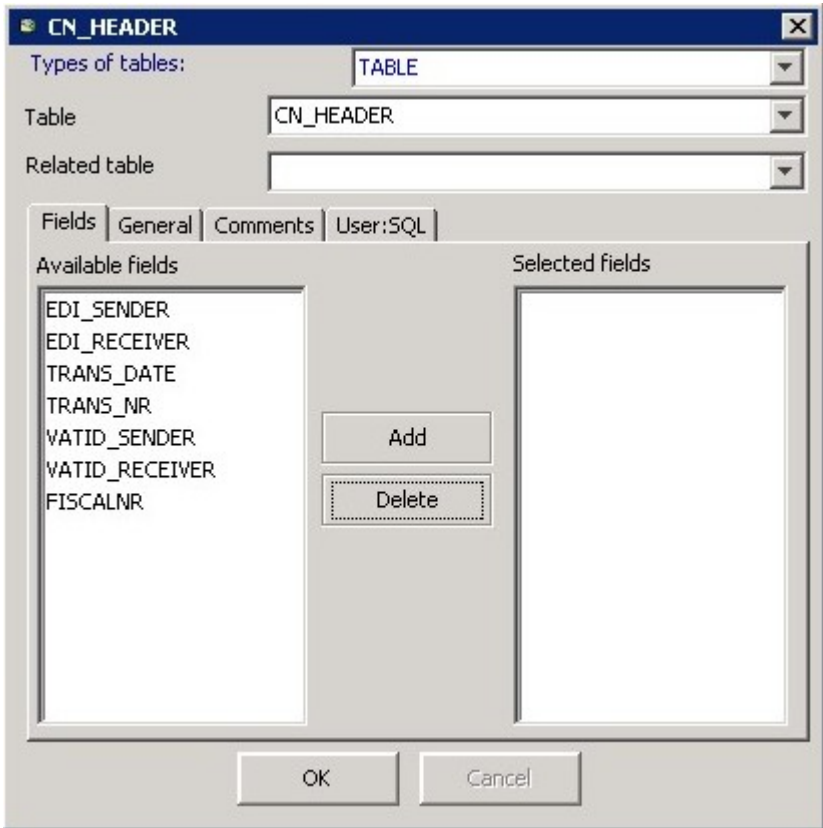
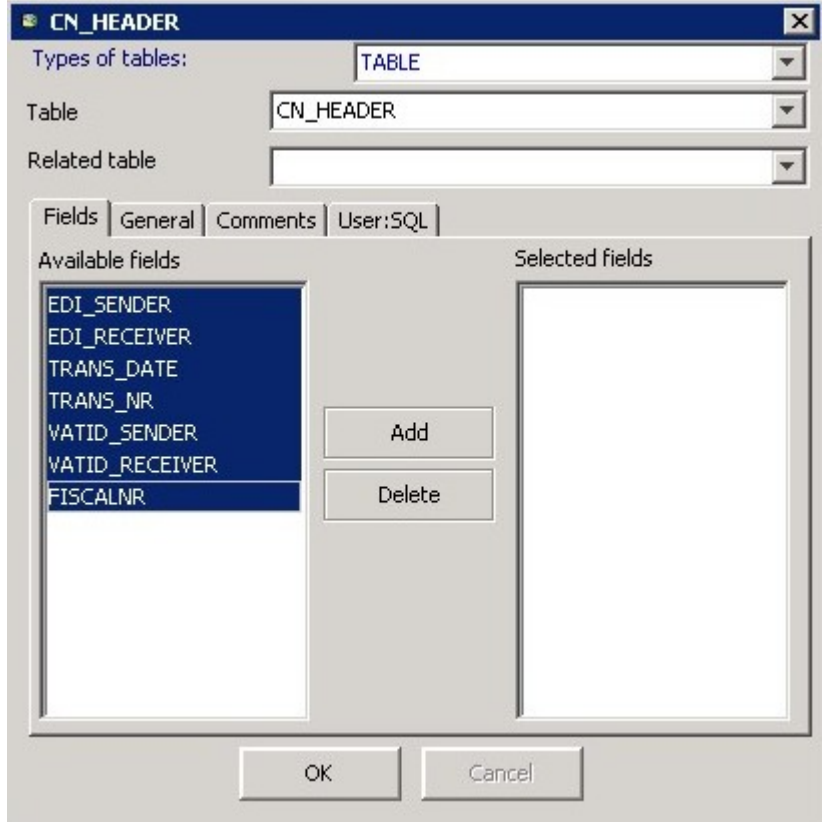
CREATE TABLE CN_HEADER (EDI_SENDER VARCHAR(16), EDI_RECEIVER VARCHAR(16),
STATUS VARCHAR(2), TRANS_DATE DATE, TRANS_NR NUMBER(5,0),
VATID_SENDER VARCHAR(20), VATID_RECEIVER VARCHAR(20), FISCALNR VARCHAR(15));
```

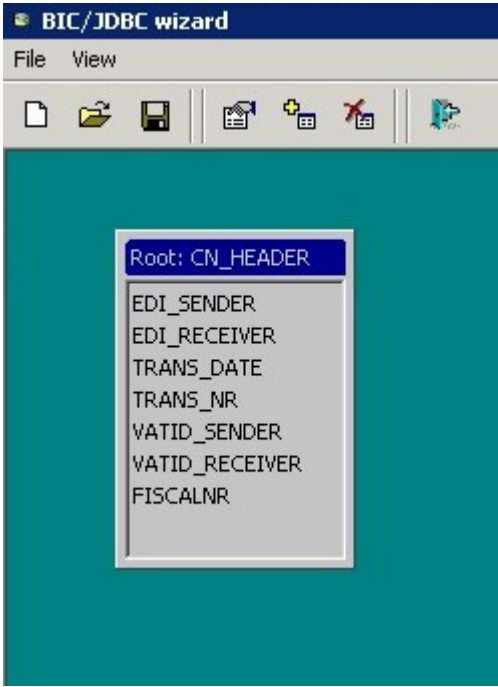
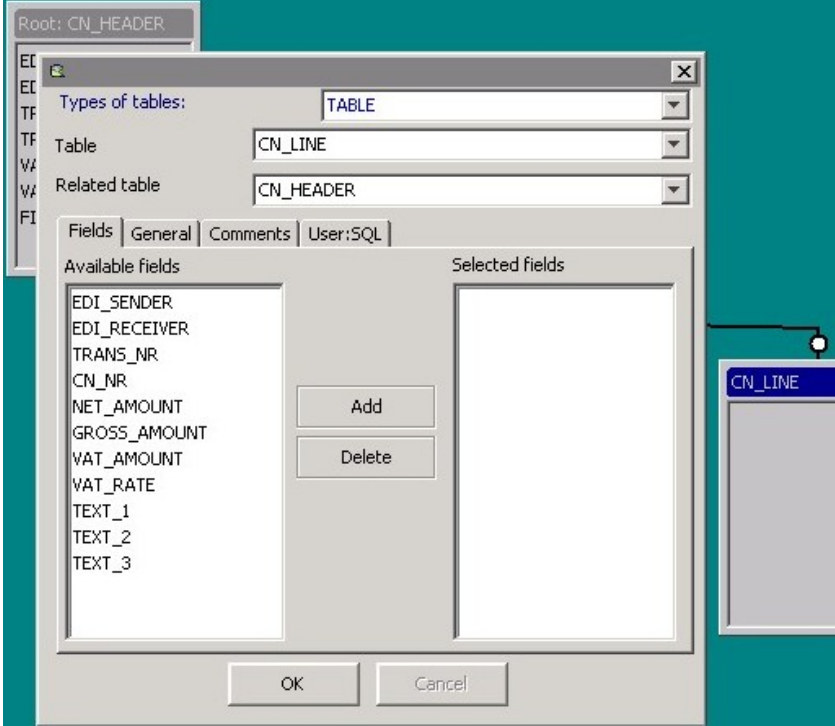
3. After the database tables have been created by a script we can go back to MappingDesigner and start with creating/referencing the table structure.
4. Please create a message CH_Message with two tables CN_Header and CN_Lines by using the JDBC Wizard.

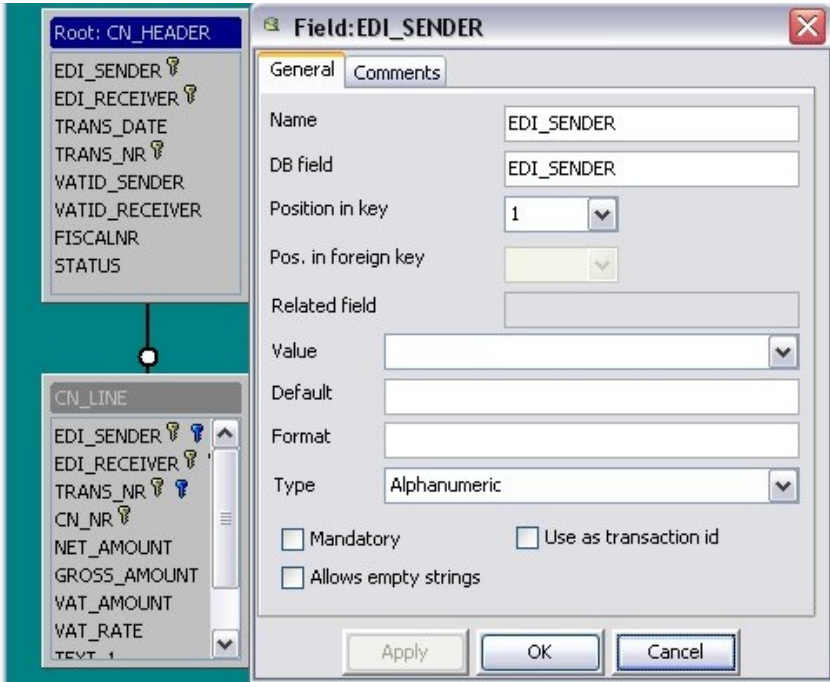
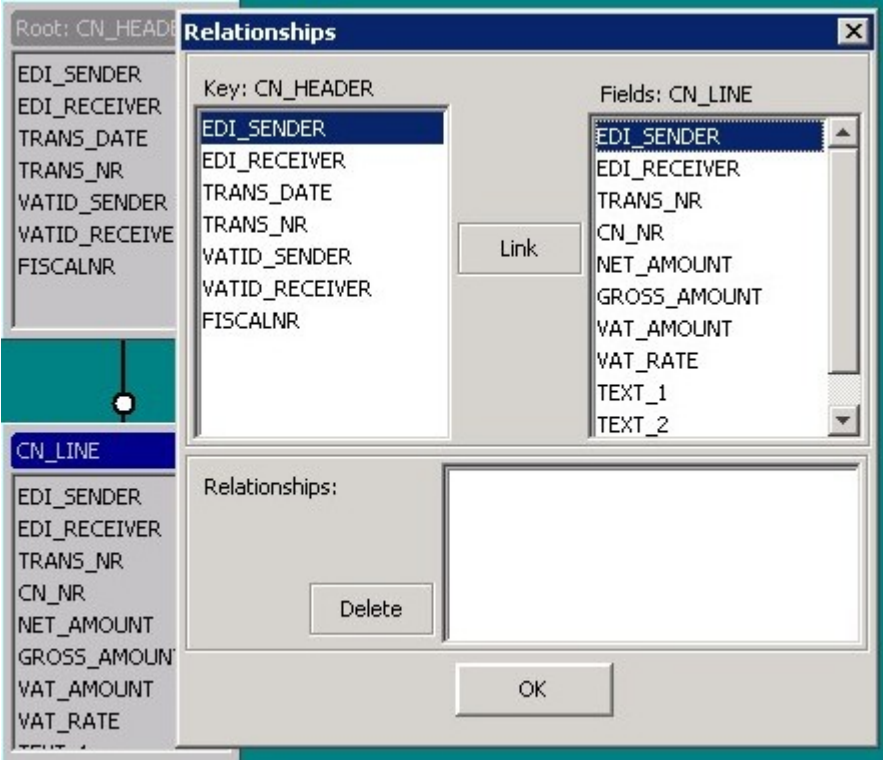
4.2.4 SQL Adapter - Tutorial 2

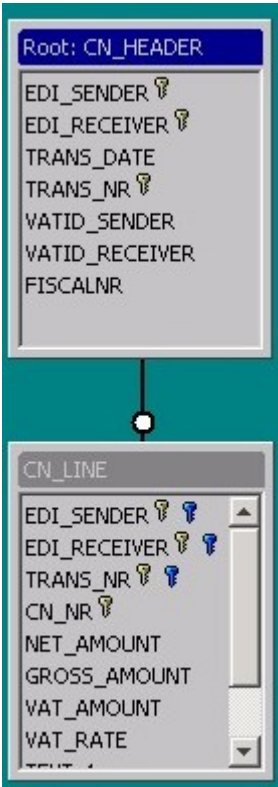
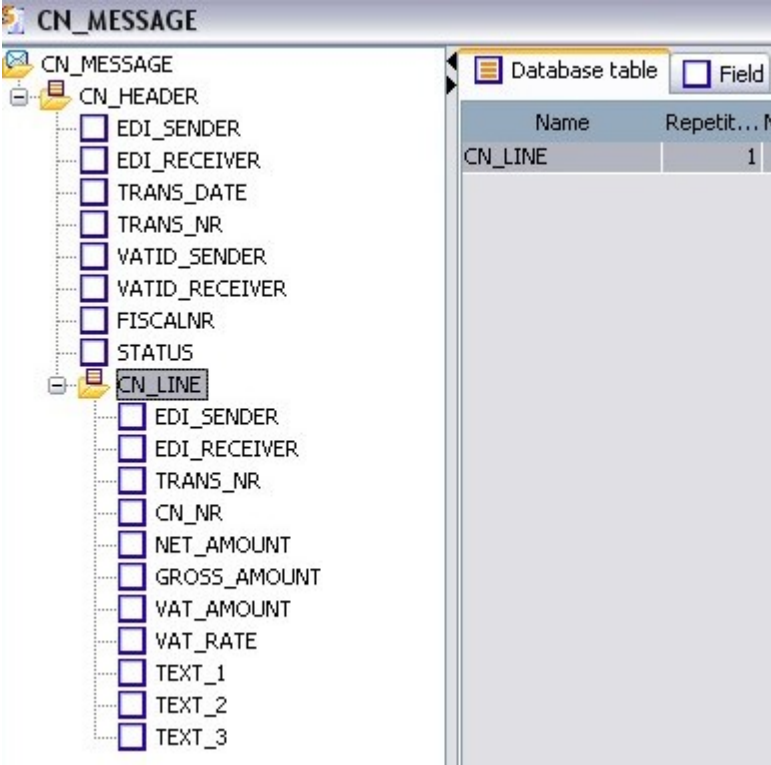
| Instruction | Illustration |
|---|--|
| <ol style="list-style-type: none"> 1. Before we are able to create the SQL structure we have to create the DB tables at the H2 database. 2. At first we have to start the H2 DB client. Please start this with <BICMD> \bin\runH2.client.bat By double clicking this file an IE window will be opened. 3. Logon details URL: jdbc:h2:tcp://localhost/H2Database user: sa Password: <empty> |  <p style="text-align: center;">Logon to H2 DB client</p> |
| <ol style="list-style-type: none"> 4. Please copy the following code to the SQL statement text field and push the button run. | <pre>CREATE TABLE CN_LINE (EDI_SENDER VARCHAR(16), EDI_RECEIVER VARCHAR(16), TRANS_NR NUMBER(5,0), CN_NR VARCHAR(20), NET_AMOUNT NUMBER(12,2), GROSS_AMOUNT NUMBER(12,2), VAT_AMOUNT NUMBER(12,2), VAT_RATE NUMBER(4,2), TEXT_1 VARCHAR(250), TEXT_2 VARCHAR(250), TEXT_3 VARCHAR(250)); CREATE TABLE CN_HEADER (EDI_SENDER VARCHAR(16), EDI_RECEIVER VARCHAR(16), STATUS VARCHAR(2), TRANS_DATE DATE, TRANS_NR NUMBER(5,0), VATID_SENDER VARCHAR(20), VATID_RECEIVER VARCHAR(20), FISCALNR VARCHAR(15));</pre> |

| Instruction | Illustration |
|---|---|
| <ol style="list-style-type: none"> 5. With the existing database tables we are able to create the SQL structure. 6. Please start your mapping designer and create a new project with the name TRAINING. 7. Below this new project you will find two subfolder, one for messages and the other one for mappings. 8. Please open the folder messages and right click at the SQL message to create a new SQL structure. 9. The JDBC wizard will be started automatically. |  <p style="text-align: center;">Create new SQL structure</p> |
| <ol style="list-style-type: none"> 10. Please start with creating the first message CN_MESSAGE 11. For each new message you have to set the main message properties. 12. Please use the data source we created at the first exercise to connect to the DB. 13. Click Connect to verify that the data source is working properly. 14. Please confirm the message properties by clicking the OK button. |  <p style="text-align: center;">Create message including the properties</p> |

| Instruction | Illustration |
|--|--|
| <p>15. Please add a new table. The first table will always be the root table.</p> <p>16. For this example the CN_HEADER should be used as root.</p> <p>17. Mark all available fields and push the button add to select this fields to the message structure.</p> | <div data-bbox="576 241 1402 1066">  <p>The screenshot shows a dialog box titled 'CN_HEADER'. It has a 'Types of tables' dropdown set to 'TABLE', a 'Table' dropdown set to 'CN_HEADER', and an empty 'Related table' dropdown. The 'Fields' tab is selected, showing a list of available fields on the left and a list of selected fields on the right. The available fields are: EDI_SENDER, EDI_RECEIVER, TRANS_DATE, TRANS_NR, VATID_SENDER, VATID_RECEIVER, and FISCALNR. The 'Add' button is visible between the two lists.</p> <p>Select the table</p> </div> <div data-bbox="576 1149 1402 1968">  <p>This screenshot is identical to the one above, showing the 'CN_HEADER' dialog box with the 'Fields' tab selected. The 'Available fields' list contains the same set of fields, and the 'Add' button is visible.</p> <p>Select all fields</p> </div> |

| Instruction | Illustration |
|---|--|
| <p>18. After setup of the first table structure you will see following screen.</p> <p>19. Please repeat the steps 14-16 for the table CN_LINE.</p> <p>I Keep in mind to define the CN_HEADER table as related table.</p> |  <p>Table structure of CN_HEADER</p> |
| <p>20. Please add a second table and choose the CN_LINE table.</p> <p>21. This table should be related to the CN_HEADER table.</p> |  <p>Select a second table</p> |

| Instruction | Illustration |
|---|--|
| <p>22. To set the primary key information you have to click to the table field which should be part of the primary key and set the position of the field at the primary key.</p> <p>23. The following fields should be marked as primary keys:</p> <p>CN_HEADER: EDI_SENDER, EDI_RECEIVER, TRANS_NR</p> <p>CN_LINE: EDI_SENDER, EDI_RECEIVER, TRANS_NR, CN_NR</p> |  |
| <p>24. To configure the relation between both tables you have to click at the node on the link line.</p> <p>25. The following fields should be linked:</p> <ul style="list-style-type: none"> • EDI_SENDER • EDI_RECEIVER • TRANS_NR <p>I Please mark EDI_SENDER at the key list and the fields list. After this you have to push the button link. Repeat this step for the other two fields.</p> |  <p style="text-align: center;">Set relation between the 2 tables</p> |

| Instruction | Illustration |
|--|--|
| <p>26. The result should look like this.</p> |  <p>Relation between two tables</p> |
| <p>27. Please save the SQL message and close the JDBC wizard. Back in MappingDesigner you will find the new SQL structure.</p> |  |
| <p>28. To double check your structure you can have a look at the trainer solution</p> | |

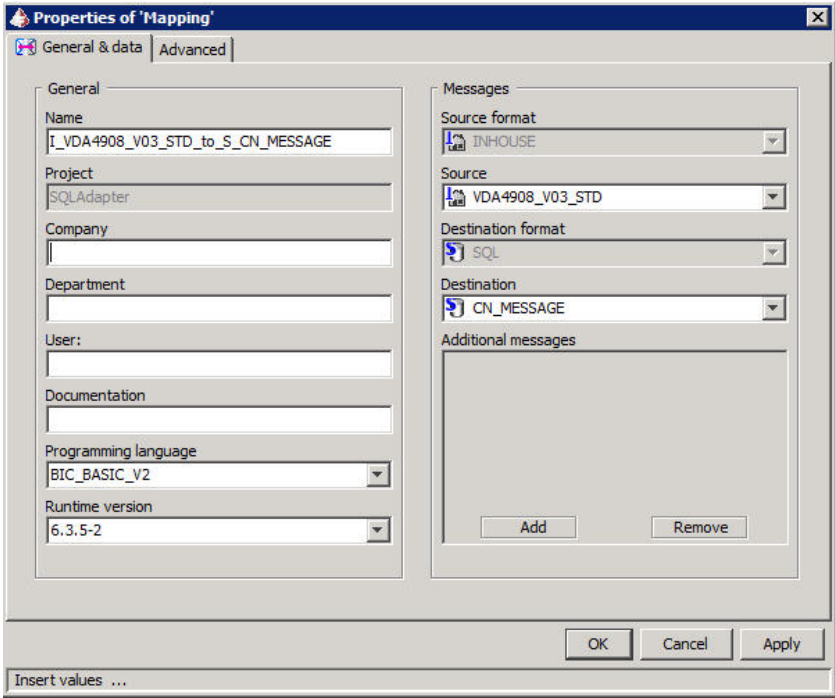
4.2.5 SQL Adapter - Exercise 3

1. After setup of the structure we will start with reading from an inhouse VDA file and writing the content to the SQL database
2. For the writing access we want to create a mapping that maps credit note information of an incoming VDA4908 to the DB structure CN_MESSAGE.
3. Please import the structure [msg_VDA4908_V03_STD.xml](#) of the class material as first step.
4. The next step is to create the mapping which uses the VDA structure as source and the SQL structure as destination message.
5. Please map the VDA fields to the corresponding destination fields.

I Hint: The H2 date format is MM/DD/CCYY and the VAT rate is a formatted amount with two decimals with a delimiter.

6. Test your mapping with the [VDA4908.txt](#) of the class material.

4.2.6 SQL Adapter - Tutorial 3

| Instruction | Illustration |
|---|--|
| 1. Please import the VDA structure VDA4908_V03_STD from the classmaterial. | |
| 2. Create a new mapping using the VDA structure VDA4908_V03_STD as source the SQL structure CN_MESSAGE as destination. |  <p style="text-align: center;">Create a new mapping</p> |
| <p>3. Please create a mapping while just coping the fields from VDA to database.</p> <p>I Hint: The H2 date format is MM/DD/CCYY and the</p> | <p>Record 821</p> <pre>copy THIS:UEBERTR_NR_NEU to CN_HEADER:TRANS_NR; //H2 DB requires the date format MM/DD/CCYY copy formatDate(THIS:UEBERTR_DATUM,"YYMMDD", "MM/DD/CCYY") to CN_HEADER:TRANS_DATE; copy THIS:UST_ID_EMPF to CN_HEADER:VATID_RECEIVER;</pre> |

| Instruction | Illustration |
|--|---|
| <p>VAT rate is a formatted amount with two decimals with a delimiter.</p> | <pre>copy THIS:UST_ID_SENDER to CN_HEADER:VATID_SENDER; copy THIS:STEUERNR to CN_HEADER:FISCALNR; copy THIS:KUNDEN_NUMMER to CN_HEADER:EDI_SENDER; copy THIS:LIEFERANTEN_NR to CN_HEADER:EDI_RECEIVER; //Set status (unread) for reading condition in the next exercises copy "U" to CN_HEADER:STATUS;</pre> <p>Record 822</p> <pre>copy THIS:SUM_UST_BETRAG to CN_HEADER.CN_LINE:VAT_AMOUNT; copy THIS:SUM_GUT_ENDWERT to CN_HEADER.CN_LINE:GROSS_AMOUNT; copy THIS:GUTSCHRIFT_NR to CN_HEADER.CN_LINE:CN_NR;</pre> <p>Record 824</p> <pre>//the VAT rate is a formatted value at the incoming VDA overwrite format(THIS:UST_SATZ,"9900","99.00") to CN_HEADER.CN_LINE:VAT_RATE;</pre> |
| <p>4. Test the mapping with the provided testfile.</p> <p>5. You can double check with the trainer solution</p> <p>I Hint: To check the values in the database, click on the tables in the H2 console and run the auto-generated SQL statement, e.g. <code>SELECT * FROM CN_HEADER</code></p> | |

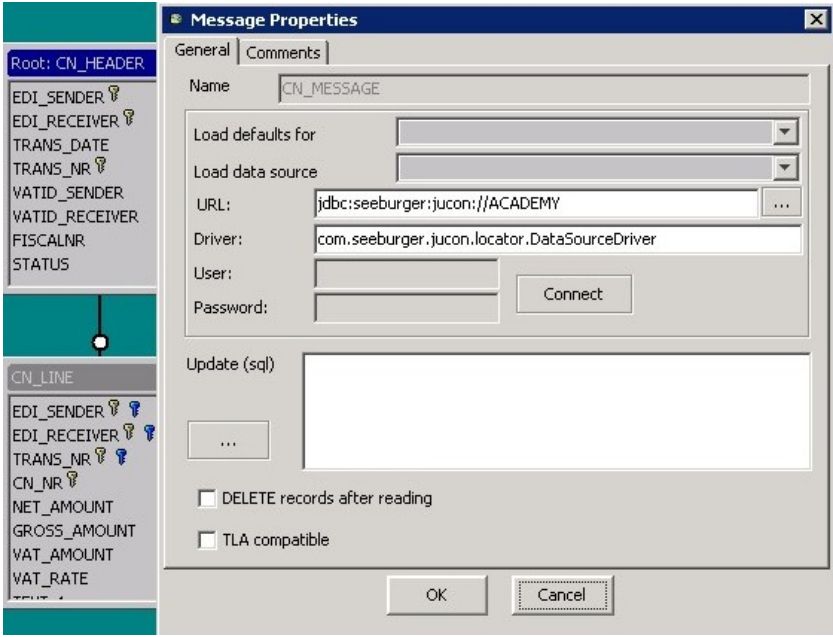
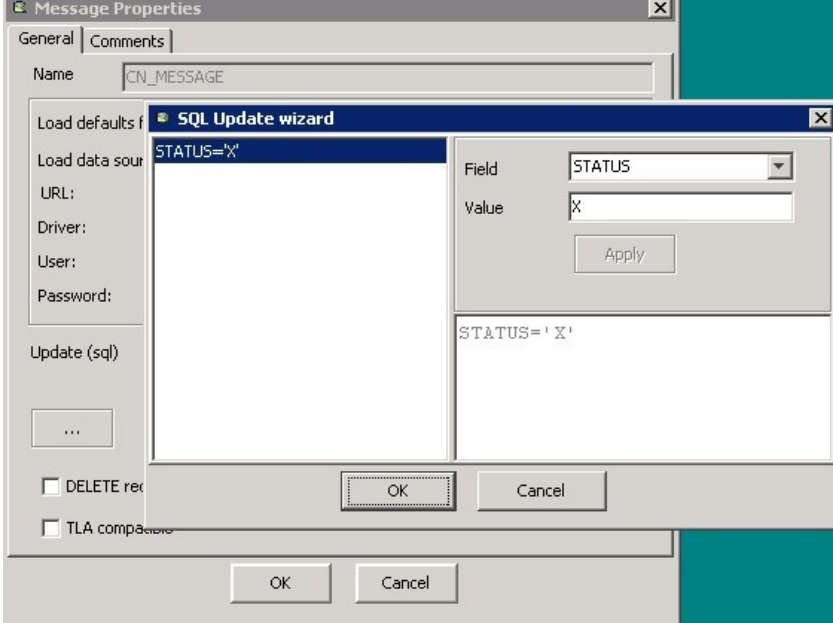
4.2.7 SQL Adapter - Exercise 4

1. For the reading access we want to create a mapping (CN_MESSAGE_TO_CN_REPORT) that maps the information from the database to a CVS file.
2. Please import the message structure [CN_REPORT](#) of the classmaterial to your mapping designer.
3. The next step is to create the mapping and map the information of the database to the corresponding CSV fields.

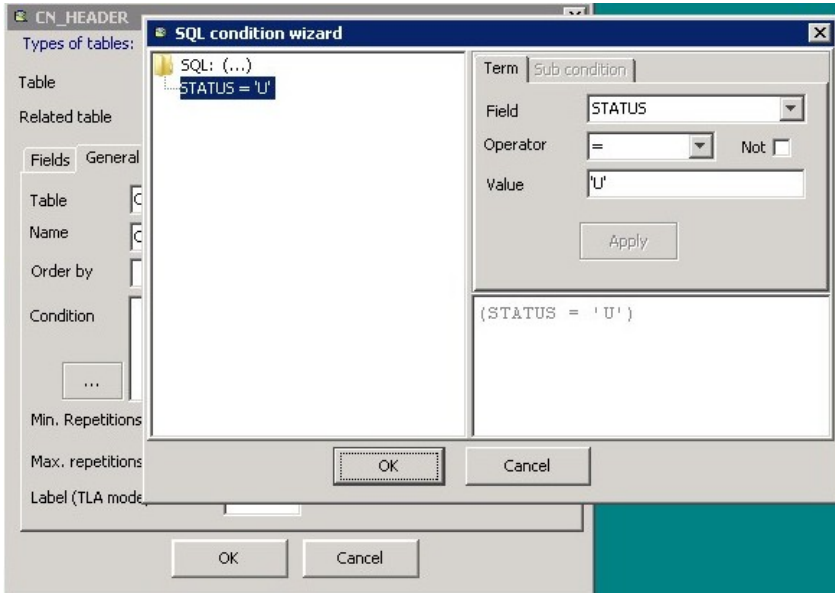
I Use for the receiver and sender name dummy values or leave it blank.

4. Adjust the DB structure to automatically update the DB entries when they are read. Set the field STATUS to X when the entry is read.
5. After this change you have to compile the mapping and run it with the datasource ACADEMY as conversion source.

4.2.8 SQL Adapter - Tutorial 4

| Instruction | Illustration |
|---|---|
| <ol style="list-style-type: none"> 1. Please open the SQL structure with the JDBC wizard. 2. Please open the message properties. |  <p>Open the message properties</p> |
| <ol style="list-style-type: none"> 3. Please insert an update information at the general tab by clicking the button with the three dots. 4. Choose the field STATUS and set the value to X. Confirm your changes with the apply button. |  <p>Insert update information</p> |

| Instruction | Illustration |
|--|--|
| <p>5. Adjust the root table to read only entries from the DB where the STATUS is U.</p> <p>6. To change this property you have to right click at the root table and choose edit table.</p> | <div data-bbox="790 241 1189 645"> </div> <p data-bbox="922 651 1054 674">Edit root table</p> <div data-bbox="582 723 1401 1541"> </div> <p data-bbox="922 1547 1054 1570">Edit root table</p> |

| Instruction | Illustration |
|--|---|
| <p>7. To add a read condition you have to click the button with the three dots.</p> <p>8. The SQL condition wizard will be opened.</p> <p>9. Choose the field STATUS, set the operator to equal and set the value to U.</p> <p>I To insert a new term please right click on the left column and click on add.</p> <p>10. Confirm your settings by clicking the apply button.</p> <p>11. Save the message adjustments and run the mapping at the test environment.</p> <p>12. You can double check with the trainer solution.</p> <p>I Hint: To run the mapping multiple times, you first have to reset the field STATUS to 'U'. Use the following SQL statement in the H2 console: UPDATE CN_HEADER SET STATUS = 'U' WHERE STATUS = 'X' "</p> |  <p style="text-align: center;">Set SQL condition</p> |

4.2.9 SQL Adapter - Exercise 5

1. The last exercise contains the BICMD commands to access the database.
2. Please open den H2 client (how to: see in exercise 2) and copy the following script to the SQL statement text field.

```
CREATE TABLE "PARTNER_ADDRESS" ("SEARCHKEY" VARCHAR2(35), "NAME1"
VARCHAR2(35), "NAME2" VARCHAR2(35), "STRASSE" VARCHAR2(35), "LAND" VARCHAR2(3),
"PLZ" VARCHAR2(9), "ORT" VARCHAR2(35));

Insert into PARTNER_ADDRESS (SEARCHKEY,NAME1,NAME2,STRASSE,LAND,PLZ,ORT)
values ('149733','Muster GmbH',null,'Hauptstraße 13','DE','75015','Bretten');

Insert into PARTNER_ADDRESS (SEARCHKEY,NAME1,NAME2,STRASSE,LAND,PLZ,ORT) values
('474843','Kleinteile Meyer GmbH',null,'Mustergasse 3','DE','75015','Bretten');
```

3. This SQL script will create a new table at the database. This table should be used to transcode the partner ID to the full name of the partner.

4. Please adjust the mapping SQL to CSV to replace the dummy names by the information of the DB table PARTNER_ADDRESS.

4.2.10 SQL Adapter - Tutorial 5

| Instructions | Illustration |
|--|---|
| 1. Please open den H2 client (how to: see in exercise 2) and copy the following script to the SQL statement text field. | <pre>CREATE TABLE "PARTNER_ADDRESS" ("SEARCHKEY" VARCHAR2(35), "NAME1" VARCHAR2(35), "NAME2" VARCHAR2(35), "STRASSE" VARCHAR2(35), "LAND" VARCHAR2(3), "PLZ" VARCHAR2(9), "ORT" VARCHAR2(35)); Insert into PARTNER_ADDRESS (SERCHKEY,NAME1,NAME2,STRASSE,LAND,PLZ,ORT) values ('149733','Muster GmbH',null,'Hauptstraße 13','DE','75015','Bretten'); Insert into PARTNER_ADDRESS (SERCHKEY,NAME1,NAME2,STRASSE,LAND,PLZ,ORT) values ('474843','Kleinteile Meyer GmbH',null,'Mustergasse 3','DE','75015','Bretten');</pre> |
| 2. For this task we can reuse the existing DB connection of the source structure. | |
| 3. You have to add the following mapping code to the additional program NewMapping, because the command reuseSourceDBConnection should be called only once. 4. The column SEARCHKEY contains the partner ID and is used to get the name of the partner. | <pre>openLookupDB("PARTNER_NAME", "SQLDocReader", "PARTNER_ADDRESS", "NAME1", "SEARCHKEY=?");</pre> <p>I The reuseSourceDBConnection() command is depreciated in 652. In future use the connection name <i>SQLDocReader</i> in the database access commands: openMapDB, openLookupDB.</p> |
| 5. Execute the lookup which was opened at the additional program. | <p>Record: CN_HEADER</p> <pre>copy THIS:EDI_SENDER to HEADER.SENDER:SENDER_ID; //execute the DB lookup executeLookupDB("PARTNER_NAME",THIS:EDI_SENDER); copy getLookupFieldDB("PARTNER_NAME", "NAME1") to HEADER.SENDER:NAME; copy THIS:VATID_SENDER to HEADER.SENDER:ID; copy THIS:EDI_RECEIVER to HEADER.RECEIVER:RECEIVER_ID; //execute the DB lookup executeLookupDB("PARTNER_NAME",THIS:EDI_RECEIVER); copy getLookupFieldDB("PARTNER_NAME", "NAME1") to HEADER.RECEIVER:NAME; copy THIS:VATID_RECEIVER to HEADER.RECEIVER:ID; create(HEADER.POSITION.POSITION_HEADER);</pre> |
| 6. You can double check with the trainer solution . | |
| I Hint: To run the mapping multiple times, you first | |

| Instructions | Illustration |
|--|--------------|
| <p>have to reset the field STATUS to 'U'. Use the following SQL statement in the H2 console: UPDATE CN_HEADER SET STATUS = 'U' WHERE STATUS = 'X' “</p> | |

5 Procedures

5.1 Notes

- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____

5.2 Exercises

5.2.1 Procedures - Exercise 1

Local procedures

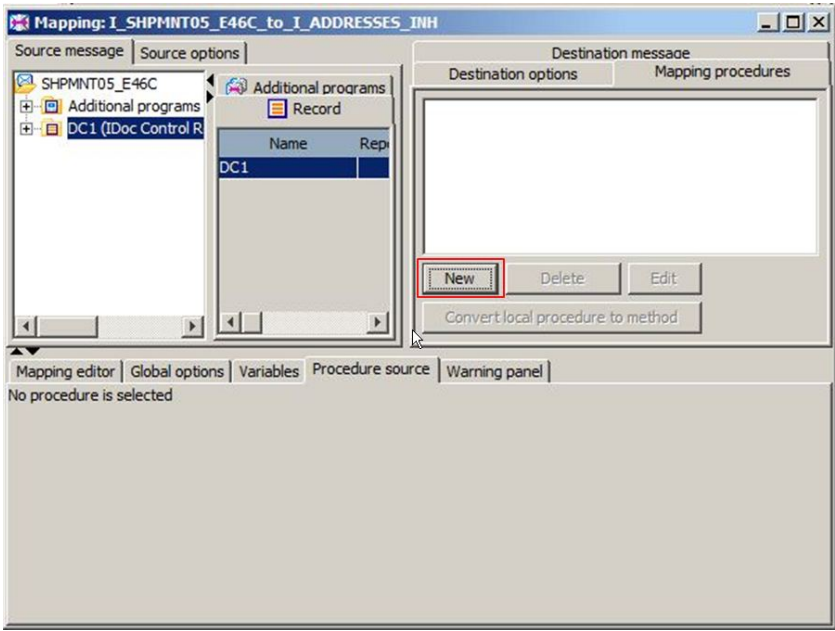
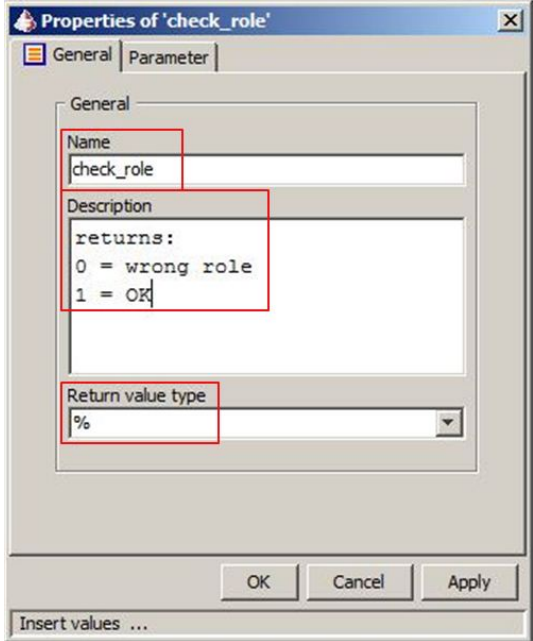
- Create a new project **Training_BIC_6_Advanced_Procedures** within the BIC MD.
- Import the mapping [I_SHPMNT05_E46C_to_I_ADDRESSES_INH_TEMPLATE](#) .
- Create a **new local procedure "check_role"** that checks if the partner role is Sales Organization, sold-to party, ship-to party or carrier.
- The following table shows the qualifiers of the partner roles that should be used in the destination file.

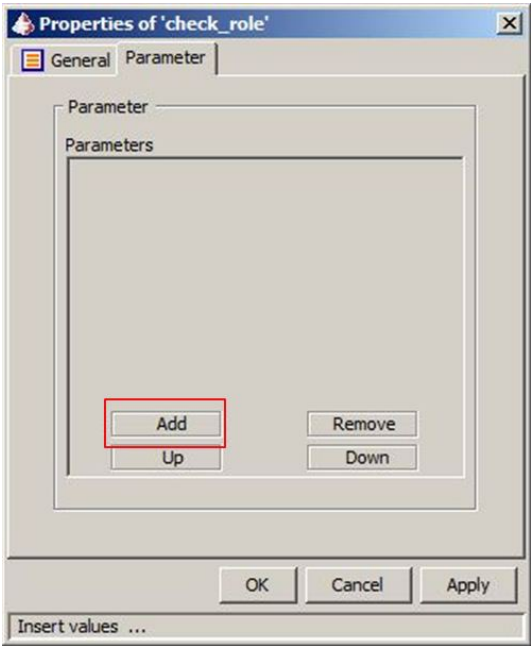
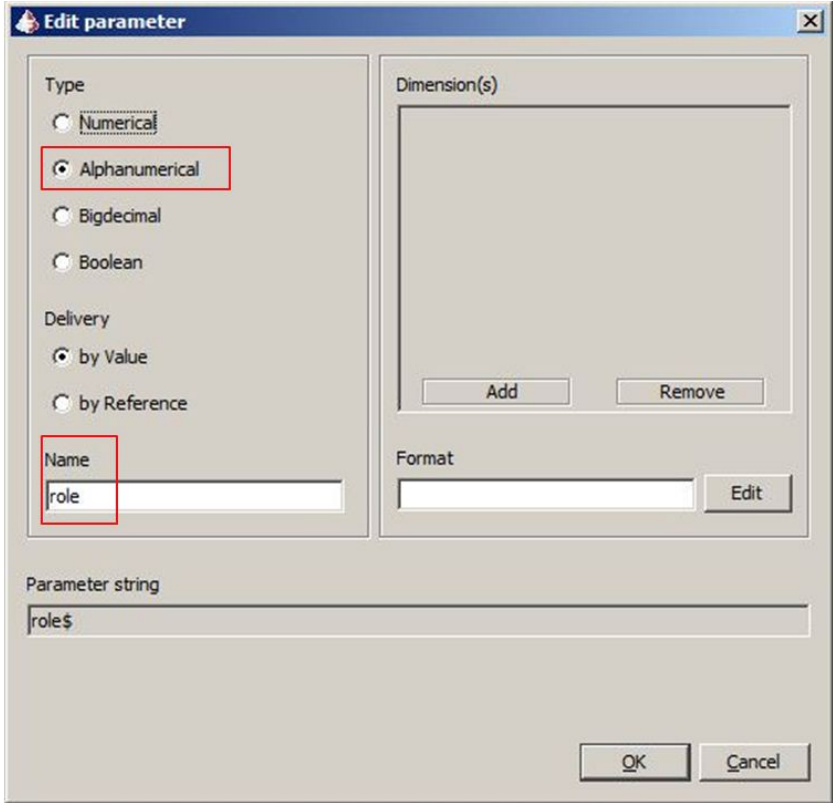
| Qualifier | partner role (eng) |
|-----------|--------------------|
| OSO | Sales Organization |
| AG | sold-to party |
| WE | ship-to party |
| SP | carrier |

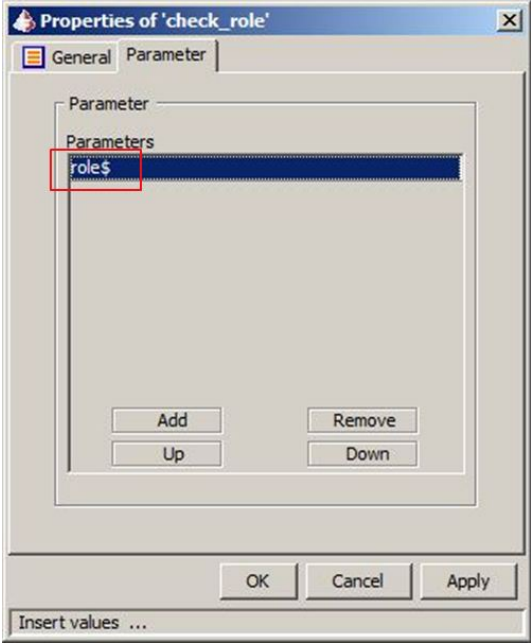
- Go to record program Additional Programms/End of MessageBlock (FileEnd) and change the mapping: local procedure should be used instead of conditional statements.
- Use the [testfile](#) to check the functionality of your procedure.

5.2.2 Procedures - Tutorial 1

| Instruction | Illustration |
|---|--|
| 1. Import the mapping map_I_SHPMNT05_E46C_to_I_ADDRESSES_INH_template.zip.bicmd from the classmaterial. | Check chapter "How to Operate BIC Mapping Designer" in Training BIC 6 Basics if you need further assistance. |
| 2. Go to record program DC1 and define additional variables. | <pre>global procedure_flag%;</pre> |

| Instruction | Illustration |
|---|---|
| <p>3. Go to the "Procedure source" tab and click the "New" button. New window will be opened.</p> |  <p style="text-align: center;">Add new local procedure</p> |
| <p>4. Define procedure properties:</p> <ul style="list-style-type: none"> • Name = check_role • Description = returns: 0 = wrong role; 1 = OK • Return value type = % (numeric) <p>5. Go to parameter tab.</p> |  <p style="text-align: center;">Add new local procedure - general properties</p> |

| Instruction | Illustration |
|---|--|
| <p>6. Use the "Add" Button to add one new line for one new parameter. Doubleclick the new line.</p> |  <p>Properties of 'check_role'</p> <p>General Parameter</p> <p>Parameter</p> <p>Parameters</p> <p>Add Remove Up Down</p> <p>OK Cancel Apply</p> <p>Insert values ...</p> <p>Add new local procedure - parameter properties</p> |
| <p>7. Define parameter details:</p> <ul style="list-style-type: none">• Type = Alphanumeric• Name = role |  <p>Edit parameter</p> <p>Type</p> <p><input type="radio"/> Numerical</p> <p><input checked="" type="radio"/> Alphanumeric</p> <p><input type="radio"/> Bigdecimal</p> <p><input type="radio"/> Boolean</p> <p>Delivery</p> <p><input checked="" type="radio"/> by Value</p> <p><input type="radio"/> by Reference</p> <p>Name</p> <p>role</p> <p>Dimension(s)</p> <p>Add Remove</p> <p>Format</p> <p>Edit</p> <p>Parameter string</p> <p>role\$</p> <p>OK Cancel</p> <p>Add new local procedure - edit parameter</p> |

| Instruction | Illustration |
|---|--|
| |  <p data-bbox="724 884 1257 913">Add new local procedure - parameter properties (role\$)</p> |
| <p data-bbox="172 931 496 1122">8. Add procedure code to "Procedure source" tab, using role information as an input parameter and result Code as an output parameter.</p> | <pre data-bbox="596 947 831 1458"> local result%; result% = 0; if role\$ = "OSO" result% = 1; endif if role\$ = "AG" result% = 1; endif if role\$ = "WE" result% = 1; endif if role\$ = "SP" result% = 1; endif if role\$ = "OSP" result% = 0; endif exitproc(result%); </pre> |
| <p data-bbox="172 1491 485 1581">9. Go to record program FileEnd and change the mapping.</p> | <pre data-bbox="596 1507 1358 1843"> local LOOP1%; for LOOP1% = 1 to COUNT_RM1% procedure_flag% = check_role(PARTNERID\$(LOOP1%)); if procedure_flag% = 1 copy PARTNERID\$(LOOP1%) to ADDRESS.ROLE:NAME; copy NAME\$(LOOP1%) to ADDRESS.NAME:NAME; copy STREET\$(LOOP1%) to ADDRESS.STREET:STREET; copy ZIP\$(LOOP1%) to ADDRESS.ZIP:ZIP; copy CITY\$(LOOP1%) to ADDRESS.CITY:CITY; copy COUNTRY\$(LOOP1%) to ADDRESS.COUNTRY:COUNTRY; endif next </pre> |
| <p data-bbox="161 1872 485 1984">10. Have a look at the destination-file and see how the local procedure worked.</p> | <p data-bbox="547 1872 1302 1895">You can compare your results with the attached trainer solution.</p> |

5.2.3 Procedures - Exercise 2

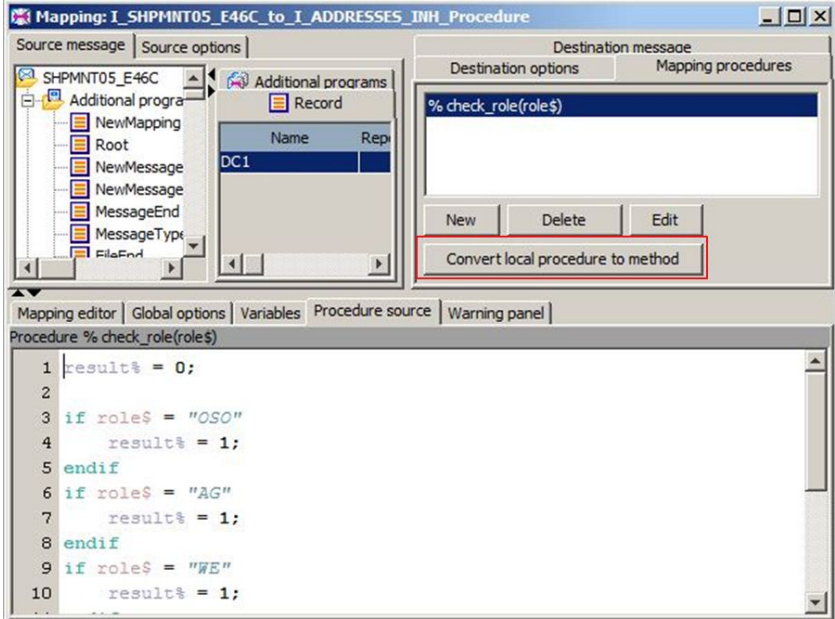

Global procedures

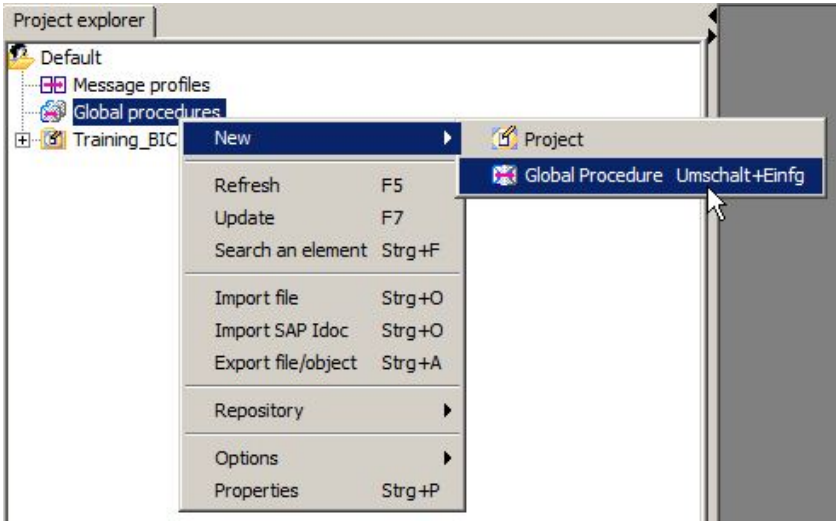
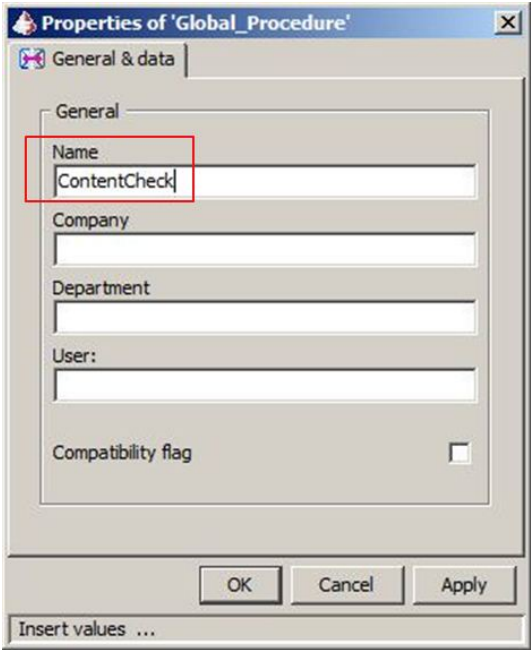
- Use the [mapping with the local procedure](#) changed in [Exercise1](#) (page 39).
- Convert local to global procedure.
- Delete the local procedure.
- What happens at compilation time?

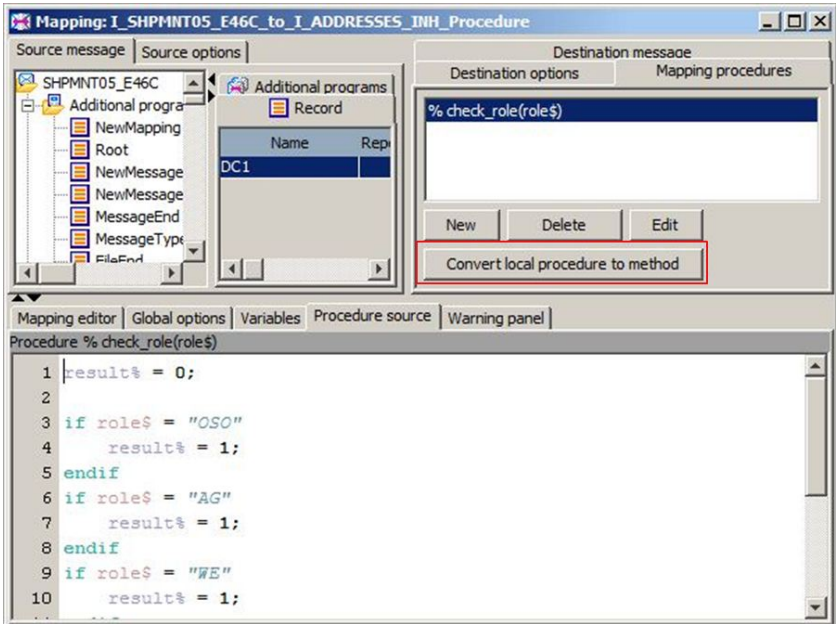

I Tip: Don't forget to compile mapping **and** global procedure!

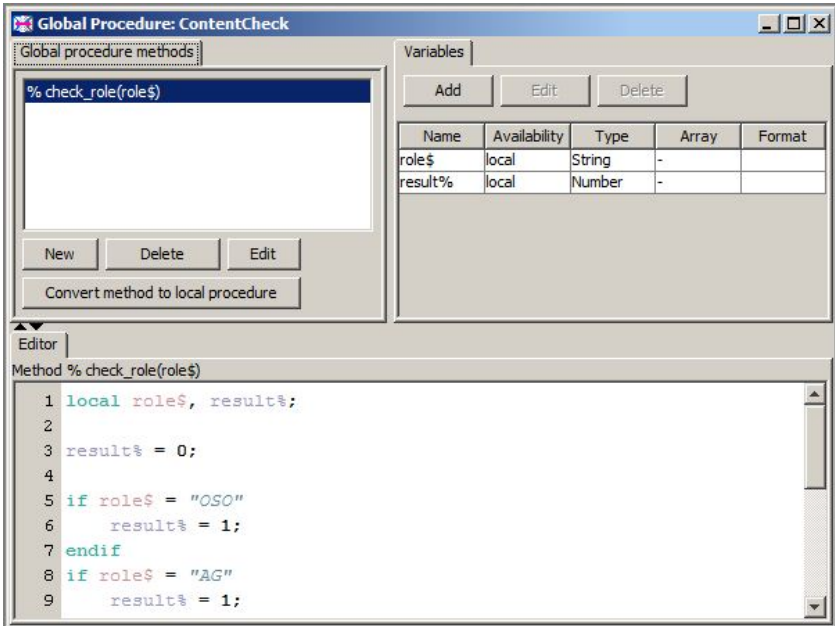
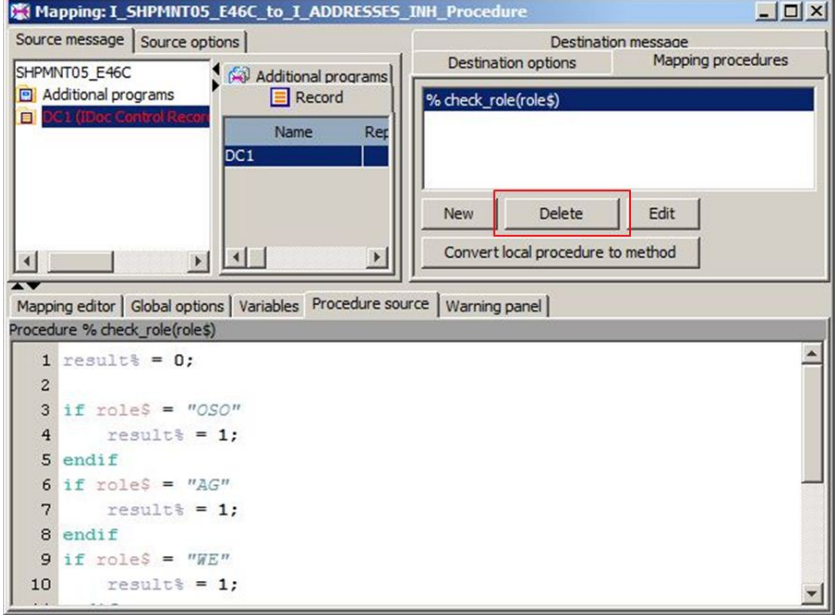
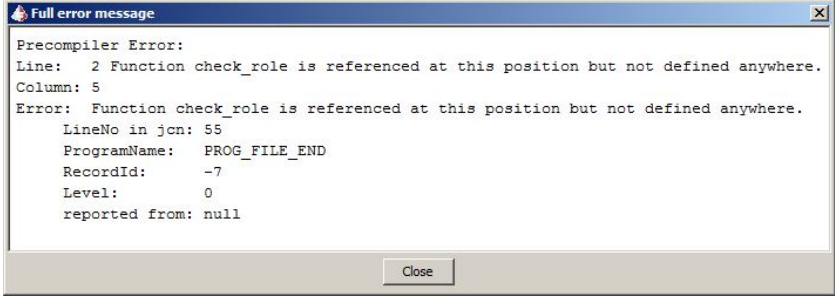
- Correct the mapping and use the [testfile](#) to check if it works.

5.2.4 Procedures - Tutorial 2

| Instruction | Illustration |
|--|---|
| 1. Open the mapping-result of Exercise1 (page 39). | Check chapter "How to Operate BIC Mapping Designer" in Training BIC 6 Basics if you need further assistance. |
| 2. Go to the "Mapping procedures" tab and click on "Convert local procedure to method" button. |  <p>Convert local procedure to method</p> |
| 3. Warning will pop up. A global procedure needs to be created first. |  <p>Warning message</p> |

| Instruction | Illustration |
|---|---|
| <p>4. Go to Project explorer, rightclick on "Global procedures". Click on "New" > "Global Procedure".</p> |  <p>The screenshot shows the 'Project explorer' window with a tree view containing 'Default', 'Message profiles', 'Global procedures', and 'Training_BIC'. A right-click context menu is open over 'Global procedures'. The 'New' option is selected, and a sub-menu is displayed with 'Global Procedure' highlighted. Other options in the sub-menu include 'Project', 'Refresh' (F5), 'Update' (F7), 'Search an element' (Strg+F), 'Import file' (Strg+O), 'Import SAP Idoc' (Strg+O), 'Export file/object' (Strg+A), 'Repository', 'Options', and 'Properties' (Strg+P).</p> <p>Add a new global procedure</p> |
| <p>5. Type in the procedures' name and click OK. 6. Go back to the "Mapping procedures" tab in the mapping.</p> |  <p>The screenshot shows the 'Properties of Global_Procedure' dialog box. The 'General & data' tab is selected. The 'Name' field is highlighted with a red box and contains the text 'ContentCheck'. Other fields include 'Company', 'Department', 'User:', and 'Compatibility flag' (unchecked). At the bottom are 'OK', 'Cancel', and 'Apply' buttons, and an 'Insert values ...' button.</p> <p>Global procedure properties</p> |

| Instruction | Illustration |
|--|---|
| <p>7. Click on "Convert local procedure to method" button.</p> |  <p>The screenshot shows the 'Mapping: I_SHPMNT05_E46C_to_I_ADDRESSES_INH_Procedure' window. The 'Mapping procedures' tab is active, displaying a list of procedures. The 'Convert local procedure to method' button is highlighted with a red rectangle. Below the window, the text 'Convert local procedure to method' is written.</p> |
| <p>8. Choose the global procedure where the local one should be converted to and click OK.</p> |  <p>The screenshot shows the 'Convert local procedure to method' dialog box. The text inside reads: 'You are going to copy the selected local procedure as method of global procedure. Choose global procedures where you want to copy this local procedure as method.' The 'ContentCheck' global procedure is selected and highlighted with a red rectangle. Below the dialog box, the text 'Select global procedure' is written.</p> |

| Instruction | Illustration |
|--|--|
| <p>9. Have a look at the global procedure now: The local procedure is converted to a global procedure method. It is ready for usage now.</p> |  <p>global procedure details</p> |
| <p>10. Delete the local procedure by using the "Delete" button in the "Mapping procedures" tab.</p> |  <p>delete local procedure</p> |
| <p>11. Compile your mapping. An error will be displayed.</p> |  <p>Error message</p> |
| <p>12. Change the procedure call.</p> | <pre>for LOOP1% = 1 to COUNT_RM1% // pick up Output-Parameter of global procedure</pre> |

| Instruction | Illustration |
|---|---|
| | <pre> procedure_flag% = ContentCheck_check_role(PARTNERID \$[LOOP1%]); if procedure_flag% = 1 copy PARTNERID\$[LOOP1%] to ADDRESS.ROLE:NAME; copy NAME\$[LOOP1%] to ADDRESS.NAME:NAME; copy STREET\$[LOOP1%] to ADDRESS.STREET:STREET; copy ZIP\$[LOOP1%] to ADDRESS.ZIP:ZIP; copy CITY\$[LOOP1%] to ADDRESS.CITY:CITY; copy COUNTRY\$[LOOP1%] to ADDRESS.COUNTRY:COUNTRY; endif next </pre> |
| <p>13. Check the procedures content. Please use only the variables that are really necessary.</p> | <pre> local role\$, result%; result% = 0; if role\$ = "OSO" result% = 1; endif if role\$ = "AG" result% = 1; endif if role\$ = "WE" result% = 1; endif if role\$ = "SP" result% = 1; endif if role\$ = "OSP" result% = 0; endif exitproc(result%); </pre> |
| <p>14. Have a look at the destination-file and see how the "global procedure" worked.</p> | <p>You can compare your results with the attached trainer solution:</p> <ul style="list-style-type: none"> • mapping • procedure |

6 Arrays and Hashmaps

6.1 Notes

- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____

6.2 Exercises

6.2.1 Arrays/Hashmaps - Exercises Overview

On the next slides/pages you will find 8 Exercises with Array and Hashmap samples.

- You will find 4 exercises with simple examples which are just to understand the handling with arrays and hasmaps.
- You will find 4 exercises with EDI examples where you can use arrays and hasmaps.

Together with your trainer you will decide which exercises you will do in more detail. You can start with the simple ones and switch to the EDI exercises later on.

6.2.2 Arrays/Hashmaps - Exercise 1

One-dimension array

- Create a new project **Training_BIC_6_Advanced_ArrayHashmap** within the BIC MD.
- Create a **noMsg to noMsg mapping “array_1_dimension”** within the BIC MD.
- Define a **one-dimension array** and fill it with the seven original titles of the Harry Potter books as displayed in the table below:

| volume number | title (ger) | ISBN (ger) | title original | ISBN (eng) | year of publication | author |
|---------------|--|--------------------|---|--------------------|---------------------|--------------|
| 1 | Harry Potter und der Stein der Weisen | ISBN 3-551-55200-2 | Harry Potter and the Philosopher's Stone | ISBN 0-7475-7360-3 | 1997 | J.K. Rowling |
| 2 | Harry Potter und die Kammer des Schreckens | ISBN 3-551-55209-6 | Harry Potter and the Chamber of Secrets | ISBN 0-7475-3849-2 | 1998 | J.K. Rowling |
| 3 | Harry Potter und der Gefangene von Askaban | ISBN 3-551-55210-X | Harry Potter and the Prisoner of Azkaban | ISBN 0-7475-4215-5 | 1999 | J.K. Rowling |
| 4 | Harry Potter und der Feuerkelch | ISBN 3-551-55253-3 | Harry Potter and the Goblet of Fire | ISBN 0-7475-4624-X | 2000 | J.K. Rowling |
| 5 | Harry Potter und der Orden des Phönix | ISBN 3-551-55297-5 | Harry Potter and the Order of the Phoenix | ISBN 0-7475-5100-6 | 2003 | J.K. Rowling |
| 6 | Harry Potter und der Halbblutprinz | ISBN 3-551-55600-8 | Harry Potter and the Half-Blood Prince | ISBN 0-7475-8108-8 | 2005 | J.K. Rowling |
| 7 | Harry Potter und die Heiligtümer des Todes | ISBN 3-551-55700-4 | Harry Potter and the Deathly Hallows | ISBN 0-7475-9106-7 | 2007 | J.K. Rowling |

- Use a loop to **trace the array data and the volume number** to the log file

6.2.3 Arrays/Hashmaps - Tutorial 1

| Instruction | Illustration |
|--|--|
| 1. Create a new No Message to No Message mapping. | Check chapter "How to Operate BIC Mapping Designer" in Training BIC 6 Basics if you need further assistance. |
| 2. Go to record program DEFAULT. | <pre> // define variables, set counter local size%, i%, volume%; volume% = 1; // define array (first index is 0) // it's not possible to use a variable for the size local harrypotter\$[7]; // fill the array harrypotter\$[0] = "Harry Potter and the Philosopher's Stone"; harrypotter\$[1] = "Harry Potter and the Chamber of Secrets"; harrypotter\$[2] = "Harry Potter and the Prisoner of Azkaban"; harrypotter\$[3] = "Harry Potter and the Goblet of Fire"; harrypotter\$[4] = "Harry Potter and the Order of the Phoenix"; harrypotter\$[5] = "Harry Potter and the Half-Blood Prince"; harrypotter\$[6] = "Harry Potter and the Deathly Hallows"; //get array size for defining how often the for loop must run //The getSize function's output is 7, but as the index starts //with 0, the highest index is 6! size% = getSize(harrypotter\$[]) - 1; traceln("Max array index: " & size%); // get array data for i% = 0 to size% // Loop effecting the read out of the elements of a one-dimensional traceln("Harry Potter Volume "& volume% & " is \"& harrypotter\$[i%] & "\"."); volume% = volume% + 1; next </pre> |
| 3. Have a look at the trace-file and find out how the one-dimensional array works. | <p>You can compare your results with the attached trainer solution.</p> <p>Traceline should look like the following lines:</p> <pre> Harry Potter Volume 1 is "Harry Potter and the Philosopher's Stone". Harry Potter Volume 2 is "Harry Potter and the Chamber of Secrets". Harry Potter Volume 3 is "Harry Potter and the Prisoner of Azkaban". Harry Potter Volume 4 is "Harry Potter and the Goblet of Fire". Harry Potter Volume 5 is "Harry Potter and the Order of the Phoenix". Harry Potter Volume 6 is "Harry Potter and the Half-Blood Prince". Harry Potter Volume 7 is "Harry Potter and the Deathly Hallows". </pre> |

6.2.4 Arrays/Hashmaps - Exercise 2

One-dimension array

- Import the message structures [msg_SHPMNT05_E46C.xml](#) and [msg_ADDRESSES_INH.xml](#) and create a **mapping** where „SHPMNT05_E46C“ is the source and „ADDRESS_INH“ is the destination.
- Go to DC1-segment and define a **one-dimension array** for partner address data (role, name, street, postal code, city and country).
- Write data from RM1-segment to the array. The following table shows the fields of the SHPMNT-Idoc (RM1) that contains the necessary information.

| field name | description |
|------------|----------------------------|
| PARNTER_Q | role of the partner |
| NAME1 | name of the partner |
| STREET1 | address data (street) |
| POSTL_COD1 | address data (postal code) |
| CITY1 | address data (city) |
| COUNTRY1 | country |

- Go to record program Additional Programs/End of Message Block (FileEnd) and fill the destination file. Only Sales Organization, sold-to party, ship-to party and carrier should be used.
- The following table shows the qualifiers of the partnerroles that should be used in the destination file.

| Qualifier | partner role (eng) |
|-----------|--------------------|
| OSO | Sales Organization |
| AG | sold-to party |
| WE | ship-to party |
| SP | carrier |

- Use the [testfile](#) (SHPMNT05.idoc) to see the results.

6.2.5 Arrays/Hashmaps - Tutorial 2

| Instruction | Illustration |
|--|---|
| <ol style="list-style-type: none"> 1. Import the message structure msg_SHPMNT05_E46C.xml from the classmaterial. 2. Import the message structure msg_ADDRESS_INH.xml from the classmaterial. 3. Create a new mapping (source = "SHPMNT05_E46C", destination = "ADDRESS_INH"). | Check chapter "How to Operate BIC Mapping Designer" in Training BIC 6 Basics if you need further assistance. |
| <ol style="list-style-type: none"> 4. Go to record program DC1 and define variables. | <pre>global PARTNERID\$[50], NAME\$[50], STREET\$[50], NO\$[50]; global ZIP\$[50], CITY\$[50], COUNTRY\$[50];</pre> |

| Instruction | Illustration |
|---|--|
| | <pre>global COUNT_RM1%, LOOP1%;</pre> |
| <p>5. Go to record program RM1, set the counter and fill the one-dimension array.</p> | <pre>COUNT_RM1% = COUNT_RM1% + 1; copy DC1.T20.L20.RM1:PARTNER_Q to PARTNERID\$(COUNT_RM1%); copy DC1.T20.L20.RM1:NAME1 to NAME\$(COUNT_RM1%); copy DC1.T20.L20.RM1:STREET1 to STREET\$(COUNT_RM1%); copy DC1.T20.L20.RM1:POSTL_COD1 to ZIP\$(COUNT_RM1%); copy DC1.T20.L20.RM1:CITY1 to CITY\$(COUNT_RM1%); copy DC1.T20.L20.RM1:COUNTRY1 to COUNTRY\$(COUNT_RM1%);</pre> |
| <p>6. Go to record program FileEnd and write the array data to the destination file.</p> | <pre>for LOOP1% = 1 to COUNT_RM1% if PARTNERID\$(LOOP1%) = "OSO" // DE: Verkaufsstelle; EN: Sales Organization copy "Sales Organization" to ADDRESS.ROLE:NAME; copy NAME\$(LOOP1%) to ADDRESS.NAME:NAME; copy STREET\$(LOOP1%) to ADDRESS.STREET:STREET; copy ZIP\$(LOOP1%) to ADDRESS.ZIP:ZIP; copy CITY\$(LOOP1%) to ADDRESS.CITY:CITY; copy COUNTRY\$(LOOP1%) to ADDRESS.COUNTRY:COUNTRY; endif if PARTNERID\$(LOOP1%) = "AG" // DE: Auftraggeber, EN: sold-to party copy "sold-to party" to ADDRESS.ROLE:NAME; copy NAME\$(LOOP1%) to ADDRESS.NAME:NAME; copy STREET\$(LOOP1%) to ADDRESS.STREET:STREET; copy ZIP\$(LOOP1%) to ADDRESS.ZIP:ZIP; copy CITY\$(LOOP1%) to ADDRESS.CITY:CITY; copy COUNTRY\$(LOOP1%) to ADDRESS.COUNTRY:COUNTRY; endif if PARTNERID\$(LOOP1%) = "WE" // DE: Warenempfänger; EN: ship-to party copy "ship-to party" to ADDRESS.ROLE:NAME; copy NAME\$(LOOP1%) to ADDRESS.NAME:NAME; copy STREET\$(LOOP1%) to ADDRESS.STREET:STREET; copy ZIP\$(LOOP1%) to ADDRESS.ZIP:ZIP; copy CITY\$(LOOP1%) to ADDRESS.CITY:CITY; copy COUNTRY\$(LOOP1%) to ADDRESS.COUNTRY:COUNTRY; endif if PARTNERID\$(LOOP1%) = "SP" // DE: Spediteur; EN: Carrier copy "Carrier" to ADDRESS.ROLE:NAME; copy NAME\$(LOOP1%) to ADDRESS.NAME:NAME; copy STREET\$(LOOP1%) to ADDRESS.STREET:STREET; copy ZIP\$(LOOP1%) to ADDRESS.ZIP:ZIP; copy CITY\$(LOOP1%) to ADDRESS.CITY:CITY; copy COUNTRY\$(LOOP1%) to ADDRESS.COUNTRY:COUNTRY; endif next</pre> |
| <p>7. Have a look at the destination-file and see how the "one-dimensional array" worked.</p> | <p>You can compare your results with the attached trainer solution.</p> |

6.2.6 Arrays/Hashmaps - Exercise 3

two-dimension array

- Create a **noMsg to noMsg** mapping “**array_2_dimension**” within the BIC MD.
- Define a **two-dimension array** and fill it with the seven original titles of the Harry Potter books and corresponding ISBN number (refer to [Exercise 1](#) (page 49) for content information).
- Use a loop to trace the **array data (Title and ISBN)** and the **volume number** to the log file

6.2.7 Arrays/Hashmaps - Tutorial 3

| Instruction | Illustration |
|---|---|
| 1. Create a new No Message to No Message mapping. | Check chapter "How to Operate BIC Mapping Designer" in Training BIC 6 Basics if you need further assistance. |
| 2. Go to record program DEFAULT. | <pre> // define variables, set counter local size%, i%, volume%; volume% = 1; // define array (first index is always 0) // it's not possible to use a variable for the size local harrypotter\$[7][2]; // fill the array // first token is the volume [0]-[6] // second token is the name of the book [0] and the ISBN [1] harrypotter\$[0][0] = "Harry Potter and the Philosopher's Stone"; harrypotter\$[0][1] = "ISBN 0-7475-7360-3"; harrypotter\$[1][0] = "Harry Potter and the Chamber of Secrets"; harrypotter\$[1][1] = "ISBN 0-7475-3849-2"; harrypotter\$[2][0] = "Harry Potter and the Prisoner of Azkaban"; harrypotter\$[2][1] = "ISBN 0-7475-4215-5"; harrypotter\$[3][0] = "Harry Potter and the Goblet of Fire"; harrypotter\$[3][1] = "ISBN 0-7475-4624-X"; harrypotter\$[4][0] = "Harry Potter and the Order of the Phoenix"; harrypotter\$[4][1] = "ISBN 0-7475-5100-6"; harrypotter\$[5][0] = "Harry Potter and the Half-Blood Prince"; harrypotter\$[5][1] = "ISBN 0-7475-8108-8"; harrypotter\$[6][0] = "Harry Potter and the Deathly Hallows"; harrypotter\$[6][1] = "ISBN 0-7475-9106-7"; //get array size for defining how often the for loop must run //The getSize function's output is 7, but as the index starts //with 0, the highest index is 6! size% = getSize(harrypotter\$[]) - 1; traceln("Max array index: " & size%); // get array data for i% = 0 to size% // Loop effecting the read out of the elements of a two-dimensional traceln("Harry Potter Volume " & volume% & " is \" & harrypotter\$[i%][0] & \" with \" & harrypotter\$[i%][1] & "."); volume% = volume% + 1; next // Excercise 3 - some additional info </pre> |
| Training BIC 6 Advanced - Handout - 6.5.2 | <pre> // The size of the first dimension traceln("getsize(harrypotter\$[]): " & getsize(harrypotter\$[])); // The size of the second dimension of the 3rd occurrence of the first dimension (0-based) traceln("getsize(harrypotter\$[2]): " & getsize(harrypotter\$[2])); </pre> |

6.2.8 Arrays/Hashmaps - Exercise 4

two-dimension array

- Import the message structures [msg_SHPMNT05_E46C.xml](#) and [msg_ADDRESSES_INH.xml](#) and create a **mapping** where „SHPMNT05_E46C“ is the source and „ADDRESS_INH“ is the destination.
- Go to DC1-segment and define a **two-dimension array** for address data and store the information of the RM1 segment into the array.
- Go to record program Additional Programs/End of Message Block (FileEnd) and fill the destination file for the roles Sales Organization, sold-to party, ship-to party and carrier.
- Use the [testfile](#) to see the results.

I What benefits can you see after using one- and two-dimensional arrays?

6.2.9 Arrays/Hashmaps - Tutorial 4

| Instruction | Illustration |
|--|--|
| 1. Import the message structure msg_SHPMNT05_E46C.xml from the classmaterial. 2. Import the message structure msg_ADDRESS_INH.xml from the classmaterial. 3. Create a new mapping (source = "SHPMNT05_E46C", destination = "ADDRESS_INH"). | Check chapter "How to Operate BIC Mapping Designer" in Training BIC 6 Basics if you need further assistance. |
| 4. Go to record program DC1 and define variables. | <pre>// define 2-dimension array global ADDRESS\$[50][6]; //define counter global COUNT_RM1%, LOOP1%; global c_PARTNER_Q%, c_NAME1%, c_STREET1%, c_POSTL_COD1%, c_City1%, c_COUNTRY1%; // initiate counter c_PARTNER_Q% = 0; c_NAME1% = 1; c_STREET1% = 2; c_POSTL_COD1% = 3; c_CITY1% = 4; c_COUNTRY1% = 5;</pre> |
| 5. Go to record program RM1, set the counter and fill the one-dimension array. | <pre>COUNT_RM1% = COUNT_RM1% + 1; copy DC1.T20.L20.RM1:PARTNER_Q to ADDRESS\$[COUNT_RM1%] [c_PARTNER_Q%];</pre> |

| Instruction | Illustration |
|---|---|
| | <pre> copy DC1.T20.L20.RM1:NAME1 to ADDRESS\$(COUNT_RM1% [c_NAME1%]; copy DC1.T20.L20.RM1:STREET1 to ADDRESS\$(COUNT_RM1% [c_STREET1%]; copy DC1.T20.L20.RM1:POSTL_COD1 to ADDRESS\$(COUNT_RM1% [c_POSTL_COD1%]; copy DC1.T20.L20.RM1:CITY1 to ADDRESS\$(COUNT_RM1% [c_CITY1%]; copy DC1.T20.L20.RM1:COUNTRY1 to ADDRESS\$(COUNT_RM1% [c_COUNTRY1%]; </pre> |
| <p>6. Go to record program FileEnd and write the array data to the destination file.</p> | <pre> for LOOP1% = 1 to COUNT_RM1% if ADDRESS\$(LOOP1%)[0] = "OSO" // DE: Verkaufsstelle; EN: Sales Organization copy "Sales Organization" to ADDRESS.ROLE:NAME; copy ADDRESS\$(LOOP1%)[1] to ADDRESS.NAME:NAME; copy ADDRESS\$(LOOP1%)[2] to ADDRESS.STREET:STREET; copy ADDRESS\$(LOOP1%)[3] to ADDRESS.ZIP:ZIP; copy ADDRESS\$(LOOP1%)[4] to ADDRESS.CITY:CITY; copy ADDRESS\$(LOOP1%)[5] to ADDRESS.COUNTRY:COUNTRY; endif if ADDRESS\$(LOOP1%)[0] = "AG" // DE: Auftraggeber, EN: sold-to party copy "sold-to party" to ADDRESS.ROLE:NAME; copy ADDRESS\$(LOOP1%)[1] to ADDRESS.NAME:NAME; copy ADDRESS\$(LOOP1%)[2] to ADDRESS.STREET:STREET; copy ADDRESS\$(LOOP1%)[3] to ADDRESS.ZIP:ZIP; copy ADDRESS\$(LOOP1%)[4] to ADDRESS.CITY:CITY; copy ADDRESS\$(LOOP1%)[5] to ADDRESS.COUNTRY:COUNTRY; endif if ADDRESS\$(LOOP1%)[0] = "WE" // DE: Warenempfänger; EN: ship-to party copy "ship-to party" to ADDRESS.ROLE:NAME; copy ADDRESS\$(LOOP1%)[1] to ADDRESS.NAME:NAME; copy ADDRESS\$(LOOP1%)[2] to ADDRESS.STREET:STREET; copy ADDRESS\$(LOOP1%)[3] to ADDRESS.ZIP:ZIP; copy ADDRESS\$(LOOP1%)[4] to ADDRESS.CITY:CITY; copy ADDRESS\$(LOOP1%)[5] to ADDRESS.COUNTRY:COUNTRY; endif if ADDRESS\$(LOOP1%)[0] = "SP" // DE: Spediteur; EN:Carrier copy "Carrier" to ADDRESS.ROLE:NAME; copy ADDRESS\$(LOOP1%)[1] to ADDRESS.NAME:NAME; copy ADDRESS\$(LOOP1%)[2] to ADDRESS.STREET:STREET; copy ADDRESS\$(LOOP1%)[3] to ADDRESS.ZIP:ZIP; copy ADDRESS\$(LOOP1%)[4] to ADDRESS.CITY:CITY; copy ADDRESS\$(LOOP1%)[5] to ADDRESS.COUNTRY:COUNTRY; endif endif next </pre> |
| <p>7. Have a look at the destination-file and see how the "two-dimensional array" worked.</p> | <p>You can compare your results with the attached trainer solution.</p> |

6.2.10 Arrays/Hashmaps - Exercise 5

hashmap

- Create a noMsg to noMsg mapping "hashmap_1_dimension" within the BIC MD.

- Define a **one-dimension hashmap** and fill it with the seven original titles of the Harry Potter books (refer to [Exercise 1](#) (page 49) for content information). Use the corresponding ISBN number as identifier (key).
- Use a loop to trace the hashmap data **for one ISBN number** to the log file.

6.2.11 Arrays/Hashmaps - Tutorial 5

| Instruction | Illustration |
|--|---|
| 1. Create a new No Message to No Message mapping. | Check chapter "How to Operate BIC Mapping Designer" in Training BIC 6 Basics if you need further assistance. |
| 2. Go to record program DEFAULT. | <pre> local harrypotter\${}; //define hashmap // fill the hashmap harrypotter{"0-7475-7360-3"} = "Harry Potter and the Philosopher's Stone"; harrypotter{"0-7475-3849-2"} = "Harry Potter and the Chamber of Secrets"; harrypotter{"0-7475-4215-5"} = "Harry Potter and the Prisoner of Azkaban"; harrypotter{"0-7475-4624-X"} = "Harry Potter and the Goblet of Fire"; harrypotter{"0-7475-5100-6"} = "Harry Potter and the Order of the Phoenix"; harrypotter{"0-7475-8108-8"} = "Harry Potter and the Half- Blood Prince"; harrypotter{"0-7475-9106-7"} = "Harry Potter and the Deathly Hallows"; // get hashmap data traceln("Title of Harry Potter book with ISBN 0-7475-4215-5 is " & harrypotter{"0-7475-4215-5"}); </pre> |
| 3. Have a look at the trace-file and find out how the hashmap works. | <p>You can compare your results with the attached trainer solution.</p> <p>Traceline should look like the following lines:</p> <pre> Title of Harry Potter book with ISBN 0-7475-4215-5 is Harry Potter and the Prisoner of Azkaban </pre> |

6.2.12 Arrays/Hashmaps - Exercise 6

hashmap

- Import the message structures [msg_SHPMNT05_E46C.xml](#) and [msg_ADDRESSES_INH.xml](#) and create a **mapping** where „SHPMNT05_E46C“ is the source and „ADDRESS_INH“ is the destination.
- Go to DC1-segment and define a **two-dimension hashmap** for address data and store the information of the RM1 segment into the hashmap. Identifier (key) should be the field name.
- Go to record program Additional Programs/End of Message Block (FileEnd) and fill the destination file for the roles Sales Organization, sold-to party, ship-to party and carrier.
- Use the [testfile](#) to see the results.

I What benefits can you see after using hashmap?

6.2.13 Arrays/Hashmaps - Tutorial 6

| Instruction | Illustration |
|--|--|
| 1. Import the message structure msg_SHPMNT05_E46C.xml from the classmaterial. 2. Import the message structure msg_ADDRESS_INH.xml from the classmaterial. 3. Create a new mapping (source = "SHPMNT05_E46C", destination = "ADDRESS_INH"). | Check chapter "How to Operate BIC Mapping Designer" in Training BIC 6 Basics if you need further assistance. |
| 4. Go to record program DC1 and define variables. | <pre>// define 2-dimension hashmap global ADDRESS\${}; // define variables and counter global PARTNERID\${50}, NAME\${50}, STREET\${50}, ZIP\${50}, CITY\${50}, COUNTRY\${50}; global COUNT_RM1%, LOOP1%;</pre> |
| 5. Go to record program RM1, set the counter and fill the one-dimension array. | <pre>COUNT_RM1% = COUNT_RM1% + 1; copy DC1.T20.L20.RM1:PARTNER_Q to ADDRESS\${"address" & COUNT_RM1%}{ "Partner"}; copy DC1.T20.L20.RM1:NAME1 to ADDRESS\${"address" & COUNT_RM1%}{ "Name"}; copy DC1.T20.L20.RM1:STREET1 to ADDRESS\${"address" & COUNT_RM1%}{ "Street"}; copy DC1.T20.L20.RM1:POSTL_COD1 to ADDRESS\${"address" & COUNT_RM1%}{ "Postl_cod"}; copy DC1.T20.L20.RM1:CITY1 to ADDRESS\${"address" & COUNT_RM1%}{ "City"}; copy DC1.T20.L20.RM1:COUNTRY1 to ADDRESS\${"address" & COUNT_RM1%}{ "Country"};</pre> |
| 6. Go to record program FileEnd and write the array data to the destination file. | <pre>for LOOP1% = 1 to COUNT_RM1% if ADDRESS\${"address" & LOOP1%}{ "Partner"} = "OSO" // DE: Verkaufsstelle; EN: Sales Organization copy "Sales Organization" to ADDRESS.ROLE:NAME; copy ADDRESS\${"address" & LOOP1%}{ "Name"} to ADDRESS.NAME:NAME; copy ADDRESS\${"address" & LOOP1%}{ "Street"} to ADDRESS.STREET:STREET; copy ADDRESS\${"address" & LOOP1%}{ "Postl_cod"} to ADDRESS.ZIP:ZIP; copy ADDRESS\${"address" & LOOP1%}{ "City"} to ADDRESS.CITY:CITY; copy ADDRESS\${"address" & LOOP1%}{ "Country"} to ADDRESS.COUNTRY:COUNTRY; endif if ADDRESS\${"address" & LOOP1%}{ "Partner"} = "AG" // DE: Auftraggeber, EN: sold-to party</pre> |

| Instruction | Illustration |
|--|--|
| | <pre> copy "sold-to party" to ADDRESS.ROLE:NAME; copy ADDRESS\${"address" & LOOP1%}{ "Name" } to ADDRESS.NAME:NAME; copy ADDRESS\${"address" & LOOP1%}{ "Street" } to ADDRESS.STREET:STREET; copy ADDRESS\${"address" & LOOP1%}{ "Postl_cod" } to ADDRESS.ZIP:ZIP; copy ADDRESS\${"address" & LOOP1%}{ "City" } to ADDRESS.CITY:CITY; copy ADDRESS\${"address" & LOOP1%}{ "Country" } to ADDRESS.COUNTRY:COUNTRY; endif if ADDRESS\${"address" & LOOP1%}{ "Partner" } = "WE" // DE: Warenempfänger; EN: ship-to party copy "ship-to party" to ADDRESS.ROLE:NAME; copy ADDRESS\${"address" & LOOP1%}{ "Name" } to ADDRESS.NAME:NAME; copy ADDRESS\${"address" & LOOP1%}{ "Street" } to ADDRESS.STREET:STREET; copy ADDRESS\${"address" & LOOP1%}{ "Postl_cod" } to ADDRESS.ZIP:ZIP; copy ADDRESS\${"address" & LOOP1%}{ "City" } to ADDRESS.CITY:CITY; copy ADDRESS\${"address" & LOOP1%}{ "Country" } to ADDRESS.COUNTRY:COUNTRY; endif if ADDRESS\${"address" & LOOP1%}{ "Partner" } = "SP" // DE: Spediteur; EN: Carrier copy "Carrier" to ADDRESS.ROLE:NAME; copy ADDRESS\${"address" & LOOP1%}{ "Name" } to ADDRESS.NAME:NAME; copy ADDRESS\${"address" & LOOP1%}{ "Street" } to ADDRESS.STREET:STREET; copy ADDRESS\${"address" & LOOP1%}{ "Postl_cod" } to ADDRESS.ZIP:ZIP; copy ADDRESS\${"address" & LOOP1%}{ "City" } to ADDRESS.CITY:CITY; copy ADDRESS\${"address" & LOOP1%}{ "Country" } to ADDRESS.COUNTRY:COUNTRY; endif next </pre> |
| 7. Have a look at the destination-file and see how the "hashmap" worked. | You can compare your results with the attached trainer solution . |

6.2.14 Arrays/Hashmaps - Exercise 7

Mixed type

- Create a **noMsg to noMsg** mapping “**mixedtype_2_dimension**” within the BIC MD.
- Define a **mixed type hashmap** and fill it with the seven original titles of the Harry Potter books and the year of publication (refer to [Exercise 1](#) (page 49) for content information). Use the volume for the array and title, published and ISBN as identifier for the hasmap.
- Use a loop to trace the mixed type data (title and year of publication) **for one ISBN number** to the log file.

6.2.15 Arrays/Hashmaps - Tutorial 7

| Instruction | Illustration |
|---|---|
| 1. Create a new No Message to No Message mapping. | Check chapter "How to Operate BIC Mapping Designer" in Training BIC 6 Basics if you need further assistance. |
| 2. Go to record program DEFAULT. | <pre> // define variables, set counter local size%, i%, volume%; volume% = 1; local harrypotter\${[]{}; //define hashmap (mixed type) // fill the hashmap harrypotter\${0}{ "title" } = "Harry Potter and the Philosopher's Stone"; harrypotter\${0}{ "published" } = "1997"; harrypotter\${0}{ "isbn" } = "0-7475-7360-3"; harrypotter\${1}{ "title" } = "Harry Potter and the Chamber of Secrets"; harrypotter\${1}{ "published" } = "1998"; harrypotter\${1}{ "isbn" } = "0-7475-3849-2"; harrypotter\${2}{ "title" } = "Harry Potter and the Prisoner of Azkaban"; harrypotter\${2}{ "published" } = "1999"; harrypotter\${2}{ "isbn" } = "0-7475-4215-5"; harrypotter\${3}{ "title" } = "Harry Potter and the Goblet of Fire"; harrypotter\${3}{ "published" } = "2000"; harrypotter\${3}{ "isbn" } = "0-7475-4624-X"; harrypotter\${4}{ "title" } = "Harry Potter and the Order of the Phoenix"; harrypotter\${4}{ "published" } = "2003"; harrypotter\${4}{ "isbn" } = "0-7475-5100-6"; harrypotter\${5}{ "title" } = "Harry Potter and the Half- Blood Prince"; harrypotter\${5}{ "published" } = "2005"; harrypotter\${5}{ "isbn" } = "0-7475-8108-8"; harrypotter\${5}{ "title" } = "Harry Potter and the Deathly Hallows"; harrypotter\${5}{ "published" } = "2007"; harrypotter\${5}{ "isbn" } = "0-7475-9106-7"; //get array size for defining how often the for loop must run //The getSize function's output is 7, but as the index starts //with 0, the highest index is 6!size% = getSize(harrypotter\${[]}) - 1; traceln("Max array index: " & size%); // get hashmap data traceln("Title of Harry Potter book volume 1 is " & harrypotter\${0}{ "title" } & ", year of publication is " & harrypotter\${0}{ "published" } & " and the isbn number is " & harrypotter\${0}{ "isbn" } & "."); </pre> |
| 3. Have a look at the trace-file and find out how the mixed type works. | <p>You can compare your results with the attached trainer solution.</p> <p>Traceline should look like the following lines:</p> <pre> Title of Harry Potter book volume 1 is Harry Potter and the Philosopher's Stone, year of publication is 1997 and the isbn number is 0-7475-7360-3. </pre> |

6.2.16 Arrays/Hashmaps - Exercise 8

Mixed type

- Import the message structures [msg_SHPMNT05_E46C.xml](#) and [msg_ADDRESSES_INH.xml](#) and create a **mapping** where „SHPMNT05_E46C“ is the source and „ADDRESS_INH“ is the destination.
- Go to DC1-segment and define a **mixed type hashmap** for address data and store the information of the RM1 segment into the mixed type hashmap. Identifier (key) should be the field name and the repetitions of RM1.
- Go to record program Additional Programms/End of MessageBlock (FileEnd) and fill the destination file for the roles Sales Organization, sold-to party, ship-to party and carrier.
- Use the [testfile](#) to see the results.

I What benefits can you see after using a mixed type hashmap?

6.2.17 Arrays/Hashmaps - Tutorial 8

| Instruction | Illustration |
|--|---|
| 1. Import the message structure msg_SHPMNT05_E46C.xml from the classmaterial. 2. Import the message structure msg_ADDRESS_INH.xml from the classmaterial. 3. Create a new mapping (source = "SHPMNT05_E46C", destination = "ADDRESS_INH"). | Check chapter "How to Operate BIC Mapping Designer" in Training BIC 6 Basics if you need further assistance. |
| 4. Go to record program DC1 and define variables. | <pre>// define 2-dimension hashmap (mixed type) global ADDRESS\${}{}; // define counter global COUNT_RM1%, LOOP1%; global c_PARTNER_Q%, c_NAME1%, c_STREET1%, c_POSTL_COD1%, c_City1%, c_COUNTRY1%; // initiate counter c_PARTNER_Q% = 0; c_NAME1% = 1; c_STREET1% = 2; c_POSTL_COD1% = 3; c_City1% = 4; c_COUNTRY1% = 5;</pre> |
| 5. Go to record program RM1, set the counter and fill the one-dimension array. | <pre>copy DC1.T20.L20.RM1:PARTNER_Q to ADDRESS\${"Partner"} [COUNT_RM1%];</pre> |

| Instruction | Illustration |
|--|---|
| | <pre> copy DC1.T20.L20.RM1:NAME1 to ADDRESS\${"Name"} [COUNT_RM1%]; copy DC1.T20.L20.RM1:STREET1 to ADDRESS\${"Street"} [COUNT_RM1%]; copy DC1.T20.L20.RM1:POSTL_COD1 to ADDRESS\${"Postl_cod"} [COUNT_RM1%]; copy DC1.T20.L20.RM1:CITY1 to ADDRESS\${"City"} [COUNT_RM1%]; copy DC1.T20.L20.RM1:COUNTRY1 to ADDRESS\${"Country"} [COUNT_RM1%]; COUNT_RM1% = COUNT_RM1% + 1; </pre> |
| <p>6. Go to record program FileEnd and write the array data to the destination file.</p> | <pre> for LOOP1% = 0 to COUNT_RM1% // DE: Verkaufsstelle; EN: Sales Organization if ADDRESS\${"Partner"}[LOOP1%] = "OSO" copy "Sales Organization" to ADDRESS.ROLE:NAME; copy ADDRESS\${"Name"}[LOOP1%] to ADDRESS.NAME:NAME; copy ADDRESS\${"Street"}[LOOP1%] to ADDRESS.STREET:STREET; copy ADDRESS\${"Postl_cod"}[LOOP1%] to ADDRESS.ZIP:ZIP; copy ADDRESS\${"City"}[LOOP1%] to ADDRESS.CITY:CITY; copy ADDRESS\${"Country"}[LOOP1%] to ADDRESS.COUNTRY:COUNTRY; endif // DE: Auftraggeber, EN: sold-to party if ADDRESS\${"Partner"}[LOOP1%] = "AG" copy "sold-to party" to ADDRESS.ROLE:NAME; copy ADDRESS\${"Name"}[LOOP1%] to ADDRESS.NAME:NAME; copy ADDRESS\${"Street"}[LOOP1%] to ADDRESS.STREET:STREET; copy ADDRESS\${"Postl_cod"}[LOOP1%] to ADDRESS.ZIP:ZIP; copy ADDRESS\${"City"}[LOOP1%] to ADDRESS.CITY:CITY; copy ADDRESS\${"Country"}[LOOP1%] to ADDRESS.COUNTRY:COUNTRY; endif // DE: Warenempfänger; EN: ship-to party if ADDRESS\${"Partner"}[LOOP1%] = "WE" copy "ship-to party" to ADDRESS.ROLE:NAME; copy ADDRESS\${"Name"}[LOOP1%] to ADDRESS.NAME:NAME; copy ADDRESS\${"Street"}[LOOP1%] to ADDRESS.STREET:STREET; copy ADDRESS\${"Postl_cod"}[LOOP1%] to ADDRESS.ZIP:ZIP; copy ADDRESS\${"City"}[LOOP1%] to ADDRESS.CITY:CITY; copy ADDRESS\${"Country"}[LOOP1%] to ADDRESS.COUNTRY:COUNTRY; endif // DE: Spediteur; EN: Carrier if ADDRESS\${"Partner"}[LOOP1%] = "SP" copy "Carrier" to ADDRESS.ROLE:NAME; copy ADDRESS\${"Name"}[LOOP1%] to ADDRESS.NAME:NAME; copy ADDRESS\${"Street"}[LOOP1%] to ADDRESS.STREET:STREET; copy ADDRESS\${"Postl_cod"}[LOOP1%] to ADDRESS.ZIP:ZIP; endif </pre> |

| Instruction | Illustration |
|--|--|
| | <pre> copy ADDRESS\${"City"}[LOOP1%] to ADDRESS.CITY:CITY; copy ADDRESS\${"Country"}[LOOP1%] to ADDRESS.COUNTRY:COUNTRY; endif next </pre> |
| <p>7. Have a look at the destination-file and see how the "mixed type" worked.</p> | <p>You can compare your results with the attached trainer solution.</p> |

7 Error Child Mapping

7.1 Notes

- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____

7.2 Exercises

7.2.1 Child mappings - Exercise

1. For the first exercise we want to use one main mapping and two different child mappings. The incoming **INVOIC D01B** should be sorted by credit note and invoice information. The destination structure will be a CSV file.
2. Please import as first step the mappings **map_MAIN_INVOIC_1.zip.bicmd** and **map_CHILD_INVOIC_CN.zip.bicmd**.
3. Your task is it to create a second child mapping to handle the invoice information of the incoming message. Please take a look to the existing child mapping. The name of the second child mapping is given by the main mapping. You will find this information at the BGM record.
4. Please map the relevant information to the Records IV_POSITION and the fields below.
5. Test your mappings with the file **INVOIC_D01B.txt**.

7.2.2 Child mappings - Tutorial

| Instruction | Illustration |
|--|--|
| 1. Please import the mappings MAIN_INVOICE_1 and CHILD_INVOICE_CN in BIC Mapping Designer. | Check chapter "How to Operate BIC Mapping Designer" in Training BIC 6 Basics if you need further assistance. |
| 2. Open mapping MAIN_INVOICE_1 and goto BGM segment. Here you will find the name of the second child mapping. | <pre>case "380": addPrefixDataForChildMapping(g_UNA\$&g_UNB\$&g_UNH\$&g_BGM\$); addSuffixDataForChildMapping(g_UNZ\$); traceIn(g_UNA\$&g_UNB\$&g_UNH\$&g_BGM\$); traceIn(g_UNZ\$); startChildMapping("CHILD_INVOIC_INV", "", "off", "", "on");</pre> |
| 3. Create a new child mapping CHILD_INVOIC_INV. Use the same source and destination messages like the child mapping CHILD_INVOICE_CN. | |
| 4. The main mapping maps the information of the sender and receiver to the header information. Please map additional information as described below. | |
| 5. Insert trace in UNB segment to identify start of child mapping. | <pre>traceIn("INVOIC MAPPING");</pre> |
| 6. Map the information of the field BGM:C106.1004 to IV_NR. | <pre>copy THIS:C106.1004 to HEADER.IV_POSITION.IV_POSITION_LINE:IV_NR;</pre> |

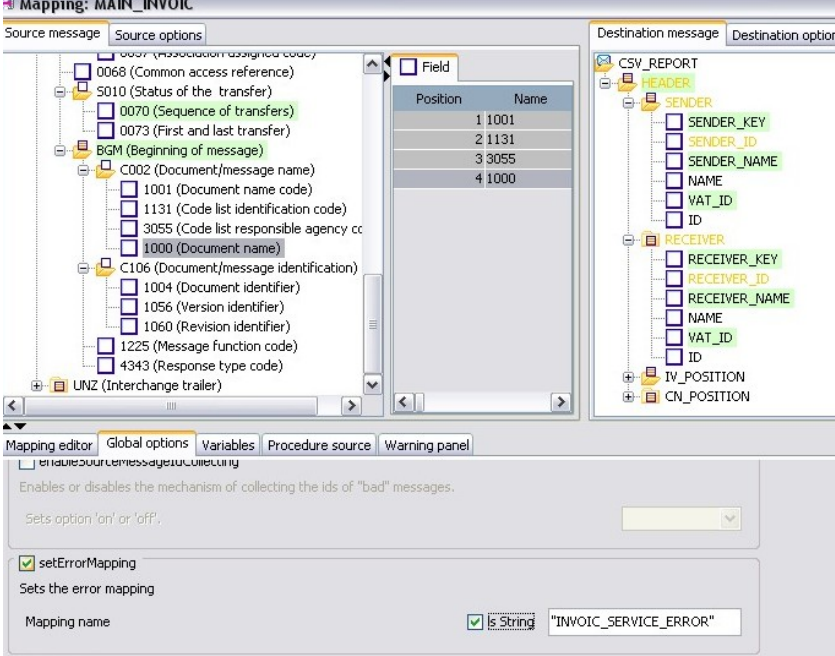
| Instruction | Illustration |
|---|--|
| 7. save address qualifier in segment SG2:NAD to variable. | <pre>g_Address_Qualifier\$ = THIS:3035;</pre> |
| 8. Map the information of SG3 RFF:C506.1154 below the NAD where the qualifier is VA. | <pre>if THIS:C506.1153 &= "VA" selectcase g_Address_Qualifier\$ case "BY": overwrite THIS:C506.1154 to HEADER.SENDER:ID; case "SU": overwrite THIS:C506.1154 to HEADER.RECEIVER:ID; endselect endif</pre> |
| 9. Map the information of the field TAX:C243.5278. | <pre>copy THIS:C243.5278 to HEADER.IV_POSITION.IV_POSITION_LINE:VAT_RATE;</pre> |
| 10. Map the information of the field MOA:C516.5004 at the segment group SG 50 where the qualifier is 77. | <pre>if THIS:C516.5025 &= "77" copy THIS:C516.5004 to HEADER.IV_POSITION.IV_POSITION_LINE:INV_AMOUNT; endif if THIS:C516.5025 &= "124" copy THIS:C516.5004 to HEADER.IV_POSITION.IV_POSITION_LINE:VAT_AMOUNT; endif</pre> |
| 11. Map the information of the field MOA:C516.5004 at the segment group SG 50 where the qualifier is 124. | |
| 12. Mapping should stop reading at segment UNT. | <pre>stopReading();</pre> |
| 13. Test your mapping with the provided test file. | |
| 14. Please double check with the trainer solution | |

7.2.3 Error mappings - Exercise

1. To test the error mapping handling you have to adjust the main mapping from the previous exercise to handle also the invoice list of the incoming file. Please delete the code line "stopreading();" for the case "393" at the BGM segment and insert an error command that this kind of messages are not supported.
2. You also have to change the option stopreadingonerror to off to handle all errors which are thrown at the main mapping.
3. As next step you have to import the error mapping from the class material ([INVOIC_SERVICE_ERROR](#)).
4. To link the error mapping to the main mapping set the global option at the corresponding tab.
5. Test your mappings with the file [INVOIC_D01B.txt](#) and check the logfile.

7.2.4 Error mappings - Tutorial

| Instruction | Illustration |
|---|--|
| 1. Use the main mapping from the previous exercise and adjust it as follows. | |
| 2. Adjust the mapping code of the record BGM at the main mapping to throw an error if | <pre>selectcase THIS:C002.1001 case "380": addPrefixDataForChildMapping(g_UNA\$&g_UNB\$&g_UNH \$&g_BGM\$);</pre> |

| Instruction | Illustration |
|--|---|
| <p>the source file contains the qualifier 393</p> <p>Use the following code line instead of the stopreading command:</p> <pre>errorln("BGM qualifier 393 not supported!");</pre> | <pre> addSuffixDataForChildMapping(g_UNZ\$); traceln(g_UNA\$&g_UNB\$&g_UNH\$&g_BGM\$); traceln(g_UNZ\$); startChildMapping("CHILD_INVOIC_INV", "", "off", "", "on"); case "381": addPrefixDataForChildMapping(g_UNA\$&g_UNB\$&g_UNH\$&g_BGM\$); addSuffixDataForChildMapping(g_UNZ\$); traceln(g_UNA\$&g_UNB\$&g_UNH\$&g_BGM\$); traceln(g_UNZ\$); startChildMapping("CHILD_INVOIC_CN", "", "off", "", "on"); case "393": errorln("BGM qualifier 393 not supported!"); endselect </pre> |
| <p>3. Import and compile the error mapping of the class material.</p> | |
| <p>4. Register the error mapping at the main mapping.</p> |  <p>Register the error mapping at the main mapping</p> |
| <p>5. Test the main and error mapping with the provided test file from the classmaterial and check the logfile.</p> | |
| <p>6. Please double check with the trainer solution</p> | |

8 BIS:MT Mappings

8.1 Notes

- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____

8.2 Exercises

8.2.1 BIS:MT - Exercise 1

- Change the datasource for DS_MT to link to a BIS:MT database. Login details can be found in the seminar opening.
- Create a new mapping "SEEEELFOR_V3_0_TO_MT_DEMO". Use [msg_SEEEELFOR_V3_0.xml](#) as source message. All informations should be added to the MT- tables *tBISMT_Workflow* and *tBISMT_Documents*.
- Map the below information to the MT- Database
- Use the test file [MT_DEMO.xml](#) to run your mapping.

Record program NewMapping:

| Source field / value | BISMT table / field | Note |
|------------------------------------|--|--|
| getinputvalue("ORIGIN_FILE_ID") | tBISMT_Workflow.cWfld | Please also write this string to a global variable g_Origin_File_ID\$ for later use. String "1234" needs to be set as input value in the test environment. |
| getinputvalue("PROCESSING_STATUS") | tBISMT_Workflow.cStatus | String "RUNNING" needs to be set as input value in the test environment. |
| 1 | tBISMT_Workflow.cMessageld | fix value for the demo mapping. |
| DEMO | tBISMT_Workflow.cPartner | fix value for the demo mapping. |
| DEMO | tBISMT_Workflow.cMessageType | fix value for the demo mapping. |
| IN | tBISMT_Workflow.cDirection | fix value for the demo mapping. |
| currentdate() & currenttime() | tBISMT_Workflow.cProcessingTime Start | Create current date and time and convert it to a timestamp. |
| currentdate() & currenttime() | tBISMT_Workflow.cProcessingTime End | Create current date and time and convert it to a timestamp. |

Write record to table tBISMT_Workflow and clear fields afterwards.

Record SEEDELFOR.CNT.GRP.MSG.ME

| Source field / value | BISMT table / field | Note |
|---|-----------------------------|---|
| increase global variable Counter_Msg% with 1 | | |
| write records to table tBISMT_Documents and clear fields afterwards. | | do this only if msg counter is greater or equal 2. |
| MESSAGE_NUMBER:value | tBISMT_Documents.cDocNumber | |
| MESSAGE_TYPE:value + MESSAGE_TYPE_VERSION:value + MESSAGE_TYPE_RELEASE:value | tBISMT_Documents.cDocType | Please create a combination of the 3 fields and use an underscore as separator. |

Record SEEDELFOR.CNT.GRP.MSG.REF

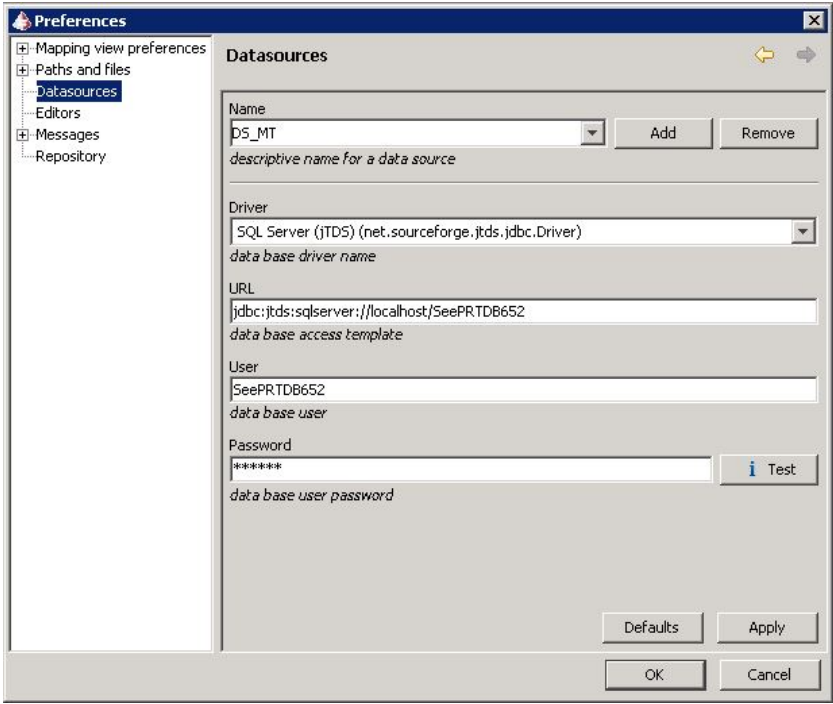
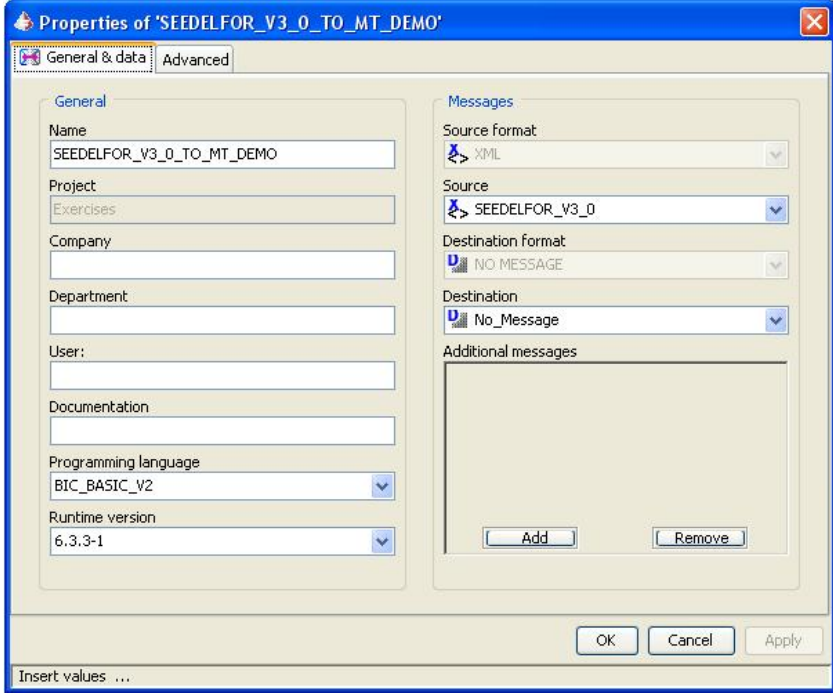
Do this only if QUALIFIER:value &= "ON"

| Source field / value | BISMT table / field | Note |
|---|--------------------------|---|
| REFERENCE_DATE:value | tBISMT_Documents.cFree15 | Please format the date as follows: DD.MM.YYYY |
| g_Origin_File_ID\$ | tBISMT_Documents.cWfld | |
| g_Origin_File_ID\$ + REFERENCE:value | tBISMT_Documents.cID | Create the combined ID as g_Comb_ID\$ |
| REFERENCE:value | tBISMT_Documents.cFree7 | |

Record program MappingEnd



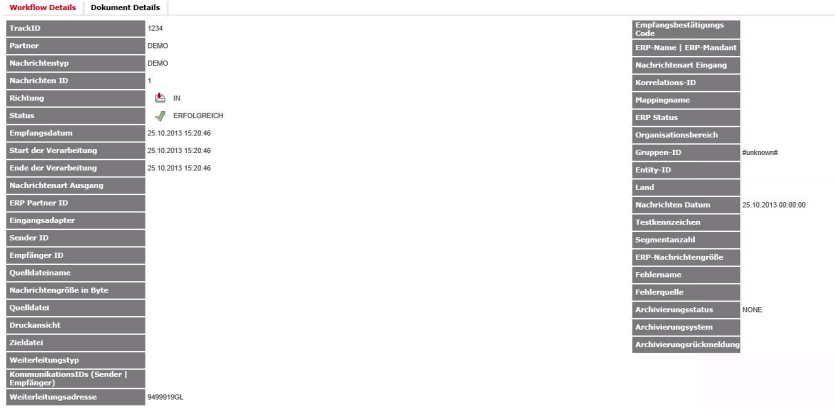
write records to table tBISMT_Documents and clear fields afterwards.

8.2.2 BIS:MT - Tutorial 1

| Instruction | Illustration |
|---|---|
| <p>1. Change the datasource SmartiDS from the H2 database to the SQL Server database:</p> <p>user: SeePRTDB652 Password: secret</p> |  <p>Change datasource SmartiDS</p> |
| <p>2. Import the provided message structure into BIC Mapping Designer</p> <p>3. Create a new Mapping with SEEDFOR as source and No_Message as destination</p> |  <p>Create a new mapping</p> |
| <p>4. Read input values ORIGIN_FILE_ID and PROCESSING_STATUS.</p> <p>I Please set them first in the test environment.</p> | <p>Record program NewMapping</p> <pre data-bbox="598 1881 1380 2029">/* Next lines are only relevant for the training environment to populate some fields in table tBISMT_Workflow. Normally these fields will be populated by the BIS process.</pre> |

| Instruction | Illustration |
|--|--|
| <p>5. Open the connection the BIS:MT database.</p> <p>6. Write the information to the BIS:MT database.</p> <p>7. Clear all MT fields.</p> | <pre> */ /* global g_Origin_File_ID\$; moved to variablePanel */ /* global g_Process_Status\$; moved to variablePanel */ /* set the Input variables in test environment if you run the mapping inside the BICMD test environment. ORIGIN_FILE_ID = 1234 PROCESSING_STATUS = RUNNING */ g_Origin_File_ID\$ = getinputValue("ORIGIN_FILE_ID"); g_Process_Status\$ = getinputValue("PROCESSING_STATUS"); openMTDBConnection(); /* Create a timestamp for processing time start and end Be carefull with the formatting of the timestamp */ local datetime\$; local datetime~; datetime\$ = currentdate() & currenttime(); datetime\$ = getTime(datetime\$, "yyyyMMddHHmmss"); copy absBig(datetime\$) to datetime~; traceln(datetime~); setMTField("cWfid", g_Origin_File_ID\$); setMTField("cMessageId", "1"); setMTField("cPartner", "DEMO"); setMTField("cMessageType", "DEMO"); setMTField("cDirection", "IN"); setMTField("cStatus", g_Process_Status\$); setMTField("cProcessingTimeStart", datetime~); setMTField("cProcessingTimeEnd", datetime~); //write record to table tBISMT_Workflow and clear fields afterwards. writeMTRecord(g_Origin_File_ID\$, null); clearMTRecord(); </pre> |
| <p>8. Write the fields to MT if msq counter is equal or greater 2.</p> <p>9. write message number and a combined message type to BIS:MT table.</p> | <p>Record SEEDELFOR.CNT.GRP.MSG.ME</p> <pre> Counter_Msg% = Counter_Msg% + 1; if Counter_Msg% >= 2 //write record to table tBISMT_Documents and clear fields afterwards. writeMTDetailRecord(g_Comb_ID\$, g_Origin_File_ID\$); clearMTRecord(); endif //Msg number setMTField("cDocNumber", THIS.MESSAGE_NUMBER:value); //Message type, version and release local l_message_type\$; copy THIS.MESSAGE_TYPE:value & "_" & THIS.MESSAGE_TYPE_VERSION:value & "_" & THIS.MESSAGE_TYPE_RELEASE:value to l_message_type\$; traceln(l_message_type\$); setMTField("cDocType", l_message_type\$); </pre> |
| <p>10. Set the MT-fields in tag REF and also concat the filename and reference for the keys.</p> | <p>Record SEEDELFOR.CNT.GRP.MSG.REF</p> <pre> local l_Date\$; if THIS.QUALIFIER:value &= "ON" l_Date\$ = formatDate(THIS.REFERENCE_DATE:value, "YYYYMMDD", "DD.MM.YYYY"); </pre> |

| Instruction | Illustration |
|--|--|
| | <pre> g_Comb_ID\$ = g_Origin_File_ID\$ & "_" & THIS.REFERENCE:value; setMTField("cWfId", g_Origin_File_ID\$); setMTField("cId",g_Comb_ID\$); setMTField("cFree15", l_Date\$); setMTField("cFree7", THIS.REFERENCE:value); //order number endif </pre> |
| 11. Write all fields to BIS:MT database. | <p>Record Program MappingEnd</p> <pre> //write record to table tBISMT_Documents and clear fields afterwards. writeMTDetailRecord(g_Comb_ID\$, g_Origin_File_ID\$); closeMTDBConnection(); </pre> |
| 12. Please test your mapping with the provided test file. | |
| 13. Please check the results in BIS:MT 14. Start the Portal application from the start menu and logon to the application with <ul style="list-style-type: none"> • user: standard • password: seeburger | |

| Instruction | Illustration |
|---|--|
| <p>15. Search for your entries in the actual timeframe.</p> |  <p>BIS:MT search for new entries</p> |
| <p>16. For each time you changed the input value ORIGIN_FILE_ID and you the mapping was executed you should see an entry in the result list.</p> |  <p>BIS:MT result</p> |
| <p>17. If you click on one result you will see more details.</p> |  <p>BIS:MT details</p> |

| Instruction | Illustration |
|--|--------------|
| 18. You can also switch between workflow and document details. | |
| 19. To double check please have a look at the trainer solution . | |

8.2.3 BIS:MT - Exercise 2

- Create a new mapping "Direct_Access_MT" that can read the *tBISMT-Workflow* table, using Direct Commands

8.2.4 BIS:MT - Tutorial 2

| Instruction | Illustration |
|---|---|
| 1. Create a new Mapping with No_Message as source and No_Message as destination | |
| 2. Start open a MT DB-connection in "NewMapping" | <pre>openMTDBConnection();</pre> |
| 3. Create a SQL- Select and combine it with the commands executeMTSQLCommandWithMultiResult, hasMoreMTRecords, nextMTRecord and getMTField | <pre>executeMTSQLCommandWithMultiResult("select * from TBISMT_WORKFLOW"); for Loop_1% = 0 to 10000 if hasMoreMTRecords() nextMTRecord(); l_Output\$ = getMTField("CPARTNER"); traceln(l_Output\$); else exitfor; endif next</pre> |
| 4. Close the connection in "MappingEnd" | <pre>closeMTDBConnection();</pre> |
| 5. Check your result in the logfile and also validate it against the trainer solution . I This trainer solution is a combination of exercise 2 and 3. | |

8.2.5 BIS:MT - Exercise 3

- Use the mapping you created in the exercise before and update the result you received before.
- You receive the field cPartner. Select one result and update the field cPartner with your Company

8.2.6 BIS:MT - Tutorial 3

| Instruction | Illustration |
|--|------------------------------|
| 1. Extend the mapping you have written before with | <pre>local l_Output\$;</pre> |

| Instruction | Illustration |
|---|---|
| the update statement and read out again. Set a clearMTRecord between your commands | <pre> executeMTSQLCommandWithMultiResult("select * from TBISMT_WORKFLOW"); for Loop_1% = 0 to 10000 if hasMoreMTRecords() nextMTRecord(); l_Output\$ = getMTField("CPARTNER"); traceLn(l_Output\$); else exitfor; endif next clearMTRecord(); executeMTSQLCommand("UPDATE TBISMT_WORKFLOW SET CPARTNER= 'Your Company' WHERE CPARTNER= 'DEMO' "); executeMTSQLCommandWithMultiResult("select * from TBISMT_WORKFLOW"); for Loop_1% = 0 to 10000 if hasMoreMTRecords() nextMTRecord(); l_Output\$ = getMTField("CPARTNER"); traceLn(l_Output\$); else exitfor; endif next </pre> |
| 2. Check your result in the logfile, BIS:MT and also validate it against the trainer solution . | |

8.2.7 BIS:MT - Exercise 4

- Create a new mapping "Read_MT" and read the *tBISMT-Workflow* table again. Trace the field cPartner

8.2.8 BIS:MT - Tutorial 4

| Instruction | Illustration |
|---|---|
| 1. Create a new Mapping with No_Message as source and No_Message as destination | |
| 2. Start open a MT DB-connection in "NewMapping" | <pre>openMTDBConnection();</pre> |
| 3. Use the command readMTRecord, "1234" is the primary key | <pre> local l_Output\$; openMTDBConnection(); readMTRecord("1234"); l_Output\$ = getMTField("CPARTNER"); traceLn(l_Output\$); clearMTRecord(); closeMTDBConnection(); </pre> |

| Instruction | Illustration |
|---|--|
| 4. Close the connection in "MappingEnd" | <pre data-bbox="587 241 1385 297">closeMTDBConnection();</pre> |
| 5. Check your results in the BIC logfile and validate it against the trainer solution . | |

9 BAPI Adapter

9.1 Notes

- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____

9.2 Exercises

9.2.1 BAPI - Exercise 1

Creation of a BAPI Mapping to call the SAP Standard BAPI **BAPI_COMPANYCODE_GETDETAIL** using simple import parameters.

To logon to the SAP system please use the following details:

server: dedcsapsoex01.seeburger.de

client: 800

system no: 00

User: academy

Password: seeburger

9.2.2 BAPI - Tutorial 1

| Instruction | Illustration |
|--|--|
| <p>1. Before you can run a bapi mapping you have to provide sapjco libraries and reference it in the CompileConvert.ini, SeeMappingsComp.ini and bic_mapping_designer.ini</p> <p>I Please do this only if haven't done it before.</p> | <p>Extract Z:_CD_Source\07_SAP\sapjco3-NTAMD64-3.0.8.zip to your harddisk and copy the following files as follows.</p> <pre>copy sapjco3.jar to C:\SEEBURGER\MappingDesigner652\lib\ext copy sapjco3.dll to C:\SEEBURGER\MappingDesigner652\runtime\jvm64\jre\bin</pre> <p>Extend the CLASSPATH of C:\SEEBURGER\MappingDesigner652\conf\bic\ini\CompileConvert.ini</p> <pre>CLASSPATH=...\%BASE%\lib/ext/sapjco3.jar;</pre> <p>I Don't forget the backslash at the end of the entry.</p> <p>Extend the CLASSPATH of C:\SEEBURGER\MappingDesigner652\conf\bic\ini\SeeMappingsComp.ini</p> <pre>Classpath=..\lib\ext\sapjco3.jar;</pre> <p>Extend the CLASSPATH of C:\SEEBURGER\MappingDesigner652\Bin\bic_mapping_designer.ini</p> <pre>Classpath=../lib/ext/sapjco3.jar;</pre> <p>Afterwards please restart your BIC Mapping Designer.</p> |
| 2. Create a new No Message to No Message mapping. | Check chapter "How to Operate BIC Mapping Designer" in Training BIC 6 Basics if you need further assistance. |
| 3. Go to Record program New Mapping | |
| 4. Define local variables | <pre>local hostname\$, systemnr\$, client\$, user\$, password\$, language\$;</pre> |
| 5. Open Bapi Connection | <pre>copy "dedcsapsoex01.seeburger.de" to hostname\$; copy "00" to systemnr\$; copy "800" to client\$; copy "academy" to user\$; copy "seeburger" to password\$; copy "EN" to language\$; openBAPICConnection("BapiConnection", hostname\$, systemnr\$, client\$, user\$, password\$, language\$); traceln("open BapiConnection test!");</pre> <p>I This code should only be used within BIC MD. If the mapping is used inside BIS you can make a reference to the masterdata.</p> |
| 6. Create Bapi Call | <pre>createBAPICall("BapiConnection", "BAPI_COMPANYCODE_GETDETAIL");</pre> |
| 7. Record program DEFAULT | |
| 8. Set the import parameter | <pre>setbapiimportvalue("COMPANYCODEID", "1000");</pre> |
| Training BIC 6 Advanced - Handout - 9. Execute the BAPI | <pre>executebapicall();</pre> |
| 10. Get the BAPI Exportparameter | <pre>local companycode_name\$, companycode_street\$, companycode_postalcode\$, companycode_city\$;</pre> |

9.2.3 BAPI - Exercise 2

Creation of a BAPI Mapping to call the SAP BAPI **/SEEAG/HF_SALES_ORDER_CREATE** using import structures and import tables.

To logon to the SAP system please use the following details:

server: dedcsapsoex01.seeburger.de

client: 800

system no: 00

User: academy

Password: seeburger

9.2.4 BAPI - Tutorial 2

This exercise will only work if the following function modul has been installed in the SAP system you will connect to: **/SEEAG/HF_SALES_ORDER_CREATE**

| Instruction | Illustration |
|--|--|
| <p>1. Before you can run a bapi mapping you have to provide sapjco libraries and reference it in the CompileConvert.ini, SeeMappingsComp.ini and bic_mapping_designer.ini</p> <p>I Please do this only if you haven't done it before.</p> | <p>Extract Z:_CD_Source\07_SAP\sapjco3-NTAMD64-3.0.8.zip to your harddisk and copy the following files as follows.</p> <pre>copy sapjco3.jar to C:\SEEBURGER\MappingDesigner652\lib\ext copy sapjco3.dll to C:\SEEBURGER\MappingDesigner652\runtime\jvm64\jre\bin</pre> <p>Extend the CLASSPATH of C:\SEEBURGER\MappingDesigner652\conf\bic\ini\CompileConvert.ini</p> <pre>CLASSPATH=...\ %BASE%/lib/ext/sapjco3.jar;</pre> <p>I Don't forget the backslash at the end of the entry.</p> <p>Extend the CLASSPATH of C:\SEEBURGER\MappingDesigner652\conf\bic\ini\SeeMappingsComp.ini</p> <pre>Classpath=...\lib\ext\sapjco3.jar;</pre> <p>Extend the CLASSPATH of C:\SEEBURGER\MappingDesigner652\Bin\bic_mapping_designer.ini</p> <pre>Classpath=../lib/ext/sapjco3.jar;</pre> <p>Afterwards please restart your BIC Mapping Designer.</p> |
| <p>2. Create a new No Message to No Message mapping.</p> | <p>Check chapter "How to Operate BIC Mapping Designer" in Training BIC 6 Basics if you need further assistance.</p> |

| Instruction | Illustration |
|-------------------------------------|--|
| 3. Go to record program New Mapping | |
| 4. Define local variables | <pre>local hostname\$, systemnr\$, client\$, user\$, password\$, language\$, j%, end%;</pre> |
| 5. Open Bapi Connection | <pre>copy "dedcsapsoex01.seeburger.de" to hostname\$; copy "00" to systemnr\$; copy "800" to client\$; copy "academy" to user\$; copy "seeburger" to password\$; copy "DE" to language\$; openBAPICConnection("BapiConnection", hostname\$, systemnr\$, client\$, user\$, password\$, language\$); traceln("open BapiConnection test!");</pre> <p>I This code should only be used within BIC MD. If the mapping is used inside BIS you can make a reference to the masterdata.</p> |
| 6. Create Bapi Call | <pre>createBAPICall("BapiConnection", "/SEEAG/ HF_SALES_ORDER_CREATE");</pre> |
| 7. Go to record program DEFAULT | |
| 8. Set the import parameter | <p>In this example the import parameters are defined as structures and tables</p> <p>import structures</p> <pre>local j%, end%, kunag\$, kunwe\$, item_number\$, article_number \$, item_quantity\$, unit\$; local sched_line\$, date\$, sched_quantity\$, ordernumber\$; kunag\$ = "0000100132"; kunwe\$ = "0000100132"; item_number\$ = "10"; article_number\$ = "0000000000000001147"; item_quantity\$ = "5"; unit\$ = "ST"; sched_line\$ = "10"; date\$ = "20131211"; sched_quantity\$ = "5"; getBAPIImportStructure("WA_ORDER_HEADER_IN"); setBAPIStructureFieldValue("DOC_TYPE", "TA"); setBAPIStructureFieldValue("SALES_ORG", "0001"); setBAPIStructureFieldValue("DISTR_CHAN", "01"); setBAPIStructureFieldValue("DIVISION", "01"); setBAPIStructureFieldValue("SD_DOC_CAT", "C"); setBAPIStructureFieldValue("PMNTTRMS", "0002"); setBAPIStructureFieldValue("INCOTERMS1", "EXW"); setBAPIStructureFieldValue("PURCH_NO_C", "BICAdvanced");</pre> <p>import tables</p> <pre>// set Partners getBapiTable("WA_ORDER_PARTNERS"); addBAPITableRow(); setBAPITableFieldValue("PARTN_ROLE", "AG"); setBAPITableFieldValue("PARTN_NUMB", kunag\$); if kunag\$ &!= kunwe\$ addBAPITableRow();</pre> |

| Instruction | Illustration |
|--|--|
| | <pre> setBAPITableFieldValue("PARTN_ROLE", "WE"); setBAPITableFieldValue("PARTN_NUMB", kunwe\$); endif // set Items getBapiTable("WA_ORDER_ITEMS_IN"); addBAPITableRow(); setBAPITableFieldValue("ITM_NUMBER", item_number\$); setBAPITableFieldValue("MATERIAL", article_number\$); setBAPITableFieldValue("TARGET_QTY", item_quantity\$); setBAPITableFieldValue("TARGET_QU", unit\$); setBAPITableFieldValue("PLANT", "0001"); setBAPITableFieldValue("GROSS_WGHT", "4"); setBAPITableFieldValue("NET_WEIGHT", "3"); setBAPITableFieldValue("UNTOF_WGHT", "G"); // set Schedules getBapiTable("WA_ORDER_SCHEDULES_IN"); addBapiTableRow(); setBapiTableFieldValue("ITM_NUMBER", item_number\$); setBapiTableFieldValue("SCHED_LINE", sched_line\$); setBapiTableFieldValue("REQ_DATE", date\$); setBapiTableFieldValue("TP_DATE", date\$); setBapiTableFieldValue("MS_DATE", date\$); setBapiTableFieldValue("REQ_QTY", sched_quantity\$); setBapiTableFieldValue("LOAD_DATE", date\$); setBapiTableFieldValue("GI_DATE", date\$); </pre> |
| 9. Execute the BAPI | <pre> traceln("Create Sales Order"); executeBapiCall(); </pre> |
| 10. Get the BAPI Exportparameter 11. Commit the imported sales order | <p>In this example the export values are defined as normal variables and as tables</p> <pre> ordernumber\$ = getBapiExportValue("WA_SALESDOCUMENT"); traceln("Order number: "&ordernumber\$); getBAPITable("WA_RETURN"); end% = getBAPITableRowCount(); for j% = 0 to end% setBAPITableRow(j%); traceln(getBAPITableFieldValue("TYPE")& "&getBAPITableFieldValue("ID")& getBAPITableFieldValue("NUMBER")& ": "&getBAPITableFieldValue("MESSAGE")); next if ordernumber\$ &!= "" createBAPICall("BapiConnection", "BAPI_TRANSACTION_COMMIT"); setBapiImportValue("WAIT", "X"); executeBapiCall(); getBAPIStructure("RETURN"); traceln(getBAPIStructureFieldValue("TYPE")&": "&getBAPIStructureFieldValue("MESSAGE")); endif </pre> |
| 12. Result can be checked using transaction /nVA03 I If you will see an incompleteness check warning in the BIC logfile and in the SAP GUI. This can be fixed with a customizing change in the | |

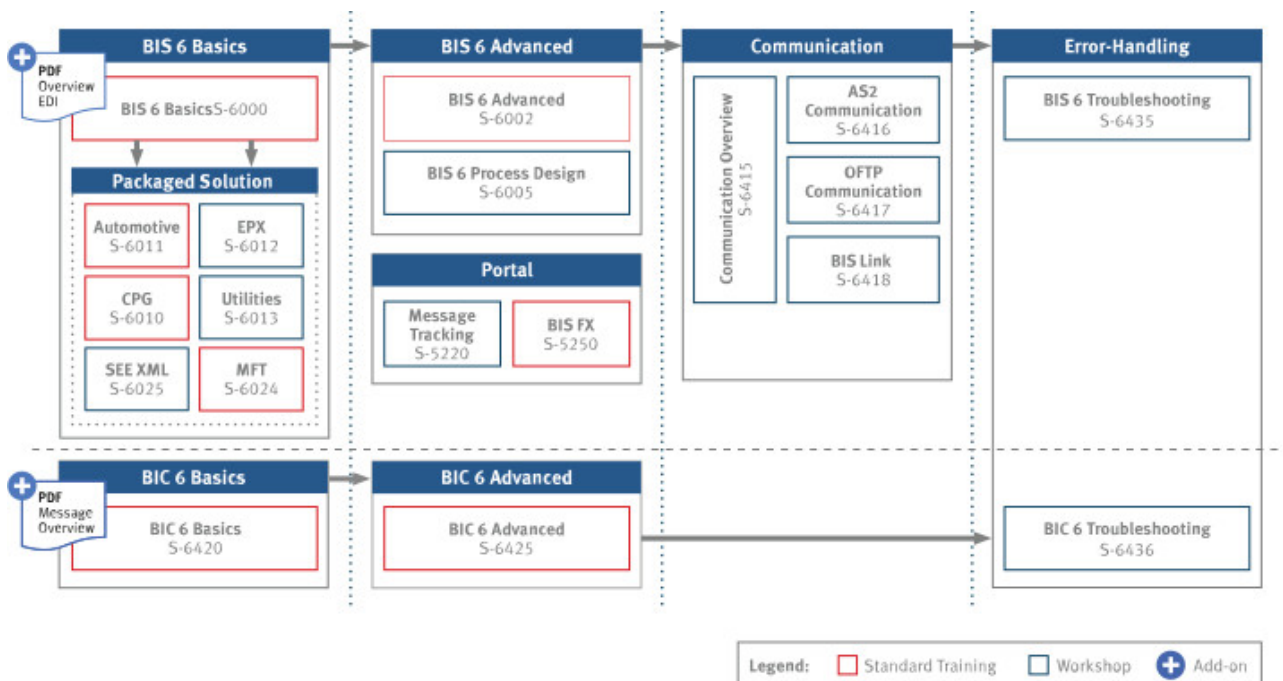
| Instruction | Illustration |
|---|--------------|
| SAP system that is not part of this training. | |
| 13. Please double check with the trainer solution | |

10 Seminar Closing

10.1 Notes

- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____

10.2 Select your next Training



Have a look at our online [Trainings finder](#)

10.3 Meet the Expert - Live Webcasts

| | | |
|--------------|--|-----------------|
| Live Webcast | Secure file transfer - ad hoc and for smart teamwork How do you exchange large files or confidential data with your partners and colleagues? | MEET THE EXPERT |
| Live Webcast | Improve data quality with the help of an EDI message validation before the transfer and avoid rejections or penalties from your customer SEEBURGER Compliance Check Adapter | MEET THE EXPERT |
| Live Webcast | Salesforce to SAP integration. SEEBURGER offers real-time integration as an on-premises solution or Cloud Service! Avoid the maintenance nightmare of "spaghetti" integration. We will show you how it works in a short demo! | MEET THE EXPERT |
| Live Webcast | Ban FTP servers! Bring security and traceability into all automated data transfers. Security is a topic that is important at the executive level (CSO) | MEET THE EXPERT |

- Sign in to the webcasts
EN: <http://www.seeburger.com/events>
DE: <http://www.seeburger.de/events>
- If you missed a webcast go to
<https://servicesdesk.seeburger.de>

| Date | Topic | Link |
|------------|--|---|
| 2016-02-09 | Need a real 24/7 monitoring of your SEEBURGER BIS? Use SEEBURGER Remote Services! | PDF VIDEO |
| 2016-02-02 | The new version 6.5 of the Business Integration Suite – highlights & planned innovations for 2016 | PDF VIDEO |
| 2016-01-26 | Successfully implement your B2B Strategy in and with China & Asia | PDF VIDEO |
| 2015-12-08 | Secure your "crown jewels" against data leakage and data theft. | PDF VIDEO |
| 2015-12-01 | Integrate Big Data Warehouses efficiently – realize ETL scenarios. | PDF VIDEO |
| 2015-11-24 | Onboard multiple partners efficiently with an integrated campaign tool. Book now online at SEEBURGER. | PDF VIDEO |
| 2015-11-17 | New revenue channels require the integration of data over all systems/channels. | PDF VIDEO |
| 2015-11-10 | Ban FTP servers! Bring security and traceability into all automated data transfers. | PDF VIDEO |
| 2015-11-03 | Still using ISO9000 based EDI communication? | PDF VIDEO |
| 2015-10-27 | Integrate your plants and branch offices using one central platform, leading to higher security, transparency and up-to-date data. | PDF VIDEO |
| 2015-10-20 | Salesforce to SAP integration. SEEBURGER offers real-time integration as an on-premise solution or Cloud Service! | PDF VIDEO |
| 2015-10-13 | Integration solutions for Industry 4.0 and IoT (Internet of Things) with a central Integration Platform | PDF VIDEO |

10.4 SEEBURGER Community

- Join the SEEBURGER Community
<https://community.seeburger.com>
- Use the SEEBURGER Community to discuss any topic related to SEEBURGER products and services.
- The service is hosted by SEEBURGER and the use is free of charge.

Join us



General

Announcements

New products, releases or services from SEEBURGER.

Social Feedback

Discussions about the SEEBURGER Community.

Customer Webcasts

Discussion about Meet-the-Expert and other Customer WebCasts.

Business Integration

BIS Operation and Development

BIS Front-End, Process Designer, Solution Configuration.

BIC Mappings and B2B Message Standards

All about mappings, guidelines and compliance checks.

Solutions for SAP

Add-ons for SAP ERP or Adapters for XI/PI.

Regional / Local

D-A-CH

German speaking countries: Local user groups and projects.

North America

Local user groups and projects.

10.5 SEEBURGER Support RSS Feed

- SEEBURGER Support RSS Feed keeps you updated regarding:
 - BIS, B2B Portal and Solution Patch News
 - Important information for daily operations (e.g. certificate change)
 - Security News for BIS and B2B Portal Releases
- RSS Feed is easy to integrate in your Outlook or other RSS reader



https://servicedesk.seeburger.de/rss/RSS_SEEBURGER_SUPPORT.xml

SEEBURGER Service & Support

SEEBURGER Support RSS Feed

[BIS 6.5.2 Patch News - 25.09.2015](#)

Freitag, 25. September 2015 17:11

In this article you will find the states of patches for the 6.5.2 Release of the SEEBURGER BIS6.

[New Telebox Certificate for Secure P7/X400 in October 2015](#)

Freitag, 18. September 2015 14:55

German Telekom plans to renew the certificate for Secure P7/X400 on 13th October 2015.

[JDK Security News for BIS 6 and B2B Portal Releases](#)

Dienstag, 1. September 2015 00:01

What is the policy for updating the Java runtime on instances of BIS and Portal?

[BIS 6.5.2 Standard Solutions News - 11.09.2015](#)

Freitag, 11. September 2015 14:57

In this article you will find the current states of solution updates for the 6.5.2 Release of the SEEBURGER BIS 6.

[BIS 6.5.2 Patch News - 07.09.2015](#)

Montag, 7. September 2015 14:56

In this article you will find the states of patches for the 6.5.2 Release of the SEEBURGER BIS6.

[BIS 6.3.5Q4 Patch News - 27.08.2015](#)

Donnerstag, 27. August 2015 15:07

In this article you will find the current states of patches for the 6.3.5Q4 Release of the SEEBURGER BIS 6.

10.6 Additional Information

- SEEBURGER Service Desk & Knowledge Base
 - <https://servicedesk.seeburger.de>
- Documentation
 - can be accessed from the BIS CD: %CDROM%\doc\index.html
 - can be accessed from the installation folder: %BISROOT%\software\spm-repository\doc\index.html
 - can be accessed via the info link in BIS Front-end or via <http://localhost:10000/seeburger/doc/index.html>