

SEEBURGER BICMD for SAP Exchange Infrastructure - Configuration Guide



Applies to:

The Business Integration Converter Mapping Designer (BIC MD) version 5.5.2/6.3.2 is a visual tool used for creating mappings, used by the BIC Adapter in the SAP XI server XI 3.0/PI 7.0. For more information, visit the [SOA Management homepage](#).

Summary

This document shows how Business Integration Converter Mapping Designer (BIC MD) can be used to create Mappings which are further used by the Seeburger BIC Adapter. This document also runs through the usage of the SDM tool for Deploying Seeburger Mappings into SAP XI.

Any enhancements to this document are welcome.

Author: Varun Reddy Kasireddy

Company: nLeague Software Solutions Private Ltd (www.nleague.com)

Created on: 23 January 2009

Author Bio

Varun Reddy has about three years of experience in the IT industry. He has been with nLeague Software Solutions Pvt Ltd. since November 2007. He is a part of nLeague's SAP Netweaver competency group and has been working on SOA for around a year now.

Table of Contents

Terms and Definitions.....	3
Overview.....	3
Integration.....	3
Features.....	4
Regular BIC Mappings.....	5
Creating 1:1 Mappings.....	5
Use.....	5
Requirements.....	5
Actions.....	5
Caution:.....	6
Deploying Mappings.....	8
Use.....	8
Actions.....	8
Caution.....	9
Additional XI Commands.....	9
Related Content.....	11
Disclaimer and Liability Notice.....	12

Terms and Definitions

ANSI X.12:	American National Standards Institute data interchange standard.
BIC	Business Integration Converter. SEEBURGER application used to convert documents from source formats to destination formats.
Business Process	(See SAP Exchange Infrastructure Documentation): A business process is an executable process (script). The process defines all steps that are executed to control the message processing on the integration server.
BIC Mapping Designer	SEEBURGER tool used to create mappings.
Communication Channel	See SAP Exchange Infrastructure Documentation.
Converter	See BIC.
CSV	Comma Separated Values format
EDI	Electronic Data Interchange.
EDIFACT	Electronic Data Interchange for Administration, Commerce and Transport (http://www.unece.org/trade/untdid/welcome.htm).
Mapping	Program, which defines how documents are converted from a given source format to a given destination format.
Module Sequence	See SAP Exchange Infrastructure Documentation
SWIFT	Banking EDI standard.
Tradacom	UK-based EDI standard
VDA	Verband der Automobilindustrie (German association of the automotive industry, http://www.vda.de).
XML document	SAP's Exchange Infrastructure message format is based on XML, Therefore, the present document refers to the messages processed by SAP Exchange Infrastructure as XML documents.
Workflow	See Business Process.

Overview

Integration

BIC MD is a separate standalone tool, which creates Software Deployment Archives (SDA). SDA files are SAP J2EE libraries that can be deployed on the SAP XI Server using the Software Deployment Manager (**SDM**).

BIC MD must be configured in the following way:

1. After installing BIC MD, import the project file `prj_GeneratorMappings.zip.bicmd` from `user\bicmd\exchange` (subdirectory in the BIC installation directory), using the menu item File | Import. The *GeneratorMappings* project will be created automatically. The project contains all service mappings required to create 1:1 mappings (see next section).
2. The project *ActiveMappings* must be created, for creating SAP deployment files (SDA files). Select menu item File | New Project. The project must be called *ActiveMappings*.

This project contains all mappings that have been deployed on the XI Server. If new mappings should be deployed, they must be imported or copied to the ActiveMappings project.

There might be another project called AdapterContent. This project contains additional mappings delivered with the adapter.

A project can be created with the following procedure:

3. Select menu item File | Import.
4. Browse to the location user\bicmd\exchange (subdirectory within the BIC installation directory).
5. Select the file called bicmd-project.zip. A mapping selection window will pop up, Select and add the mappings from the popup window and click ok.

Note: Mappings can be modified and tested in this project, but they must be moved to the Active Mappings project in order to create the deployment archive (SDA file).

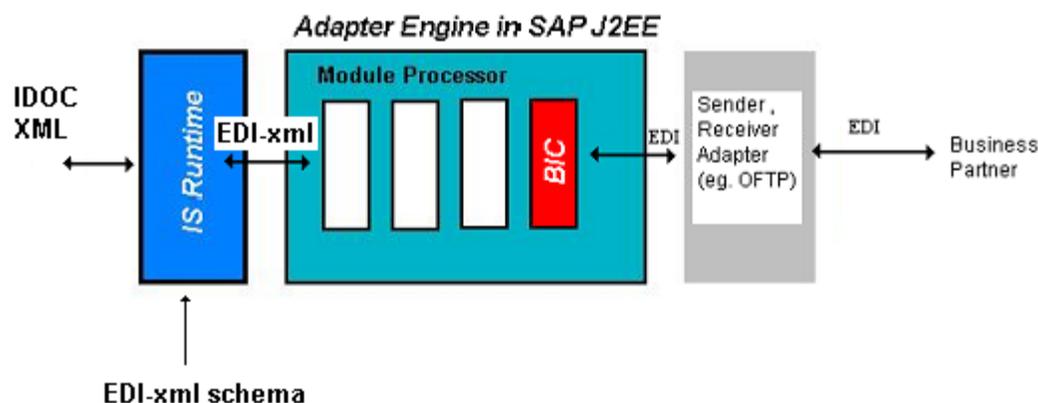
Features

BIC MD allows the creation of two mapping types:

1:1 mappings

These mappings are conversions from XML to EDI documents and vice versa. The mappings are automatically created and perform a simple 1:1 mapping process, with each source field in the XML (or EDI) document having an equivalent field in the destination EDI (or XML) document.

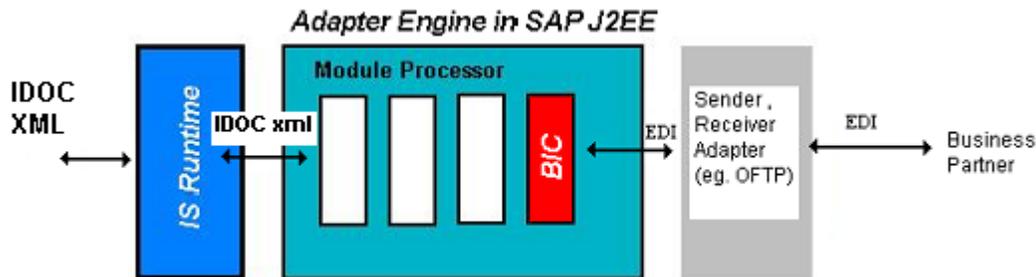
The following diagram shows a possible use of a 1:1 mapping. The XI Server receives an IDOC XML document, which is mapped to an EDI-XML structure using the XI Mapper. The EDI-XML structure is an XML representation of the EDI document. This means, it contains the same fields and values as the final EDI document. As a final step, before transmission, the EDI-XML document is mapped to the EDI document using the BIC Adapter. Using this approach the XI Mapper must be configured to convert from IDOC to EDI- XML format, while the final mapping EDI-XML to EDI can be created automatically and does not require deep knowledge of BIC MD.



Regular BIC Mappings

These mappings are created manually and allow the employment of the BIC programming language to perform complex operations during the mapping process

The following diagram shows a possible use of a regular mapping. The XI Server gets an IDOC XML document that is passed directly to the BIC adapter, which will convert it to an EDI document. In this case, the XI Mapper does not perform any task, but the BIC adapter will require a complex mapping.



Creating 1:1 Mappings

Use

This feature is used to create 1:1 XML to EDI mapping (or EDI to XML). These mappings are made of:

- The XML schema of the EDI- XML document (XML representation of the EDI document)
- The mapping itself

These mappings can be created automatically and perform a 1:1 mapping between source and destination fields.

Requirements

Before creating 1:1 mappings, the following steps have to be carried out:

1. Install BIC Mapping Designer (see Seeburger's Business Integration Server documentation).
2. After installation, the GeneratorMappings project should contain the following service mappings:

- CreateMappingEdiToXml
- CreateMappingXmlToEdi
- CreateXMLMessage

Actions

Now BIC MD will be used to create all data structures that BIC or the XI Server will need for converting XML documents into EDI documents.

Creating the XML Schema for the EDI Document

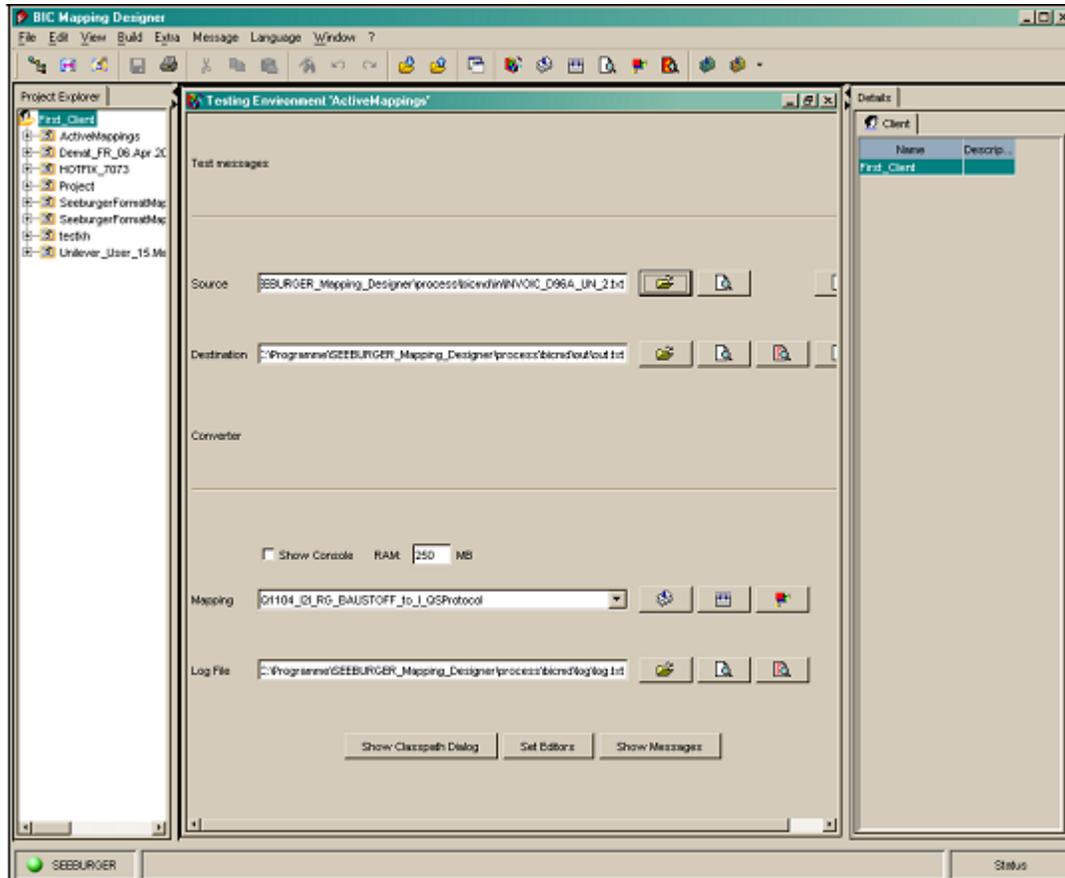
As already stated BIC MD will create an XML schema, which is an XML representation of the EDI. We assume the EDI message has already been imported into BIC MD. In this case, the process for creating the XML schema is:

1. Export the EDI document by selecting the document in the project window followed by the

menu item File|Export. In the screenshot an Edifact message called INVOIC_UN_D96A is being imported and will be saved to the file msg_INVOIC_UN_D96A.xml.

2. Configure the CreateXMLMessage mapping in the project GeneratorMappings as follows:

Click on the “Test Environment” button and select the mapping, the source and destination files should be filled out as shown in the screenshot:



Caution:

Destination file names starting with “See_” are forbidden!

3. Compile the mapping, if this has not been done yet and then run it (these are the buttons next to the Mapping list box.

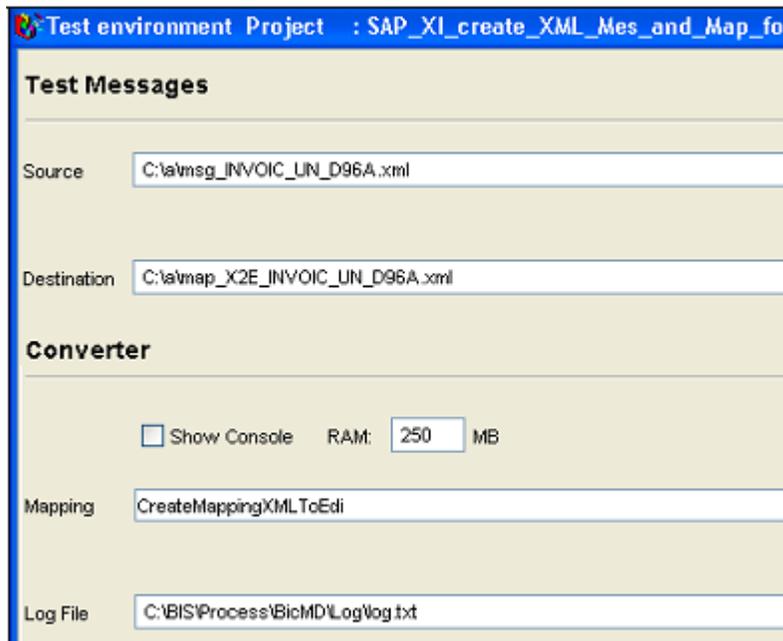
The msg_XML_name.xml file (msg_XML_INVOICE_UN_D96A.xml in the example) has now been created. This file is required for creating the 1:1 mapping and the XML schema for the XI Mapper. Import msg_XML_INVOICE_UN_D96A.xml using Main menu | File | Import.

4. Create the XML schema by selecting the XML_name (XML_INVOIC_UN_D96A) message and exporting it (select XSD in the export window).
5. This file can be imported in the XI Integration Builder: Design using the Interface Objects | External Definitions item. It can then be used in XI Mappings.

Creating XML-to/from-Edifact Mappings

Once the XML representation of the EDI document (file msg_XML_INVOIC_UN_D96A in our example) has been created, the 1:1 mapping can now be created.

1. Configure the CreateMappingXMLtoEdi mapping in the BIC MD project GeneratorMappings as follows: Click on the “Test Environment” button and select the mapping and the source and destination files as illustrated in the screenshot:



2. Compile the mapping if this has not been done yet and then run it (these are the buttons next to the Mapping list box).
3. The map_X2E_name.xml file (map_X2E_INVOIC_UN_D96A.xml) has now been created. These steps must be repeated with the CreateMappingEditoXml, in order to create the map_E2X_name.xml file.
4. Import both files , map_X2E_name.xml and map_E2X_name.xml into BIC MD using the menu item File | Import.

Note: At this point. The Test Environment window of BUC MD can be used for compiling and running the created mappings

5. Deploy the mappings as described in the next section.

Deploying Mappings

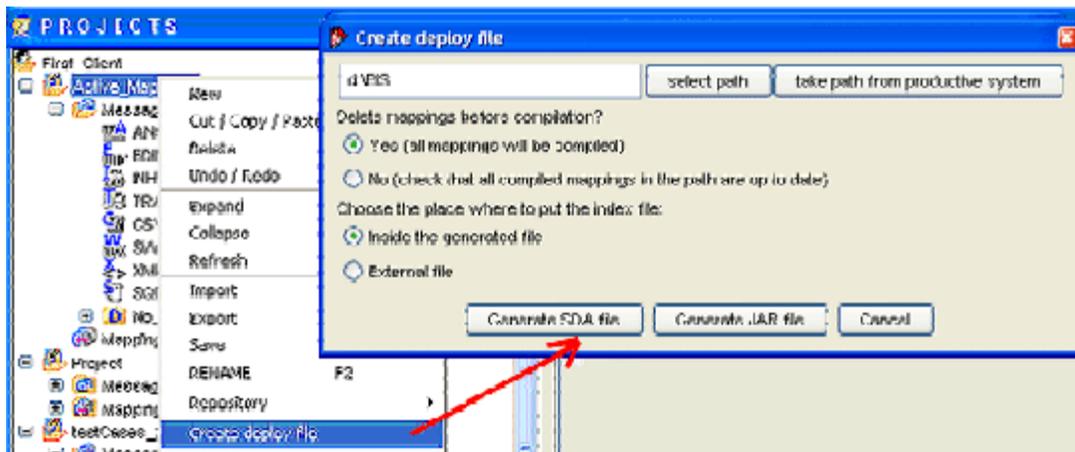
Use

Once the mappings have been created using BIC MD, either as 1:1 mappings or as regular mappings, a method of deploying them on the XI Server is required, so that they can be used by the BIC Adapter. Deployment files can be created with BIC MD, as these are required by SAP XI to perform this task.

Actions

Software Deployment Archives (SDA files) can be created by BIC MD in the following way:

1. Move or copy mapping to the Active mappings project by means of the copy & paste feature.
2. Select the ActiveMappings project.
3. Select the Create deploy file item in the popup menu (mouse's right button).



4. Enter the directory where the file is to be created and click on the Generate SDA button.

If the Yes option is selected, all previously created mappings will be deleted, compiled in a selectable directory and a SDA (i.e. jar) will be created. The created files will not be deleted afterwards. This may take several hours.

If the No option is selected, all compiled mappings in the selected directory will be copied to SDA (or jar). They will not be examined whether they are up to date. Individual classes can be updated/refreshed with the “Compile Mappings to Productive System” feature. For this, the “Productive System” path and the path that is input into the input field for SDA creation have to correspond.

It can be set automatically with the Take path from productive system button.

The SDA file will be called Mappings_user.sda.

5. A CSV file called filecontent.csv will be created either within Mappings_user.sda, or as an extra file in addition to the Mappings_user.sda.

1. Start the Software Deployment Manager and deploy the SDA file: Click on the + (Add) button; select the SDA file from the hard disk and click on Next.



Caution

The second one will overwrite the first one, if 2 Mappings_user.sda files are being deployed. Therefore be sure that all mappings are located in one BIC MD project (the ActiveMappings project) only and exported together in the same SDA file.

Additional XI Commands

It is possible to manipulate the Message Tracking database with the following commands and thus enables message tracking (MT). XIMessageID, XIDocumentID and XIPrimaryMTKey (XI Primary Message Tracking Key: comprised of XIMessageID and XIDocumentID) can be of greater interest. Follows an example of how to make this information accessible in a BIC mapping.

Example:

```
local xiMessageID$;
local xiDocumentID$;
local xiPrimaryMTKey$;
xiMessageID$ = getInputValue("xiMessageID");
xiDocumentID$ = getInputValue("xiDocumentID");
xiPrimaryMTKey$ = getInputValue("xiPrimaryMTKey");
```

The XIDocumentID with the xiDocumentID key is to be passed on as a module-parameter in the module-chain. The xiMessageID is automatically set in the BIC-mapping and the xiPrimaryMTKey is generated dynamically.

Additional parameters can be set in the module chain over key/value pairs and can be queried with the BIC-Basic command getInputValue(KEY) in the mapping. An example is shown in the figure below.

Module Configuration

Module Key	Parameter Name	Parameter Value
bic	mappingName	EdifactMapping
bic	destTargetMsg	MainDocument
bic	destSourceMsg	MainDocument
bic	anOwnParameter	valueForBICMapping

BIC-Basic code for reading out the anOwnParameter parameter in the BIC mapping:

```
local xiParameter$;
xiParameter$ = getInputValue("anOwnParameter");
```

You can also read and write XI message dynamic properties using the getInputValue and setOutputValue() functions. For more information, refer to the corresponding sections of the Business Integration Converter Adapter Configuration Guide.

The following XI Commands are described in detail in the BIC Mapping Designer Command Reference:

Command	Purpose
OpenMTDBConnection	Opens a connection to the Message Tracking database (if not yet opened). In case of an error a BIC runtime error is generated.
CloseMTDBConnection	Closes a connection to the Message Tracking database (if still open). In case of an error a BIC runtime error is generated
SetMTField	Fills a field/ column in the current message Tracking Database record. If no record is available, a new one is generated.
GetMTField	Gets a field/ column value from the current Message Tracking database record. If no record is available, a BIC error is generated.
WriteMTRRecord	Writes the current Message Tracking database record into the master Message Tracking database. If the record already exists, an update is executed, otherwise an insert.
WriteMTDetailRecord	Writes the current Message Tracking database record into the detail Message Tracking database. If the record already exists, an update is executed, otherwise an insert.
ReadMTRRecord	Reads the current Message Tracking database record from the master Message Tracking database.
ReadMTDetailRecord	Reads the current message tracking database record from the detail Message Tracking database
ExecuteMTSQLCommand	Executes an SQL command in the Message Tracking database.
ClearMTRRecord	Clears the fields of the current Message Tracking record.

Related Content

[SAP PI/XI : Content conversion using Generator mapping functionality of SeeBurger : Part 1.](#)

[SAP PI/XI : Content Conversion using Generator mapping functionality of SeeBurger : Part 2.](#)

[Handling EDI interfaces using Seeburger BIC modules](#)

For more information, visit the [SOA Management homepage](#).

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.