Chapter 2. Mapping

Chapter 2 deals with mapping development on both stacks (ABAP, JAVA). Here you will find information on how to test your mappings in real life scenarios in order to avoid surprises later on, as well as a report for testing ABAP mappings which you are likely to use instead of the standard transaction for testing ABAP mappings. You will also find a tip on how to throw exceptions with information about the cause of the error inserted directly into the error segment of the PI message header. For those of you using the latest edition of SAP Process Integration, we have included an article about the latest functionality – parameterized mappings, together with a simple trick that helps with checking if your mappings will still work after the upgrade.

2.1 How to test your mapping (in real life scenarios)

Article

This article shows how to correctly perform real life mapping tests within the SAP NetWeaver PI environment.

This article shows a very simple path on how to test them with two scenarios:

1. if you're not using BPM (integration process)
2. if you are using BPM

When you send the data to XI you might see that your mapping program doesn't always work the way you'd like it to work. Very often it's because either your Data Type or your inbound message does not have a correct format. Your mapping program appears to work when you test it from the Test Tab (inside Integration Repository), but you wonder why it shows mapping errors when you send a real message.

The best and possibly only way to resolve such errors is to use the **REAL** inbound message.

Testing mapping programs without BPM

Step 1 (Figure 67)

- go to TCODE - SXMB_MONI
- choose your message
- choose the payload of the inbound message

Figure 67

Step 2 (Figure 68)

- copy the source XML of your message

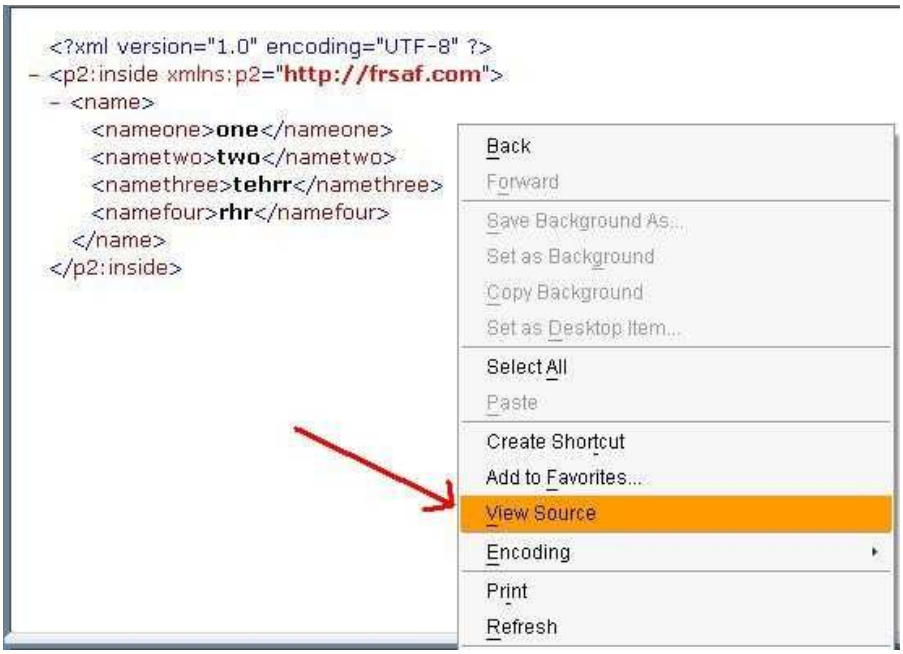

Figure 68

Step 3 (Figure 69)

- paste the source of your message into your mapping program

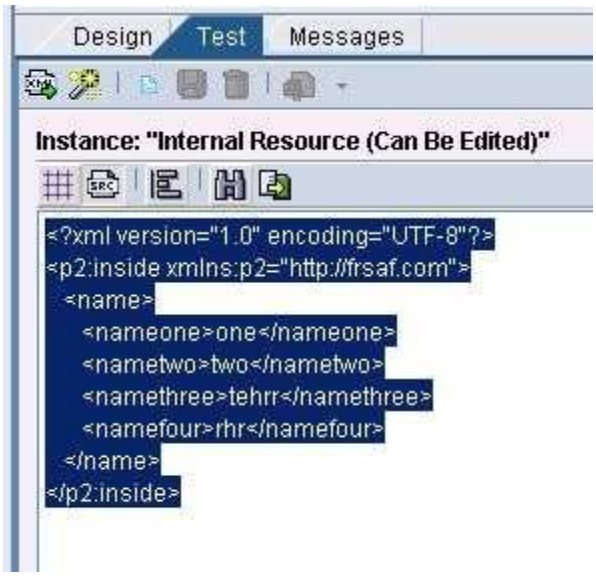

Figure 69

Now you can be sure that you test a correct message.

Testing mapping programs with the use of BPM

Step 1 (Figure 70)

- open the **Technical workflow log** of your BPM
- choose **Show container** option (from your mapping step)
- choose your inbound message

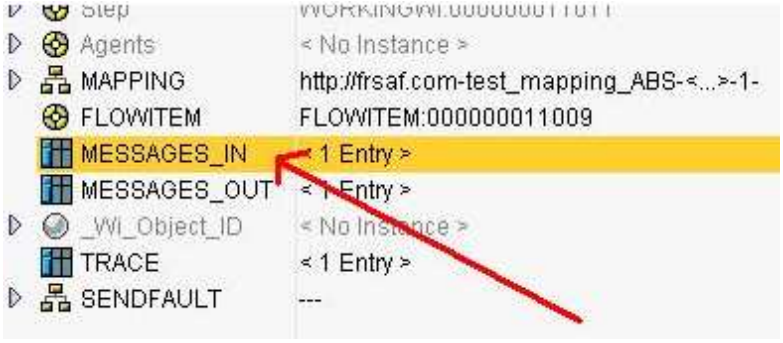

Figure 70

Step 2 (Figure 71)

- open your message and follow the same steps as for testing without BPM

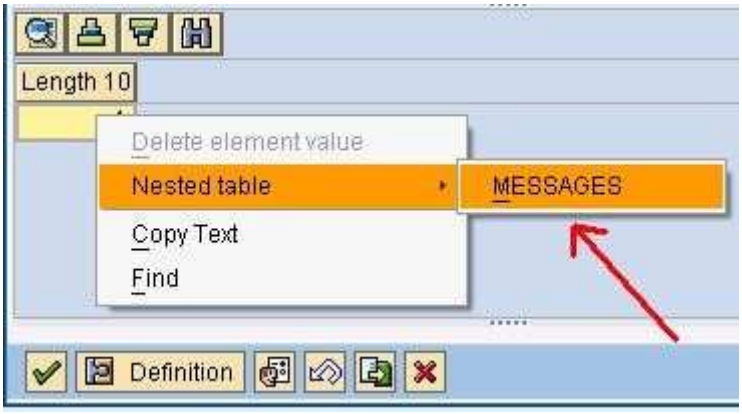

Figure 71

With these few steps you can ensure that the message you test has a correct format - exactly the same as the message that was sent to your adapter, so you can easily debug it inside your mapping program.

End Notes

Basically there are no other ways to test your mapping. Creating a sample file, which should emulate the real system, can be done during the start of development. The more realistic the sample, the greater the chance that the mapping will work as it should. Hence it is good practice to test mappings as outlined in this article.

2.2 XML node into a string with graphical mapping?

Article

This article shows how you can transform an XML node into a string with graphical mapping.

There are times when you need to send a XML message inside a XML tag of the other message. There are at least two approaches when dealing with such a scenario:

- you can change the XML message's tags into something else and change it back later.
- you can include the XML message inside a CDATA tag.

Graphical mapping can still be used, but we have to combine it with any other kind of mapping. PI/XI gives us the option of using many mappings with one transformation - **Interface mapping**. Bear in mind that **Source message** to **Target message** transformation can only be done once. The rest of the transformations are **Target message** to **Target message** only (Figure 72).

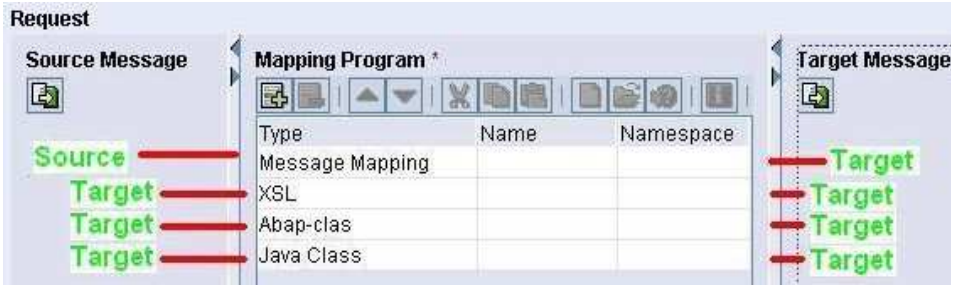

Figure 72

In this case we want to do the whole mapping inside the first graphical mapping and then just transform the result a little.

So how do we go about this?

- use graphical mapping to transform our source message to our target message (target message contains all tags - in our example it looks exactly the same as the source message).
- then we want to put the WHOLE "name" node inside the "name" tag within CDATA.

The first part is simple, but how do we complete the second part?

```
<?xml version='1.0' ?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:p2="http://frik.bcc.com.pl">
<xsl:template match="/">
<p2:inside xmlns:p2="http://frik.bcc.com.pl">
<name>
```