

Chapter 6. Non categorized articles

In this chapter you will find a collection of articles that did not fit specifically into any category or applied to more than one. This includes information about industry adapters (Rosetta net), restarting successfully processed messages (in development environments) or sending messages without the use of an adapter. SAP Process Integration version 7.1 released in 2009 is currently the most recent version available. This chapter will also discuss details on some of the new functionalities together with some strategies for objects naming conventions already designed for SAP PI 7.1. One of the highlights of this chapter is an article showing how to use fault messages correctly with ABAP proxy implementations – something that every SAP PI developer needs to know, as ABAP proxy interfaces are becoming more and more popular and will likely replace the well known IDOCs in the future.

6.1 Usage of Fault Messages

Article

Fault messages are designed for application-specific errors that occur on the inbound side and are reported back to the sender or recorded in monitoring. Fault message can be used in both asynchronous and synchronous scenarios. This article will explain the use of fault messages based on an ABAP proxy scenario.

The use of fault messages will be described using an ABAP proxy scenario, focusing on asynchronous communication.

Note: An Extract from SAP online help says "The fault message of the application for an asynchronous ABAP server proxy is persisted for monitoring. In the case asynchronous of Java server proxies, the fault message is part of a negative application acknowledgement."

Let us begin with the scenario of a File -> XI -> ABAP Proxy. The scenario is designed in such a way that the file will contain a SAP user name which will be passed to the ABAP proxy via XI. Inside the ABAP proxy, two types of checks are done to trigger the fault message:

- There is no entry for the user in the file.
- The user specified in the file does not exist in the USR01 table.

The Development:

Data Type (Figure 155):

DT_UserName

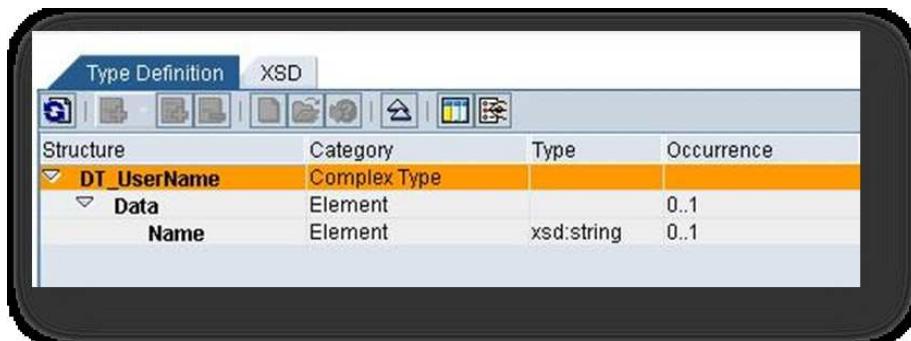


Figure 155

Message type (Figure 156):

- MT_UserNameFromFile
- MT_UserName

Use DT_UserName for these message types.

Fault Message type

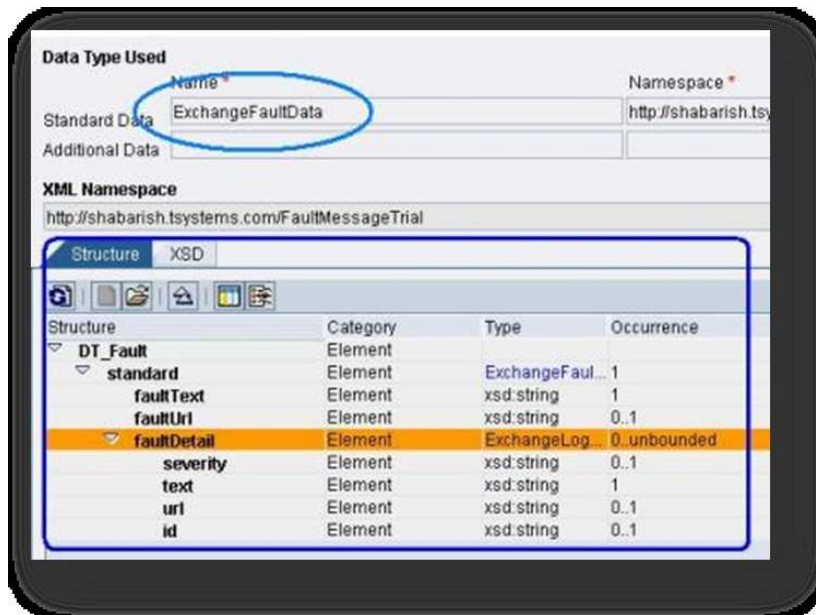


Figure 156

Message Interface (Figure 157):

- MI_UserNameFile

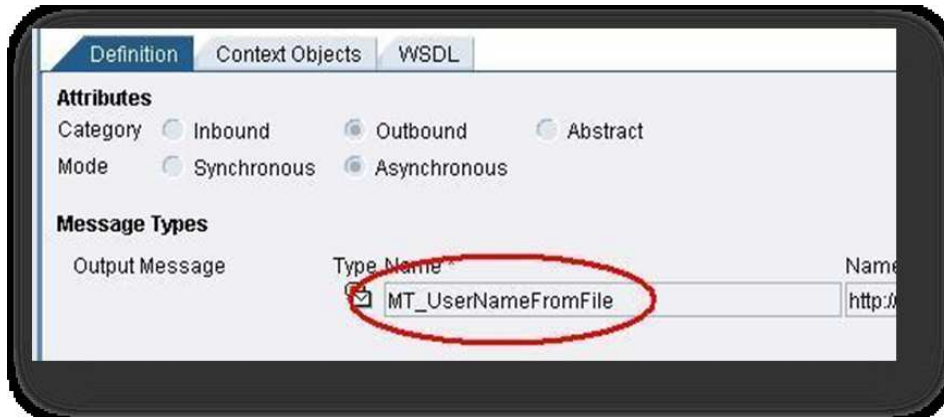


Figure 157

- MI_UserNameAndException (Figure 158)

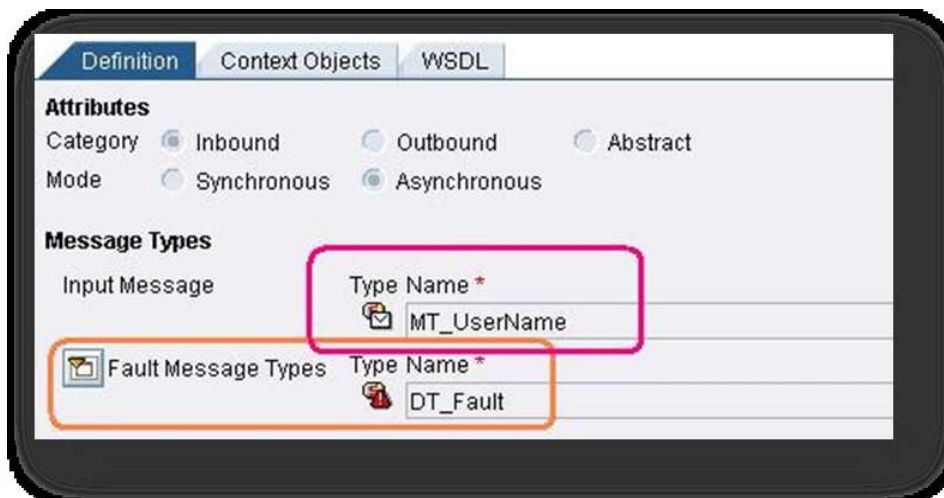


Figure 158

Create the message mapping between MT_UserNameFromFile & MT_UserName, as well as the corresponding Interface mapping.

For the configuration, configure as any File to Proxy scenario.

In the R3, generate proxy for the inbound interface, MI_UserNameAndException.

Figure 159 below shows that the fault message type that was used in the inbound interface is reflected in the exceptions tab under 'Methods', in the Class generated for the inbound proxy.

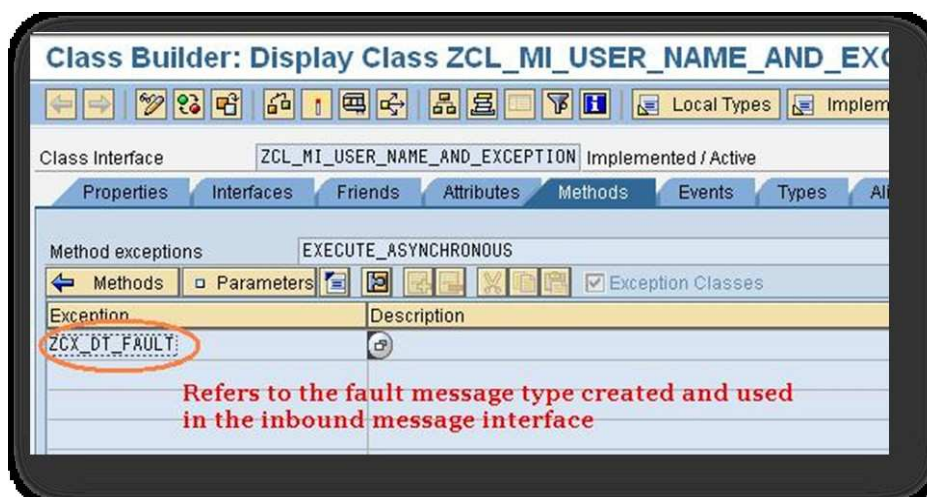


Figure 159

ZEXCHANGE_FAULT_DATA inside ZCX_DT_FAULT (Figure 160)

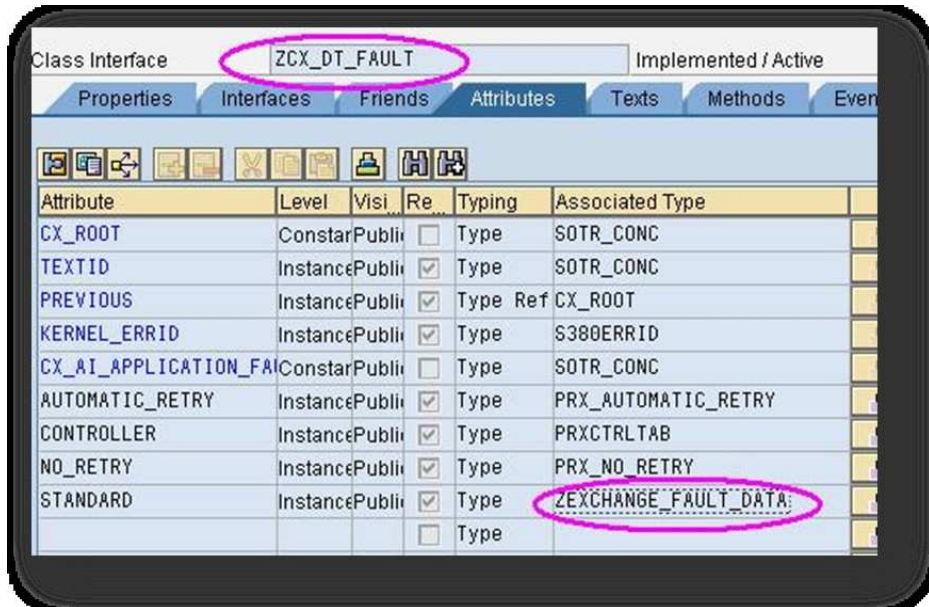


Figure 160

ZEXCHANGE_FAULT_DATA structure inside the proxy (Figure 161)