SAP® SOA Integration – Enterprise Service Monitoring

- Tools for Monitoring SOA Scenarios
- SAP Application Interface Framework (AIF) Customization Best Practices
- Forward Error Handling (FEH) and Error Conflict Handler (ECH) Configuration Tips
- Detailed Message Monitoring and Reprocessing Examples
# Table of Contents

**Foreword**  
7

**Preface**  
9

## 1 What is SOA integration monitoring and error handling?  
13  
1.1 SOA Integration  
13  
1.2 SOA monitoring and error handling  
15  
1.3 Tools for monitoring SOA scenarios  
20

## 2 Local Integration Engine Monitor  
25  
2.1 Activation and initial configuration  
25  
2.2 Message monitoring and error monitoring  
25  
2.3 Message reprocessing  
28  
2.4 Content-based Message Search  
30

## 3 Forward Error Handling and Error and Conflict Handler  
39  
3.1 Concept  
39  
3.2 Activation and initial configuration  
41  
3.3 Message monitoring and error monitoring  
42  
3.4 Message reprocessing  
54  
3.5 Content-based message search  
59
# Table of Contents

## 3.6 Worklists

## 3.7 Object linking

## 4 SAP Application Interface Framework

### 4.1 Activation and initial configuration

### 4.2 Additional configuration for Interface Monitor

### 4.3 Interface Monitor

### 4.4 Message monitoring and error monitoring

### 4.5 Hints for error monitoring

### 4.6 Content-based message search

### 4.7 Object linking

### 4.8 Hiding structures

### 4.9 Defining changable fields

### 4.10 Snapshots

## 5 Summary

## A The Author

## B Index

## C Disclaimer
2 Local Integration Engine Monitor

Local Integration Engine Monitor is the most well known and most commonly used application for monitoring SOA integration scenarios in backend application systems. There’s very little configuration required to make it run, but it has some features which can make it much easier to use. We will explore those features in this chapter.

2.1 Activation and initial configuration

In standard SAP, the Local Integration Engine Monitor does not require any activation and both kinds of asynchronous web services (mediated and point-to-point) will run through this monitor.

2.2 Message monitoring and error monitoring

Message monitoring and error monitoring is done with the same tool—transaction SXMB_MONI • MONITOR FOR PROCESSES XML MESSAGES, or directly from transaction SXI_MONITOR.

There are a number of standard selection criteria like sender/receiver interface name, namespace, system, as well as some more advanced selection criteria like Message ID, Error ID, etc. (see Figure 2.1).
After we run the selection, we can see all of the messages. If we double-click on any of them, we can see the contents of the message in the **INBOUND MESSAGE • PAYLOADS • MAIN DOCUMENT** section (see Figure 2.2).

In the event of an error, we will be able to see the red flag and short status on the first screen of the application.
If the web service was using a fault message type and was programmed correctly, then we can see some additional details by clicking on the red flag as shown in Figure 2.3 below.

**Figure 2.3: Error monitoring**

Fault message types are structures in which detailed information on an error is propagated to. All custom asynchronous web services should always use fault message types as this facilities message monitoring (among other things).

A more detailed error description, or any other error information, can always be found by drilling down into the message content (double-click on the message) and by opening the error section of the SOAP HEADER as shown in Figure 2.4.
As we can see, message and error monitoring with the use of Local Integration Engine Monitor is primarily done with the use of XML messages. The expectation is that the user knows how to work with XML standard. For that reason, it is not very user friendly compared to other SAP SOA integration monitoring tools.

2.3 Message reprocessing

Message reprocessing is one of the most typical tasks during day-to-day work with SOA integration scenarios. Messages tend to fail for different reasons and they either need to be reprocessed (which in some cases means that the content of the message also needs to be changed), or cancelled. A message which cannot or should not be reprocessed needs to be cancelled. A cancelled message is the only identification that an error was checked and proper action was taken. You can cancel messages directly from monitoring transaction by selecting the message that you want to cancel and EDIT • CANCEL MESSAGE WITH ERRORS from the menu (see Figure 2.5).

It's also possible to cancel messages from a special program called RSXMBCANCEL_MESSAGES. In the event that you need to cancel a lot of messages, it is advisable to run it with this program.
Each time you use this special program to cancel messages, you should run it first in the test mode to see if the selection criteria you’ve used selects the correct messages. Only after you’re sure that the selection was correct, run the program once more without test mode turned on.

If we know that the message can be reprocessed manually, which means that the error should no longer happen, we can deal with it similarly as cancelling the message. We need to select the message and click on the RESTART button in the menu (see Figure 2.6).
B  Index

/A
/AIF/CUST_FUNC 115
/AIF/CUST_HINTS 101
/AIF/CUST_IF 71
/AIF/ERR/ 87
/AIF/ERROR_CHANGE
   _REPORT 117
/AIF GENERATE_MSG
   _STAT_SNAP 119
/AIF/IDXTBL 108
/AIF/IFMON 82
/AIF/PERS_CGR 97
/AIF/REP_AC_ASGN
   96, 97
/AIF/REP_AC_DEF 96
/AIF/RESTART_MSG
   97
/AIF/STD_IDX_TBL
   103
/AIF/TEMPL_RESTART
   _AUTO_REPR 97
/AIFX/ERR_WEB 82, 95
/SAPPO/WL_CHANGE
   64

A
acknowledgment 40

Application
Integration
Framework 20
Application Interface
Framework 23

B
Backend application
system 17

E
ECH_MONI_SEL 44
Enabler 69
Enterprise service bus
17
Error and Conflict
Handler 22

F
Forward Error Handling
22, 39
Forward Error
Handling/Error and
Conflict Handler 20

H
Hierarchical Derivation
Service 41
I
index table 103
Interface Monitor 82

L
Local Integration
Engine monitor 20

M
Mediated SOA
integration scenario 18
multiple selection index
table 110

P
Persistent mode 57
Post Processing Office 41
Post-mapping payload 50
Pre-mapping payload 50

R
RSXMB_CANCEL_ME
SSAGES 28

S
Service Interface 46
Service oriented
architecture 13
SOA integration
scenario 16
SXI_MONITOR 25
SXMS_EXTRACT_MESSAGES 34
SXMS_LMS_CONF 31
Synchronous scenario 16

T
Transient mode 57

W
Worklist 62