

SOA Suite for healthcare integration Series

Externalise, Repair and Resubmit “Broken” Messages

michael@czapski.id.au

January 2013

Table of Contents

Introduction	1
Solution Overview.....	2
Start the “Receiving” “External Systems”	2
Clear Instances from the Repository.....	3
Send an invalid ADT message	4
Externalise invalid ADT message’s payload	6
“Repair” the message	7
Resubmit the message.....	8
Summary	10

Introduction

Every now and then a situation may arise where a message sent to a downstream system is rejected or a message arriving from an external system, is rejected. Some of the possible issue could be “fixable” and others not. If message is “fixable”, in the sense that its content or structure can be modified and the re-submitted modified message will be accepted, then the operator must follow the steps necessary to extract, repair and resubmit the message. In other circumstances it may be necessary to resubmit a specific batch of messages to an external system, for a particular time period or of particular kind.

In this article we will walk through the process of externalizing a message, “fixing” it and resubmitting it for processing. We will also briefly touch on other uses of the resubmit utility.

More details on the technical aspects of this topic can be found in the Oracle “Technical Note: 11g_005 Command-Line Utilities”, section “Resubmit Utility”, available at <http://www.oracle.com/ocom/groups/public/@otn/documents/webcontent/326860.pdf>.

This article assumes that the reader has the SOA Suite for healthcare integration environment with all necessary components installed and ready to use. The Bill of Materials for such an environment and a discussion on where the components can be obtained is provided in the earlier article, “SOA Suite for healthcare integration Series - Overview of the Development Environment”, to be found at

<http://blogs.czapski.id.au/2012/08/soa-suite-for-healthcare-integration-series-overview-of-the-development-environment>.

This article assumes that the reader completed the solution discussed in the earlier article, "SOA Suite for healthcare integration Series – Exception Handling – Processing Endpoint Errors", to be found at

<http://blogs.czapski.id.au/2013/01/soa-suite-for-healthcare-integration-series-exception-handling-processing-endpoint-errors>.

Solution Overview

All infrastructure components, except the tool which we use for "fixing" a "broken" message, are a part of the SOA Suite for healthcare integration environment. We will not build any applications or configure any components. Even though this is the case, the exercise relies on the availability of a functioning solution which was developed in all the articles leading up to and including the "SOA Suite for healthcare integration Series – Exception Handling – Processing Endpoint Errors".

Start the "Receiving" "External Systems"

We will use the CMDHL7Listener command line client to receive HL7 ADT messages and look at the output in the output directory specified on the listener's command line – for me c:\hl7\received. Each CMDHL7Listener will display a trace of message exchange in the console window. The SOA Suite for healthcare integration will record message tracking information which we will also look at.

Please note that in this solution the CMDHL7Listener returns an ACK as soon as it gets the message.

- Check that your configured output directory is empty
- To start the listener for the "HA08_SystemA_Out" sender endpoint, in a command / terminal window execute the following command

```
java -jar c:\tools\CMDHL7\CMDHL7Listener_v0.7.jar -c ID_ -p 22100 -s c:\hl7\received
```

- To start the listener for the "BcastB_SystemA_Out" sender endpoint, in a command / terminal window execute the following command

```
java -jar c:\tools\CMDHL7\CMDHL7Listener_v0.7.jar -c ID_ -p 22200 -s c:\hl7\received
```

- Inspect the CMDHL7Listener console output making sure the listener started and is listening on the appropriate port

```
08/12/2012 10:45:36 AM au.id.czapski.hl7.CMDHL7Listener main
INFO: Port: 22200
08/12/2012 10:45:36 AM au.id.czapski.hl7.CMDHL7Listener main
INFO: Store in: c:\hl7\received
08/12/2012 10:45:36 AM ca.uhn.log.HapiLogImpl info
INFO: au.id.czapski.hl7.SimpleACKApplication registered to handle **
messages
08/12/2012 10:45:36 AM ca.uhn.log.HapiLogImpl info
INFO: SimpleServer running on port 22200
```

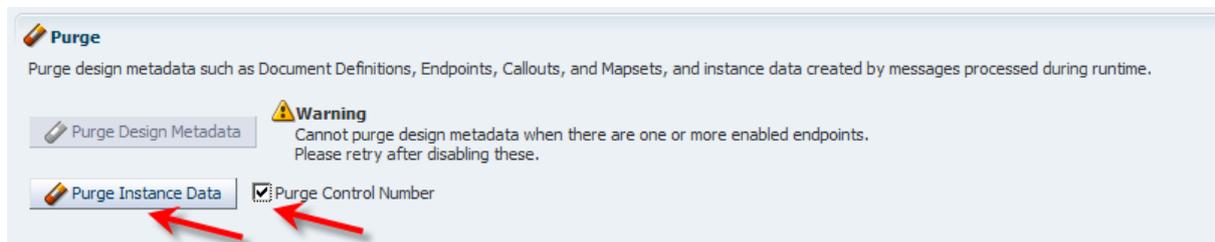
Clear Instances from the Repository

To start with a “clean slate” we will use the Healthcare Integration Console to clear out all old message tracking information. Bear in mind that this is irreversible and one should think carefully about clearing instance repository in production systems.

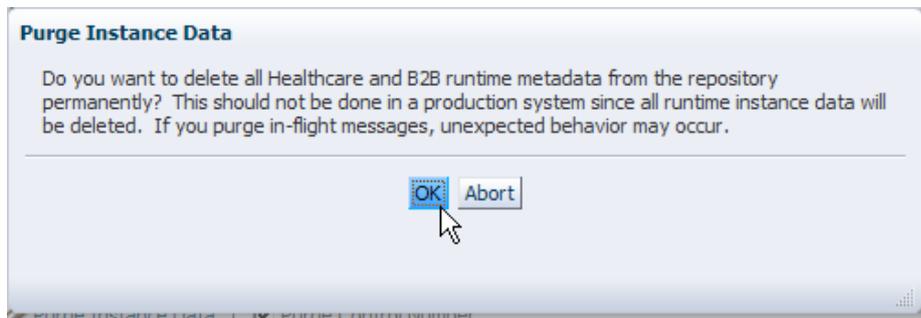
- Start the Healthcare Integration Console, <http://localhost:7001/healthcare>, and log in with your credentials, perhaps weblogic/welcome1
- Click “Designer” Tab, click “Administration” Tab and double-click on the “Repository Management” node



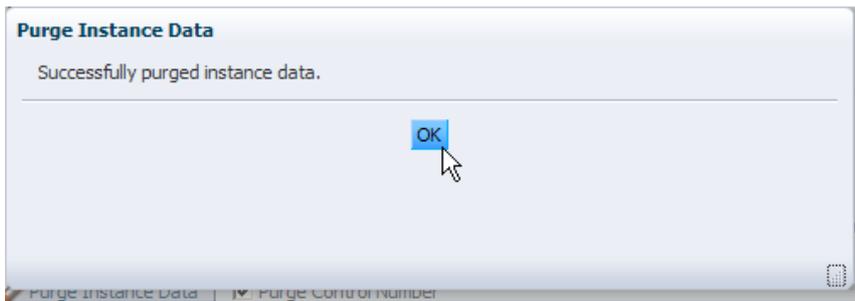
- Check the “Purge Control Number” checkbox, if appropriate, and click the “Purge Instance Data” button



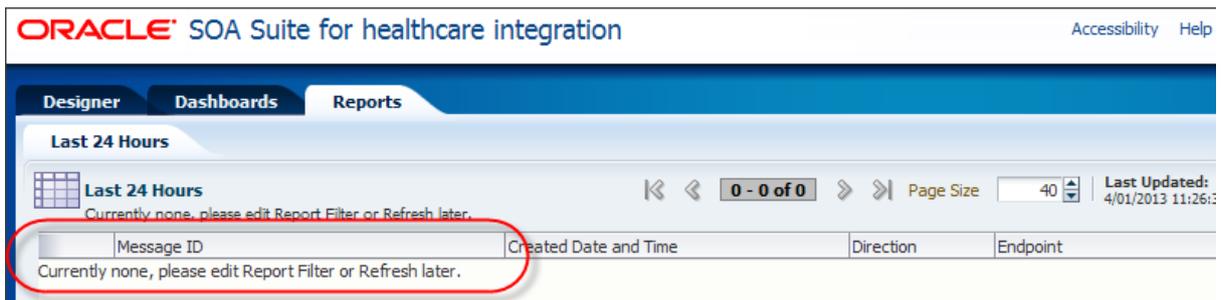
- Click “OK” to purge messages and message tracking information from the repository



- Once the process is completed, dismiss the wizard by clicking “OK”



- Click on the "Reports" Tab, look at the last 24-hours report and note that there are no messages in the repository



Send an invalid ADT message

We will use the CMDHL7Sender command line client to read a file containing a single HL7 ADT message, which is invalid, and submit it to the ADT Receiver endpoint. We will look at the exception file in our configured output directory – for me c:\hl7\received, and then review message tracking information in the Healthcare Integration Console.

- Copy the ADT_A01_output_1.hl7 file and give a copy a new name ADT_A01_broken.hl7 – if you have this file from the previous article's exercise use that
- Using a text editor which does not change line endings, for example "Notepad++", edit the message and change the name of the EVN segment to read EVX instead. An example is shown below.

```
MSH|^~\&|SystemA|HosA|PI|MDM|2008090801529||ADT^A01|000000_CTLID_2008
090801529|P|2.3.1||AL|NE
EVX|A01|2008090801529||JavaCAPS6^^^^^^USERS
PID|1||A000010^^^^HosA^MR^HosA|Kessel^Abigail||19460101123045|M|||7
South 3rd Circle^^Downham Market^England -
Norfolk^30828^UK|||||A2008090801529
PV1|1|I||I||FUL^Fulde^Gordian^^^^^^^^^^MAIN||EMR||||||V200809080
1529^^^^VISIT|||||||||||||||||2008090801529
```

- In a command / terminal window execute the following command

```
java -jar c:\tools\CMDHL7\CMDHL7Sender_v0.7.jar -a SystemA -b HosA -c ID_
-n 1 -d \r\r\n -p 22222 -h localhost -t 30000 -f
c:\hl7\adt\sources\ADT_A01_broken_1.hl7
```

- Inspect the Negative ACK which the command line tool displays in the console window

```
MSH|^~\&|PI|MDM|SystemA|HosA|20130103||ACK^A01|1001|P|2.3.1
MSA|AE|ID__000000
```

```
ERR|^^^100&0x810070:Rejecting:Unrecognized data was found in the data
file. The last~^^^100&0x810070:Rejecting:Unrecognized data was found in
the data file. The last~EVN^3^^100&0x81004d:Rejecting:Segment EVN (EVN
- event type segment) at guidelin~PID^3^^100&0x81004d:Rejecting:Segment
PID (PID - patient identification segment)
```

- Inspect the content of the exception file, ipException_nn.xml – for me it looks like the text below

```
<?xml version="1.0" encoding="UTF-8" ?><Exception
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://integration.oracle.com/B2B/Exception">
  <correlationId>null</correlationId>
  <b2bMessageId>C0A8E9E913BFE096F4E000001BB82D63-1</b2bMessageId>
  <errorCode>B2B-51507</errorCode>
  <errorText>
    Validator error - Extra data was encountered.
    Validator error - Extra data was encountered.
    A data segment with 'Mandatory' status is missing.
    A data segment with 'Mandatory' status is missing.

  </errorText>
  <errorDescription>
    Machine Info: (R1PS5HCI)
    Unrecognized data was found in the data file. The last known Segment was MSH at
    guideline position 001.{br}{br}This error was detected at:{br}{tab}Segment
    Count: 2{br}{tab}Characters: 80 through 125
    Unrecognized data was found in the data file. The last known Segment was MSH at
    guideline position 001.{br}{br}This error was detected at:{br}{tab}Segment
    Count: 3{br}{tab}Characters: 126 through 275
    Segment EVN (EVN - event type segment) at guideline position 002 is missing. This
    Segment's standard option is 'Required'.{br}{br}This Segment was expected
    after:{br}{tab}Segment Count: 3{br}{tab}Character: 275
    Segment PID (PID - patient identification segment) at guideline position 003 is
    missing. This Segment's standard option is 'Required'.{br}{br}This Segment was
    expected after:{br}{tab}Segment Count: 3{br}{tab}Character: 275

  </errorDescription>
  <errorSeverity>2</errorSeverity>
  <errorDetails>
    <parameter name="hc.messageId"
    value="C0A8E9E913BFE096F4E000001BB82D63-1"/>
    <parameter name="hc.documentTypeName" value="ADT"/>
    <parameter name="hc.documentProtocolVersion" value="2.3.1"/>
    <parameter name="hc.documentDefinitionName" value="CMM_v1.0"/>
    <parameter name="hc.documentProtocolName" value="HL7"/>
    <parameter name="hc.messageType" value="1"/>
    <parameter name="hc.fromEndpoint" value="HosA_SystemA_In"/>
  </errorDetails>
</Exception>
```

- In the "Healthcare Integration Console" switch to "Reports" Tab and select the entry with the state of "MSG_ERROR"

ORACLE SOA Suite for healthcare integration Accessibility Help Logout Logged in as weblogic since 3/01/2013 10:51:14 AM GMT+11:00

Designer Dashboards Reports

Last 24 Hours

Last 24 Hours Found 6 results.

Message ID	Created Date and Time	Direction	Endpoint	State	Document Type
1 > COA8E9E9138FE09705800001BB82D68	3/01/2013 12:32:09 PM GMT+11:00	OUTBOUND	HosA_SystemA_In	MSG_COMPLETE	HL7-2.3.1-ACK
2 > COA8E9E9138FE096F4E000001BB82D63-1	3/01/2013 12:32:07 PM GMT+11:00	INBOUND	HosA_SystemA_In	MSG_ERROR	HL7-2.3.1-ADT

HosA_SystemA_In → Wire Message → Business Message → Application Message → Internal Delivery Channel

Business Message Details

Identification

Endpoint: HosA_SystemA_In Message ID: COA8E9E9138FE096F4E000001BB82D63-1
 State: MSG_ERROR Label: soa_b2b_ - Thu Jan 03 12:28:43 EST 2013 - 2
 Direction: INBOUND Interchange Control Number
 Message Type: REQ Group Control Number
 ID: COA8E9E9138FE096F5E000001BB82D64 Transaction Set Control Number: ID__0000000

Communication and Protocol

Document Protocol: HL7 Sent Date: 3/01/2013 12:32:07 PM GMT+11:00
 Protocol Version: 2.3.1 Received Date: 3/01/2013 12:32:07 PM GMT+11:00
 Document Type: ADT Acknowledgement Mode: NONE
 Document Definition: CMM_v1.0 Response Mode: ASYNC
 Document Retry Interval: 0 Transport Protocol Name: HLLP
 Document Remaining Retry: 0 Transport Protocol Version: 1.0

Content

Native Message Size: 397 Translated Message Size: 0

Payload Key Fields

- Click at the "Error" "button" between the "Write Message" and the "Business Message" and inspect the exception

ORACLE SOA Suite for healthcare integration Accessibility Help Logout Logged in as weblogic since 3/01/2013 10:51:14 AM GMT+11:00

Designer Dashboards Reports

Last 24 Hours

Last 24 Hours Found 6 results.

Message ID	Created Date and Time	Direction	Endpoint	State	Document Type
1 > COA8E9E9138FE09705800001BB82D68	3/01/2013 12:32:09 PM GMT+11:00	OUTBOUND	HosA_SystemA_In	MSG_COMPLETE	HL7-2.3.1-ACK
2 > COA8E9E9138FE096F4E000001BB82D63-1	3/01/2013 12:32:07 PM GMT+11:00	INBOUND	HosA_SystemA_In	MSG_ERROR	HL7-2.3.1-ADT

HosA_SystemA_In → Wire Message → Business Message → Application Message → Internal Delivery Channel

Error Details

Error Text: Validator error - Extra data was encountered. Validator error - Extra data was encountered. A data segment with 'Mandatory' status is missing. A data segment with 'Mandatory' status is missing.

Error Code: B2B-51507

Error Severity: ERROR

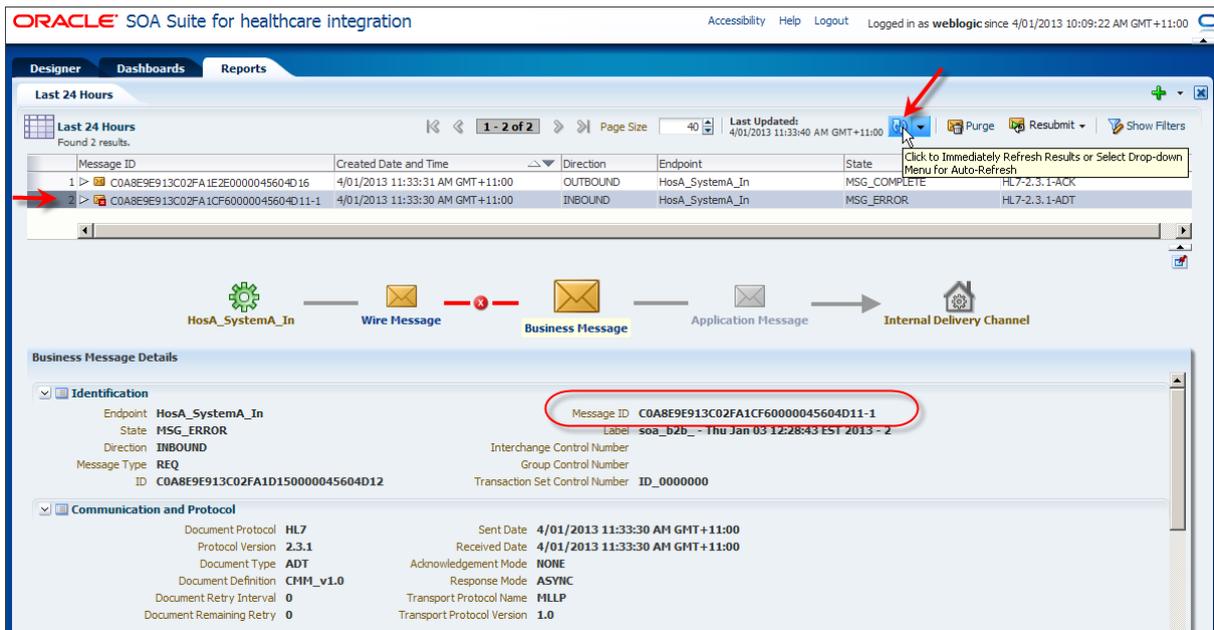
Error Level: ERROR_LEVEL_COLLABORATION

Error Description: Machine Info: (R1PSSHCI) Unrecognized data was found in the data file. The last known Segment was MSH at guideline position 001. This error was detected at: Segment Count: 2 Characters: 80 through 125 Unrecognized data was found in the data file. The last known Segment was MSH at guideline position 001. This error was detected at: Segment Count: 3 Characters: 126 through 275 Segment EVN (EVN - event type segment) at guideline position 002 is missing. This Segment's standard option is 'Required'. This Segment was expected after: Segment Count: 3 Character: 275 Segment PID (PID - patient identification segment) at guideline position 003 is missing. This Segment's standard option is 'Required'. This Segment was expected after: Segment Count: 3 Character: 275

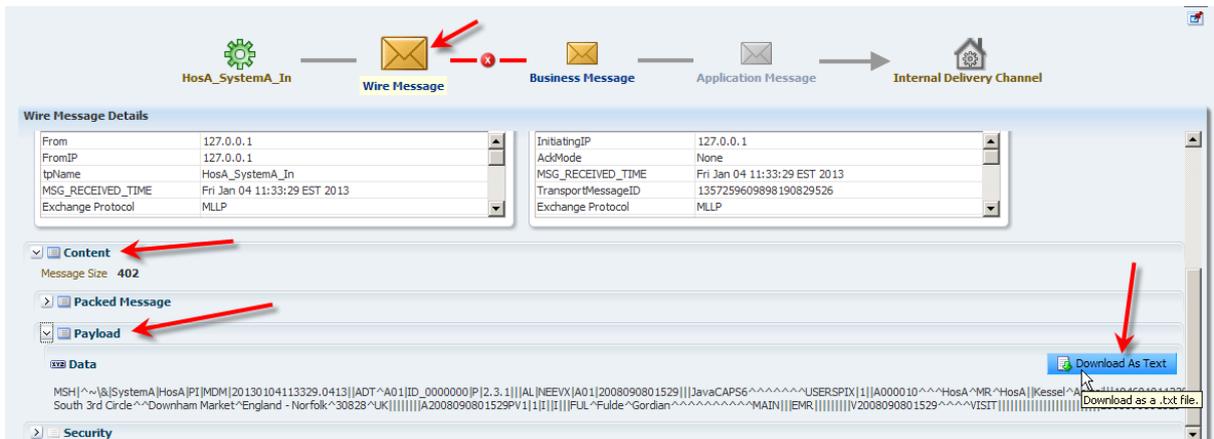
Externalise invalid ADT message's payload

To "repair" the message we need to extract the payload into a file in the file system.

- In Healthcare Integration Console, Reports Tab, refresh the display by clicking the "Refresh" button, select the message in MS_ERROR state by clicking on it, locate the Message ID and copy the ID to the clipboard or to a file for later use



- Click on the “Wire Message” icon, scroll the display to show the “Payload” section, expand and select the “Payload” node, click on the “Download as Text” button and save the payload as default “wmPayload.txt” file, perhaps in the file system directory C:\hl7\repair – *I work on Windows for this series of articles – if you work on another platform adjust file paths as appropriate*



- If it helps you, change the name of the file so that it includes the message id copied to the clipboard earlier – perhaps c:\hl7\repair\wmPayload_COA8E9E913C02FA1CF60000045604D11-1.txt

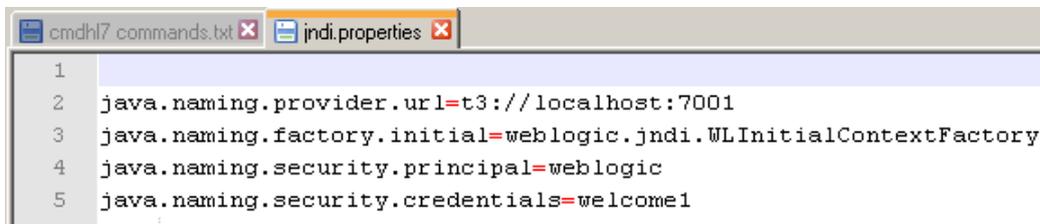
“Repair” the message

Using a text editor which does not modify line endings, perhaps “Notepad++”, edit the extracted payload file, changing “EVX” to “EVN” and making sure the file is otherwise valid.

Resubmit the message

We will use the command line tools discussed in the Oracle "Technical Note: 11g_005 Command-Line Utilities", section "Resubmit Utility", available at <http://www.oracle.com/ocom/groups/public/@otn/documents/webcontent/326860.pdf>.

- Before the Ant script can be used the environment must be correctly configured. Locate the directory "Oracle_SOA1" in the file system and record the full file system path which includes this directory – for me this is "C:\oracle\Middleware\Oracle_SOA1" – this path is the path to which the environment variable "ORACLE_HOME" will be set in subsequent commands
- Find the file "jndi.properties", typically found in %ORACLE_HOME%\bin (for me C:\oracle\Middleware\Oracle_SOA1\bin)
- Edit the jndi.properties file to make sure the username, password and port are correct for your environment



```
1
2 java.naming.provider.url=t3://localhost:7001
3 java.naming.factory.initial=weblogic.jndi.WLInitialContextFactory
4 java.naming.security.principal=weblogic
5 java.naming.security.credentials=welcome1
```

- Open a command window (I work on Windows for this series of articles – if you work on a different platform adjust paths and command as appropriate)
- Enter the following commands to set the environment for running Ant scripts, adjusting paths as required and pointing to the version of Java which you are using

```
set ORACLE_HOME=C:\oracle\Middleware\Oracle_SOA1
set ANT_HOME=%ORACLE_HOME%\..\modules\org.apache.ant_1.7.1
set JAVA_HOME=c:\jdk1.6.0_30
```

- Assuming your payload has been "fixed", is located in C:\hl7\repair, is in a file named wmPayload_C0A8E9E913C02FA1CF60000045604D11-1.txt and pertains to message with Message ID of C0A8E9E913C02FA1CF60000045604D11-1, execute the following command (where ^ is a Windows command line continuation character – adjust this and paths for other operating systems as required)

```
%ANT_HOME%\bin\ant ^
-f %ORACLE_HOME%\bin\ant-b2b-util.xml b2bresubmit ^
-Dmsgsource=WIREMSG ^
-Ddirection=INBOUND ^
-Dmsgid=C0A8E9E913C02FA1CF60000045604D11-1 ^
-Dpayloadpath=c:\hl7\repair\wmPayload_C0A8E9E913C02FA1CF60000045604D11-1.txt
```

- Inspect the output, looking for a successful submission feedback

```

C:\Documents and Settings\Administrator>set ORACLE_HOME=C:\oracle\Middleware\Oracle_SOA1
C:\Documents and Settings\Administrator>set ANT_HOME=%ORACLE_HOME%\..\modules\org.apache.ant_1.7.1
C:\Documents and Settings\Administrator>set JAVA_HOME=c:\jdk1.6.0_30
C:\Documents and Settings\Administrator>%ANT_HOME%\bin\ant ^
More? -f %ORACLE_HOME%\bin\ant-b2b-util.xml b2bresubmit ^
More? -Dmsgsource=WIREMSG ^
More? -Ddirection=INBOUND ^
More? -Dmsgid=COA8E9E913C02FA1CF60000045604D11-1 ^
More? -Dpayloadpath=c:\hl7\repair\wmPayload_COA8E9E913C02FA1CF60000045604D11-1.txt
Buildfile: C:\oracle\Middleware\Oracle_SOA1\bin\ant-b2b-util.xml

b2bresubmit:

weblogic_dependencies:
[echo]
[echo] Weblogic server location: C:\oracle\Middleware\Oracle_SOA1\bin\..\..\wls_server_10.3
[echo] JNDI properties folder location: C:\oracle\Middleware\Oracle_SOA1\bin
[echo]

verifyargs:

utility:
[echo]
[echo] exitonerror=${exitonerror}
[echo]
[echo] java.naming.provider.url=t3://localhost:7001
[echo] java.naming.factory.initial=weblogic.jndi.WLInitialContextFactory
[echo] java.naming.security.principal=weblogic
[echo] java.naming.security.credentials=*****
[echo]
[echo] args=resubmit ${propfile}
[echo]
[java] Commandline arguments 2: [resubmit, ${propfile}]
[java] Resubmit: : Result:
[java] Following messages got resubmitted : [COA8E9E913C02FA1CF60000045604D11-1].
[java] Total Number of messages : 1
[java] *** Resubmit operation completed successfully ***
[java] Time elapsed:1219ms

BUILD SUCCESSFUL
Total time: 5 seconds
C:\Documents and Settings\Administrator>

```

- If the command is incorrect, for example the message ID is wrong, a failure feedback will be provided, for example like the one below

```

[echo]
[java] Commandline arguments 2: [resubmit, ${propfile}]
[java] Resubmit: : Result:
[java] No Messages to resubmit for [direction, messageId] with the values [INBOUND, COA8E9E913C02FA1CF60000045604D11-199999]
[java] *** Resubmit operation completed successfully ***
[java] Time elapsed:1203ms

```

- Switch to the Healthcare Integration Console, click the "Reports" Tab, refresh the list and note the repaired message being received again and sent to the two configured external systems

ORACLE SOA Suite for healthcare integration Accessibility Help Logout Logged in as weblogic since 4/01/2013 10:09:00

Designer Dashboards **Reports**

Last 24 Hours

Last 24 Hours Found 6 results. Page Size 40 Last Updated: 4/01/2013 12:12:16 PM GMT+11:00 Purge Resubmit

Message ID	Created Date and Time	Direction	Endpoint	State	Document Type
1 > 2d383330363033323732333133353630	4/01/2013 12:12:14 PM GMT+11:00	OUTBOUND	HA08_SystemA_Out	MSG_COMPLETE	HL7-2.3.1-ADT
2 > 2d383438353837373436373733323835	4/01/2013 12:12:13 PM GMT+11:00	OUTBOUND	BcastB_SystemA_Out	MSG_COMPLETE	HL7-2.3.1-ADT
3 > COA8E9E913C031D84970000045604D36-1	4/01/2013 12:12:10 PM GMT+11:00	INBOUND	HosA_SystemA_In	MSG_COMPLETE	HL7-2.3.1-ADT
4 > COA8E9E913C0311A2810000045604D26-1	4/01/2013 11:59:12 AM GMT+11:00	INBOUND	HosA_SystemA_In	MSG_ERROR	HL7-2.3.1-ADT

In the steps above we externalized, "repaired" and resubmitted an inbound wire message – HL7 v2 delimited message which was broken. The command line identified the message as INBOUND and WIREMSG, providing a message ID which identified the specific inbound wire message being resubmitted and the payload file to use. There are several other options and value alternatives which can be used to control re-submission of inbound and outbound, wire and application messages, period of time, message state and so on. Oracle

"Technical Note: 11g_005 Command-Line Utilities", section "Resubmit Utility" provides details of these and example commands. Some of the things one can do with this utility is to resubmit outbound messages to a specific external system if it was down and missed messages from a particular time period, "repair" and resubmit an outbound message, and similar kinds of things that operations staff must contend with in normal operations.

Summary

In this article we walked through the process of externalizing a message, "fixing" it and resubmitting it for processing. We also briefly touched on other uses of the resubmit utility.