Stonebranch Solutions

Infitran

Infitran – SAP PI Integration

infitran-sap pi-integration

Document Name	Infitran – SAP PI Integration
Document ID	infitran-sap pi-integration
Products	Stonebranch Solutions

Stonebranch, Inc.
950 North Point Parkway, Suite 200
Alpharetta, GA 30005 USA
Tel: (678) 366-7887
Fax: (678) 366-7887

Stonebranch, Inc.® makes no warranty, express or implied, of any kind whatsoever, including any warranty of merchantability or fitness for a particular purpose or use

The information in this documentation is subject to change without notice.

Stonebranch shall not be liable for any errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this document.

All products mentioned herein are or may be trademarks of their respective owners.

SAP Certified Integration

Contents

Contents.......... 3

Overview 4

Components Required for the Example.......... 5
Licensed Stonebranch Solutions Components 5
Indesca Solution Components 5
Infitran Solution Components.......... 5
SAP PI Integration Content.......... 6
Stonebranch Solutions.......... 6
SBSPI Example .zip file Contents.......... 7
Script Files.......... 7
Other Files 8

Set-up 9
External System Set-up.......... 9
SAP Process Integrator Setup 11
Import SBS SLD Content 11
Import SBS Integration Repository Design Content 11
Transport SBS Integration Scenarios into PI Integration Directory 11
SAP Back-End System Setup.......... 12

Process 13

Executing the Example 15

Overview

This document is intended to discuss the integration possibilities between the Stonebranch Solutions Indesca and Infitran and SAP's Process Integrator (PI).

To accomplish this, an example is used that demonstrates both inbound and outbound communication between Stonebranch Solutions and the SAP Process Integrator. This example can be used as a starting point for design and development of a tailored integration implementation by an SAP solutions consultant based on organizational need.

In this example:

- An Infitran file transfer process will be initiated from a message issued by the SAP Process Integrator (PI). The PI message will contain all the information required for Infitran to carry out some unit of work. As Infitran performs the work, appropriate logging information will be sent back to the PI.
- Indesca for SOA will be used as the originator of the message that requests the Infitran work via PI. However, in a real-world implementation of this concept, an SAP component such as the Solution Manager would be a more likely originator of the message.
- The destination of the logging messages sent by Infitran to PI will be spool files on an SAP back-end system. This is used solely to demonstrate the concept. In a real-world implementation of this concept, a dedicated logging application or database table would be a more likely destination.

The example presented in this document is designed to work on Windows platforms. However, the concepts are applicable to any platform where Stonebranch Solutions is available.

Components Required for the Example

The integration example described in this document is comprised of several components. In addition to the licensed Stonebranch Solutions components, items such as scripts and SAP Integration Content are provided to demonstrate the integration concept.

Licensed Stonebranch Solutions Components

Indesca Solution Components

Universal Command Agent for SOA

Universal Command Agent for SOA (UCA for SOA) will be used to send a message to the SAP PI. In the example that follows, UCA for SOA will send the message that initiates the Infitran work request.

Infitran Solution Components

Universal Event Monitor for SOA

Universal Event Monitor for SOA (UEM for SOA) will be used to listen for the message and receive the payload from the SAP PI system. The payload will contain all the information required for Infitran to carry out some unit of work. Specifically, in this example, UEM for SOA will be receiving messages routed through the SBS_SendMessage integration scenario.

Universal Event Monitor

Universal Event Monitor (UEM) is used to detect the payload file and process the SOAP messages that are received from the SAP PI via UEM for SOA and initiate the Infitran work request.

Universal Data Mover

Universal Data Mover (UDM) is the file transfer component of Infitran and the target component of the work request sent through the SAP PI system in this example. This component will perform the requested file transfer work and issue logging messages back through the PI system.

SAP PI Integration Content

Stonebranch Solutions

Stonebranch Solutions is a software product that gets installed to the SAP System Landscape Directory.

This software product contains two integration scenarios:

- SBS_SendMessage
  This integration scenario is used to rout messages through the SAP PI to UEM for SOA (the Stonebranch Solutions SOA listener) on an external system.
- SBS_WriteToSpool
  This integration scenario is used to write spool files to an SAP back-end system.

SBSPI Example .zip file Contents

Script Files

Startexample

This batch file is used to automate the example. It performs three operations:

1. Calls UCA for SOA to send a message to the SAP PI system requesting work from Infitran.
2. Starts a UEM demand-driven server to monitor for the Infitran work request.
3. Opens a log file that contains the output from the UEM event handler and the UDM process.

\SBSPI\scripts\piudm.bat

The batch file piudm.bat is called from the UEM event handler as part of the SOAP message file processing. This script is used to coordinate payload extraction with the initiation of the UDM process.

Script "\SBSPI\scripts\getpld.ps1" (described below) is called to perform the payload extraction. The payload contains a UDM script that will be used to control the work request. UDM is then called to process the extracted script.

\SBSPI\scripts\getpld.ps1

This file is a Windows PowerShell script that is used to extract a UDM script from the SOAP message received from the SAP PI. The getpld.ps1 script is called by the piudm.bat batch file as part of the UEM event handler processing.

Other Files

\SBSPI\work\udmworkrequest1.xml

This is an XML file that contains the work request for Infitran. The work request is in the form of a pre-formatted UDM script file. The UDM script contains instructions for transferring a file and for sending logging information back to the SAP PI system.

\SBSPI\scripts\http-SBS_SendMessage.txt

This is a UCA for SOA script file that will be used to send an Infitran work request to the SAP Process Integrator.

\SBSPI\scripts\http-SBS_WriteToSpool.txt

This is a UCA for SOA script file that will be used to send UDM log entries to the SAP Process Integrator.

Set-up

This section discusses the set-up steps that must be taken for this example to work.

The example involves communication between an external system (where Indesca and Infitran are installed), an SAP Process Integration system, and an SAP back-end system (where spool files will be created as log entries).

There are set-up steps required for each of these systems.

External System Set-up

The external system, in this example, is the system where Infitran and Indesca are installed. To keep the example as simple as possible, the remote system will be used both to send the Infitran work request and to execute the Infitran work.

As a prerequisite, Stonebranch Solutions (Infitran and Indesca) and the Stonebranch Agent for SOA must be installed on the external system.

1. Extract the “SBSPIExample.zip” file to the root directory of the Windows system that will be used for execution of this example. This will create the directory hierarchy and collateral that will be used in this example.
2. From the extracted content, open file “SBSPI\work\startexample.bat”. Change “**myuserid**” and “**mypwd**” to appropriate values for a valid user account on the external system where this example is installed. Note that both myuserid and mypwd occur twice in the file; once for the “ucmd” command and once for the “uem” command.
3. From the extracted content, open file “SBSPI\work\ udmworkrequest1.xml”. Change “**myuserid**” and “**mypwd**” to appropriate values for a valid user account on the external system where this example is installed. Note that both myuserid and mypwd occur twice in the file; once for the “open” command and once for the “exec” command.
4. From the extracted content, open file “SBSPI\work\http-SBS_SendMessage.txt”.
    a. Change “**mysapuserid**” and “**mysappwd**” to appropriate values for a valid user account on the SAP PI system.
    b. Change “**pihostname**” to the hostname for the target SAP PI system.
    c. Change “**httpport**” to the port number for http requests on the PI system.

5. From the extracted content, open file “SBSPI\work\http-SBS_WriteToSpool.txt”.
    a. Change “**mysapuserid**” and “**mysappwd**” to appropriate values for a valid user account on the SAP PI system.
    b. Change “**pihostname**” to the hostname for the target SAP PI system.
    c. Change “**httpport**” to the port number for http requests on the PI system.
6. Add a SOAPListener entry in the UAC.xml file to listen for the message for the Infitran work requests sent from the SAP PI. The work requests are sent in the form of SOAP messages. For detail information on configuration of a SOAPListener entry in the UAC.xml file, see Chapter 2 Configuration of the Universal Event Monitor for SOA Reference Guide.

```
<?xml version="1.0" encoding="UTF-8"?>
<sb:UAC xmlns:sb="http://com.stonebranch/UAC/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://com.stonebranch/UAC/ UAC.xsd">
   <sb:SOAPConnection>
     <sb:URI>/axis2/services/UACInbound</sb:URI>
     <sb:Listeners>
       <sb:SOAPListener>
         <sb:Operation>Process</sb:Operation>
         <sb:Actions>
           <sb:SOAPFileWriter>
           <sb:Directory>C:\SBSPI\PIMsgIn</sb:Directory>
           <sb:FilenamePattern>SBSPIMsg%Seq%.xml</sb:FilenamePattern>
           <sb:StartSequenceNumber>1</sb:StartSequenceNumber>
           <sb:WriteEnvelope>false</sb:WriteEnvelope>
           </sb:SOAPFileWriter>
         </sb:Actions>
       </sb:SOAPListener>
     </sb:Listeners>
   </sb:SOAPConnection>
</sb:UAC>
```

SAP Process Integrator Setup

The following steps assume the user has some degree of familiarity with SAP Process Integrator.

Import SBS SLD Content

Log on to the SLD and go to “Administration -> Content Import (Upload CIM data)”. Browse and select the SLD content zip file “SBSPI\PIContent\SBS_PI_Product.zip”.

Import SBS Integration Repository Design Content

Import the PI content file “SBSPI\PIContent\SBS_PI_Content.tpz” into the PI Integration Repository:

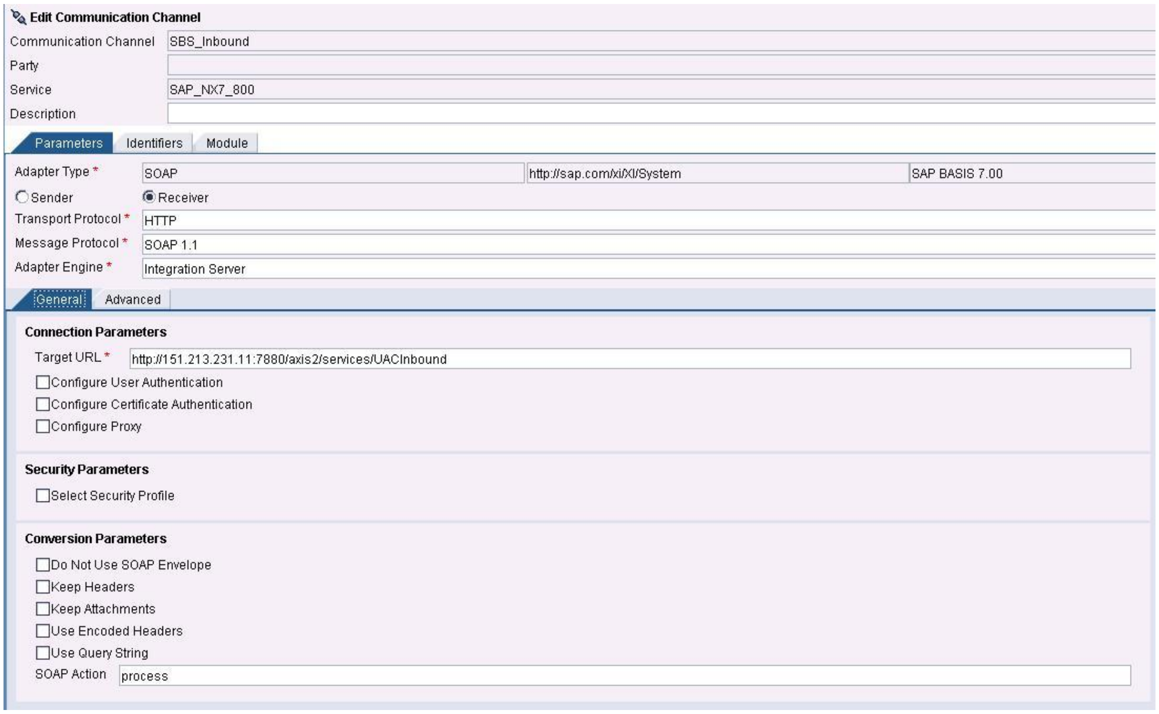
1. Copy the SBSPI\PIContent\SBS_PI_Content.tpz file into the repository import directory on the PI server. This is usually similar to: \usr\sap\<sysid>\SYS\global\xi\repository_server\import.
2. Log on to the Integration Repository and select “Tools -> Import Design Objects”. There you can select the SBSPI\PIContent\SBS_PI_Content.tpz content file.

Transport SBS Integration Scenarios into PI Integration Directory

- Log on to the PI Integration Directory and clear the SLD cache with “Environment -> Clear SLD Data Cache”.
- Transfer the SBS_SendMessage Integration Scenario from the Integration Repository into the Integration Directory. This can be done with “Tools -> Transfer Integration Scenario from the Integration Repository”.
- The “Integration Scenario Configuration Wizard” will start automatically. Complete the Configuration Wizard by assigning the services, configuring the connections and generating all the communication agreements.

- Open the SBS_Inbound communication channel and modify the connection parameters to match your environment.

  The IP address and port in the URL shown below should be changed to values that point to the external system where this example is installed.



Edit Communication Channel

Communication Channel	SBS_Inbound
Party	
Service	SAP_NX7_800
Description	

Adapter Type *	SOAP	http://sap.com/xi/XI/System	SAP BASIS 7.00
○ Sender	● Receiver		
Transport Protocol *	HTTP		
Message Protocol *	SOAP 1.1		
Adapter Engine *	Integration Server		

Connection Parameters

Target URL * http://151.213.231.11:7880/axis2/services/UACInbound

□ Configure User Authentication
□ Configure Certificate Authentication
□ Configure Proxy

Security Parameters

□ Select Security Profile

Conversion Parameters

□ Do Not Use SOAP Envelope
□ Keep Headers
□ Keep Attachments
□ Use Encoded Headers
□ Use Query String

SOAP Action process

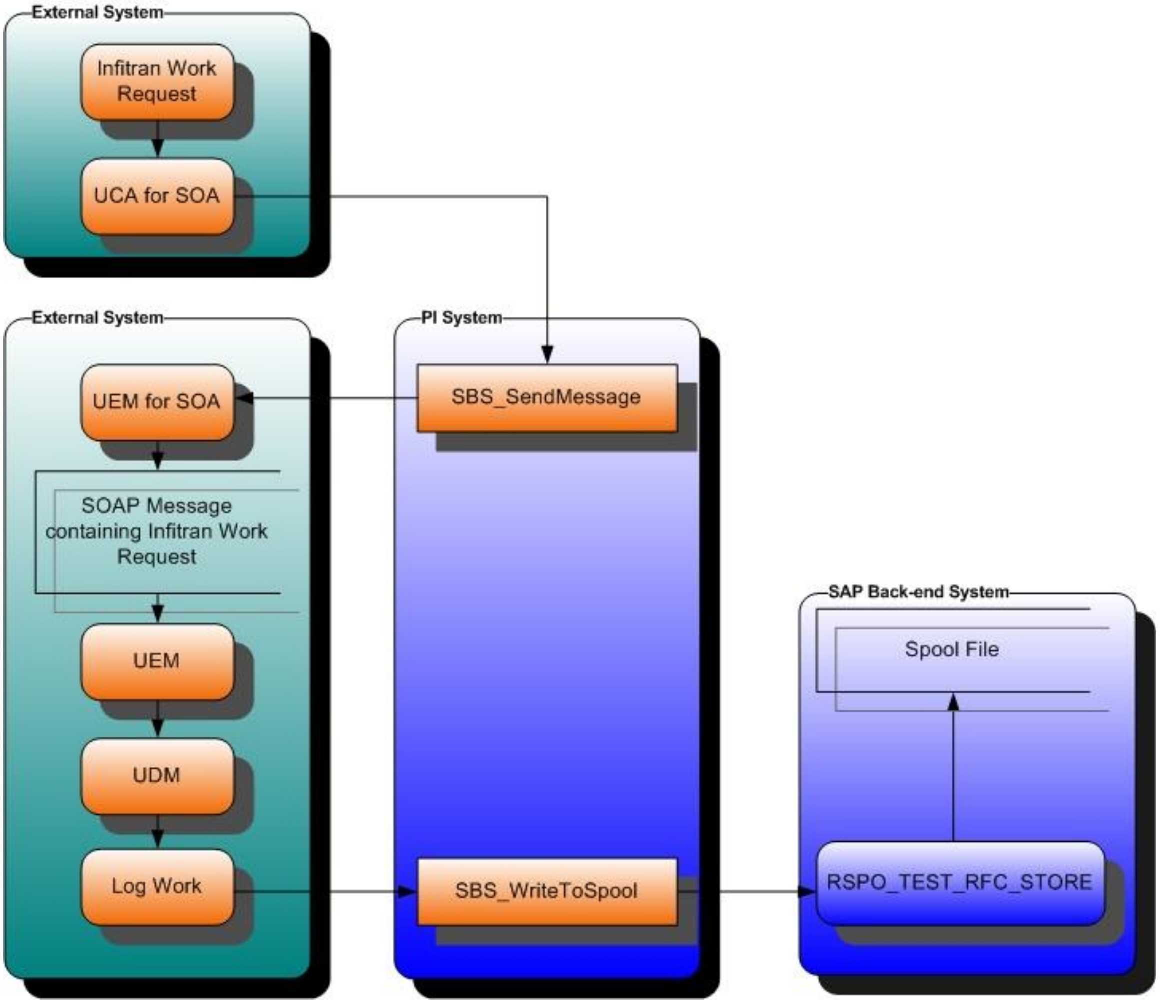
- Transfer the SBS_WriteToSpool Integration Scenario from the Integration Repository into the Integration Directory. This can be done with “Tools -> Transfer Integration Scenario from the Integration Repository”.
- The “Integration Scenario Configuration Wizard” will start automatically. Complete the Configuration Wizard by assigning the services, configuring the connections and generating all the communication agreements.
- Open the SBS_RFC_Inbound communication channel and enter the “RFC Client Paremeter” values for the SAP back-end system where spool files will be created for logging Infitran work.

SAP Back-End System Setup

An RFC destination that points to the PI system must be created in the SAP back-end system that will be used to store spool files created when logging Infitran work. Log on to the back-end system and use transaction SM59 to create the new destination.

Process

This section provides a detailed discussion of what takes place during the execution of the example. The discussion will follow the order of events as they occur in the example.



1. UCA for SOA will be used to send a message to the SAP PI system requesting Infitran work. Specifically, the message will be sent to the “SendMessage” interface of the SBS_SendMessage Integration Solution. The message sent by UCA for SOA contains a UDM script. The UDM script contains the information needed to carry out file transfer work. In addition, instructions for sending logging information back to the PI system are embedded in the UDM script.
2. The PI system receives the message sent by UCA for SOA and directs it to the SendMessage interface of the SBS_SendMessage Integration Solution.
3. The Integration Solution will, in turn, package the message as a SOAP request and send it to a listening UEM for SOA agent on the external system.
4. The UEM for SOA agent receives the SOAP message and stores it as a file in the location specified in the UAC.xml configuration file.
5. A UEM server, which has been configured to watch for the SOAP message file, is triggered and its handler processes the message.
6. The UEM handler calls the piudm.bat batch file, which in turn calls the getpld.ps1 PowerShell script. These scripts will perform the following tasks:
    a. Extract the UDM script from the SOAP message received from the PI system.
    b. Execute UDM (passing the extracted script as a parameter).
7. As UDM process the script, custom logging statements are used to build a logging payload.
8. After building the logging payload, the UDM script issues an EXEC command to send the logging payload to the SAP PI via UCA for SOA. Specifically, UCA for SOA will send a message to the SBS_WriteToSpool solution.
9. The PI receives the logging payload and routs it to the SBS_WriteToSpool solution, which in turn calls a function module on the SAP back-end system to create a spool file.

Executing the Example

At this point, all systems involved in the example should be properly configured. To execute the example, perform the following steps:

1. Open a command prompt window and CD to the working directory for the example (cd C:\SBSPI\work).
2. Execute the startexample.bat file: "C:\SBSPI\work\startexample". This will:
   a. Send an Infitran work request to the SAP PI system via UCA for SOA.
   b. Start a UEM demand-driven server to monitor for the PI SOAP messages that will be sent containing the Infitran work request.
3. Wait for the UEM demand-driven server to trigger the event handler and initiate the Infitran work by executing UDM.
4. Examine the output.
   a. The Notepad application will open automatically on the local system when the example completes. The contents of the Notepad application show the command line that executed the "\SBSPI\scripts\getpld" script, followed by the UDM execution that performed the Infitran work request. The UDM output should show successful completion.
   b. Log onto the PI system and enter transaction "SXMB_MONI". Execute "Monitor for Processed XML Messages". Select an appropriate time frame and execute the program to see the XML messages that were processed for this example.
   c. Log onto the SAP back-end system (using the user id from file SBSPI\work\http-SBS_WriteToSpool.txt) and enter transaction SP02 . Here you can find the spool files created to log the Infitran work process.

stonebranch

950 North Point Parkway, Suite 200
Alpharetta, Georgia 30005
U.S.A.