Gentran Integration Suite

Build Updates

Version 4.2

Contents

Introduction

This document provides information about fixes and enhancements provided in Gentran Integration Suite™ Version 4.2. These builds are cumulative and include all fixes and enhancements contained in the previous build.

Build 4220 or Higher

Authentication Using Multiple SSH/SFTP Keys

Authentication for SSH/SFTP connections is performed by the exchange of session keys for the server and the client. This assures that both parties know who they are exchanging data with.

Gentran Integration Suite now has the ability to associate multiple SSH Authorized User Keys with one Gentran Integration Suite account for authentication. This will allow you the ability to use one common user account, but with several keys.

You can assign multiple SSH Authorized User Keys from the User Accounts page in Gentran Integration Suite. From this page, you can associate the SSH Authorized User Keys already checked into Gentran Integration Suite to the user.

To use multiple SSH Authorized User Keys:

1. From the **Accounts** menu, select **User Accounts**.
2. Complete the User Accounts page. For additional information, see *Managing User Accounts*.
3. Click **Next**. The SSH User Key page appears.
4. On the SSH Users Key page, assign multiple SSH Authorized Keys by clicking the arrow buttons to move keys from the **Available** list to the **Assign** list.
5. Click **Next** and continue completing your User Account. For additional information, see *Managing User Accounts*.

Build 4207 or Higher

File Transfer with Connect:Direct on z/OS

The Connect:Direct Server adapter supports document transfer between Gentran Integration Suite and Connect:Direct on z/OS platforms. Files created in z/OS have attributes, named DCB attributes, that define the physical file structure. The DCB attributes are required to create or access the file content.

The Connect:Direct Server adapter provides these attributes when sending a document from Gentran Integration Suite to the z/OS platform.

Effective with Build 4207, which includes the CDJava.jar version 113, Gentran Integration Suite collects DCB attributes from files received from remote Connect:Direct z/OS PNODEs and stores them in the ProcessData of the currently executing business process. The DCB attributes are available to other services running in that business process. The following DCB attributes are collected:

- DSORG
- LRECL
- RECFM
- BLKSIZE

Accessing the DCB Attributes

Define a business process to access the DCB attributes on an incoming file. The following sample business process, CaptureProcessData, writes the incoming file to the /tmp directory:

```
<process name="CaptureProcessData">
  <sequence>
    <operation name="File System Adapter">
      <participant name="CDInteropTestFSA"/>
      <output message="FileSystemInputMessage">
        <assign to="Action">FS_EXTRACT</assign>
        <assign to="appendOnExtract">false</assign>
        <assign to="assignFilename">false</assign>
        <assign to="bootstrap">false</assign>
        <assign to="extractionFolder">/tmp</assign>
        <assign to="useSubFolders">false</assign>
        <assign to="." from="*"></assign>
      </output>
```

```
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

The remote z/OS node must specify the destination file name with the /businessprocess/<*bp-name*>/ path prefix. In the following example, the remote Connect:Direct process specifies the /businessprocess/CaptureProcessData path prefix in the destination file name.

```
/*BEGIN_REQUESTER_COMMENTS
    $PNODE$="CD.DALLAS " $PNODE_OS$="OS390"
    $SNODE$="CDSA-GIS" $SNODE_OS$="UNIX"
    $OPTIONS$="WDNO,WDOS"
  END_REQUESTER_COMMENTS*/

STARTBP PROCESS
    SNODE=CDSA-GIS
    SNODEID=(cduser1,password)

STEP1 COPY
    FROM (
        FILE="JMCGE1.LRECLXX.SAMPLE.FILE"
    )
    TO (
        FILE=/businessprocess/CaptureProcessData/sample.txt
        DISP=RPL
    )
PEND
```

Access the attributes using XPath syntax (//CDServerNodeData/DCB) within the business process.

The DCB attributes are stored in ProcessData when the incoming file is received from a Connect:Direct PNODE on the z/OS platform. The following is an example of ProcessData:

```
<?xml version="1.0" encoding="UTF-8"?>
<ProcessData>
  <PrimaryDocument SCIObjectID="myhost:a1807c:113f8fcbdbb:-78a0"/>
  <CDServerNodeData>
    <RemoteFileName/>
    <PnodeName>CD-WINDOWS</PnodeName>
    <DCB>lrecl=80,blksize=4160,recfm=vb,dsorg=ps</DCB>
    <LocalFileName>/businessprocess/CaptureProcessData/sample.txt</LocalFileName>
    <SnodeName>CDSA-GIS</SnodeName>
  </CDServerNodeData>
  <TRACKINGID>myhost:a1807c:113f8fcbdbb:-7890</TRACKINGID>
</ProcessData>
```

Example Business Process with DCB Attributes

In the following example, the CaptureProcessData business process copies the incoming file to a second remote node, CD.CHICAGO. The DCB parameter in the CopyTo Service obtains its value from ProcessData.

```
process name="CaptureProcessData">
```

```
  <sequence>
    <operation name="CD Server Begin Session Service">
      <participant name="CDServerBeginSession"/>
      <output message="CDServerBeginSessionServiceTypeInputMessage">
        <assign to="LocalCDNodeName">CDSA-GIS</assign>
        <assign to="RemoteCDNodeName">CD.CHICAGO</assign>
        <assign to="RemotePasswd">cduser1</assign>
        <assign to="RemoteUserId">password</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation name="CD Server CopyTo Service">
      <participant name="CDServerCopyTo"/>
      <output message="CDServerCopyToServiceTypeInputMessage">
        <assign to="RemoteFileName">JMCGE1.LRECLXX.SAMPLE.FILE</assign>
        <assign to="DCB">//CDServerNodeData/DCB/text()</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation name="CD Server End Session Service">
      <participant name="CDServerEndSession"/>
      <output message="CDServerEndSessionServiceTypeInputMessage">
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

SSL/TLS Client-Authentication with the Connect:Direct Server Adapter

The Connect:Direct Server adapter supports secure connections to and from Connect:Direct servers using the Secure+ protocol. Secure+ implements secure connections using either SSL or TLS. The SSL/TLS handshake phase of client-server negotiation requires the server to send its certificate to the client for server authentication. The server may request the client's certificate for client-authentication, but this is not required.

A new parameter is available for the Connect:Direct Server adapter:

Field	Description
Require Client Authentication	Indicates if Secure+ requires the client's certificate for client-authentication. Only displayed if "Encryption using Secure+" is enabled. Required if "Encryption using Secure+" is enabled. Valid values are Yes and No. Default is Yes.

By default, the Connect:Direct Server adapter is configured to request client-authentication. To disable client-authentication for a specific Connect:Direct Server adapter or specific target remote nodes, configure the Connect:Direct Server adapter using the following values:

- Encryption Using Secure+ - Enabled
- Enable Netmap Node Override - Yes
- Require Client Authentication - No

CDJava.jar version 113 (included with Gentran Integration Suite build 4207) is required for the option to disable client-authentication.

Build 4204 or Higher

Enhancement to Connect:Direct Interoperability with SPOE

In previous versions of Gentran Integration Suite, the Connect:Direct Server adapter supported Secure Point-of-Entry (SPOE) for outbound connection requests. With Build 4204, inbound requests from remote Connect:Direct PNODEs are supported.

Secure Point Of Entry (SPOE) support as a PNODE

Connect:Direct has an optional security configuration called Secure Point-of-Entry (SPOE). To increase security when communicating with known and expected nodes, a Connect:Direct server acting as an SNODE is configured to accept an inbound connection from a designated NodeName. Connections are made with PNODE userid's that are known to the SNODE and authenticated by the PNODE. SPOE uses the process PNodeId to map to a local operating system userid for any necessary authorizations. Gentran Integration Suite supports SPOE as a PNODE and automatically supplies the Gentran Integration Suite user executing the business process as the PNodeId.

Gentran Integration Suite also supports Connect:Direct User Proxies as a PNODE whereby the BP can designate a RemoteUserId and not provide a RemotePassword.

Secure Point Of Entry (SPOE) support as an SNODE

The Connect:Direct Server adapter can be the SNODE and accept inbound requests from remote Connect:Direct PNODES with SPOE. SPOE maintains the privacy of sensitive user account information by associating a pseudo account with an authentic one. Remote Connect:Direct trading partners establish sessions using pseudo accounts rather than authentic ones. SPOE is disabled by default.

Enable SPOE Authentication from Remote Connect:Direct PNODEs

SPOE policy can be modified while Gentran Integration Suite is running. To enable SPOE authentication:

1. Modify the cdinterop-spoe-policy.properties file. This file is located in the following directory: *install_dir*/properties
2. Uncomment the spoe.policy=yes property.
3. Modify other settings in the cdinterop-spoe-policy.properties file necessary for your situation.

4. Modify the cdinterop-spoe-auth.properties file. Add entries for each remote trading partner allowed to connect to this site.

See *Using Property Files*.

Enhancements to SWIFTNet Client Service for SWIFTNet Version 6.0

Overview

The SWIFTNet Client service now supports the SWIFTNet Version 6.0 mandatory fields for FileAct only. The two mandatory field requirements are as follows:

- Mandatory use of structured request types (Request Type field)
- Mandatory use of file compression indication (File Info field)

How the Enhancements Work

For the Request Type parameter, when you are configuring the SWIFTNet Client service you need to indicate the request type supported by the message exchange. This parameter is optional for InterAct and required for FileAct in SWIFTNet 6.0. For SWIFTNet 6.0 FileAct the format convention is as follows:

```
<business_area>.<type_of_syntax>.<detailed_syntax_and_format>
```

Note: This format starts with a four-characters business area code, followed by a period (dot), followed a three-character code that designates the type of syntax (which can be <nnn> , FIN, or xxx), followed by another period (dot), and then followed by a more detailed indication of syntax and format.

For the File Info parameter (only configurable through the Gentran Integration Suite GPM), when you are configuring the SWIFTNet Client service for a FileAct PUT, you need to specify whether the file will be compressed. For SWIFTNet 6.0 FileAct, the format convention is as follows:

```
SwCompression=<value>
```

Valid values are SwCompression=None (default) or SwCompression=ZIP.

Note: If you specify to use compression, you must have compressed the file before sending it to the SWIFTNet Server adapter.