First Look: Connecting Systems Using Java Business Hosts

Version 2018.1
2018-10-22

InterSystems Corporation 1 Memorial Drive Cambridge MA 02142 www.intersystems.com

First Look: Connecting Systems Using Java Business Hosts
InterSystems IRIS Data Platform Version 2018.1 2018-10-22

For Support questions about any InterSystems products, contact:

InterSystems Worldwide Response Center (WRC)
Tel: +1-617-621-0700
Tel: +44 (0) 844 854 2917
Email: support@InterSystems.com

Table of Contents

First Look: Connecting Systems Using Java Business Hosts

This First Look guide helps you develop interfaces in Java that connect systems together with an InterSystems IRIS™ interoperability production. An interoperability production is an interoperability framework for rapid connectivity and the development of new connectable applications. The production provides built-in connections to a wide variety of message formats and communications protocols. You can easily add other formats and protocols and use a graphic interface to define business logic and message transformations. Productions provide persistent storage of messages, which allow you to audit whether a message is successfully delivered. A production consists of business services, processes, and operations. Business services connect with external systems and receive messages from them. Business processes allow you to define business logic including routing and message transformation. Business operations connect with external systems and send the messages to them.

1 Solving the Problem of Connecting Systems

When connecting systems together, it can be challenging to get them to understand the other system's messages and documents. For example, consider the following problem:

- You have two separate systems: one is collecting data from multiple networked devices and the other is a work order system that tracks broken devices and the repair process.
- The current process depends on human intervention to monitor the devices and initiate the repair process. This has caused delays and is unreliable.
- You have been given the task to connect the two systems together: to monitor the data being collected and to automate initiating the repair process. You know how to detect faulty devices in the data collection system and know how to initiate a repair, but the two systems store data in incompatible formats even when the data represents the same item.
- You also need to record the actions when a repair is initiated from the data collection system.

You can solve this problem using an InterSystems IRIS production. It provides the framework for defining an interface that accepts messages from the data collection system, transforming the message into one that can be understood by the repair system, and then sending it to the repair system. It also stores a record of the message path.

In this guide, you will learn how to connect two Java programs with a simple production. For demonstration purposes, this document uses very simple Java code. A Java program for the data collection system or the work order system would be more complex and require a DTD schema, but you would use the same procedure to connect them with InterSystems IRIS.

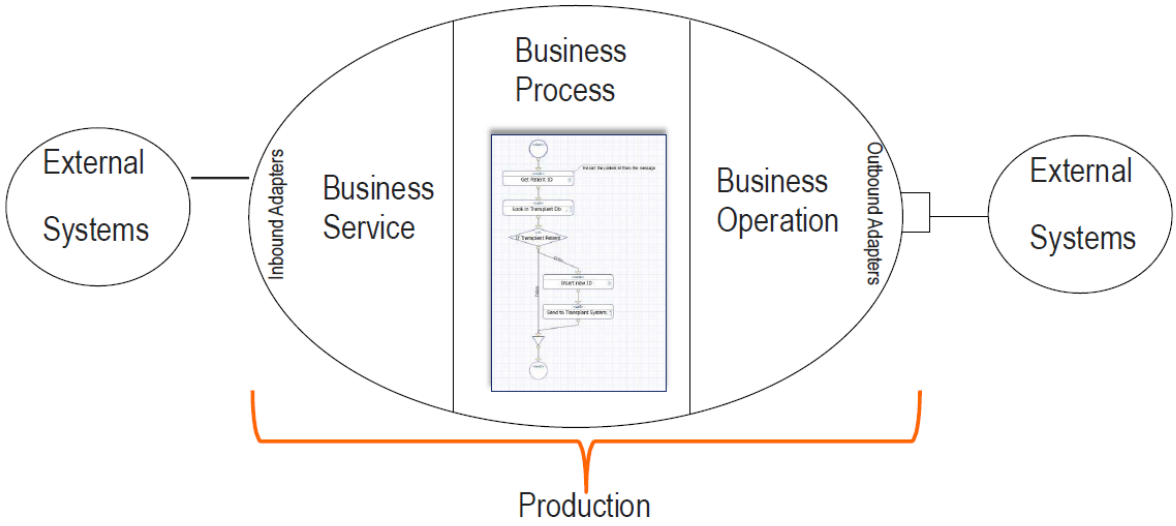
2 How Interoperability Productions Connect Systems

In its simplest form, a production consists of:

- A business service that provides the interface for a message coming from an external system.
- A business process that provides any needed business logic and message transformation.

- A business operation that provides the interface for a message going to an external system.

The following illustrates a simple production:



There are some business services and operations provided with InterSystems IRIS. If it has one that supports the message format that a system uses, you can avoid custom coding. But in many cases you will have to develop a custom business service and operation. You can develop these using the InterSystems IRIS ObjectScript or using Java.

Typically, the reason you choose to develop in Java is one of the following:

- There is an available Java library that parses the message format used by the system, and it is quicker to use the library rather than custom coding a parser for the message format.
- You prefer to develop custom code in Java rather than in InterSystems IRIS ObjectScript.

If you are developing a business service or operation in Java, you can use the Java Business Hosts feature to connect your Java code with the production. This allows you to do all of your business service and business operation coding in Java. The following illustration shows how the Java code connects to the InterSystems IRIS production:

You can use Java Business Hosts with the following kinds of messages:

- Plain text
- XML
- X12
- EDIFACT

To connect your Java code to the production, you have to implement the following classes and methods.

- For receiving messages from an external service, you implement a Java application that listens to messages and includes the Java class:

  `com.intersystems.gateway.bh.BusinessService`

  with the following methods:

  – OnInit — this method is called when the production starts or the business service is enabled. It typically starts a listener that will receive messages. The listener receives the messages from the external service and then sends them to the business service in the production by calling the method Production.SendRequest(). The production is passed in as an argument to OnInit. Your code should save it so that it can call SendRequest in the listener.

  – OnTearDown — this method is called when the production is stopped or the business service is disabled. It typically stops the listener.

- For sending messages from the production to an external service, you implement a Java application, which includes the Java class:

  `com.intersystems.gateway.bh.BusinessOperation`

  with the following methods:

  – OnInit — this method is called when the business operation starts. It typically initializes any structures needed by the OnMessage method. The OnInit method has an argument that specifies settings and values. These settings make it possible for your code to access the values set by the administrator in the Production Configuration page.

  – OnMessage — this method is called when the business operation receives a message. It is responsible for sending the message to the external service.

  – OnTearDown — this method is called when the business operation ends. It typically releases any structures created by the OnInit method.

The Production object is provided as a parameter to the BusinessService OnInit method. It provides the following methods:

- SendRequest — Sends a request message to the target configuration item of the Business Service.
- GetSetting — Gets the value for the specified Business Service setting.
- SetStatus — Sets the status of the Business Service configuration item and changes the color of the item on the Production Configuration page.
- LogMessage — Writes a message to the production log. You can use this to report errors or to help debug code.

3 Trying Connecting Systems for Yourself

In this section, you will connect two Java hosts in a production. For demonstration purposes, these are very simple Java programs. Rather than getting messages from an external service, the business service just generates a random message.

And the business operation writes the message to a log. Connecting to an external server requires more complex Java code, but you would follow the same process to connect the Java code to the production.

3.1 Getting Your System Ready

Before creating this example, you should do the following:

- Install a running, licensed instance of InterSystems IRIS. For instructions on how to install and license a development instance of InterSystems IRIS, see Quick Start: InterSystems IRIS Installation.
- Ensure that your system has the Java run-time environment and a Java development environment.
- Clone or download the FirstLook-JavaHosts sample code from github: https://github.com/intersystems/FirstLook-JavaHosts. Build the jar files from the sources or use the ones provided in the github Releases for the repository.
- Create an interoperability-enabled namespace if you don't have one. Ensure that there is no running production in the namespace. See First Look: Connecting Systems Using Interoperability Productions for details.

3.2 Creating the Credentials

The Java code needs credentials to have access to the production. For this example, you can use the same InterSystems IRIS account that you use to develop a production. For a live system, you would create an account that has the privileges needed to run the production, but not any extra privileges.

To create the credentials, in the Management Portal:

1. Select an interoperability-enabled namespace.
2. Select **Interoperability** > **Configure** > **Credentials**.
3. Specify an ID, such as JavaHostsCredentials, and a user name and password for an account on the InterSystems IRIS system. Then select **Save**.

3.3 Creating the Production and the Initiator and Generating the Business Hosts

In this step, you will create a new production, include the Java Business Host initiator, and generate the business hosts. In the Management Portal:

1. Select **Interoperability** > **Build** > **Java Business Hosts**.
2. Select **Start New Production**, give the production a name, such as JavaHostsProd, leave the other fields with the default values, and select **OK** twice. This creates a new production, adds the EnsLib.JavaGateway.Initiator component to it, and starts the production. If this step succeeds, the Java Business Hosts page will have a message indicating that the production is running and contains a Java Gateway Service. If you don't get this message, you may have a problem with the environment variables or Java JDK installation.
3. Configure the Java Gateway Initiator by selecting **Interoperability** > **Configure** > **Production** and then select the Initiator in the production diagram. You may need to set the following settings depending on your Java environment and environment variables:
   a. Java Home — Specifies the location of the JVM.
   b. Class Path — Specifies the jar files imported in the Java code. This sample only imports the java.io.FileOutputStream, java.io.PrintWriter, and java.util.Random classes, which are included in the Java system jar files. It also uses the intersystems-gateway-3.0.0.jar file which is provided in the *install-dir*\dev\java\lib\JDK18 directory.

c. JVM Args — Specifies any arguments you need to specify for you JVM.

d. If you have specified values for any settings, select **Apply**.

4. Return to the Java Business Hosts page by selecting **Interoperability** > **Build** > **Java Business Hosts** and generate the business service host by:

   a. Select **Browse** and select the jar file generated for the business service.

   b. Select the name of the Java class, such as JavaHosts.JavaHostsService, from the drop-down menu.

   c. Accept the default ObjectScript class name, such as JBH.JavaHosts.JavaHostsService.

   d. For this sample, accept the default Format of Incoming Data, Plain Text.

   e. Select the credentials that you created in the previous step from the drop-down menu.

   f. Select **Generate**.

5. Then generate the business operation host by:

   a. Select **Browse** and select the jar file generated for the business operation.

   b. Select the name of the Java class, such as JavaHosts.JavaHostsOperation, from the drop-down menu.

   c. Accept the default ObjectScript class name, such as JBH.JavaHosts.JavaHostsOperation.

   d. Select **Generate**.

You have completed creating the production and generating the business hosts. In the next section, `you add the business hosts to the production and configure them.

3.4 Configuring the Production

In this step, you will add the business operation and business service to the production and configure them. In the Management Portal:

1. Select **Interoperability** > **Configure** > **Production**.
2. Select the **Operations** plus sign to display the Business Operation Wizard.

   a. In the **Operation Class** drop-down menu, select the business operation, JBH.JavaHosts.JavaHostsOperation, that you generated using the Java Business Hosts page.

   b. Leave the **Enabled** check box clear.

   c. Select **OK**.

3. Select the JBH.JavaHosts.JavaHostsOperation in the production diagram and then in the **Settings** tab:

   a. Expand **Additional Settings** and in the **LogFile** field, enter a file path for the log file, such as c:\practice\javhost-slog.txt. Create the directory if it does not exist.

   b. Select the **Enabled** check box.

   c. Select **Apply**.

4. Select the **Services** plus sign to display the Business Service Wizard.

   a. In the **Service Class** drop-down menu, select the business service, JBH.JavaHosts.JavaHostsService, that you generated using the Java Business Hosts page.

   b. Leave the **Enabled** check box clear. You will enable the service in the next step.

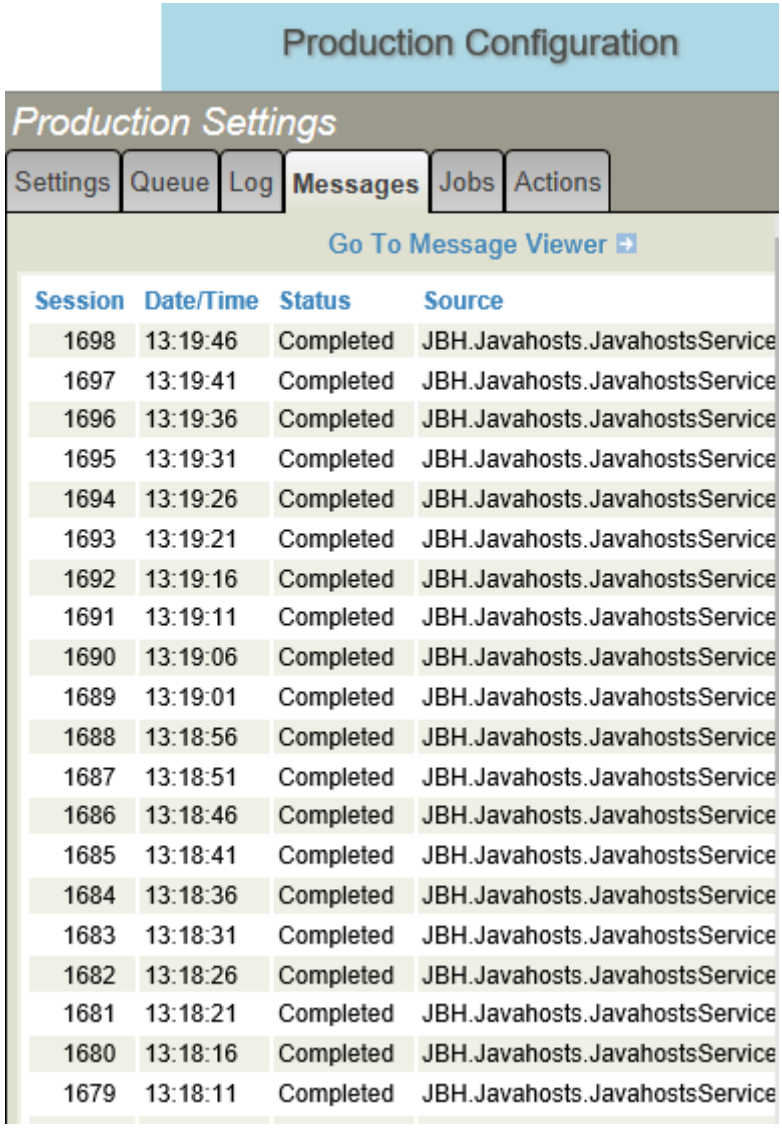
c. Select **OK**.

5. Select the JBH.JavaHosts.JavaHostsService service in the production diagram and then in the **Settings** tab:
   a. In the **Target Config Names** drop-down menu, select the JBH.JavaHosts.JavaHostsOperation operation.
   b. Optionally, specify an integer value for the Min and Max settings.
   c. Select the **Enabled** check box.
   d. Select **Apply**.

You've finished configuring the business hosts and the production. All the business hosts in the production diagram should be green and the production should be running. In the next section you will examine the messages.

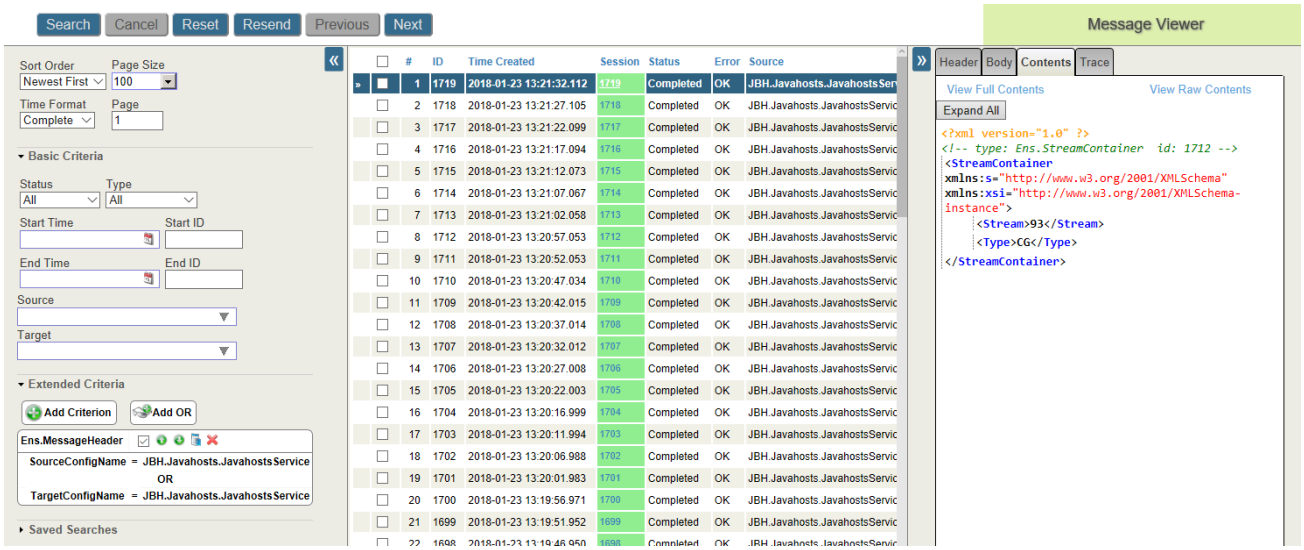
3.5 Running the Production and Examining the Messages

Once you enabled the business service, the production started sending messages. To see the messages, select the **Messages** tab on the Production Configuration page. The messages are displayed as shown by:



Session	Date/Time	Status	Source
1698	13:19:46	Completed	JBH.Javahosts.JavahostsService
1697	13:19:41	Completed	JBH.Javahosts.JavahostsService
1696	13:19:36	Completed	JBH.Javahosts.JavahostsService
1695	13:19:31	Completed	JBH.Javahosts.JavahostsService
1694	13:19:26	Completed	JBH.Javahosts.JavahostsService
1693	13:19:21	Completed	JBH.Javahosts.JavahostsService
1692	13:19:16	Completed	JBH.Javahosts.JavahostsService
1691	13:19:11	Completed	JBH.Javahosts.JavahostsService
1690	13:19:06	Completed	JBH.Javahosts.JavahostsService
1689	13:19:01	Completed	JBH.Javahosts.JavahostsService
1688	13:18:56	Completed	JBH.Javahosts.JavahostsService
1687	13:18:51	Completed	JBH.Javahosts.JavahostsService
1686	13:18:46	Completed	JBH.Javahosts.JavahostsService
1685	13:18:41	Completed	JBH.Javahosts.JavahostsService
1684	13:18:36	Completed	JBH.Javahosts.JavahostsService
1683	13:18:31	Completed	JBH.Javahosts.JavahostsService
1682	13:18:26	Completed	JBH.Javahosts.JavahostsService
1681	13:18:21	Completed	JBH.Javahosts.JavahostsService
1680	13:18:16	Completed	JBH.Javahosts.JavahostsService
1679	13:18:11	Completed	JBH.Javahosts.JavahostsService

To see the contents of a message, select **Go To Message Viewer**. Select **Search** in the message viewer, select a message, and select the **Contents** tab. The Message Viewer shows you the following:



The production continues to send messages. To stop the production:

- Select **Stop** on the Production Configuration page to stop the production.
- You can restart the production by selecting **Start**.

4 Learn More About Java Business Hosts and Productions

Java Business Hosts provides an easy way to create business services and operations in Java. It uses the InterSystems IRIS Java Gateway to do this. Although it is more work to use the Java Gateway directly, it provides more options and capabilities than Java Business Hosts. For more information on Java Business Hosts and the Java Gateway,, see:

- Java Business Hosts Presentation
- Developing Productions with Java Business Services and Operations
- Javadocs Reference for Java Business Hosts Classes
- Using the Java GatewayUsing the Java Gateway

For more information about productions, see:

- Introducing Interoperability Productions
- Developing Productions
- Configuring Productions