InterSystems
IRIS Data Platform

InterSystems IRIS Basics: Connecting an IDE

Version 2019.4
2020-01-28

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

InterSystems IRIS Basics: Connecting an IDE

This document tells you how to quickly connect your integrated development environment (IDE) to an instance of InterSystems IRIS® and verify that you have successfully connected, using template files downloaded from GitHub.

The instructions cover IDEs in the following categories:

- IDEs from InterSystems
- Java IDEs
- .NET IDEs
- Python
- Node.js

The next two sections cover the following:

- Deploying InterSystems IRIS

  Lists the options for deploying both licensed and free evaluation instances of InterSystems IRIS. If you do not yet have an instance to work with, review this section.

- InterSystems IRIS Connection Information

  Details the information you need to connect an IDE to a licensed or free InterSystems IRIS instance, and to interact with it in other ways, and explains get the information for your instance. Review the section covering the type of instance you are working with, regardless of which IDE you want to connect to InterSystems IRIS.

1 Deploying InterSystems IRIS

This section covers both licensed instances and free evaluation instances.

1.1 Deploying Licensed Instances

If you have purchased one or more licenses from InterSystems, you can deploy a licensed InterSystems IRIS instance using one of these methods:

- InterSystems Cloud Manager
- an InterSystems IRIS Docker image
- an InterSystems IRIS installation kit

1.1.1 InterSystems Cloud Manager

InterSystems Cloud Manager (ICM) is the InterSystems IRIS end-to-end provisioning and deployment solution. Using ICM, you can provision infrastructure and deploy containerized services on public cloud platforms such as Google Cloud Platform, Amazon Web Services, and Microsoft Azure, or in your private VMware vSphere cloud. and you can also deploy

services on existing virtual or hardware systems. You can deploy your custom and third-party containers alongside those from InterSystems, and ICM can also install from an InterSystems IRIS installation kit.

For detailed ICM documentation, see the *InterSystems Cloud Manager Guide*. You can try out ICM using *First Look: InterSystems Cloud Manager*. For information on how to obtain and deploy the ICM image, see Launch ICM in the former or the first few subsections of the Try It! section in the latter.

1.1.2 InterSystems IRIS Docker Image

Docker images from InterSystems let you use your own tools and methods to deploy InterSystems IRIS in containers on Linux hosts — cloud, virtual, or physical. Persistent storage of instance-specific data makes containerized InterSystems IRIS easy to upgrade, so you can always move your existing configurations to the latest version of InterSystems IRIS with little trouble and minimal downtime. You can easily customize InterSystems IRIS container images, including by adding your own application and configuring the InterSystems IRIS instance accordingly.

For a guide to using InterSystems IRIS images, see *Running InterSystems Products in Containers*. For a hands-on experience with InterSystems IRIS images and container, try *First Look: InterSystems Products in Docker Containers*. For information on how to obtain the ICM image, see Using InterSystems IRIS Docker Images in the former or the first few subsections of the Try It! section in the latter. (You can also use these documents with a free InterSystems IRIS Community Edition image from the Docker Store.)

1.1.3 InterSystems IRIS Installation Kit

You can install and license a development instance of InterSystems IRIS on your local machine or on another on your network by obtaining the latest installation kit for your platform from the InterSystems Worldwide Response Center (WRC) download area. For basic installation instructions, see *InterSystems IRIS Basics: Installation*; for detailed information, see the *Installation Guide*.

1.2 Deploying Free Evaluation Instances

If you are not yet an InterSystems IRIS user, you can explore its many features and try it out with your preferred languages and tools and your own code by deploying a free evaluation instance. There are several options for doing this:

- Deploy an instance of InterSystems IRIS Community Edition in the cloud or on your own system.

- Launch an InterSystems Learning Labs instance on the web.

1.2.1 InterSystems IRIS Community Edition

InterSystems IRIS Community Edition comes with a free built-in 13-month license (and some functionality restrictions). You can deploy Community Edition in two ways:

- Provision a cloud node hosting a running InterSystems IRIS Community Edition container on the Google Cloud Platform, Microsoft Azure, or Amazon Web Services public cloud platforms.

- Pull the container image from the Docker Store and deploy an InterSystems IRIS Community Edition container on the system of your choice.

For instructions for deploying and using InterSystems IRIS Community Edition in the cloud or on your own system, see Getting Started with InterSystems IRIS Community Edition.

1.2.2 InterSystems Learning Labs

The InterSystems Learning Labs web page lets you quickly and easily launch a lab instance of InterSystems IRIS for training, development, and testing, with a free 30-day license. Your InterSystems Learning Labs instance includes an integrated IDE and plenty of samples to work with, and you can connect your own IDE.

To launch an InterSystems Learning Labs instance, you must be logged into learning.intersystems.com; you can easily create an account if you do not have one. (Like Community Edition, the Learning Labs instance has some functionality restrictions.)

2 InterSystems IRIS Connection Information

To connect an IDE to an InterSystems IRIS instance, you first need to select the instance, then determine the needed connection information, as follows:

- Host identifier: the IP address or hostname of the instance's host.

- Superserver and web server port numbers for the instance.

- Valid credentials for the instance.

- The name of an existing namespace on the instance.

It is also helpful to know the URL of the instance's web-based Management Portal, and how to open the InterSystems Terminal on the instance, which requires the instance's name.

How you determine this information depends on how you deployed the instance you are working with, as follows:

- InterSystems Cloud Manager

- InterSystems IRIS in a Container

- InterSystems IRIS Installation Kit

- InterSystems IRIS Community Edition

- InterSystems Learning Labs

2.1 InterSystems Cloud Manager

For InterSystems IRIS instances in containers deployed by ICM, the host identifier is the IP address or hostname of the provisioned or existing node the container is deployed on. You can always review the hostnames and IP addresses of the nodes in your deployment using the **icm inventory** command.

The default superserver and web server port numbers are 51773 and 52773, respectively; you can specify different numbers by including the SuperServerPort and WebServerPort properties in the defaults.json file. The Management Portal URL portal incorporates the host identifier and the web server port, for example http://40.122.36.160:52773/csp/sys/UtilHome.csp.

To open the Terminal for an instance deployed by ICM, you can use the **icm session** command with the **-interactive** option and the **-machine** option specifying the node hosting the instance.

The credentials for all of the above purposes are one of the predefined user accounts and the default password you set when deploying services on your infrastructure with the **icm run** command, or an account and password you have previously added to the instance.

You can always connect your IDE to the **USER** namespace that exists in all InterSystems IRIS instances, but if you used the Namespace property in the definitions.json file or the **-namespace** option when deploying to specify a different default execution namespace, you may want to connect to that instead. You can also connect to a different namespace you previously created using the Management Portal.

2.2 InterSystems IRIS in a Container

For an InterSystems IRIS instance in a container deployed by you from an InterSystems image, the host identifier is the IP address or hostname of the system the container is running on.

The superserver and web server port numbers depend on how you publish the instance's ports when you create and start the container with the **docker run** command; for more information, see Deploying InterSystems IRIS Community Edition on Your Own System in *Getting Started with InterSystems IRIS Community Edition*. As an example, if you publish the instance's web server port (always 52773) as 52773 (as described in the cited section) and your browser is running on the same system as the container, the Management Portal URL is http:/localhost/:52773/csp/sys/UtilHome.csp.

To open the Terminal for an instance in a container, you can use the **docker exec** command to run the **iris terminal** command within the container; for an example, see Interacting Using the InterSystems IRIS Terminal in *Getting Started with Inter-Systems IRIS Community Edition*. You can also use **docker exec** to open a shell within the container and run **iris terminal** from that. To use the **iris terminal** command you need the instance name, which in any container from InterSystems is always `IRIS`.

The credentials for all purposes are one of the predefined user accounts and the default password `SYS` (you are prompted to immediately change the password for the account after doing this), or an account and password you have previously added to the instance.

You can always connect your IDE to the `USER` namespace that exists in all InterSystems IRIS instances, but you can also connect to a different namespace you previously created using the Management Portal.

2.3 InterSystems IRIS Installation Kit

For an InterSystems IRIS instance installed from a kit, the host identifier is the hostname or IP address of the system the instance is running on; you can use **localhost** if it is installed locally.

InterSystems IRIS installation sets the superserver and web server port numbers to 51773 and 52773 by default. However, if you have more than one instance of InterSystems IRIS installed on your system, the superserver and web server ports of the instances vary; to display the port numbers for all of the instances, you can use the iris list command on the operating system command line.

On a Windows system, you can open the Management Portal by clicking the InterSystems IRIS icon in the system tray and selecting **Management Portal**. On all systems, you can use the URL http://*host*:*webserverport*/csp/sys/UtilHome.csp, where *host* is the host identifier and *port* is the web server port number, for example http://localhost:52773/csp/sys/UtilHome.csp.

To open the Terminal for an installed instance:

- On Windows systems, you can select the **Terminal** option on the InterSystems IRIS launcher menu.

- On all systems, you can use the iris terminal command on the operating system command line, with the instance name as an argument. The instance name is set when you install the instance and cannot be changed; you can use the **iris list** command to display it.

The credentials for all purposes are one of the predefined user accounts and the default password `SYS` (you are prompted to immediately change the password for the account after doing this), or an account and password you have previously added to the instance.

You can always connect your IDE to the `USER` namespace that exists in all InterSystems IRIS instances, but you can also connect to a different namespace you previously created using the Management Portal.

2.4 InterSystems IRIS Community Edition

You can deploy Community Edition in one of two ways:

- When you provide a cloud node hosting a running InterSystems IRIS Community Edition container on a public cloud platform, the IP address (host identifier) of the cloud node. the instance's superserver and web server port numbers (always 51773 and 52773, respectively), and the Management Portal URL for the instance are displayed when you first connect to the node using SSH, and can be displayed at any time by entering **iris info** on the Linux command line. The portal URL incorporates the IP address and the web server port, for example http://40.122.36.160:52773/csp/sys/UtilHome.csp.

- When you pull the Community Edition image from the Docker Store and deploy an InterSystems IRIS Community Edition container on a system of your choice, the host identifier is the IP address or hostname of the system the container is running on. The superserver and web server port numbers depend on how you publish the instance's ports when you create and start the container with the **docker run** command; for more information, see Deploying InterSystems IRIS Community Edition on Your Own System in *Getting Started with InterSystems IRIS Community Edition*. As an example, if you publish the instance's web server port (always 52773) as 52773 (as described in the cited section) and your browser is running on the same system as the container, the Management Portal URL is http:/localhost/:52773/csp/sys/UtilHome.csp.

To information about opening the Terminal for any Community Edition instance, see Interacting Using InterSystems IRIS Terminal in *Getting Started with InterSystems IRIS Community Edition*.

The credentials for all purposes are one of the predefined user accounts and either the new default password you provided when changing the password on first connecting to the cloud node (see Changing the InterSystems IRIS Password in *Getting Started with InterSystems IRIS Community Edition*) or the default password **SYS**, if you are on your own system or logging into the cloud instance without having connected to the node first. When you use **SYS**, you are prompted to immediately change the password for the account you are using. You can also use an account you previously created on the instance.

You can always connect your IDE to the **USER** namespace that exists in all InterSystems IRIS instances, but you can also connect to a different namespace you previously created using the Management Portal.

2.5 InterSystems Learning Labs

All connections to the containerized InterSystems IRIS instance that is part of your InterSystems Learning Labs configuration are set up during the Learning Labs launch, and all the connection information you need is displayed on the launch page, as follows:

- The URL for the integrated IDE.

- A set of credentials for all purposes.

- The Management Portal URL.

- The hostname for Atelier only, and the web server port number (always 80).

- Under **External Connections**, the server's IP address and the superserver port number (labeled **InterSystems IP** and **InterSystems Port**); you can use these to connect any IDE other than Atelier, including InterSystems Studio, to the instance.

Important: The hostname provided for Atelier and included in the Management Portal URL is not a standard DNS name and cannot be used for other purposes.

A Terminal option is available on the **InterSystems** menu in the integrated IDE (as is a **Management Portal** option), so you don't need to keep track of the instance name.

3 IDEs from InterSystems

This section covers Atelier and Studio, which you can use to develop ObjectScript code on an InterSystems IRIS instance.

3.1 Atelier

The Atelier plug-in from InterSystems enables you to take advantage of the widely-known Eclipse open-source development environment to develop applications using InterSystems technology. The extensive array of plug-ins and extensions available for Eclipse means you can use whatever additional technologies your project requires, all in a single development environment. Eclipse also provides plug-ins exist for virtually all widely-used version control systems (VCS) and a powerful built-in debugging capability that is fully exploited by Atelier. The Atelier home page explains how to install Atelier and provides links to other information about Atelier.

Note: For information about connecting Eclipse to InterSystems IRIS as a Java IDE, see Eclipse.

The video Setting Up Atelier with InterSystems Products takes you step-by-step through the process of downloading Eclipse, adding the Atelier plug-in, and connecting Atelier to an instance of InterSystems IRIS. The document First Look: Atelier with InterSystems Products also describes the procedures for connecting Atelier to InterSystems IRIS, and includes a hand-on example of using Atelier to create and run a simple class file. It is important to note that Atelier connects using the webserver port (see InterSystems IRIS Connection Information), rather than using the superserver port like other IDEs.

3.2 Studio

InterSystems IRIS application developers can also the Studio IDE, a client application running on Windows systems; see *Using Studio* for detailed information. Studio can connect to any InterSystems IRIS instance regardless of the platform the instance is running on, using the DNS name or IP address of the host, the name of the instance, and the instance's superserver port number.

4 Java IDEs

This section provides connection instructions for Eclipse, IntelliJ, and NetBeans, which you can use to develop Java code that interacts with InterSystems IRIS using the InterSystems Native API, JDBC, and the Java Gateway.

4.1 Eclipse

To connect Eclipse to InterSystems IRIS, use the following steps. (Click **Next** to advance to the next panel of each dialog as needed.)

First, import the sample package,

1. Select **File > Import > Git > Projects from Git**.

2. Choose **Clone URI**. Copy and paste https://github.com/intersystems/Samples-java-helloworld into the **URI** field.

3. Select the **master** branch, configure local storage, then choose **Import existing Eclipse projects**.

4. Confirm the import by clicking **Finish**.

Next, update the sample code and run it.

1. Open **samples-java-helloworld > src > main.java.com.intersystems.samples > HelloWorld.java** and make the following changes:

    a. Change the package declaration at the top to **package main.java.com.intersystems.samples;**.

    b. Configure the username, password, IP address, and port for your instance (see InterSystems IRIS Connection Information) in the variable declarations at the top of the main method. These values are used to construct the InterSystems JDBC connection string, in the form jdbc:IRIS://*ipAddress*:*superserverPort*/*namespace*, for example jdbc:IRIS://12.345.678.910:51773/USER.

2. Run the code by selecting **Run > Run**.

3. The **Console** tab displays the message `Hello World! You have successfully connected to InterSystems IRIS via JDBC.`

If you are not successful, confirm that the Java execution environment is set properly by doing the following:

1. Right-click the **samples-java-helloworld** project in the Package Explorer pane and choose **Build Path > Configure Build Path...**.

2. On the **Libraries** tab, choose **Add Library** > **JRE System Library** > **Execution environment** > and choose an execution environment, such as **JRE-1.1 (jre 1.8.0_172)**. Click **Finish** and then **Apply and Close**.

4.2 IntelliJ

To connect IntelliJ to InterSystems IRIS, use the following steps. (Click **Next** to advance to the next panel of each dialog as needed.)

First, create the sample project.

1. Select **VCS > Checkout from Version Control > Git**.

2. On the Clone Repository dialog,

    a. Copy and paste https://github.com/intersystems/Samples-java-helloworld into the **Clone URI** field.

    b. For the **Directory** field, enter the path to the location where you want the local GitHub repository created.

3. Click **Clone**, then choose **Yes** in the popup to create a project based on this source.

Next, update the sample code and run it.

1. Open src/main/java.com.intersystems.samples.helloworld.java and configure the username, password, IP address, and port for your instance (see InterSystems IRIS Connection Information) in the variable declarations at the top of the main method. These values are used to construct the InterSystems JDBC connection string, in the form jdbc:IRIS://*ipAddress*:*superserverPort*/*namespace*, for example jdbc:IRIS://12.345.678.910:51773/USER.

2. In the **Project** pane, right-click **HelloWorld.java** and choose **Run 'HelloWorld.main()'**.

3. The **Output** pane displays the message `Hello World! You have successfully connected to InterSystems IRIS via JDBC.`

If you are not successful, confirm that the Java execution environment is set properly by doing the following:

1. Choose **Build Path > Build Project**.

2. Edit the configuration and add a new application configuration, selecting **com.intersystems.samples.HelloWorld** for the main class.

3. If the error `Error:java: invalid source release: 9` is displayed, change the project SDK and project language level to **1.8** in the following locations:

- **File > Project Structure** > **Project Settings**

- **File** > **Project Structure** > **Module Settings** > **Sources: Language Level** tab

- **File** > **Project Structure** > **Module Settings** > **Dependencies: Module SDK** tab

4.3 NetBeans

To connect NetBeans to InterSystems IRIS, use the following steps. (Click **Next** to advance to the next panel of each dialog as needed.)

First, create the sample project.

1. Select **Team > Git > Clone**.

2. Copy and paste https://github.com/intersystems/Samples-java-helloworld into the **Repository URL** field.

3. Select **Master** as the branch to be fetched, choose the location where you want the local GitHub repository to be created, and click **Finish**.

Next, update the sample code and run it.

1. Open HelloWorld.java and configure the username, password, IP address, and port for your instance (see InterSystems IRIS Connection Information) in the variable declarations at the top of the main method. These values are used to construct the InterSystems JDBC connection string, in the form jdbc:IRIS://*ipAddress*:*superserverPort*/*namespace*, for example jdbc:IRIS://12.345.678.910:51773/USER.

2. In the **Project** pane, open the Dependencies folder, then right-click **intersystems-jdbc-3.0.0.jar** and choose **Manually install artifact**. Navigate to the folder you recently cloned, select **intersystems-jdbc-3.0.0.jar**, and click **Install Locally**.

3. In the **Project** pane, right-click HelloWorld.java and choose **Run File**.

4. The **Output** pane displays the message `Hello World! You have successfully connected to InterSystems IRIS via JDBC.`

5 .NET IDEs

This section provides connection instructions for Microsoft's Visual Studio and Visual Studio Code, which you can use to develop .NET code that interacts with InterSystems IRIS through the InterSystems Native API, the ADO.NET Managed Provider for .NET, InterSystems XEP, and the .NET Gateway.

5.1 Visual Studio

To connect Visual Studio to InterSystems IRIS, use the following steps. (Log into GitHub if necessary.)

First, create the sample solution.

1. Select **View > Team Explorer**.

2. In the Team Explorer – Connect pane, select **Local Git Repositories > Clone**, copy and paste https://github.com/inter-systems/Samples-dotnet-helloworld into the URL box, and click **Clone**.

Next, update the sample code and run it.

1. Double-click HelloWorld.sln to see the files in the solution, then open helloworld.cs.

2. Configure the username, password, IP address, and port for your instance (see InterSystems IRIS Connection Information) using the variable declarations,

3. Press the F5 key to run the program.

4. A command window displays the message `Hello World! You have successfully connected to InterSystems IRIS. Press any key to continue.`

5.2 Visual Studio Code

To connect Visual Studio Code to InterSystems IRIS, use the following steps. (Log into GitHub if necessary.)

1. In your web browser, download or clone https://github.com/intersystems/Samples-dotnet-helloworld.

2. In Visual Studio Code, use **File > Open Folder** to open the folder you recently downloaded or cloned from GitHub.

3. Expand **samples-dotnet-helloworld** and select **helloworld.cs**.

4. Configure the username, password, IP address, and port for your instance (see InterSystems IRIS Connection Information) using the variable declarations.

5. Use **View > Terminal** to open the terminal and execute the following commands:

```
csc /reference:InterSystems.Data.IRISClient.dll helloworld.cs
.\helloworld.exe
```

6. The Output pane displays the message `Hello World! You have successfully connected to InterSystems IRIS. Press any key to continue.`

6 Python

This section provides instructions for connecting your favorite Python-focused IDE, such as PyCharm, Spyder, IDLE/IdleX, or Vim to InterSystems IRIS. When you have connected using the following steps (log into GitHub if necessary). you can explore the InterSystems IRIS Native API for Python using First Look: InterSystems IRIS Native API for Python.

1. In your web browser, download or clone https://github.com/intersystems/Samples-python-helloworld.

2. In your IDE, open the folder you recently downloaded or cloned from GitHub, then open hello_world.py.

3. Configure the username, password, IP address, and port for your instance (see InterSystems IRIS Connection Information) using the variable declarations.

4. In the Terminal of your IDE, do the following:

    a. Install the InterSystems Native API for Python using one of the following commands:

      • Microsoft Windows:

        pip install wheel\irisnative-1.0.0-cp34.cp35.cp36.cp37-none-win_amd64.whl

      • UNIX®/Linux:

        pip install wheel/irisnative-1.0.0-cp34-abi3-linux_x86_64.whl

      • Apple macOS:

        pip install wheel/irisnative-1.0.0-cp34-abi3-macosx_10_13_x86_64.macosx_10_14_x86_64.whl

   b.   Run the code with the command `python hello_world.py`.

5.   The Output pane displays the message `Hello World! You have successfully connected to InterSystems IRIS`.

Note: The Python code sample and supporting wheel files are designed for Python 3 because Python 2 will retire in 2020; Python 2 versions are available from InterSystems Learning Services.

7 Node.js

This section provides instructions for connecting your favorite IDE for Node.js, such as Visual Studio Code, Vim, or Webstorm, to InterSystems IRIS. When you have connected using the following steps (log into GitHub if necessary). you can explore the InterSystems IRIS Native API for Node.js using First Look: InterSystems IRIS Native API for Node.js.

1.   In your web browser, download or clone https://github.com/intersystems/Samples-nodejs-helloworld.

2.   In your IDE, open the folder you recently downloaded or cloned from GitHub, then open hello_world.js.

3.   Configure the username, password, IP address, and port for your instance (see InterSystems IRIS Connection Information) using the variable declarations.

4.   In the Terminal of your IDE, do the following:

   a.   Install the InterSystems Native API for Node.js with the command `npm install --save intersystems-iris-native`.

   b.   Run the code with the command `node hello_world.js`.

5.   The Output pane displays the message `Hello World! You have successfully connected to InterSystems IRIS`.