



# InterSystems IRIS 基础： 连接一个 IDE

版本 2021.1

2021-07-21

*InterSystems IRIS 基础: 连接一个 IDE*

InterSystems IRIS 数据平台 版本 2021.1 2021-07-21

版权所有 © 2021 InterSystems 公司

保留所有权利。

InterSystems、InterSystems IRIS、InterSystems Caché、InterSystems Ensemble 以及 InterSystems HealthShare 均为 InterSystems 公司注册商标。

在此使用或涉及到的所有其他品牌或产品名称均为各公司或机构所有的商标或注册商标。

本文件所含商业机密和机密信息，属 InterSystems 公司（马萨诸塞州剑桥纪念大道 1 号，邮编 02142）或其关联公司财产，仅出于 InterSystems 公司产品运营及维护目的而提供。未经 InterSystems 公司事先书面同意，该文件任何部分均不得用于其他目的，亦不可以任何形式、任何方式全部或部分地对该文件进行重制、复制、披露、传输、存储在检索系统中或翻译为任何其他人类或计算机语言。

禁止复制、使用和处置本文件和本文中描述的软件程序，除非在 InterSystems 公司涵盖该等程序和相关文档的标准软件许可协议中所规定的有限范围内。除了标准软件许可协议中规定的声明和保证外，InterSystems 公司对此类软件程序不作任何声明和保证。此外，InterSystems 公司对与使用该等软件程序有关的或因使用该等软件程序而产生的任何损失或损害的责任，按照该等标准软件许可协议所规定的方式加以限制。

以上概括描述了 InterSystems 公司对其计算机软件的使用和责任所施加的限制。完整的信息应参考 InterSystems 公司的标准软件许可协议，该协议的副本将根据要求提供。

InterSystems 公司对本文中可能出现的错误不承担责任，并保留在不另行通知的情况下自行决定对本文中描述的产品和实践进行替换和修改的权利。

有关 InterSystems 产品的技术支持问题，请联系：

**InterSystems 全球响应中心 (WRC)**

Tel: +1-617-621-0700

Tel: +44 (0) 844 854 2917

Email: [support@InterSystems.com](mailto:support@InterSystems.com)

# 目录

<b>InterSystems IRIS 基础：连接一个 IDE.....</b>	<b>1</b>
1 部署 InterSystems IRIS.....	1
1.1 部署已授权的实例.....	1
1.2 部署免费的评估实例.....	3
2 InterSystems IRIS 连接信息.....	3
2.1 InterSystems 云管理器.....	4
2.2 在一个容器中部署 InterSystems IRIS.....	4
2.3 从套件中安装 InterSystems IRIS .....	5
2.4 InterSystems IRIS 社区版.....	5
2.5 InterSystems 学习实验室（Learning Labs）.....	6
3 ObjectScript IDE.....	6
3.1 Visual Studio Code.....	6
3.2 Studio.....	6
4 Java IDE .....	7
4.1 Visual Studio Code.....	7
4.2 Eclipse.....	7
4.3 IntelliJ .....	8
4.4 NetBeans.....	9
5 .NET IDE.....	9
5.1 Visual Studio.....	9
5.2 Visual Studio Code.....	10
6 Python IDE.....	10
7 Javascript IDE.....	11



# InterSystems IRIS 基础：连接一个 IDE

本文档告诉您如何快速连接您的集成开发环境（IDE），用于 [ObjectScript](#)、[Java](#)、[.NET](#)、[Python](#) 或 [JavaScript/Node.js](#) 的集成开发环境连接到 InterSystems IRIS® 的一个实例，并使用从 [GitHub](#) 下载的模板文件验证您是否已经成功连接。

此外，接下来的两节还包括以下内容：

- [部署 InterSystems IRIS](#)  
列出用于部署 InterSystems IRIS 的已授权和免费评估实例的选项。如果您还没有一个可以使用的实例，请查看本节。
- [InterSystems IRIS 连接信息](#)  
详细介绍了将 IDE 连接到已授权或免费的 InterSystems IRIS 实例，以及以其他方式与之进行交互所需的信息，并解释了如何获得实例的信息。无论您想用哪个 IDE 连接到 InterSystems IRIS，都要查看涉及您所使用的实例类型的部分。

## 1 部署 InterSystems IRIS

本节涵盖了 [已授权的实例](#) 和 [免费的评估实例](#)。

### 1.1 部署已授权的实例

如果您已经从 InterSystems 购买了一个或多个许可证，您可以使用以下方法之一来部署一个已授权的 InterSystems IRIS 实例：

- [InterSystems 云管理器](#)
- [InterSystems Kubernetes Operator](#)
- [InterSystems IRIS 容器镜像](#)
- [InterSystems IRIS 安装套件](#)

当从容器镜像或安装套件进行部署时，您可以使用 [配置合并功能](#) 来支持自动部署。

#### 1.1.1 InterSystems 云管理器

InterSystems 云管理器（ICM）是 InterSystems IRIS 的端到端配置和部署解决方案。使用 ICM，您可以在公共云平台（如谷歌云平台、亚马逊网络服务和微软 Azure）或您的私有 VMware vSphere 云中配置基础设施和部署容器化服务，您也可以在现有的虚拟或硬件系统中部署服务。您可以将您的自定义和第三方容器与 InterSystems 的容器一起部署，而且 ICM 还可以从 InterSystems IRIS 安装套件中安装。

关于 ICM 的详细文档，请参见 *InterSystems Cloud Manager Guide*（《*InterSystems 云管理器指南*》）。您可以通过 *First Look: InterSystems Cloud Manager*（《*技术概要：InterSystems 云管理器*》）试用 ICM。关于如何获得和部署 ICM 镜像的信息，请参见 *InterSystems Cloud Manager Guide*（《*InterSystems 云管理器指南*》）中的 [Launch ICM](#)（启动 ICM）。

### 1.1.2 InterSystems Kubernetes Operator

**Kubernetes** 是一个开源的编排引擎，用于自动化部署、扩展和管理容器化工作负载和服务。您可定义您想要部署的容器化服务和想要使用的治理策略；Kubernetes 以最有效的方式透明地提供所需的资源，在部署偏离规格时修复或恢复，并自动或按需扩展。InterSystems Kubernetes Operator (IKO) 通过 IrisCluster 自定义资源扩展了 Kubernetes API，该资源可以在任何 Kubernetes 平台上作为 InterSystems IRIS 分片集群、分布式缓存集群或独立实例（所有可选镜像）进行部署。IKO 还为 Kubernetes 增加了 InterSystems IRIS 特定的集群管理功能，实现了任务的自动化，如向集群添加节点，否则您必须通过与实例直接交互来手动完成。关于使用 IKO 的信息，请参见 [Using the InterSystems Kubernetes Operator](#)（《*使用 InterSystems Kubernetes Operator*》）。

### 1.1.3 InterSystems IRIS 容器镜像

InterSystems 的 Docker 镜像让您可以使用自己的工具和方法将 InterSystems IRIS 部署到 Linux 主机——云、虚拟或物理主机上的容器中。对特定实例数据的持久性存储使容器化的 InterSystems IRIS 易于升级，因此您可以随时将现有的配置转移到最新版本的 InterSystems IRIS 中，而几乎不存在任何问题，停机时间也很少。

关于使用 InterSystems IRIS 镜像的详细指南，请参见 [Running InterSystems Products in Containers](#)（《*在容器中运行 InterSystems 产品*》）。如果想亲身体验一下 InterSystems IRIS 镜像和容器，请尝试 [First Look: InterSystems Products in Containers](#)（《*技术概要：容器中的 InterSystems 产品*》）。关于如何获得 InterSystems IRIS 镜像的信息，请参见 [Running InterSystems Products in Containers](#)（《*在容器中运行 InterSystems 产品*》）中的 [Downloading the InterSystems IRIS Image](#)（《*下载 InterSystems IRIS 镜像*》）。

您可以使用安装在谷歌云平台、微软 Azure 或亚马逊网络服务公共云平台上的 InterSystems IRIS 容器镜像和 Docker 来配置 BYOL（自带许可证）云节点，然后添加您的许可并按照 [Running InterSystems Products in Containers](#)（《*在容器中运行 InterSystems 产品*》）或 [First Look: InterSystems Products in Containers](#)（《*技术概要：容器中的 InterSystems 产品*》）中的描述，从该镜像中运行 InterSystems IRIS 容器。欲了解更多信息，请参见 [Deploy and Explore InterSystems IRIS](#)（《*部署和探索 InterSystems IRIS*》）中的 [Deploy InterSystems IRIS BYOL on a Cloud Node](#)（《*在云节点上部署 InterSystems IRIS BYOL*》）。（您也可以使用免费的 InterSystems IRIS 社区版镜像来使用这些文档。）

### 1.1.4 InterSystems IRIS 安装套件

您可以通过从 [InterSystems 全球响应中心 \(WRC\)](#) 下载区获得您的平台的最新安装套件，在您的本地机器或网络上的另一台机器上安装和许可 InterSystems IRIS 的开发实例。有关基本的安装说明，请参见 [InterSystems IRIS Basics: Installation](#)（《*InterSystems IRIS 基础：安装*》）；详细信息请参见 [安装指南](#)。

### 1.1.5 使用配置合并进行部署

配置合并功能在 Linux 和 UNIX® 系统上可用，让您可以根据同一镜像部署的 InterSystems IRIS 容器的配置，或从同一套件安装的本地实例，只需简单地将所需的声明性配置合并文件应用于部署的每个实例。

这个合并文件，也可以在重启现有实例或使用 [iris 合并命令](#) 时应用，更新实例的 *配置参数文件 (CPF)*，其中包含了大部分的配置设置；这些设置在每次启动时都会从 CPF 读取，包括实例部署后的第一次启动。当您在部署过程中应用配置合并时，实际上是用您自己的更新版本替换了随实例提供的默认 CPF。

[InterSystems 云管理器](#) 和 [InterSystems Kubernetes Operator](#) 都包含了配置合并功能。

有关使用配置合并的信息，请参见 *Configuration Parameter File Reference*。（[配置参数文件参考](#)）中的 [Using Configuration Merge to Deploy and Reconfigure InterSystems IRIS](#)（使用配置合并来部署和重新配置 InterSystems IRIS）和 [Create and Use a Configuration Merge File](#)（创建和使用配置合并文件）。

## 1.2 部署免费的评估实例

如果您还不是 InterSystems IRIS 的用户，您可以通过部署一个免费的评估实例，来探索它的许多功能，并使用您喜欢的语言和工具以及您自己的代码来试用它。有以下几个选项可以做到这一点：

- 在云或者您自己的系统上部署 [InterSystems IRIS 社区版实例](#)。
- 在网上启动一个 [InterSystems 学习实验室（Learning Labs）实例](#)。

### 1.2.1 InterSystems IRIS 社区版

InterSystems IRIS 社区版有一个免费的内置 13 个月的许可证（和一些功能限制）。您可以通过两种方式部署社区版：

- 在谷歌云平台、微软 Azure 或亚马逊网络服务公共云平台上提供一个托管运行社区版容器的云节点。
- 从 [InterSystems Container Registry \(ICR\)](#) 或 [Docker Store](#) 中提取社区版镜像，并使用它在您选择的系统上部署容器。

有关在云或自己的系统上部署和使用 InterSystems IRIS 社区版的说明，请参见 [Deploy and Explore InterSystems IRIS](#)（《部署和探索 InterSystems IRIS》）。

### 1.2.2 InterSystems 学习实验室（Learning Labs）

[InterSystems 学习实验室（Learning Labs）](#) 网页可以让您快速、轻松地启动 InterSystems IRIS 实验室实例，用于培训、开发和测试，并有 30 天的免费许可。您的 InterSystems 学习实验室实例包括一个集成的 IDE 和大量的示例，您可以连接您自己的 IDE。

要启动 InterSystems 学习实验室实例，您必须登录 [learning.intersystems.com](https://learning.intersystems.com)；如果您没有账户，可以轻松创建一个。（与社区版一样，学习实验室实例有一些功能限制。）

## 2 InterSystems IRIS 连接信息（Connection Information）

要将一个 IDE 连接到一个 InterSystems IRIS 实例，首先需要选择实例，然后确定所需的连接信息，如下所示：

- 主机标识符：实例主机的 IP 地址或主机名。
- 实例的超级服务器和 web 服务器端口号。
- 实例的有效凭证。
- 实例上现有命名空间的名称。

了解实例基于 Web 管理门户的 URL，以及如何在实例上打开 [InterSystems 终端](#)（这需要实例的名称），这也是很有帮助的。

如何确定这些信息取决于您如何部署您正在使用的实例，如下所示：

- InterSystems 云管理器 (Cloud Manager)



- [在一个容器中部署 InterSystems IRIS](#)
- [从套件中安装 InterSystems IRIS](#)
- [InterSystems IRIS 社区版](#)
- [InterSystems 学习实验室 \(Learning Labs\)](#)

## 2.1 InterSystems 云管理器 (Cloud Manager)

对于由 ICM 部署的容器中的 InterSystems IRIS 实例，主机标识符是容器所部署的配置或现有节点的 IP 地址或主机名。您可以随时使用 `icminventory` 命令查看您部署的节点的主机名和 IP 地址。

默认的超级服务器和 Web 服务器端口号分别为 1972 和 52773；您可以通过在 `defaults.json` 文件中加入 `SuperServerPort` 和 `WebServerPort` 属性来指定不同的端口号。管理门户的 URL 门户包含了主机标识符和 web 服务器端口，例如 `http://40.122.36.160:52773/csp/sys/UtilHome.csp`。IDE 连接或管理门户的凭证是您在基础设施上用 `icm run` 命令部署服务时设置的预定义用户账户之一和默认密码，或者是您先前添加到实例的账户和密码。

要打开 ICM 部署的实例的终端，可以使用带有 `-interactive` 选项和 `-machine` 选项的 `icm session` 命令，指定托管实例的节点；不需要验证。

您始终可以将您的 IDE 连接到所有 InterSystems IRIS 实例中存在的 `USERnamespace`，但是如果您在部署时使用了 `definitions.json` 文件中的 `Namespace` 属性或 `-namespace` 选项以指定不同的默认执行命名空间，您可能希望连接到该 `namespace`。您还可以连接到以前使用管理门户创建的其他名称空间。

## 2.2 在一个容器中部署 InterSystems IRIS

对于您从 InterSystems 镜像（包括在 BYOL 云节点上）部署的容器中的 InterSystems IRIS 实例，主机标识符是容器所运行的系统的 IP 地址或主机名。

超级服务器和 Web 服务器的端口号取决于您在使用 `docker run` 命令创建和启动容器时如何发布实例的端口；更多信息，请参见 *Running InterSystems Products in Containers*（《在容器中运行 InterSystems 产品》）中的 [Running InterSystems IRIS Containers](#)（运行 InterSystems IRIS 容器）。举个例子，如果您把实例的 Web 服务器端口（始终是 52773）发布为 52773（如引用部分所述），并且您的浏览器与容器在同一个系统上运行，那么管理门户的 URL 是 `http://localhost:52773/csp/sys/UtilHome.csp`。IDE 连接或管理门户的凭证是您在创建容器时设置的预定义用户账户之一和默认密码[请参见 *Running InterSystems Products in Containers*（《在容器中运行 InterSystems 产品》）中的 [Authentication and Passwords](#)（验证和密码）]，或者是您先前添加到实例中的账户和密码。

要为容器中的实例打开 Terminal，您可以使用 `docker exec` 命令在容器中运行 `iris terminal` 命令，您也可以使用 `docker exec` 在容器中打开一个 shell，并从那里运行 `iris terminal`；有关示例，请参见 *Deploy and Explore InterSystems IRIS*（《部署和探索 InterSystems IRIS》）中的 [Interacting Using the InterSystems Terminal](#)（使用 InterSystems Terminal 进行交互）。要使用 `iris terminal`，您需要实例名称，在 InterSystems 的任何容器中，该名称始终为 `IRIS`。如 *Running InterSystems Products in Containers*（《在容器中运行 InterSystems 产品》）中的 [Ownership and Directories](#)（所有权和目录）所述，使用 `docker exec` 从 InterSystems IRIS 容器外发出的命令在容器内以 `irisowner` 身份执行，不需要身份验证，因此您无需登录 Terminal。

您始终可以将您的 IDE 连接到存在于所有 InterSystems IRIS 实例中的 `USER namespace`，但您也可以连接到您先前使用管理门户创建的不同 `namespace`。

## 2.3 从套件中安装 InterSystems IRIS

对于从套件中安装的 InterSystems IRIS 实例，主机标识符是该实例运行的系统的主机名或 IP 地址；如果是本地安装，可以使用 `localhost`。

InterSystems IRIS 的安装将超级服务器和 web 服务器的端口号默认设置为 1972 和 52773。但是，如果您在系统中安装了一个以上的 InterSystems IRIS 实例，那么这些实例的超级服务器和 web 服务器的端口是不同的；要显示所有实例的端口号，您可以在操作系统的命令行中使用 `iris list` 命令。

在 Windows 系统中，您可以通过点击系统托盘中的 InterSystems IRIS 图标并选择 **Management Portal** 来打开管理门户。在所有系统上，您都可以使用 URL `http://host:webserverport/csp/sys/UtilHome.csp`，其中 `host` 是主机标识符，`port` 是 web 服务器端口号，例如 `http://localhost:52773/csp/sys/UtilHome.csp`。

要打开已安装实例的终端：

- 在 Windows 系统上，您可以选择 InterSystems IRIS 启动菜单上的 **Terminal** 选项。
- 在所有的系统上，您可以在操作系统的命令行上使用 `iris terminal` 命令，把实例名称作为参数。实例名称是在您安装实例时设置的，不能更改；您可以使用 `iris list` 命令来显示它。

用于所有目的的凭证是预定义用户账户（**predefined user accounts**）之一和默认密码 **sys**（这样做后会提示您立即更改账户的密码），或者是您先前添加到实例的账户和密码。

您始终可以将您的 IDE 连接到存在于所有 InterSystems IRIS 实例中的 **USER namespace**，但您也可以连接到您先前使用管理门户创建的不同的 namespace。

## 2.4 InterSystems IRIS 社区版

您可以通过以下两种方式之一部署社区版：

- 当您在公共云平台上配置托管运行的 InterSystems IRIS 社区版容器的云节点时，云节点的 IP 地址（主机标识符）、实例的超级服务器和 Web 服务器端口号（始终分别为 1972 和 52773），以及实例的管理门户 URL 在您首次使用 SSH 连接到节点时显示，并且可以通过在 Linux 命令行上输入 `iris info` 随时显示。门户网站 URL 包含了 IP 地址和 web 服务器端口，例如 `http://40.122.36.160:52773/csp/sys/UtilHome.csp`。
- 当您从 [InterSystems Container Registry](#) 或 [Docker Store](#) 提取社区版镜像并在您选择的系统上部署 InterSystems IRIS 社区版容器时，主机标识符是容器运行所在系统的 IP 地址或主机名。超级服务器和 Web 服务器的端口号取决于您在使用 `docker run` 命令创建和启动容器时如何发布实例的端口，如在 *Deploy and Explore InterSystems IRIS*（《部署和探索 InterSystems IRIS》）中的 [Deploying InterSystems IRIS Community Edition on Your Own System](#)（在您自己的系统上部署 InterSystems IRIS 社区版）中所述。举个例子，如果您把实例的 web 服务器端口（始终是 52773）发布为 52773（如引用部分所述），并且您的浏览器与容器在同一个系统上运行，那么管理门户的 URL 是 `http://localhost:52773/csp/sys/UtilHome.csp`。

有关为任何社区版实例打开终端的信息，请参见 *Deploy and Explore InterSystems IRIS*（《部署和探索 InterSystems IRIS》）中的 [Interacting Using InterSystems Terminal](#)（使用 InterSystems 终端进行交互）。

IDE 连接和管理门户的凭证是预定义用户账户之一，以及您在第一次连接云节点修改密码时提供的新的默认密码 [请参见 *Deploy and Explore InterSystems IRIS*（《部署和探索 InterSystems IRIS》）中的 [Change the Default Passwords](#)（修改默认密码）]，或者默认密码 **sys**，如果您在自己的系统上或者在没有先连接到节点的情况下登录到云实例。当您使用 **sys** 时，系统会提示您

立即修改您所使用的账户的密码。您也可以使用您以前在实例上创建的账户。

您始终可以将您的IDE 连接到存在于所有InterSystems IRIS 实例中的**USER namespace**，但您也可以连接到您以前使用管理门户创建的不同namespace。

## 2.5 InterSystems 学习实验室（Learning Labs）

到容器化 InterSystems IRIS 实例的所有连接都是您的 **InterSystems 学习实验室（learning labs）** 配置的一部分，在学习实验室启动期间，您需要的所有连接信息都显示在启动页面上，如下所示：

- 集成 IDE 的 URL。
- 一套用于所有目的的凭证。
- 管理门户 URL。
- Web 服务器端口号（始终是 80）。
- 在**外部连接**下，服务器的 IP 地址和超级服务器的端口号（标记为**InterSystems IP** 和 **InterSystems 端口**）；您可以使用它们将 IDE（如 InterSystems Studio 或 Visual Studio Code）与 ObjectScript 扩展连接到实例。

在集成 IDE 的 **InterSystems** 菜单上有一个 **Terminal** 选项（**Management Portal** 选项也是如此），所以您不需要跟踪实例名称。

# 3 ObjectScript IDE

本节介绍了您可以用来在 InterSystems IRIS 实例上开发 ObjectScript 代码的 IDE：带有 ObjectScript 扩展的 **Visual Studio Code** (VS Code)，以及 **Studio**。

## 3.1 Visual Studio Code

Visual Studio Code (VS Code) 是微软为 Windows、Linux 和 macOS 制作的一个免费源代码编辑器。虽然它可以作为带有 InterSystems IRIS 的 **.NET**、**Python** 或 **JavaScript** IDE 开箱即用，但可用的扩展功能使您能够使用 VS Code 开发 ObjectScript 代码，这些代码在客户端系统上存储和版本化，在 InterSystems IRIS 服务器上编译、运行和调试。关于安装扩展，连接到一个或多个 InterSystems IRIS 实例，以及使用 VS Code for ObjectScript 的信息，请访问 <https://intersystems-community.github.io/vscode-objectscript/>。VS Code 可以连接到任何 InterSystems IRIS 实例，无论该实例在哪个平台上运行，使用主机的 DNS 名称或 IP 地址、实例的名称和实例的超级服务器端口号，默认为 1972（但如果系统上安装了多个实例，则可能不同）。

## 3.2 Studio

InterSystems IRIS 应用程序开发人员还可以使用 **Studio** IDE，这是一个在 Windows 系统上运行的客户端应用程序；详细信息请参见 [使用 Studio](#)。Studio 可以连接到任何 InterSystems IRIS 实例，无论该实例在哪个平台上运行，使用主机的 DNS 名称或 IP 地址、实例的名称和实例的超级服务器端口号，默认为 1972（但如果系统上安装了多个实例，则可能有所不同）。

## 4 Java IDE

本节提供了 [Visual Studio Code](#)（带有 Java 编码包或 Java 扩展包）、[Eclipse](#)、[IntelliJ](#) 和 [NetBeans](#) 的连接说明，您可以用它们来开发 Java 代码，使用 [InterSystems Native API](#)、[JDBC](#) 和 [Java 网关](#) 与 InterSystems IRIS 交互。

### 4.1 Visual Studio Code

要将 Visual Studio Code 连接到 InterSystems IRIS，请使用以下步骤。(如有必要，请登录 GitHub。)

1. 在您的 web 浏览器中，下载或克隆 <https://github.com/intersystems/Samples-java-helloworld>。
2. 在 Visual Studio Code 中，使用 **File > Open Folder** 来打开您最近从 GitHub 下载或克隆的文件夹。
3. 选择并打开 `src/main/java.com.intersystems.samples.helloworld.java`。
4. 在主要方法的顶部的变量声明中，为您的实例配置用户名、密码、IP 地址和端口（请参见 [InterSystems IRIS Connection Information 《InterSystems IRIS 连接信息》](#)）。这些值用于构造 InterSystems JDBC 连接字符串，格式为 `jdbc:IRIS://ipAddress:superserverPort/namespace`，例如 `jdbc:IRIS://12.345.678.910:51773/USER`。
5. 使用 **View > Terminal** 来打开终端并执行以下命令：

```
javac -cp ".:intersystems-jdbc-3.2.0.jar" HelloWorld.java
java -cp ".:intersystems-jdbc-3.2.0.jar" HelloWorld
```

6. 输出窗格显示信息 `Hello World! You have successfully connected to InterSystems IRIS via JDBC.`

### 4.2 Eclipse

要将 Eclipse 连接到 InterSystems IRIS，请使用以下步骤。(点击 **Next**，根据需要推进到每个对话框的下一个面板。)

首先，导入示例包，

1. 选择 **File > Import > Git > Projects from Git**。
2. 选择 **Clone URI**。将 <https://github.com/intersystems/Samples-java-helloworld> 复制并粘贴到 **URI** 字段中。
3. 选择 **master** 分支，配置本地存储，然后选择 **Import existing Eclipse projects**。
4. 点击 **Finish** 确认导入。

接下来，更新示例代码并运行它。

1. 打开 `samples-java-helloworld > src > main.java.com.intersystems.samples > HelloWorld.java` 并做如下修改：
  - a. 将顶层包声明更改为 `package main.java.com.intersystems.samples;`
  - b. 在主要方法的顶部变量声明中，为您的实例配置用户名、密码、IP 地址和端口（请参见 [InterSystems IRIS Connection Information 《InterSystems IRIS 连接信息》](#)）。这些值用于构造 InterSystems JDBC 连接字符串，格式为 `jdbc:IRIS://ipAddress:superserverPort/namespace`，例如 `jdbc:IRIS://12.345.678.910:1972/USER`。
2. 通过选择 **Run > Run** 来运行代码。

3. **Console** 标签显示的信息是 Hello World! You have successfully connected to InterSystems IRIS via JDBC.

如果您没有成功，请通过以下操作确认 Java 执行环境设置正确：

1. 右键点击 Package Explorer 窗格中的 **samples-java-helloworld** 项目，然后选择 **Build Path > Configure Build Path...**
2. 在 **Libraries** 标签上，选择 **Add Library > JRE System Library > Execution environment** 并选择一个执行环境，如 **JRE-1.1 (jre 1.8.0\_172)**。点击 **Finish**，然后点击 **Apply and Close**。

## 4.3 IntelliJ

要将 IntelliJ 连接到 InterSystems IRIS，请使用以下步骤。（点击 **Next**，根据需要推进到每个对话框的下一个面板。）

首先，创建示例项目。

1. 选择 **VCS > Checkout from Version Control > Git**。
2. 在克隆存储库对话框中，
  - a. 将 <https://github.com/intersystems/Samples-java-helloworld> 复制并粘贴到 **Clone URI** 字段中。
  - b. 对于 **Directory** 字段，输入要创建本地 GitHub 存储库的位置的路径。
3. 点击 **Clone**，然后在弹出的窗口中选择 **Yes**，以创建一个基于此源的项目。

接下来，更新示例代码并运行它。

1. 在主要方法的顶部的变量声明中，打开 `src/main/java.com.intersystems.samples.helloworld.java` 并为您的实例配置用户名、密码、IP 地址和端口（请参见 [InterSystems IRIS Connection Information 《InterSystems IRIS 连接信息》](#)）。这些值用于构造 InterSystems JDBC 连接字符串，格式为 `jdbc:IRIS://ipAddress:superserverPort/namespace`，例如 `jdbc:IRIS://12.345.678.910:1972/USER`。
2. 在 **Project** 窗格中，右键单击 `HelloWorld.java` 并选择 **Run 'HelloWorld.main()'**。
3. **Output** 窗格显示的信息是 Hello World! You have successfully connected to InterSystems IRIS via JDBC.

如果您没有成功，请通过以下操作确认 Java 执行环境设置正确：

1. 选择 **Build Path > Build Project**。
2. 编辑配置并添加一个新的应用程序配置，选择 `com.intersystems.samples.HelloWorld` 作为主类。
3. 如果错误 `Error:java: invalid source release: 9` 显现，在以下位置将项目 SDK 和项目语言级别修改为 **1.8**：
  - **File > Project Structure > Project Settings**
  - **File > Project Structure > Module Settings > Sources: Language Level** 标签
  - **File > Project Structure > Module Settings > Dependencies: Module SDK** 标签



## 4.4 NetBeans

要将 NetBeans 连接到 InterSystems IRIS，请使用以下步骤。(点击 **Next**，根据需要推进到每个对话框的下一个面板。)

首先，创建示例项目。

1. 选择 **Team > Git > Clone**。
2. 将 <https://github.com/intersystems/Samples-java-helloworld> 复制粘贴到 **Repository URL** 字段中。
3. 选择 **Master** 作为要提取的分支，选择创建本地 GitHub 存储库的位置，然后点击 **Finish**。

接下来，更新示例代码并运行它。

1. 在主要方法的顶部的变量声明中，打开 `HelloWorld.java` 并为您的实例配置用户名、密码、IP 地址和端口（请参见 [InterSystems IRIS Connection Information 《InterSystems IRIS 连接信息》](#)）。这些值用于构造 InterSystems JDBC 连接字符串，格式为 `jdbc:IRIS://ipAddress:superserverPortInnamespace`，例如 `jdbc:IRIS://12.345.678.910:1972/USER`。
2. 在 **Project** 窗格中，打开 **Dependencies** 文件夹，然后右键单击 `intersystems-jdbc-3.0.0.jar` 并选择 **Manually install artifact**。导航到您最近克隆的文件夹，选择 `intersystems-jdbc-3.0.0.jar`，然后点击 **Install Locally**。
3. 在 **Project** 窗格中，右键单击 `HelloWorld.java` 并选择 **Run File**。
4. **Output** 窗格显示的信息是 `Hello World! You have successfully connected to InterSystems IRIS via JDBC.`

## 5 .NET IDE

本节为微软的 [Visual Studio](#) 和 [Visual Studio Code](#) 提供连接说明，您可以用它来开发 .NET 代码，这些代码通过 [InterSystems Native API](#)、[ADO.NET Managed Provider for .NET](#)、[InterSystems XEP](#) 和 [.NET 网关](#) 与 InterSystems IRIS 交互。

### 5.1 Visual Studio

要将 Visual Studio 连接到 InterSystems IRIS，请使用以下步骤。(如有必要，请登录 GitHub。)

首先，创建示例解决方案。

1. 选择 **View > Team Explorer**。
2. 在 **Team Explorer - Connect** 窗格中，选择 **Local Git Repositories > Clone**，将 <https://github.com/intersystems/Samples-dotnet-helloworld> 复制粘贴到 URL 框中，并点击 **Clone**。

接下来，更新示例代码并运行它。

1. 双击 `HelloWorld.sln` 来查看解决方案中的文件，然后打开 `helloworld.cs`。
2. 使用变量声明，为您的实例配置用户名、密码、IP 地址和端口（请参见 [InterSystems IRIS Connection Information 《InterSystems IRIS 连接信息》](#)）。
3. 按 F5 键，运行该程序。

- 命令窗口显示的信息是Hello World! You have successfully connected to InterSystems IRIS. Press any key to continue.

## 5.2 Visual Studio Code

通过 ObjectScript 扩展，Visual Studio Code 可以用来在 InterSystems IRIS 上开发 ObjectScript 代码；请参见 [ObjectScript IDE](#)。

要将 Visual Studio Code 连接到 InterSystems IRIS，请使用以下步骤。(如有必要，请登录 GitHub。)

- 在您的 web 浏览器中，下载或克隆 <https://github.com/intersystems/Samples-dotnet-helloworld>。
- 在 Visual Studio Code 中，使用 **File > Open Folder** 来打开您最近从 GitHub 下载或克隆的文件夹。
- 拓展 **samples-dotnet-helloworld** 并选择 **helloworld.cs**。
- 使用变量声明，为您的实例配置用户名、密码、IP 地址和端口（请参见 [InterSystems IRIS Connection Information 《InterSystems IRIS 连接信息》](#)）。
- 使用 **View > Terminal** 来打开终端并执行以下命令：

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

- 输出窗格显示的信息是Hello World! You have successfully connected to InterSystems IRIS.Press any key to continue.

## 6 Python IDE

本节提供了将您最喜欢的以 Python 为中心的 IDE（如 Visual Studio Code、PyCham、Spyder、IDLE/IdleX 或 Vim）连接到 InterSystems IRIS 的说明。当您使用以下步骤连接后（如有必要，请登录 GitHub），您可以使用 [技术概要:InterSystems IRIS Native API for Python](#) 来探索 InterSystems IRIS Native API for Python。

- 在您的 web 浏览器中，下载或克隆 <https://github.com/intersystems/Samples-python-helloworld>。
- 在您的 IDE 中，打开您最近从 GitHub 下载或克隆的文件夹，然后打开 **hello\_world.py**。
- 使用变量声明，为您的实例配置用户名、密码、IP 地址和端口（请参见 [InterSystems IRIS Connection Information 《InterSystems IRIS 连接信息》](#)）。
- 在您的 IDE 的终端，做如下操作：
  - 使用以下命令之一安装 InterSystems Native API for Python:

- 微软 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
```

- 使用命令 `python hello_world.py` 运行代码。

5. 输出窗格显示信息Hello World! You have successfully connected to InterSystems IRIS.

**注意：** Python 代码示例和支持轮文件是为 Python3 设计的，因为 Python2 将在 2020 年退休；Python2 版本可从 InterSystems Learning Services 获得。

## 7 Javascript IDE

本节提供了将您最喜欢的 JavaScript 和 Node.js 的 IDE（如 Visual Studio Code、Vim 或 Webstorm）连接到 InterSystems IRIS 的说明。当您使用以下步骤连接后（如有必要，请登录 GitHub），您可以使用[技术概要：InterSystems IRIS Native API for Node.js](#) 来探索 InterSystems IRIS Native API for Node.js。

1. 在您的 web 浏览器中，下载或克隆 <https://github.com/intersystems/Samples-nodejs-helloworld>。
2. 在您的 IDE 中，打开您最近从 GitHub 下载或克隆的文件夹，然后打开 hello\_world.js。
3. 使用变量声明，为您的实例配置用户名、密码、IP 地址和端口（请参见 [InterSystems IRIS Connection Information 《InterSystems IRIS 连接信息》](#)）。
4. 在您的 IDE 的终端，做如下操作：
  - a. 使用命令 `npm install --save intersystems-iris-native` 安装 InterSystems Native API for Node.js。
  - b. 使用命令 `node hello_world.js` 运行代码。
5. 输出窗格显示信息Hello World! You have successfully connected to InterSystems IRIS.



