First Look: Atelier with InterSystems Products

Version 2018.1
2018-11-30

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

First Look: Atelier with InterSystems Products
InterSystems IRIS Data Platform Version 2018.1 2018-11-30
Copyright © 2018 InterSystems Corporation
All rights reserved.

InterSystems, InterSystems Caché, InterSystems Ensemble, InterSystems HealthShare, HealthShare, InterSystems TrakCare, TrakCare, InterSystems DeepSee, and DeepSee are registered trademarks of InterSystems Corporation.

InterSystems IRIS Data Platform, InterSystems IRIS, InterSystems iKnow, Zen, and Caché Server Pages are trademarks of InterSystems Corporation.

All other brand or product names used herein are trademarks or registered trademarks of their respective companies or organizations.

This document contains trade secret and confidential information which is the property of InterSystems Corporation, One Memorial Drive, Cambridge, MA 02142, or its affiliates, and is furnished for the sole purpose of the operation and maintenance of the products of InterSystems Corporation. No part of this publication is to be used for any other purpose, and this publication is not to be reproduced, copied, disclosed, transmitted, stored in a retrieval system or translated into any human or computer language, in any form, by any means, in whole or in part, without the express prior written consent of InterSystems Corporation.

The copying, use and disposition of this document and the software programs described herein is prohibited except to the limited extent set forth in the standard software license agreement(s) of InterSystems Corporation covering such programs and related documentation. InterSystems Corporation makes no representations and warranties concerning such software programs other than those set forth in such standard software license agreement(s). In addition, the liability of InterSystems Corporation for any losses or damages relating to or arising out of the use of such software programs is limited in the manner set forth in such standard software license agreement(s).

THE FOREGOING IS A GENERAL SUMMARY OF THE RESTRICTIONS AND LIMITATIONS IMPOSED BY INTERSYSTEMS CORPORATION ON THE USE OF, AND LIABILITY ARISING FROM, ITS COMPUTER SOFTWARE. FOR COMPLETE INFORMATION REFERENCE SHOULD BE MADE TO THE STANDARD SOFTWARE LICENSE AGREEMENT(S) OF INTERSYSTEMS CORPORATION, COPIES OF WHICH WILL BE MADE AVAILABLE UPON REQUEST.

InterSystems Corporation disclaims responsibility for errors which may appear in this document, and it reserves the right, in its sole discretion and without notice, to make substitutions and modifications in the products and practices described in this document.

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: Atelier with InterSystems Products 1
1 Atelier Features 1
2 Trying Atelier for Yourself 1
2.1 Creating a Server Connection 2
2.2 Creating a Project 3
2.3 Creating a Class File 5
2.4 Adding a Class Method 7
2.5 Running the Class 9
3 Learn More About Atelier 9

First Look: Atelier with InterSystems Products

This First Look guide introduces the Atelier plug-in for Eclipse. Eclipse is a popular open-source development environment with an extensible architecture that enables support for a wide variety of languages and technologies. The Atelier plug-in extends that support to the InterSystems IRIS Database Platform™, bringing InterSystems IRIS™ into the Eclipse 'ecosystem'. When you develop applications with Atelier, you edit and save source files on your client machine. You compile, run, and debug source files on a server instance, using a connection you create and maintain in Atelier.

1 Atelier Features

Atelier provides a powerful and flexible development environment. Key advantages include:

- Eclipse is a widely-known open-source development environment.
- The Eclipse Consortium and other parties provide extensive learning resources.
- The Atelier plug-in enables you to develop applications in Eclipse 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.
- Atelier is able to draw on the powerful debugging capability built into Eclipse.
- Eclipse plug-ins exist for virtually all widely-used version control systems (VCS), which enables you to use the VCS of your choice to manage development.

2 Trying Atelier for Yourself

In order to try Atelier, you must first download and install Eclipse. Then you can install the Atelier plug-in. The Atelier Download page describes the steps necessary to get up and running with Atelier.

The first time you start Eclipse, it asks you to select a workspace. An Eclipse workspace is just a directory on your local disk where Eclipse stores the resources you use for development. Once you are running the Eclipse application, you have a number of resources available to guide you through your first steps:

- **Cheat Sheets** that provide step-by-step guidance through basic Atelier tasks. Select **Help > Cheat Sheets...** from the main menu, open the **Atelier** folder, and select a topic of interest. You may want to work through the cheat sheets in the order listed, as each one builds on previously covered material.
- **Product Documentation**, which is available by selecting **Help > Help Contents** from the main menu.
  - The **Getting Started** section provides an introduction to Atelier and the Eclipse UI.
  - The **Concepts** section introduces the major components and capabilities of Atelier.
  - The **Tasks** section contains topics that walk you through a number of useful tasks with Atelier.
  - The **Reference** section provides more detailed technical information about Atelier.

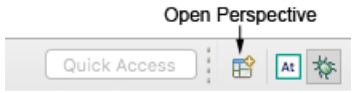
The next several sections describe how to perform some basic tasks with Atelier.

2.1 Creating a Server Connection

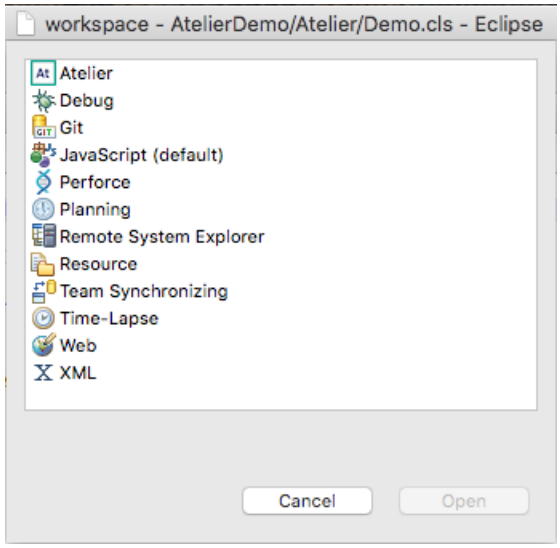
Development with Atelier requires a connection to an InterSystems server. One of the first tasks you must perform when getting started with Atelier is to define one or more server connections. The following steps describe how to create a connection.

1. Your first step is to open the **Atelier perspective**. An Eclipse perspective is a set of UI resources, or views, that help you perform a group of related tasks. In this case, we need the **Server Explorer** view, which is part of the Atelier perspective.

   To open the **Atelier perspective**, click on the **Open Perspective** button, shown in the following screen shot:

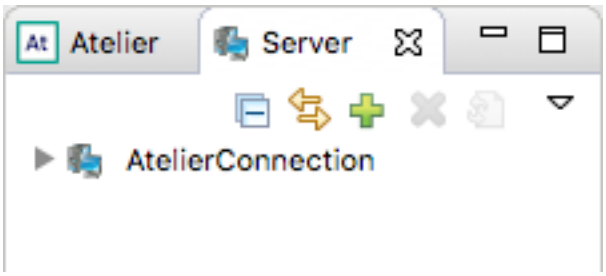
   

   This button opens a dialog, shown in the next screen shot, that lists available perspectives.

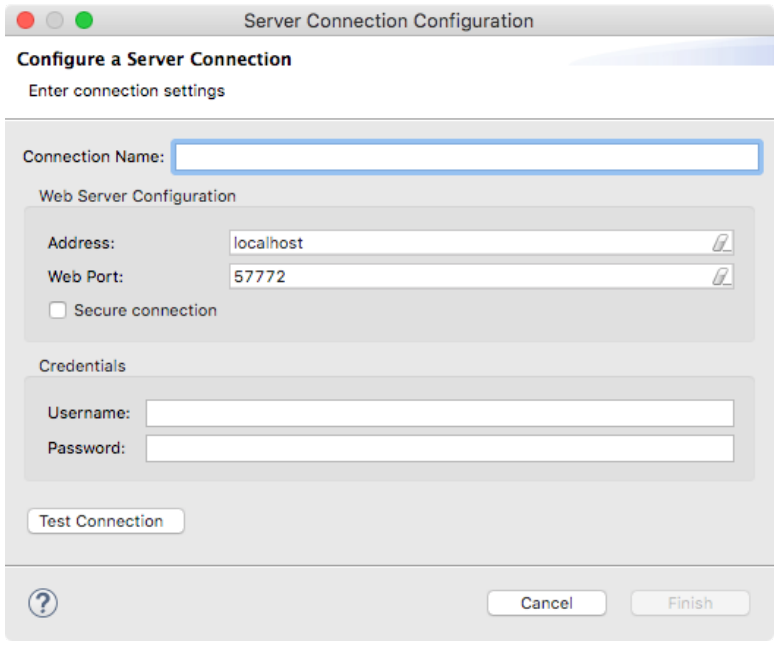
   

   Select **Atelier** and click **Open**.

2. Once you're in the **Atelier perspective**, click on the tab for the **Server Explorer** view. The next screen shot shows the **Server Explorer** view selected, with one existing connection:

3. You can create a new connection by clicking the green plus sign (+) in the view toolbar. This action opens the **Server Connection Configuration** wizard:



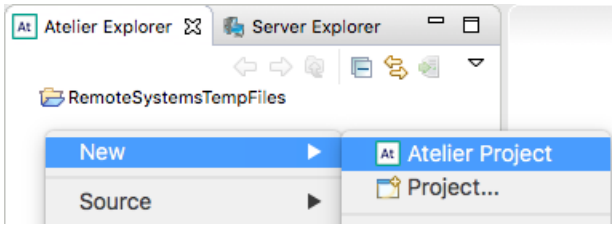
4. Now, fill in the fields in the connection wizard:
   a. Connection Name: Enter a name to identify the new connection. The connection name cannot contain spaces.
   b. Next, fill in Web Server Configuration section. If you don't have the required information, you need to get it from a qualified person in your company.
      1. Address: Enter the address of the host where the server for this connection is running. The default value is localhost. This field supports IPV4 and IPV6 IP addresses in addition to server names.
      2. Web Port: Enter the WebServer port number used by the server. The default value is 52773. Note that this is not the port number for the SuperServer port.
      3. Secure Connection: Leave this box unchecked. In a real-world application, you would want to use a secure connection, but that's not necessary for this quick introduction.
   c. Next, fill in Credentials section:
      1. Username: A valid user ID for the instance specified in the **Web Server Configuration** section. The user must have the required minimum privileges. See Minimum Required Privileges.
      2. Password: A valid password for the specified user.
   d. Click the **Test Connection** button to test the connection. You see either "Connection succeeded." or an error message.
   e. Once the connection test had completed successfully, click the **Finish** button to create the connection.

The newly created connection is listed in the **Server Explorer**. You can open the connection node and drill down to examine the resources provided by the connection.

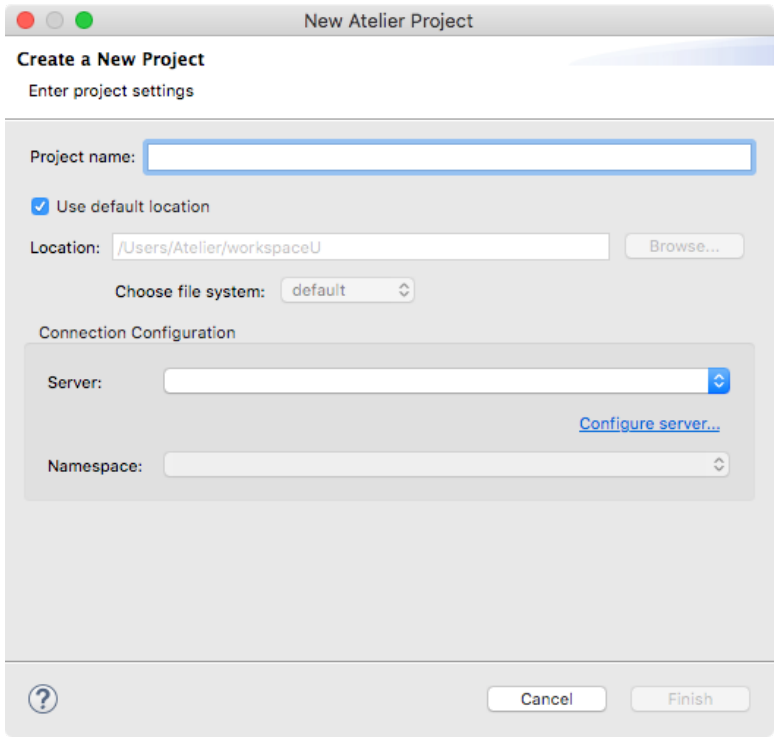
2.2 Creating a Project

Atelier uses projects to organize your work, so your next step is to create a project.

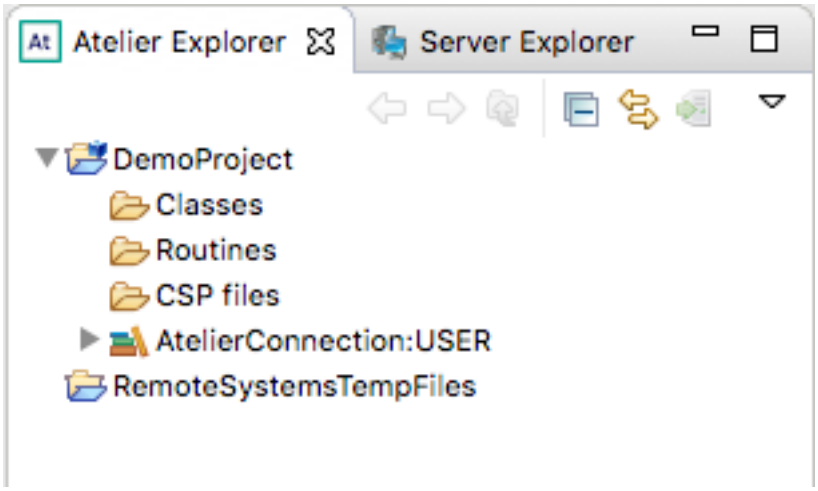
1. Right-click in the **Atelier Explorer** view to open the context menu and select **New > Atelier Project**, as illustrated in the following screen shot:



This action opens the **New Atelier Project** wizard:



2. Next, fill in the fields in the Atelier project wizard:
   a. Enter a name for the project in the **Project name** field. We'll use `DemoProject`.
   b. Leave the check box **Use default location** selected, which puts your new project in your current workspace.
   c. Next, fill in the **Connection Configuration** section:
      1. **Server**: Use this drop-down list to select the server you configured in the previous section. If you need to create a new server connection at this point, the button labeled **Configure server...** opens the **Server Connection Configuration** wizard for you.
      2. Once you have selected a server, use the **Namespace** drop-down list to select a namespace from those available on the server.
   d. Click **Finish** to create the project.
3. You can now see your new project listed in the **Atelier Explorer**, as shown in the following screen shot:

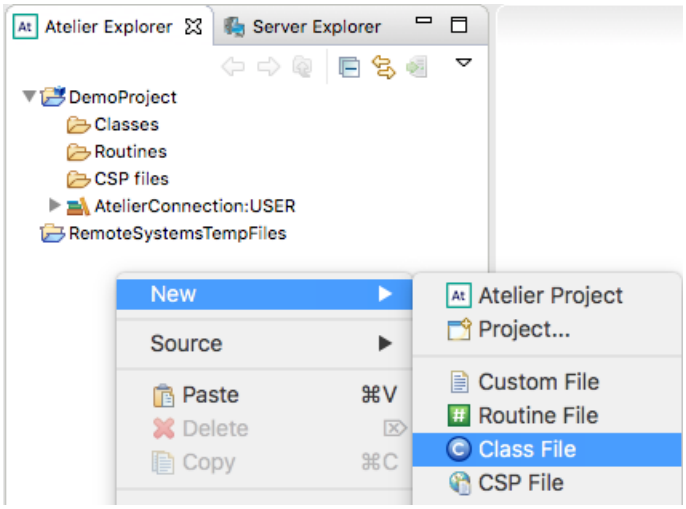


.

2.3 Creating a Class File

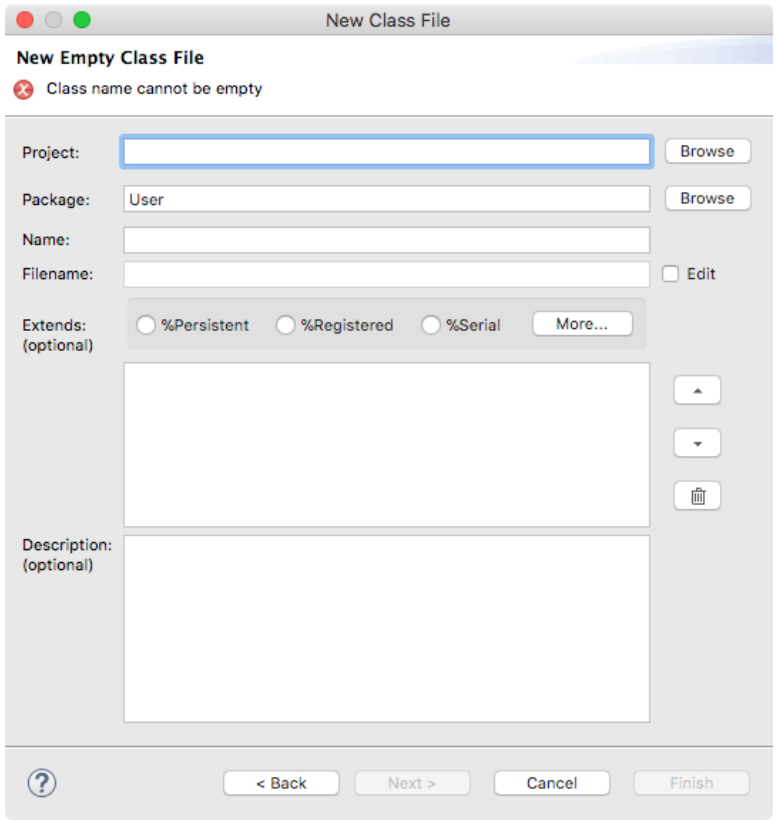
Files that contain Atelier classes end with a `.cls` extension, and are referred to as class files. Follow these steps to create an Atelier class file:

1. **Right-click** in the **Atelier Explorer** view to open the context menu and select **New > Class File**, as illustrated in the following screen shot:



   This action opens the **New Class File** wizard.

2. The first page of the wizard lets you select a file template. Select the **Empty Class** template and click **Next**. The next page lets you configure the class:



3. Fill in the fields in the Atelier class wizard:

   a. In the **Project** field, use the **Browse** button to select the project you just created.

   b. In the **Package** field, keep the default package name `User`.

   c. In the **Name** field, enter a name for the class. We'll use `MyClass`.

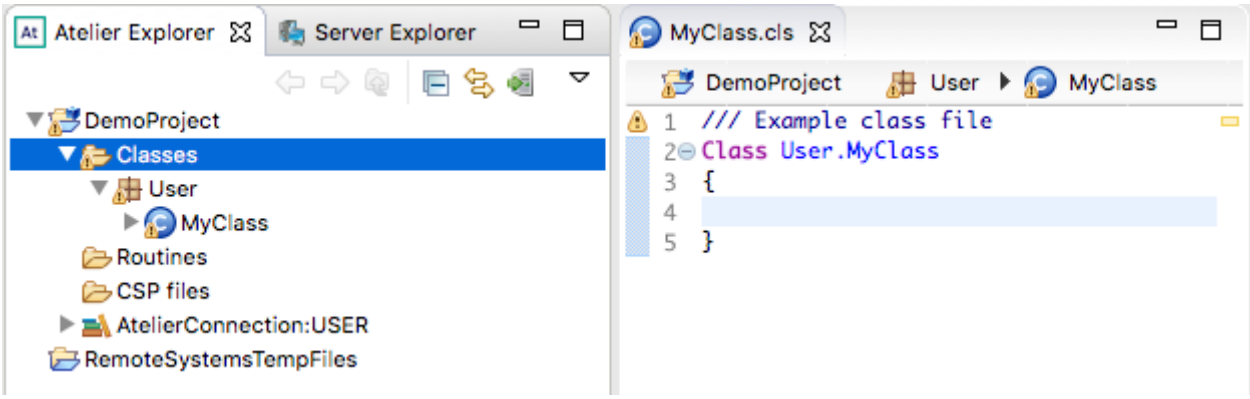
   d. The **Filename** field contains the name of the class file as it appears in the local file system. The default value is the class name you just entered, with the extension `.cls`.

   e. The **Extends** field lets you specify parent classes. This field is optional, and we'll ignore it.

   f. In the **Description** field, enter `Example class file`. This field is also optional.

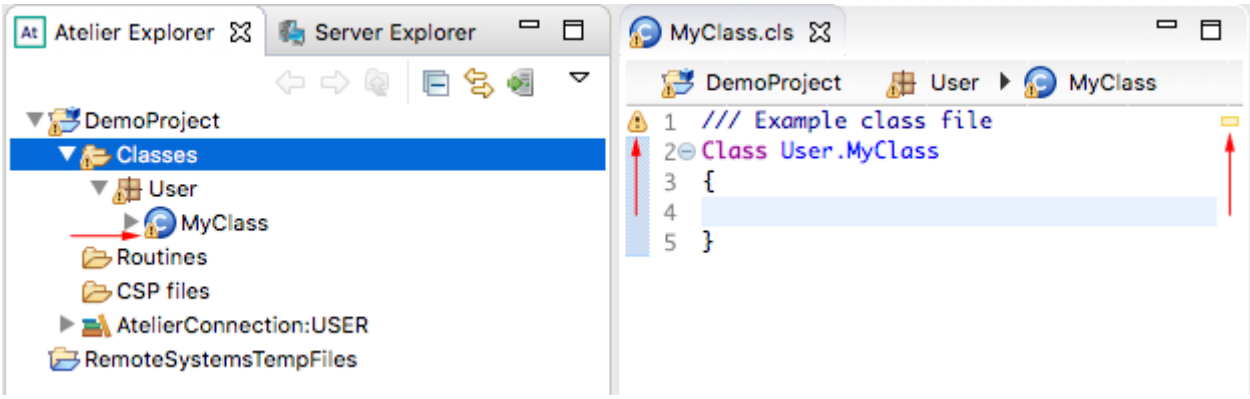
   g. Click **Finish** to create the class file.

4. At this point, your new class is listed in the **Atelier Explorer** and open in the editor, as shown in the following screen shot:



.

Note that there is a warning badge on the name of the new class in the **Atelier Explorer** and in the editor, pointed out by the red arrows in this screen shot:

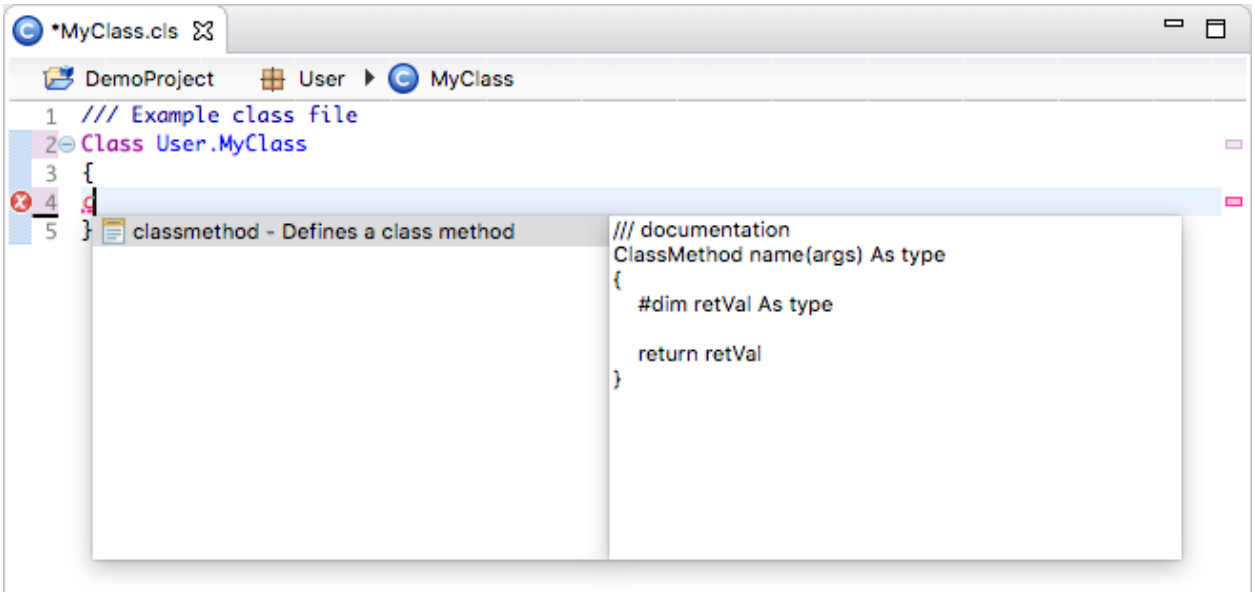


If you hover the cursor over the small yellow rectangle in the upper-right margin of the editor area, you see the warning: `Resource is out of sync in namespace`. This problem results from the file having been created and saved on the client, but not yet synchronized to the server. This issue will be resolved when you edit and save the file in a later step.

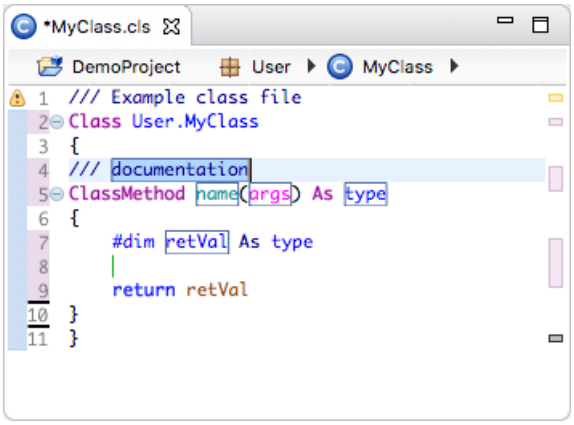
2.4 Adding a Class Method

You can now add a class method to the class:

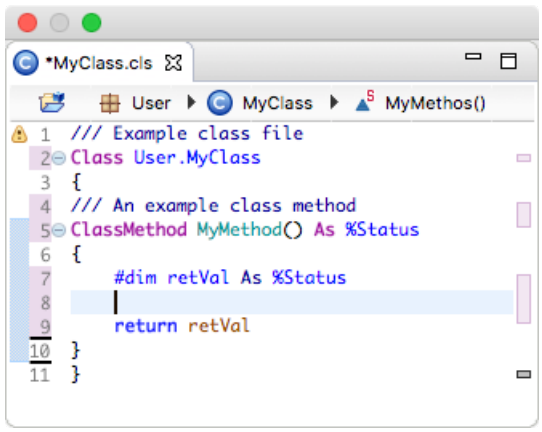
1. You can use the Eclipse content assist feature to help you add a class method. In the editor, position the cursor in the body of the class and type `c` followed by **CTRL+Space**. A content-assist pop-up appears that suggests possible completions given cursor position and what you've already typed. The following screen shot shows the pop-up:



Select the **classmethod** template and type **Enter**. The class editor adds the class method template, as shown in the next screen shot.



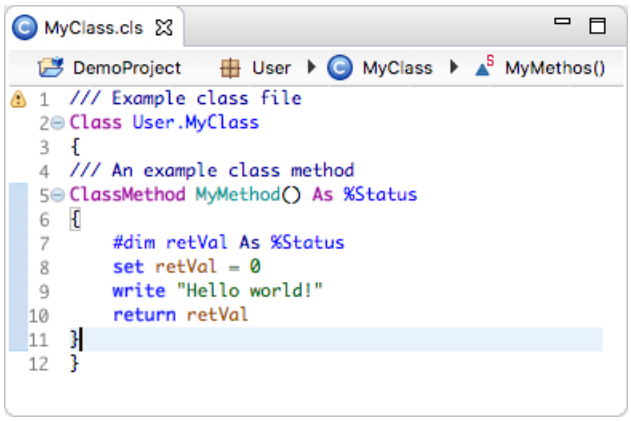
2. Next, fill in the template. First replace the **documentation** placeholder with `An example class method`, then use **tab** to advance to the next field. Replace **name** with `MyMethod`. In the next field, remove the placeholder value for arguments. Then set **type** to `%Status`. Again, content assist can help you. Type `%St` followed by **CTRL+Space**. Select **%Status** and type **Enter**. Now, the class method looks like this:



3. In this step, you fill in the rest of the content for this sample class. Above the return statement, add the following lines:

```
set retVal = $$$OK
write "Hello world!"
```

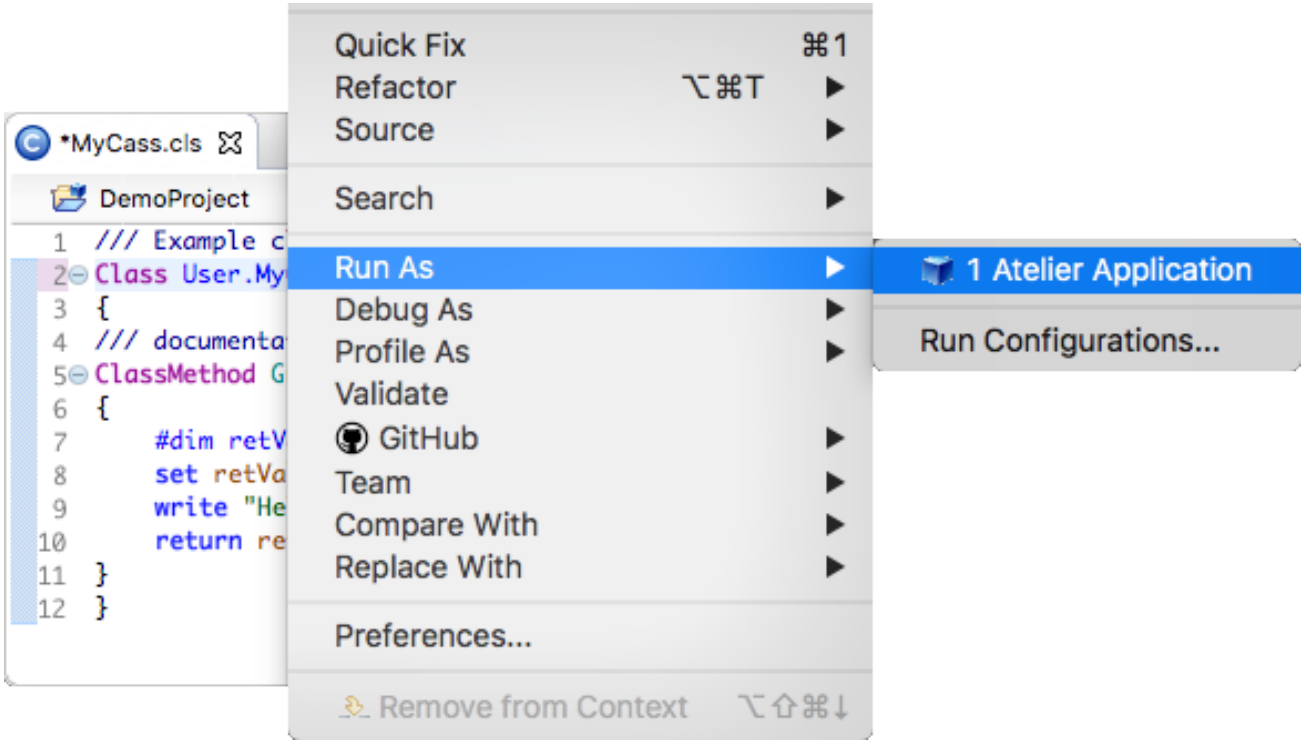
Now, the finished class method looks like this:



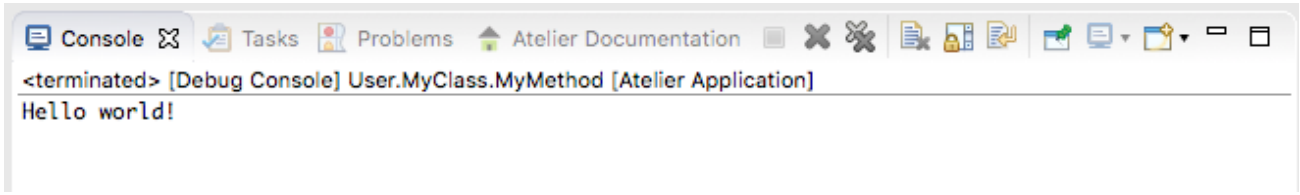
4. Right-click and select **Save**. This action saves the file to the local file system, synchronizes the file with the server, and compiles it.

2.5 Running the Class

Now you can use Atelier to run the class you have just created. **Right-click** in the **Atelier Editor** view and select **Run As > Atelier Application**, as shown in the following screen shot:



Atelier runs the class, and you can see the output of the class method `MyMethod` in the **Console** view:



3 Learn More About Atelier

InterSystems provides several resources to learn more about Atelier:

- Atelier Videos — Step-by-step videos that guide you through a number of basic development tasks in Atelier.
- Atelier Documentation
- Atelier Group — An online forum in the InterSystems Developer Community where you can connect with other developers, ask questions, and read articles others have posted.