



Unit Testing with %UnitTest, Perforce, and Studio

Version 5.2
01 September 2006

Unit Testing with %UnitTest, Perforce, and Studio
Caché Version 5.2 01 September 2006
Copyright © 2006 InterSystems Corporation.
All rights reserved.

This book was assembled and formatted in Adobe Page Description Format (PDF) using tools and information from the following sources: Sun Microsystems, RenderX, Inc., Adobe Systems, and the World Wide Web Consortium at www.w3c.org. The primary document development tools were special-purpose XML-processing applications built by InterSystems using Caché and Java.



The Caché product and its logos are registered trademarks of InterSystems Corporation.



The Ensemble product and its logos are registered trademarks of InterSystems Corporation.



The InterSystems name and logo are trademarks of InterSystems Corporation.

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.

Caché, InterSystems Caché, Caché SQL, Caché ObjectScript, Caché Object, Ensemble, InterSystems Ensemble, Ensemble Object, and Ensemble Production are trademarks of InterSystems Corporation. All other brand or product names used herein are trademarks or registered trademarks of their respective companies or organizations.

For Support questions about any InterSystems products, contact:

InterSystems Worldwide Customer Support

Tel: +1 617 621-0700
Fax: +1 617 374-9391
Email: support@InterSystems.com

Table of Contents

- Unit Testing with %UnitTest, Perforce, and Studio..... 1**
- 1 Set Up 2
- 2 General Use 2
- 3 Procedure Suggested by QD 3

Unit Testing with %UnitTest, Perforce, and Studio

This procedure is written for Intersystems developers. It describes a way to set up Studio and Perforce so that you can create and run unit tests easily (using %UnitTest) as you change or develop code. You are strongly encouraged to create unit tests as you code and to use the structure described here.

The development environment suggested here for unit tests is similar to the one used at Intersystems for %SYS.

This procedure uses Perforce as the source code control system (sccs). If you are a developer that is not at Intersystems, use this procedure as a model for developing your own structure for use with your own source code control system.

This procedure

- Makes Quality Development's job easier.
- Helps ensure that you submit code that works.
(You'll unit test your code before you submit it.)
- Helps ensure builds that work.
(You'll run ALL the tests in the Perforce branch after you submit yours.)
- Improves the speed of code development.

The decision to implement unit testing is something that you can discuss with your manager or the QD manager, but you should at least set up this configuration and run the existing tests from time to time. Some developers have found some rather obscure bugs doing this.

More information on using the **Manager.RunTest** method and creating TestCase classes can be found in the class documentation for the %UnitTest package and in the [%UnitTest tutorial](#).

1 Set Up

1. Create two namespaces, one for developing unit test classes, called **UT**, and one for running debugging the tests, called **TEST**. (Names here are suggestions.)
2. In the **UT** namespace, set **^SourceRoot** to the correct Perforce Branch, such as `c:/perforce/dev/latest/`.
3. Set **^UnitTestRoot** to the same value as **^SourceRoot**, with the addition of `/internal/testing/unit_tests`. These three directory levels are mandatory. For example, with the Perforce branch above, the **^UnitTestRoot** would be:

`c:/perforce/dev/latest/internal/testing/unit_tests`
4. Synchronize the `unit_tests` directory with Perforce as needed, such as `dev` and `build`, with **Sync to Head Revision**.
5. Load files into the **UT** namespace. (You'll run `d ^%RI` the first time, but not thereafter.)

```
d ^%RI
d ^build
```

(Intersystems developers can find more info on Perforce/Studio integration on this Wiki page: <http://turbo.iscinternal.com/twiki/bin/view/ISC/StudioSourceControlIntegration>.)

2 General Use

With the Perforce/Studio integration, when you open a class or routine in Studio, relevant files are automatically checked out of Perforce and imported into Studio. They are automatically exported when you save.

1. Create all your unit tests in the directory `/unit_tests/dev` directory (which QD has reserved for the development unit tests).
2. One suggested method is to create your test classes in a package name in the form `dev.obj.devchangekey`. (SQL developers could use `dev.sql.devchangekey`.)
3. Run your tests with the command

```
d ##class(%UnitTest.Manager).RunTest("dev/obj:devchangekey")
```

4. After you run your own test successfully, run all the tests in the dev directory to ensure that your code has not broken something.
5. Submit your change and your test in Perforce.
6. Use Perforce to port to other branches.

3 Procedure Suggested by QD

The procedure is this:

1. Open a ProdLog issue for edit.
2. Click the **Dev Change** link to create a Dev Change to get a Dev Change key.
3. Change to the **UT** namespace.
4. Create a new class with the package name `dev.obj.devchangekey` (or `dev.sql.devchangekey`) that extends `UnitTest.TestCase`. When it asks if you want to add it to Source Code Control, click **Yes**.
5. When you submit your change, submit your tests at the same time. (You might create a changelist that includes the code and the tests to make submission easy.)

