

# **The Gallio Book**

**The Gallio Documentation Team**

---

# **The Gallio Book**

by The Gallio Documentation Team

3.1 beta

Copyright © 2010 Gallio Project

---

---

---

---

# Table of Contents

|  |      |
|--|------|
| Preface .....                            | vi   |
| The Aim Of This Book .....               | vi   |
| Who Should Read This Book? .....         | vi   |
| How To Read This Book .....              | vii  |
| Structure Of The Book .....              | viii |
| This Book Is Free .....                  | ix   |
| What Is MbUnit? .....                    | ix   |
| What Is Gallio? .....                    | x    |
| The History Of Gallio & MbUnit .....     | x    |
| What's New In Gallio & MbUnit? .....     | xi   |
| I. Using MbUnit and Gallio .....         | 1    |
| 1. Installing & Configuring Gallio ..... | 3    |
| Installing Gallio .....                  | 3    |
| Using the MSIs .....                     | 3    |
| Completing the Installation .....        | 13   |
| Using the ZIP Installer .....            | 14   |
| Configuring Gallio .....                 | 16   |
| Runtime Settings .....                   | 16   |
| Icarus Settings .....                    | 17   |
| AutoCAD Settings .....                   | 24   |
| TestDriven.NET Settings .....            | 25   |
| Ambience Settings .....                  | 26   |
| Conclusion .....                         | 27   |
| 2. My First Tests .....                  | 28   |
| 3. MbUnit Fundamentals .....             | 29   |
| 4. Testing Recipes 1 .....               | 30   |
| 5. Testing Recipes 2 .....               | 31   |
| 6. Domain Testing .....                  | 32   |
| 7. Test Reports .....                    | 33   |
| 8. Extending MbUnit .....                | 34   |
| 9. Automated Testing With Gallio .....   | 35   |
| II. Developing MbUnit and Gallio .....   | 36   |
| 10. Gallio Development Basics .....      | 38   |
| 11. Creating Gallio Plugins .....        | 39   |
| 12. Creating Test Frameworks .....       | 40   |
| III. Appendices .....                    | 41   |
| A. Migration Guide .....                 | 43   |
| B. Creative Commons License .....        | 44   |
| Creative Commons Notice .....            | 48   |
| Glossary .....                           | 49   |

---

## List of Figures

|  |    |
|--|----|
| 1.1. Starting the Gallio Installer .....                 | 4  |
| 1.2. Choosing Setup Type .....                           | 5  |
| 1.3. The Custom Setup Dialog .....                       | 6  |
| 1.4. Choosing Gallio Options To Install .....            | 7  |
| 1.5. Choosing MbUnit v3 Options To Install .....         | 8  |
| 1.6. Choosing Test Framework Adapters To Install .....   | 9  |
| 1.7. Choosing Test Runners To Install .....              | 10 |
| 1.8. Choosing Test Tools To Install .....                | 11 |
| 1.9. Choosing Documentation And Samples To Install ..... | 12 |
| 1.10. Choosing Extras To Install .....                   | 13 |
| 1.11. The Gallio Start Menu .....                        | 14 |
| 1.12. The Contents Of The Gallio Zip Download .....      | 15 |
| 1.13. Configuring Additional Gallio Plugins .....        | 17 |
| 1.14. Configuring Icarus Progress Monitoring .....       | 18 |
| 1.15. Configuring Icarus Test Status Colours .....       | 19 |
| 1.16. Configuring Icarus Test Explorer Settings .....    | 20 |
| 1.17. The Icarus Test Explorer Window .....              | 21 |
| 1.18. Configuring Test View Categories .....             | 22 |
| 1.19. Configuring Test Reports .....                     | 23 |
| 1.20. Configuring Icarus Startup Actions .....           | 24 |
| 1.21. Configuring AutoCAD Integration .....              | 25 |
| 1.22. Configuring TestDriven.NET Integration .....       | 26 |
| 1.23. Accessing The Ambience Service .....               | 27 |

---

# Preface

MbUnit is unit testing on crack.  
--Scott Hanselman, Microsoft

In the world of software development, the concept of *test-driven development* (TDD) is not a new one. Conceived in 1999 as part of the *Extreme Programming* process, its popularity within the .NET world began to rise in October 2002 when *nUnit* v2.0 was released. A completely reworked version of *jUnit* written entirely in C#, it started many .NET developers down the agile programming path. It was simple to learn, easy to use, and quite versatile. However, just as *nUnit* had many advantages, it also had several disadvantages and other viable test frameworks such as *MbUnit* and *xUnit* have emerged.

MbUnit began life in 2004 when Marc Clifton wrote a series of articles proposing the formalization of various test patterns beyond basic test-driven development. A young developer by the name of *Jonathan "Peli" de Halleux* read them while recovering from surgery and proceeded to write the first prototype of MbUnit (or *gUnit* as it was first called) before he had left hospital. MbUnit wasn't designed as a replacement for *nUnit*, but thanks to very smart developers such as *Jamie Cansdale* and *Andy Stopford*, it became a definite, more powerful alternative to its kindred framework.

In Autumn 2007, a new face took the reins of the MbUnit project – *Jeff Brown* – and started a ground-up rewrite of MbUnit as two separate projects. **MbUnit v3** would remain the test framework that developers knew and used, but the automation platform underneath the framework was to be isolated and made framework-neutral. This new platform, codenamed **Gallio** (because of a typo), would aim to run any framework's tests – for example *nUnit*, *nBehave* or *csUnit* – and allow them to be run from any host – for example *Resharper*, *TeamCity* or *Visual Studio*. It was agreed that in doing this, other open-source and commercial projects would be able to leverage the platform's services to create rich, interoperable and extensible testing solutions, thereby adding great value to the community. Going forward, the Gallio project continues to become visible to other open source projects so that the capabilities of the platform can bring unity and value to the many projects in the testing space.

This book is written to document the 3.1 series of the MbUnit test framework and Gallio Automation Platform. We have made every attempt to make it as accurate as possible. However, both projects are in active development, so while we will attempt to keep this text up to date with the software, please note that future versions may change some of the functionality detailed in this text.

## The Aim Of This Book

This book has two aims:

- to bring the reader up to speed with the capabilities of MbUnit v3 and Gallio
- to make it easier to start contributing back to the project

It is not our intention to look at the theory of unit testing or the application generic unit test patterns to your own applications beyond the basic introduction to test-driven development given in Chapter 2, *My First Tests*.

## Who Should Read This Book?

This book is aimed at .NET developers with an interest in writing tests for their applications. It is written on the premise that the reader is coming to unit testing for the first time. Those readers who have used

nUnit or MbUnit v2 before will be able to use our Appendix A, *Migration Guide* as a quick reference to find the information they need to use MbUnit v3. All our examples will initially be written in C# and subsequently translated into Visual Basic .NET, F# and whatever other flavours of CLR language people kindly volunteer to translate them into.

Even if you don't intend to use MbUnit as your test framework, you might still want to use Gallio to run your tests, so you may also find parts of the text useful. Similarly, while Gallio does allow for the running of tests within other applications, we'll focus on using the two applications it ships with to run tests:

- *Echo* is a command-line based application
- *Icarus* is a GUI-based application

Note also that the source code examples used throughout the book are only examples. While they will compile with the proper compiler incantations, they are intended to illustrate a particular scenario and not necessarily to serve as examples of good programming style or practices.

Books like Roy Osherove's *The Art Of Unit Testing* [<http://www.manning.com/osherove/>] and Gerard Meszaros's *Unit Test Patterns* [<http://xunitpatterns.com>] teach various types of testing so rather than be a full-on introduction to testing in general, it will stay very specific to what is available in MbUnit v3 supported by Gallio.

## How To Read This Book

Because this book is aimed at several different groups of readers, the structure of this book is aimed at what we perceive to be the largest of these groups – the newcomer to test-driven development who wants to write tests and learn by doing. That doesn't preclude anyone else from reading this however and we'd suggest the following routes through the book for various types of readers:

Newcomers to Test-Driven Development & MbUnit

If you're just starting out with test-driven development, install the default options shown in Chapter 1, *Installing & Configuring Gallio* and start reading from Chapter 2, *My First Tests* to the end of the book.

Those who want to move to MbUnit v3 from another test framework or MbUnit v2

The assumption here is that you are already familiar with a test framework and want to know how to switch to MbUnit v3. Use Appendix A, *Migration Guide* to see how the attributes and features in your framework compare up to MbUnit and Chapter 3, *MbUnit Fundamentals*, Chapter 4, *Testing Recipes 1*, Chapter 5, *Testing Recipes 2*, and Chapter 6, *Domain Testing* to find more detail on the specific implementation in MbUnit v3. Chapter 7, *Test Reports* demonstrates how to interpret the results of your tests once you are up and running and Chapter 2, *My First Tests* shows how you can run your current tests in Gallio while you make the switch. Chapter 8, *Extending MbUnit* covers how to extend the base MbUnit functionality with your own custom code.

Those who just want to use Gallio to run their tests

If you're happy using your current test framework but would like to leverage Gallio as the platform to run your tests, Chapter 8, *Extending MbUnit* shows how you can run your current tests in Gallio, and Chapter 2, *My First Tests*, how to debug into your code using Gallio. Chapter 7, *Test Reports* covers how to use the reports generated by Gallio after a test report and Chapter 9, *Automated Testing With Gallio* looks at how to integrate it into an automated test build.

If you'd like to contribute to the Gallio or MbUnit code base with your own fixes and features, check out Part II, "Developing MbUnit and Gallio" on how to start.

## Structure Of The Book

This book is broadly divided into three sections:

1. Section 1 is all about using Gallio and MbUnit to build your own applications.

|                               |   |
|-------------------------------|---|
| Installation                  | Walks you through the various options you have to install Gallio and MbUnit and the applications it can hook into.  |
| My First Tests                | Explains the basics of test-driven development and uses MbUnit to write tests for a basic application. You'll see how to run those tests first in Gallio's native applications, Icarus and Echo, and then within other applications for which Gallio has a plug-in. You'll also see tests written with other frameworks all run together within Gallio.   |
| MbUnit Fundamentals           | Building on Chapter 2, you'll look at the basic class and method attributes provided in MbUnit v3 for test construction. You'll see how to structure your application projects for testing, and step into your own code from running tests to debug it when your tests fail.  |
| Testing Recipes 1             | Discusses some of the core features in MbUnit which have made it one of the more popular test frameworks today. You'll look at row tests, how to supply metadata to tests from various resources, creating test factories, contract verifiers and how to order tests. We'll also cover how to run your tests in parallel and the problems that can create, as well as how to dynamic generate tests within 'test suites'. |
| Testing Recipes 2             | Looks at the various options you have to influence the tests you've written from outside the test classes themselves. You'll see how to use test assembly config files, the assembly decorators provided by MbUnit v3, inclusion and exclusion filters, and mock objects.   |
| Domain Testing                | Covers scenarios where MbUnit is being used to test specific domains. We'll look at testing database scripts and data access layers, web testing, and unit testing a more generic domain model with Moq and Rhino.Mocks. Finally, we'll look at how to deal with elevating privileges and the dreaded Vista\Win 7 UAC in your tests.  |
| Test Reports                  | Looks at the various types of report that Gallio generates after a test run, how to navigate them and how to customize their contents to your requirements.   |
| Extending MbUnit              | Takes the bonnet off the MbUnit engine and looks at how you can extend its core functionality with your own.  |
| Automated Testing With Gallio | For those who use automated test servers or who would like to get started building one, this chapter looks at how to control  |



Gallio with various build engines and how to integrate it into complementary applications such Cruise Control, TeamCity or nCover.

2. Section 2 is all about contributing to the MbUnit and Gallio project and will be developed further once section 1 is complete.
  - Development Basics
  - Creating Gallio Plug-ins
  - Creating Test Frameworks
3. Section 3 is the reference area containing a number of appendices for quick reference.
  - Migration Guide From MbUnit v2 to MbUnit v3
  - A copy of the Creative Commons Attribution License, under which this book is licensed.
  - Glossary of Terms

## This Book Is Free

One of the biggest problems in many open source projects is the lack of proper documentation. This is a shame because there is no point in implementing a thousand features if no one knows about them. The Gallio development team was very aware of this issue, so documentation was given a high priority from the beginning of the project. The API reference documentation, which is automatically generated from the source code, was the first sign of this concern, but the team knew users need more: they need a guide to show them how to use the multiple features, the runners and so on under different scenarios. So the idea of writing a book was there from the beginning, in our wishlist, but we knew it was a big task so it was postponed.

This book is an evolution of the documentation written in various forms (wikis, blog posts, etc) by MbUnit and Gallio developers into a single work. As such, it has always been under a free license (see Appendix B, *Creative Commons License*) and two things will remain true:

- You will always find the latest version of this book in the book's own Subversion repository [<http://code.google.com/p/mb-unit/source/browse/>].
- You can make changes to this book and redistribute it however you wish—it's under a free license. Your only obligation is to maintain proper attribution to the original authors. Of course, we'd much rather you send feedback and patches to the Gallio developer community, instead of distributing your private version of this book.

The online home of this book's development and most of the volunteer-driven translation efforts regarding it is <http://www.gallio.org/Docs.aspx>. There you can find links to the latest releases and tagged versions of the book in various formats, as well as instructions for accessing the book's Subversion repository (where its DocBook XML source code lives). Feedback is welcomed—encouraged, even. Please submit all comments, complaints, and patches against the book sources to [gallio-book@googlegroups.com](mailto:gallio-book@googlegroups.com) [<mailto:gallio-book@googlegroups.com>].

## What Is MbUnit?

MbUnit is a unit testing framework in the tradition of xUnit frameworks such as JUnit. In addition, MbUnit includes a rich suite of features designed to simplify other automation tasks that arise during integration

testing. The present incarnation of MbUnit, MbUnit v3, represents a complete rewrite and redesign of Peli's original work to improve the end-user experience, consolidate features, enhance extensibility, and enable advanced integration testing and reporting.

- MbUnit v3 is a .Net 2.0 based framework. It uses generic types and methods where possible to encourage code reuse. It also provides additional features for .Net 3.5 clients in a separate assembly. MbUnit v3 leverages the Gallio test automation platform heavily to provide integration with numerous other tools and to enable functionality such as rich reporting.
- MbUnit v2 is a .Net 1.1 based framework with a few .Net 2.0 add-ons. It is stand-alone framework that includes its own suite of test runners. Gallio includes an adapter plugin so that Gallio-based tools may also be used with MbUnit v2 (when running tests in a .Net 2.0 environment). However, since MbUnit v2 was not originally designed for Gallio, it does not provide as many advanced features as MbUnit v3.

MbUnit v2 is being maintained concurrently with MbUnit v3 for the benefit of existing projects based on MbUnit v2 that have not yet migrated to MbUnit v3. For new projects, we recommend adopting MbUnit v3.

MbUnit v3 is mostly backwards compatible with MbUnit v2 except for some APIs that have been renamed or redesigned. Transitioning to MbUnit v2 is relatively straightforward.

For more information about the differences between MbUnit v2 and v3, please refer to Appendix A, *Migration Guide*.

## What Is Gallio?

The Gallio Automation Platform is an open, extensible, and neutral system for .NET that provides a common object model, runtime services and tools (such as test runners) that may be leveraged by any number of test frameworks.

## The History Of Gallio & MbUnit

In January 2004 Marc Clifton, a frequent contributor at Codeproject, wrote a series of articles that sought to expand the unit testing discussion. Among other things, Marc proposed a formalization of various test patterns beyond basic TDD. Marc then took his ideas into code as AUT (Advanced Unit Testing), an independent project that you can find at Codeproject.

Two months later, Jonathan "Peli" de Halleux took a look at Marc's proposals and created gUnit (which was later renamed to MbUnit) while recovering from surgery in a hospital. In fact, Peli wrote most of MbUnit while still in the hospital.

MbUnit had some new ideas and concepts and it caught the attention of Jamie Cansdale who while on a trip to Brussels hooked up with Peli to work on an add-on for TD.net. TD.net started life as a NUnit project and so this made MbUnit the next framework after NUnit to be supported by TD.net, as such since the very early days of this great tool there has been MbUnit support.

In 2005 Peli made MbUnit opensource and continued working on the framework while finishing his PhD. Shortly after completing his PhD he accepted a job with Microsoft as a *SDET* on the CLR team. Unable to carry on MbUnit, he handed it over to Jamie Cansdale as short time caretaker. Peli blogged about needing someone to take on MbUnit and shortly after Andy Stopford as a long time MbUnit user read this and stepped up.

Since then MbUnit has grown as a framework and project, with two major releases and triple the downloads per release it has firmly rooted itself in main stream Microsoft .net culture as a viable unit test framework next to NUnit.

In the autumn of 2007, MbUnit v3 - a ground up rewrite of MbUnit, started. In one of those funny turn of events, v3 was to be code named *Galileo* but due to a typo became *Gallio*. The name stuck and development continued on MbUnit v3, code name: Gallio.

With MbUnit v3 development well under way, long time MbUnit core member Jeff Brown attended the *Alt.Net* conference in Austin, Texas. Following discussions with other programmers at the conference, Jeff made the case to the MbUnit team that there was value to the community at large in isolating the test runner capabilities of the system to create a neutral platform upon which MbUnit could then be hosted as one of many supported frameworks. Other open-source and commercial projects would be able to leverage the platform's services to create rich, interoperable and extensible testing solutions, thereby adding great value to the community.

After much discussion, the decision was made to separate the test runner from MbUnit and Gallio the Automation Platform was born.

Going forward the Gallio Project seeks to become visible to other open source projects so that the capabilities of the platform can bring unity and value to the many projects in the testing space.

## What's New In Gallio & MbUnit?

The first version of this book is aimed for release during the v3.1 life-cycle. A quick run-down of new features in subsequent version will be found here. Full release notes can be found at <http://www.gallio.org/Docs.aspx>.

---

# **Part I. Using MbUnit and Gallio**

---

---

## Table of Contents

|  |    |
|--|----|
| 1. Installing & Configuring Gallio ..... | 3  |
| Installing Gallio .....                  | 3  |
| Using the MSIs .....                     | 3  |
| Completing the Installation .....        | 13 |
| Using the ZIP Installer .....            | 14 |
| Configuring Gallio .....                 | 16 |
| Runtime Settings .....                   | 16 |
| Icarus Settings .....                    | 17 |
| AutoCAD Settings .....                   | 24 |
| TestDriven.NET Settings .....            | 25 |
| Ambience Settings .....                  | 26 |
| Conclusion .....                         | 27 |
| 2. My First Tests .....                  | 28 |
| 3. MbUnit Fundamentals .....             | 29 |
| 4. Testing Recipes 1 .....               | 30 |
| 5. Testing Recipes 2 .....               | 31 |
| 6. Domain Testing .....                  | 32 |
| 7. Test Reports .....                    | 33 |
| 8. Extending MbUnit .....                | 34 |
| 9. Automated Testing With Gallio .....   | 35 |

---

# Chapter 1. Installing & Configuring Gallio

Before you can do anything, you'll need to install Gallio & MbUnit, and if required configure it to your needs. In this chapter, we'll look at both installation and configuration.

## Installing Gallio

To install Gallio and MbUnit v3, you'll need to visit [gallio.org/downloads.aspx](http://gallio.org/downloads.aspx) [<http://gallio.org/downloads.aspx>] and download the version you need for your system. There are three options:

- An installer for 64-bit systems (`GallioBundle-3.x.y.z-Setup-x64.msi`)
- An installer for 32-bit systems (`GallioBundle-3.x.y.z-Setup-x86.msi`)
- A zip file of the code for you to install as you will (`GallioBundle-3.x.y.z.zip`)

A fourth option is to download the source code for Gallio from its Google Code project (<http://code.google.com/p/mb-unit>) and build it manually. You'll find instructions on how to do this in Part II, “Developing MbUnit and Gallio” .

## Using the MSIs

Let's start by using the installer. Once you've downloaded the correct installer for your system, double-click it to start the installation. You'll soon see the start screen as shown in Figure 1.1, “Starting the Gallio Installer”.

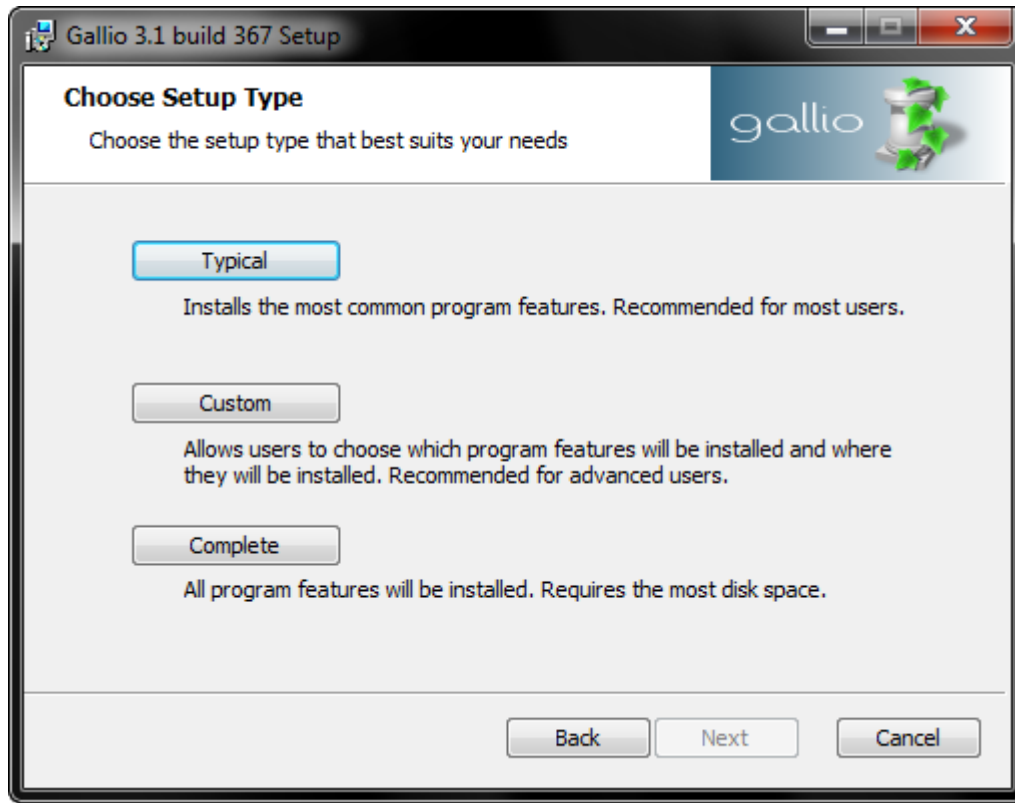
**Figure 1.1. Starting the Gallio Installer**



Click Next. The second screen shows you that Gallio and MbUnit are licensed under the Apache Licence v2.0 and where to get a copy of it. For reference, you can find it at <http://www.apache.org/licenses/LICENSE-2.0>. Check the box accepting the terms of the agreement and then hit Next.

The third screen, shown in Figure 1.2, “Choosing Setup Type” offers three levels of setup option.

**Figure 1.2. Choosing Setup Type**

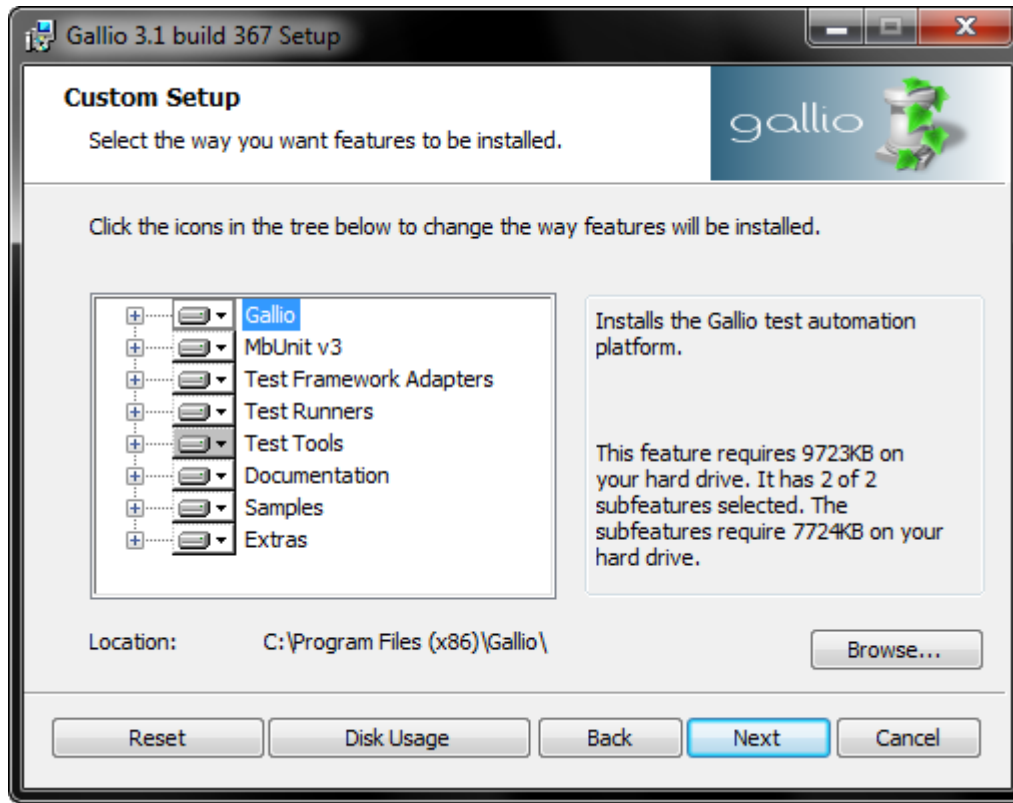


Clicking Custom will show you all the possible options that can be installed as part of this setup. By default, those already selected are the options installed as part of the Typical setup. Selecting all the options is the same as choosing the Complete setup option.

## Choosing Your Options

The custom setup option, shown in Figure 1.3, “The Custom Setup Dialog” allows you to install only the elements of Gallio that you need for your system along with any documentation or samples you might need. In this section we’ll run through the various options this custom setup allows you to pick.



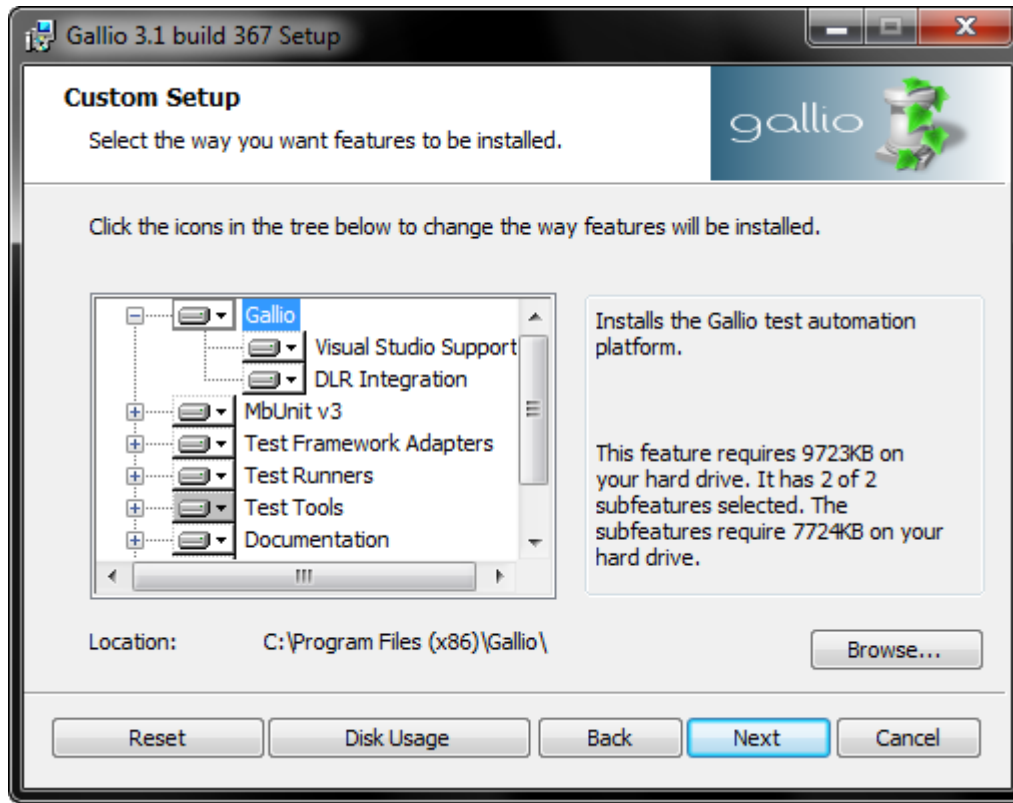
**Figure 1.3. The Custom Setup Dialog**

We'll walk through them in order.

## Gallio

The first option, shown in Figure 1.4, “Choosing Gallio Options To Install”, installs the main Gallio automation framework and optionally the libraries that allow Gallio to interact with Visual Studio for debugging through tests and for navigating to source. You can also install:

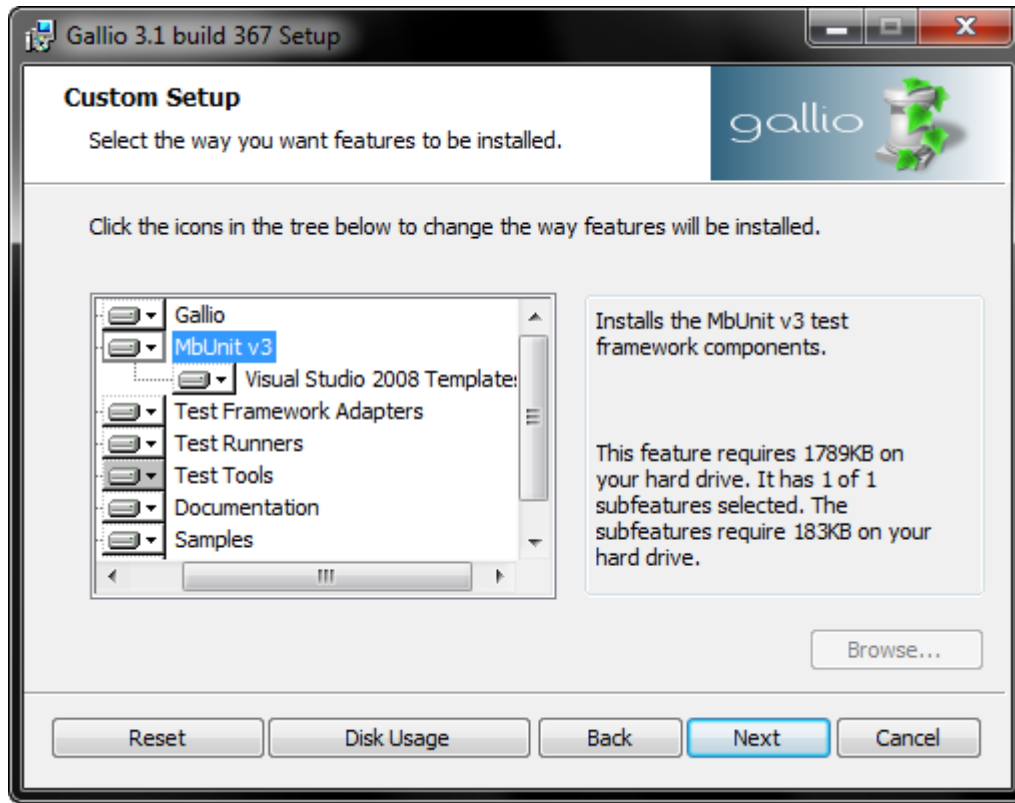
- A set of libraries that allow Gallio to work with the dynamic library runtime enabling use with *IronPython.NET* and *IronRuby.NET*, for example.

**Figure 1.4. Choosing Gallio Options To Install**

### MbUnit v3

The second option, shown in Figure 1.5, “Choosing MbUnit v3 Options To Install”, installs the libraries for MbUnit v3. Optionally, you can also install

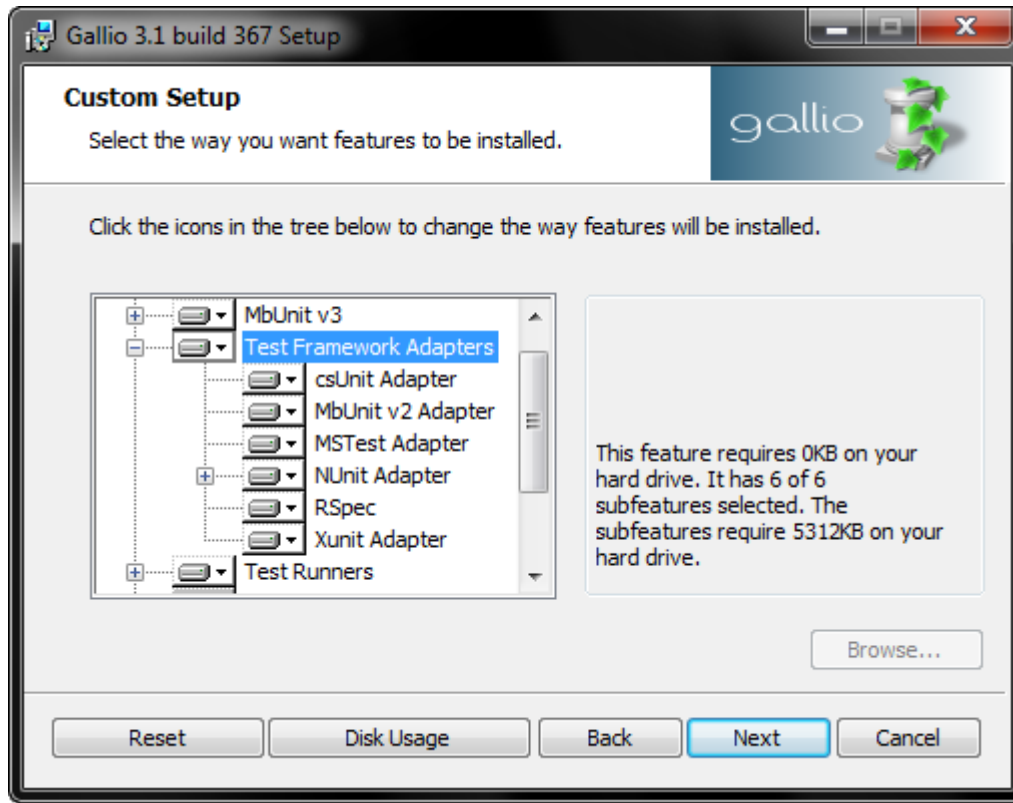
- A set of project templates for Visual Studio that allow for easier set up of test projects with MbUnit v3.

**Figure 1.5. Choosing MbUnit v3 Options To Install**

## Test Framework Adapters

One half of Gallio's flexibility comes from its ability to run tests written with any framework. All it needs is an adapter written to tell it how to run those tests and interpret the results. The third option, shown in Figure 1.6, "Choosing Test Framework Adapters To Install", allows you to select from those adapters already written.

- csUnit : Allows you to run csUnit tests with Gallio
- MbUnit v2: Allows you to run MbUnit v2 tests with Gallio
- MSTest: Allows you to run *MSTest* tests with Gallio (only if you have Visual Studio 2008 / Visual Studio Team System installed)
- NUnit: Allows you to run NUnit tests with Gallio
- RSpec: Allows you to run *RSpec* tests with Gallio
- xUnit.NET: Allows you to run xUnit tests with Gallio

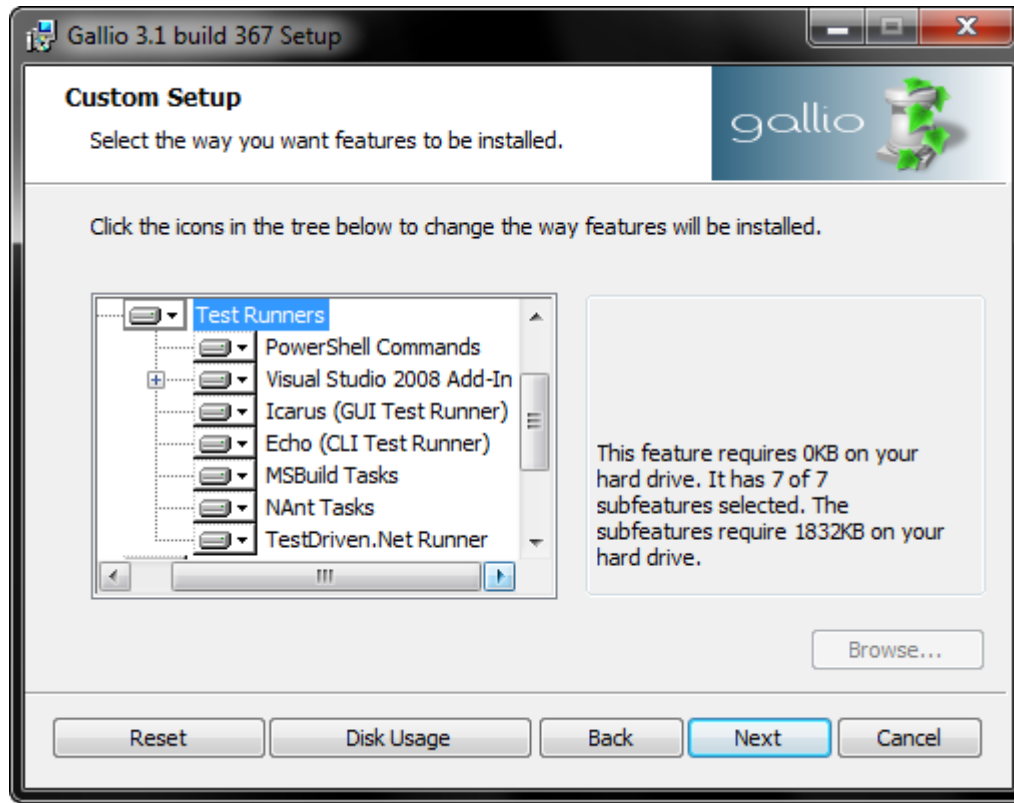
**Figure 1.6. Choosing Test Framework Adapters To Install**

Note that unless you need to run tests created with these frameworks, you don't need to install these plugins.

## Test Runners

The other half of Gallio's flexibility is its test runners. *Runners* are programs or plug-ins for other programs that provide different ways to run tests through Gallio. The fourth option, shown in Figure 1.7, "Choosing Test Runners To Install", allows you to select which runners to install for other programs already installed on your machine. You have the choice of:

- PowerShell commands: Allows you to run Gallio from a PowerShell script
- ReSharper v4.5 plug-in: Allows you to run Gallio from ReSharper
- Visual Studio 2008 Add-in: Allows you to run Gallio from Visual Studio's Team System
- Icarus: Gallio's own graphical test runner
- Echo: Gallio's own console test runner
- MSBuild tasks: Allows you to run Gallio from an *MSBuild* build script
- NAnt tasks: Allows you to run Gallio from a *NAnt* build script
- TestDriven.NET runner: Allows you to run tests with Gallio from the *TestDriven.NET* Visual Studio add-in

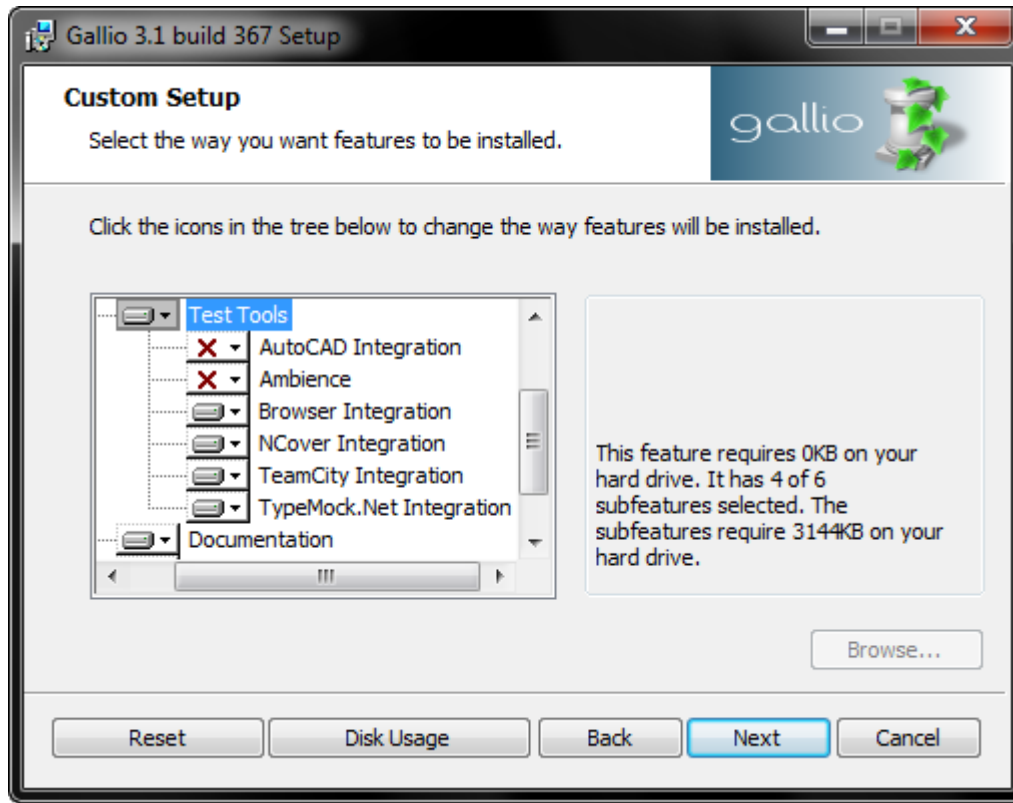
**Figure 1.7. Choosing Test Runners To Install**

Note that no matter how you choose to run Gallio, you will always be able to run tests written against any of the frameworks you have installed adapters for. For example, you'll be able to run csUnit tests through Resharper as long as you have installed the Resharper test runner and the csUnit framework adapter.

## Test Tools

The fifth set of options, shown in Figure 1.8, “Choosing Test Tools To Install”, allows some integration between Gallio and other tools you have installed separately on your machine.

- **AutoCAD Integration:** Allows tests that depend on the AutoCAD API (*ObjectARX*) to run inside the AutoCAD process.
- **Ambience:** *Ambience* is Gallio's own utility for storing persistent data across multiple test runs. We'll look at using *Ambience* in Chapter 5, *Testing Recipes 2*
- **Browser Integration:** Allows you to navigate directly from Gallio test reports to the source code the report is highlighting.
- **NCover integration:** Allows you to enable code coverage with *NCover* by simply setting a property
- **TeamCity integration:** Allows you to integrate Gallio reports into *TeamCity*.
- **TypeMock integration:** Allows you to enable *TypeMock.NET* by simply setting a property

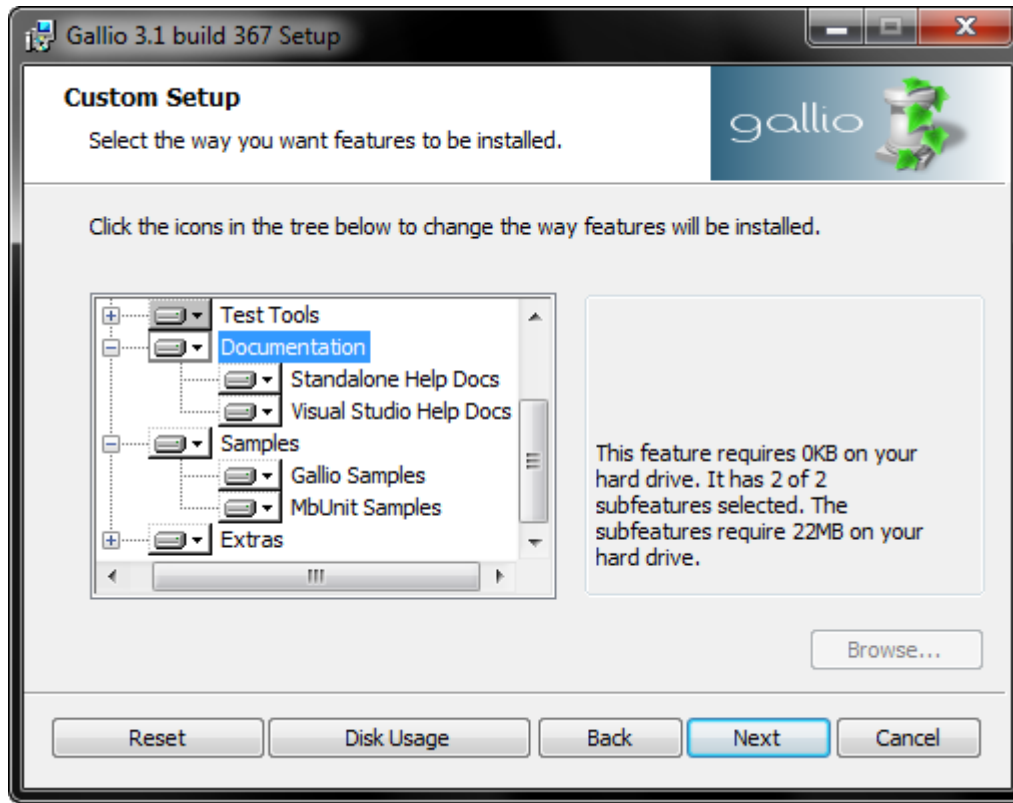
**Figure 1.8. Choosing Test Tools To Install**

We'll look at the various ways that Gallio reports integrate into the browser and more in Chapter 7, *Test Reports*.

## Documentation & Samples

The Gallio installer also comes with a variety of documentation and code samples for your reference. Figure 1.9, "Choosing Documentation And Samples To Install" shows the options. You have the choice of:

- Standalone Help Docs: Puts the standalone CHM documentation file on your system
- Visual Studio Help Docs: Installs the integrated documentation for Visual Studio
- Gallio Samples: Puts the (to be written) Gallio code samples on your system.
- MbUnit Samples: Puts the MbUnit code samples on your system.

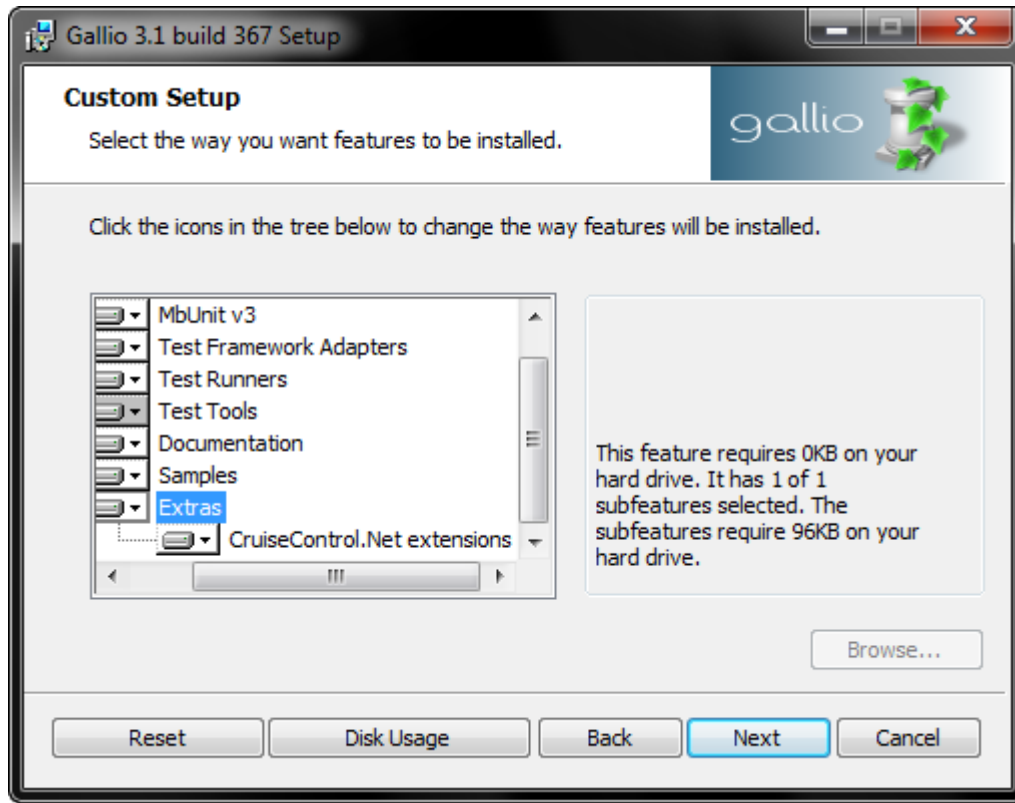
**Figure 1.9. Choosing Documentation And Samples To Install**

The samples and standalone documentation can be found at `%programfiles%\Gallio\samples` and `%programfiles%\Gallio\docs` for your use later on. The samples are saved in zip files for you to extract yourself.

## Extras

The final set of options, shown in Figure 1.10, “Choosing Extras To Install”, contain just one feature at the moment.

- **CruiseControl .Net extensions:** Provides an extension to allow downloading attachment from the *CCNet* build report. It's only useful if you are implementing a build server.

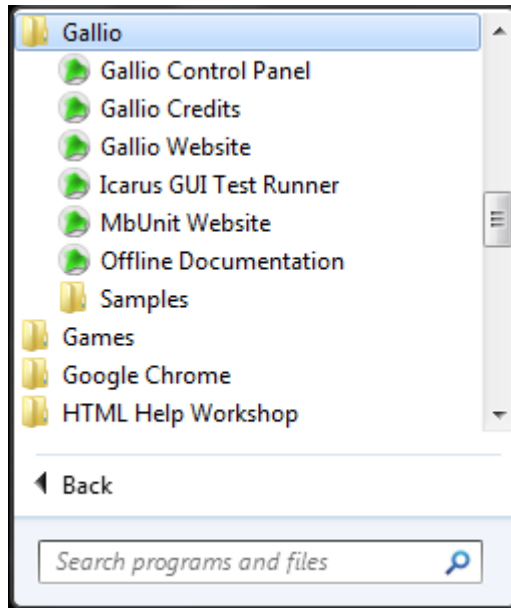
**Figure 1.10. Choosing Extras To Install**

## Completing the Installation

The wizard is now ready to install your options. Click Install to have it run. If you ran the setup program as a standard user in Windows Vista or Windows 7, User Account Control will ask you for an administrator's account and password before installation begins. When everything is installed, the wizard lets you know.

Click Finish to close the setup wizard. You'll find that the installer has created a folder in the Start Menu (Figure 1.11, "The Gallio Start Menu", where you will see shortcuts for the Icarus GUI Test Runner, the MbUnit Website, the Offline Documentation, and the online documentation amongst others.

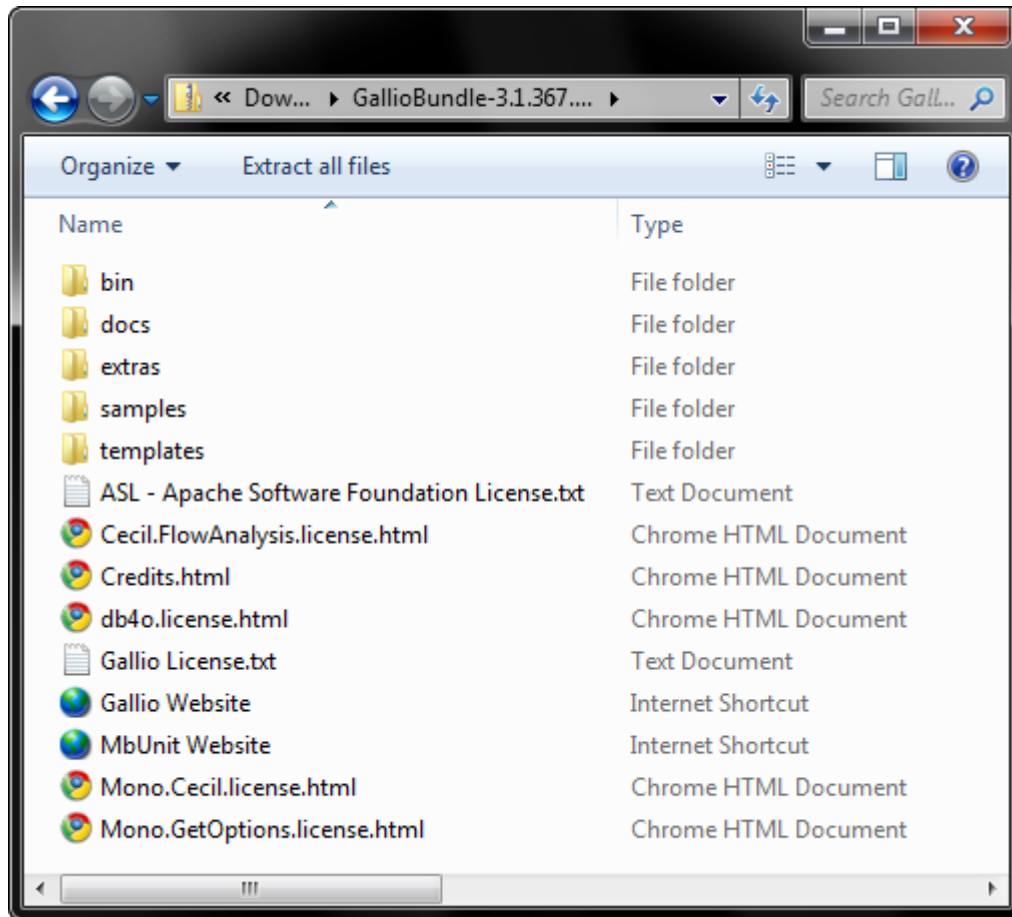


**Figure 1.11. The Gallio Start Menu**

## Using the ZIP Installer

If you need access to the Gallio samples, docs and DLLs but don't want it registered on your system – for example, you just need the DLLs for source code check in or a build server – you can download a zipped archive of all the Gallio files instead of the standard MSI installer. If you visit [gallio.org/downloads.aspx](http://gallio.org/downloads.aspx) [<http://gallio.org/downloads.aspx>] you'll find that each release has a zip file alternative for download as well.

The contents of the zip file, shown in Figure 1.12, “The Contents Of The Gallio Zip Download”, mirror those installed to %programfiles%\Gallio by the MSI installer.

**Figure 1.12. The Contents Of The Gallio Zip Download**

Besides the root folder which contains the license files for the various pieces of software used by Gallio and shortcuts for the Gallio website, you'll find four folders.

- The `bin` folder contains the main Gallio and MbUnit v3 DLLs along with the Echo and Icarus test runner applications.
- The various `bin\xyz` subfolders each contain one of the test framework adapters, runners or tools mentioned earlier.
- The `bin\Resources` subfolder also contains the various files needed for the HTML and XML reports that Gallio builds after a test run.
- The `docs` folder contains the API documentation for Gallio and MbUnit . It is available as a standalone `.chm` file, as files for integration into Visual Studio help (and as Help 3 files – forthcoming). The `h2reg.exe` program for integrating these files is in the `extras` folder.
- The `extras` folder contains the aforementioned `h2reg.exe` program, project templates for Visual Studio (2005/8/10) and files for incorporating Gallio into your CruiseControl.net build server.
- The `samples` folder contains a zip folder of an example C# solution demonstrating various features in MbUnit v3. Once you've extracted the solution, note that `MbUnit.Samples.sln` is a VS2005 file. `MbUnit.Samples.2008.sln` is for VS2008 users.

## Configuring Gallio

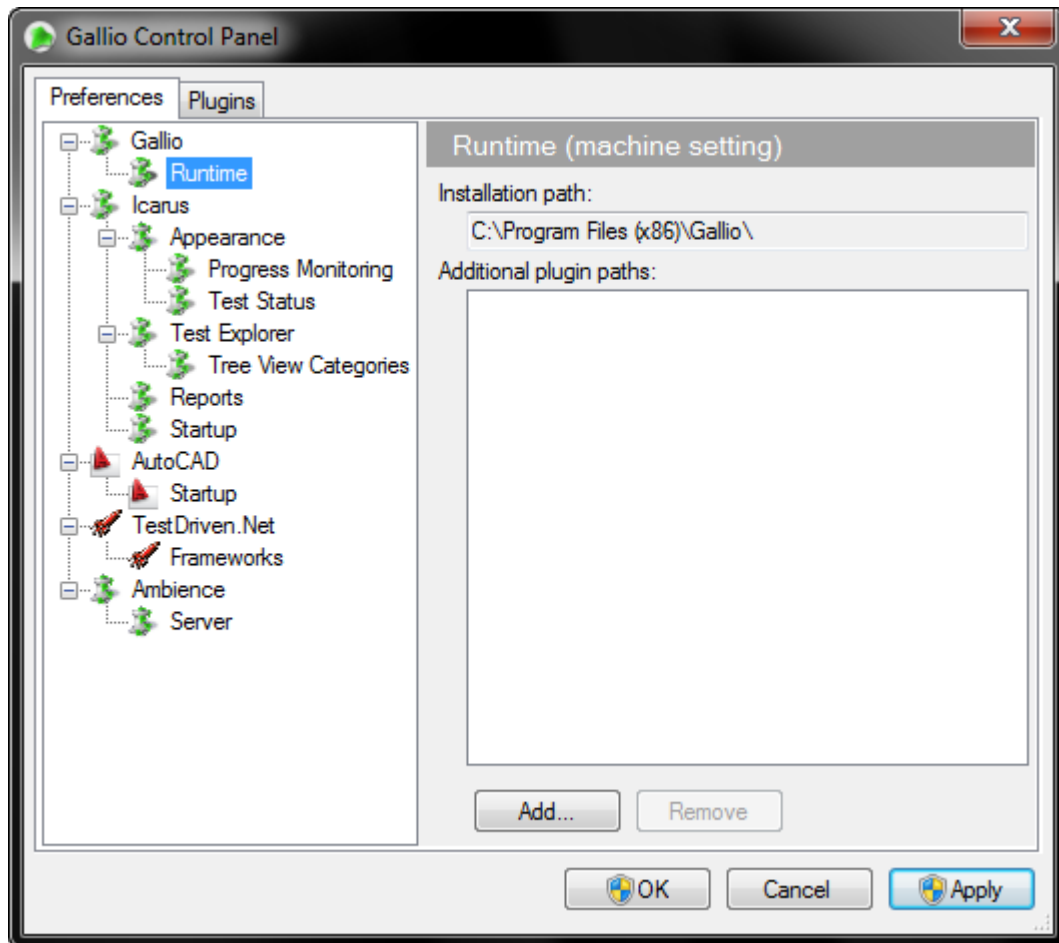
Once Gallio is installed, you can use the Gallio Control Panel to:

- Integrate any of your own plugins into Gallio
- Change various settings for Icarus, Gallio's GUI-based test runner
- Specify which test frameworks Gallio will run for TestDriven.NET and which it will let TestDriven.NET run itself.
- Configure the startup mode for Ambience, Gallio's database server.
- Check which versions of the various framework adapters, test runners and tools are currently working with your version of Gallio.

If you've not used Gallio before, try working with the default options before tweaking things with the Control Panel. You can find it under the Start Menu # All Programs # Gallio # Gallio Control Panel.

## Runtime Settings

The *Runtime* tab, shown in Figure 1.13, “Configuring Additional Gallio Plugins”, allows you to specify the path to any additional Gallio plugins that you may want to use beyond those in the standard installation.

**Figure 1.13. Configuring Additional Gallio Plugins**

To add a new plugin directory:

1. Click Add
2. Navigate to the directory holding the plug-in
3. Click OK.

The directory will be listed under “Additional plugin paths”.

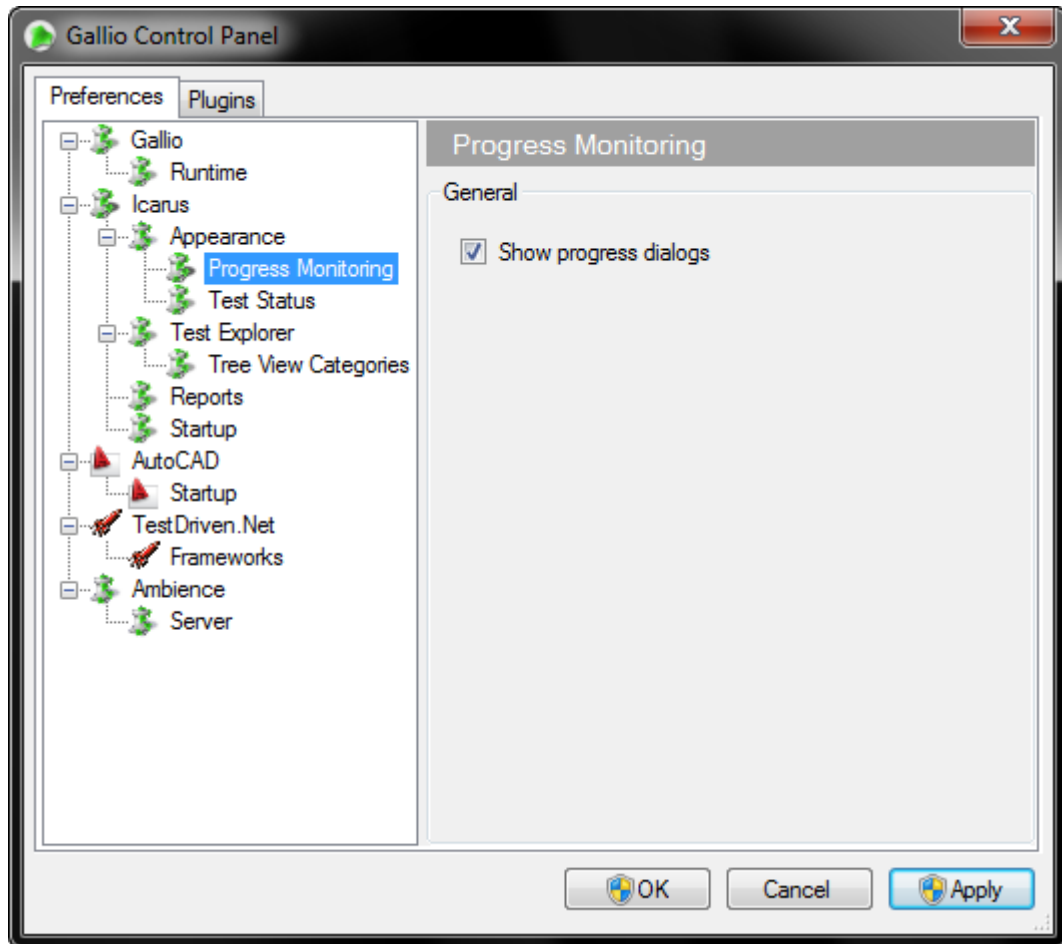
To remove the directory from the list, select it and click Remove.

## Icarus Settings

*Icarus* is Gallio's own GUI-based test runner. Gallio also provides a console-based test runner called Echo, but Echo's settings must always be specified when it is run. *Icarus* on the other hand stores some of its default settings on disk to be altered by the Gallio Control Panel. You can access the Control Panel from *Icarus* directly via Tools # Options.

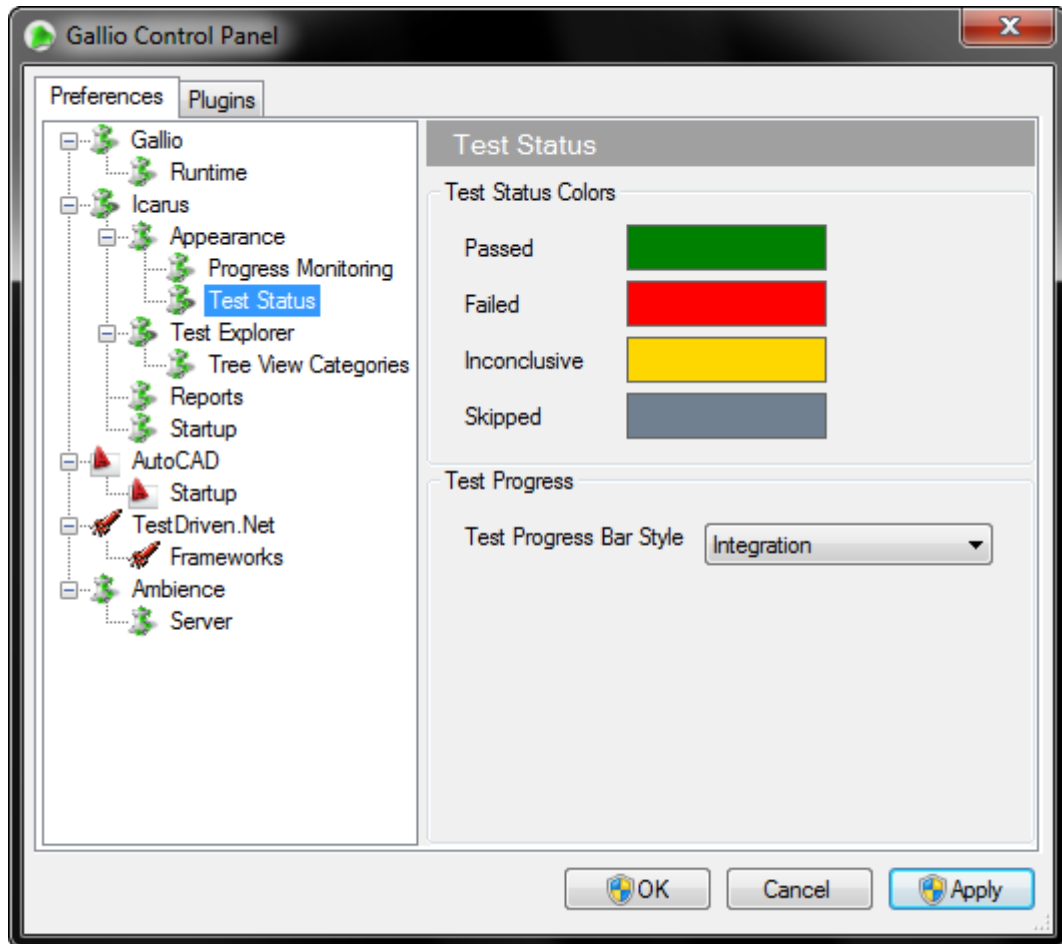
## Progress Monitoring

The *Progress Monitoring* tab, shown in Figure 1.14, “Configuring *Icarus* Progress Monitoring”, has one option which lets you set whether or not *Icarus* should show progress dialogs as tests are run.

**Figure 1.14. Configuring Icarus Progress Monitoring**

## Colours and Progress Bar Style

The *Appearance \ Test Status* dialog, shown in Figure 1.15, “Configuring Icarus Test Status Colours”, allows you to specify the colours associated with the four states of a test and the style of the test progress bar used in Icarus.

**Figure 1.15. Configuring Icarus Test Status Colours**

By common association, the colours green, red and yellow represent a test's success, failure or inconclusive ending. However, if you need to change them, for example because of colour-blindness, click on the colour block for each state and choose another colour more to your preference. The changes in colour will affect the progress bar, the test tree in Icarus (and the report).

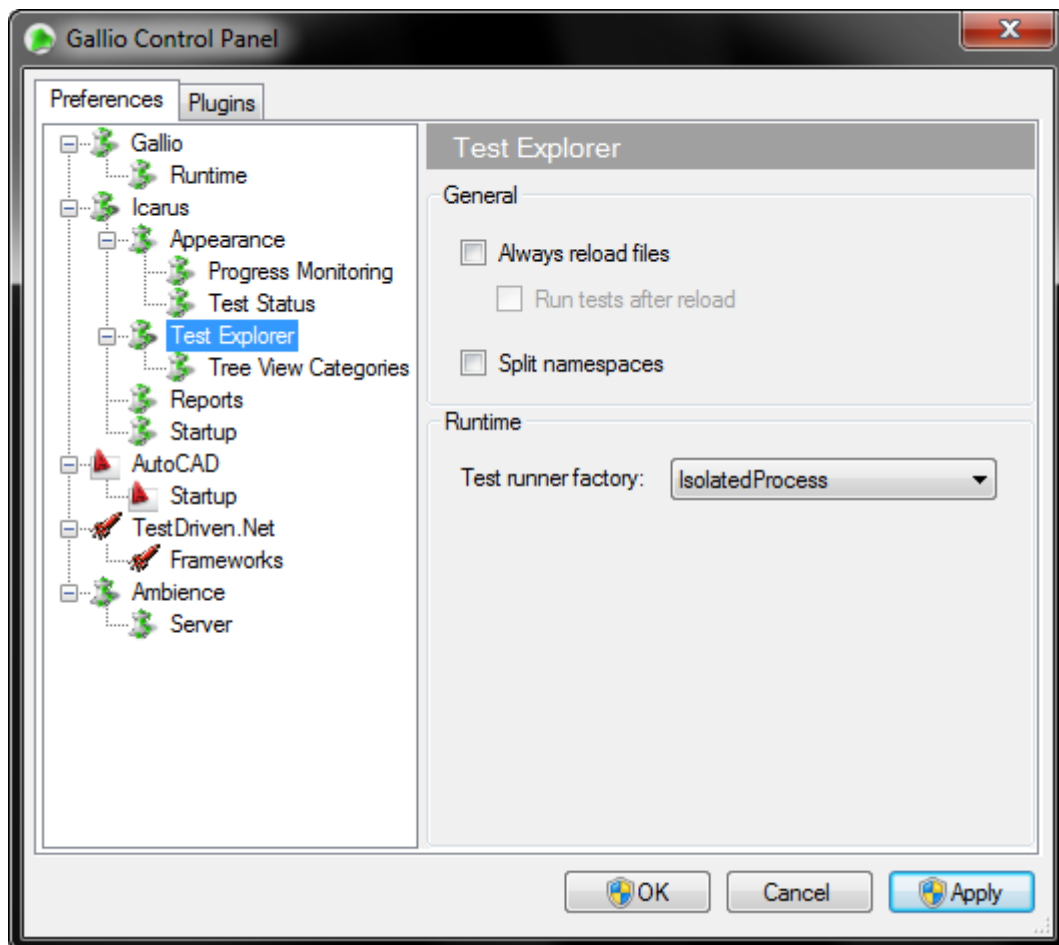
The test progress bar in Icarus has two modes:

- In “Integration” mode, the progress bar displays red, green and yellow sections in proportion to the number of failed, passed and inconclusive tests just run.
- In “Unit” mode, the progress bar displays fully green until a test fails at which point it turns totally red.

Select the one you want to use and click Apply to have Icarus use it.

## Test Explorer Settings

The *Test Explorer* dialog, shown in Figure 1.16, “Configuring Icarus Test Explorer Settings”, lets you set several options directly related to the execution of tests by Gallio.

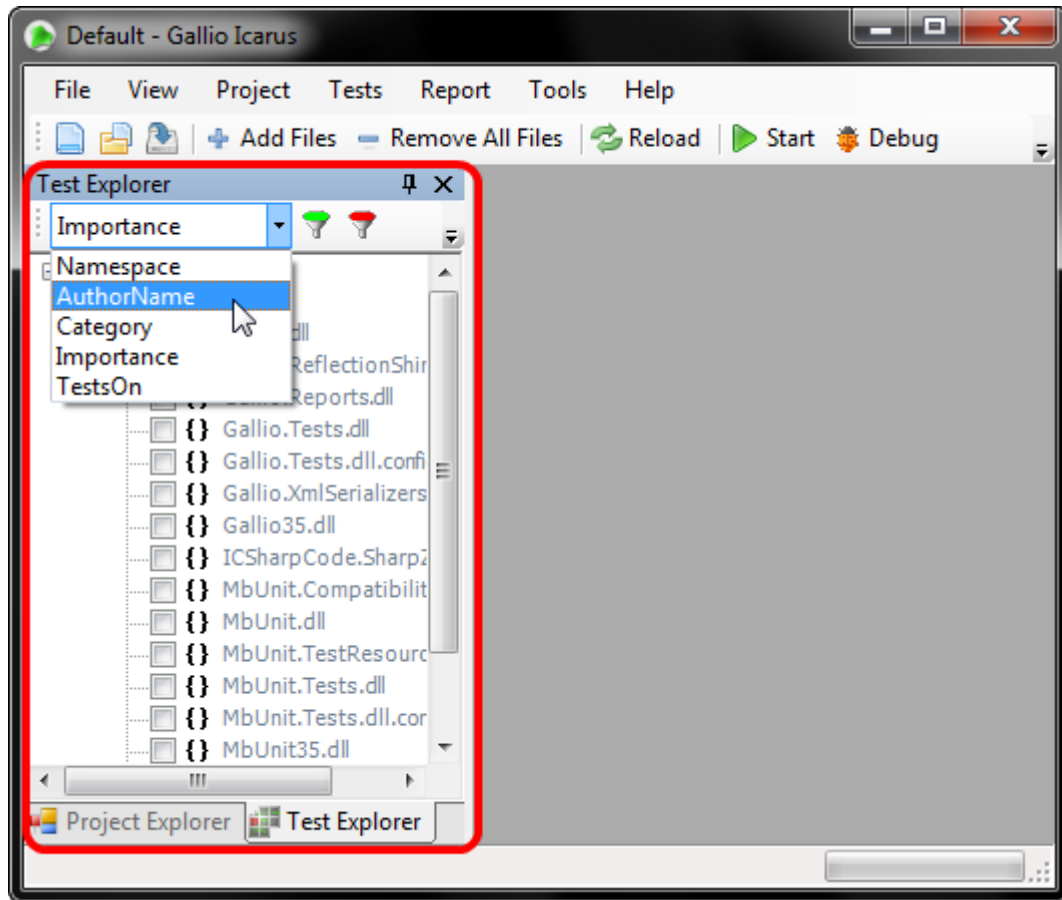
**Figure 1.16. Configuring Icarus Test Explorer Settings**

From top to bottom, the options allow you to

- Force Icarus to reloading your test DLLs and associated files when any of them has changed (e.g. when an assembly is rebuilt)
- Make the Icarus test tree split at the dots in the tree.
- Choose which application will generate and process the test runner factory to run the tests for your application. The default is for Gallio to use its own test runner factory within an Isolated Process.

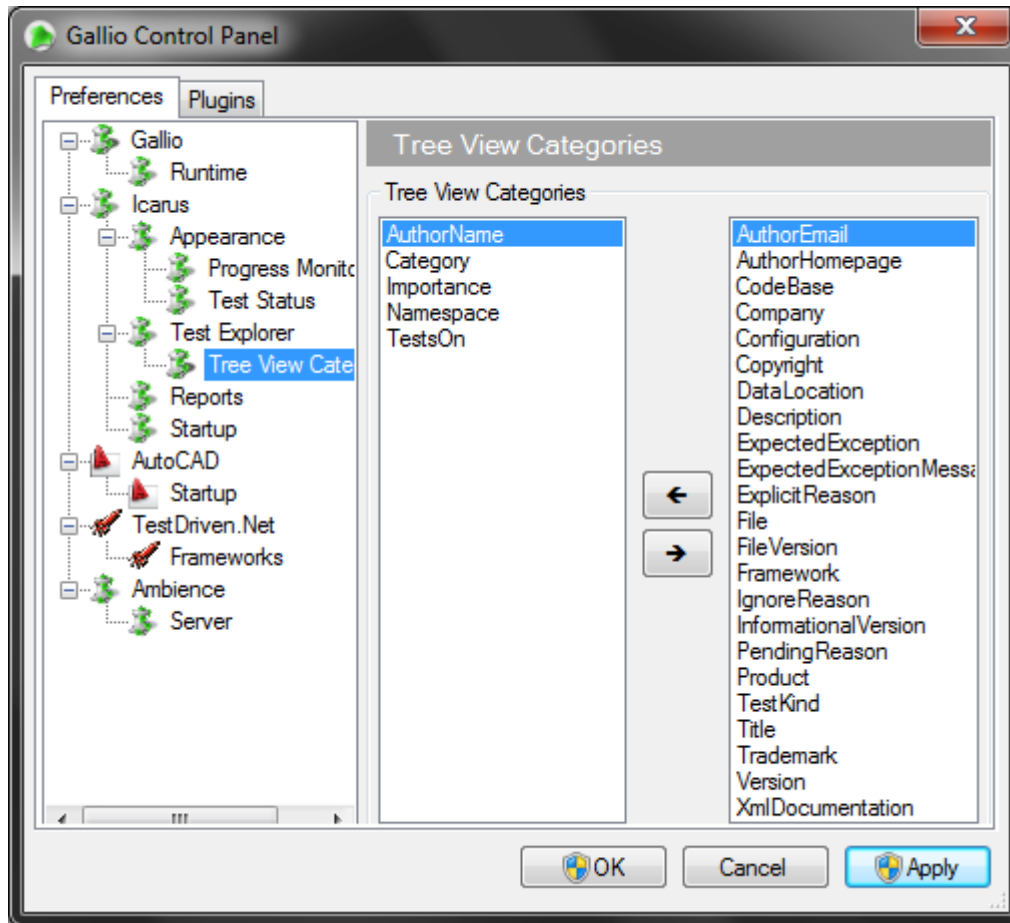
## Tree View Categories

The *Test Explorer* window, shown in Figure 1.17, “The Icarus Test Explorer Window”, allows you to view the tests currently loaded into Icarus grouped in a tree view based on one of a number of options you can choose from a drop-down.

**Figure 1.17. The Icarus Test Explorer Window**

The *Tree View Categories* dialog in the Gallio control panel, shown in Figure 1.18, “Configuring Test View Categories”, allows you to set the available categories by which the test explorer will attempt to order your tests.

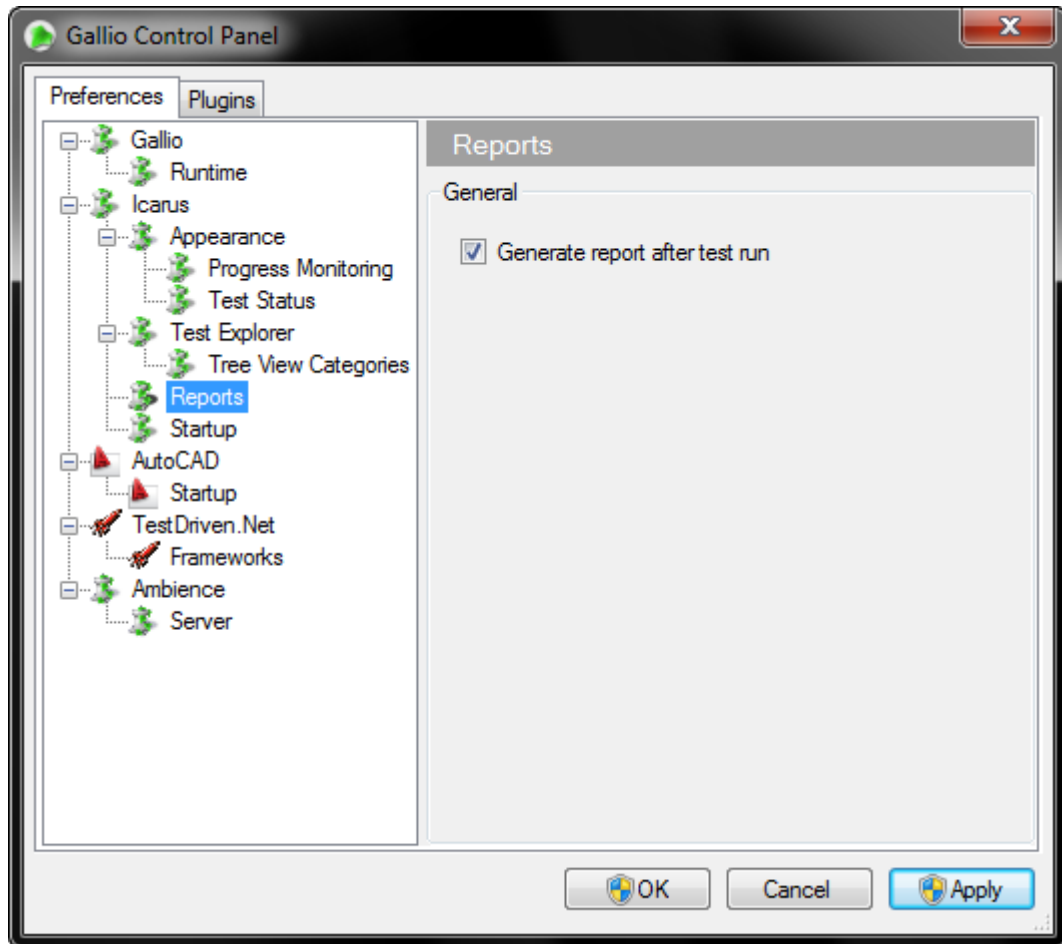


**Figure 1.18. Configuring Test View Categories**

The categories in the left hand column are those that will appear in the test explorer drop down list. We'll look in more detail at how each category works in Chapter 2, *My First Tests*.

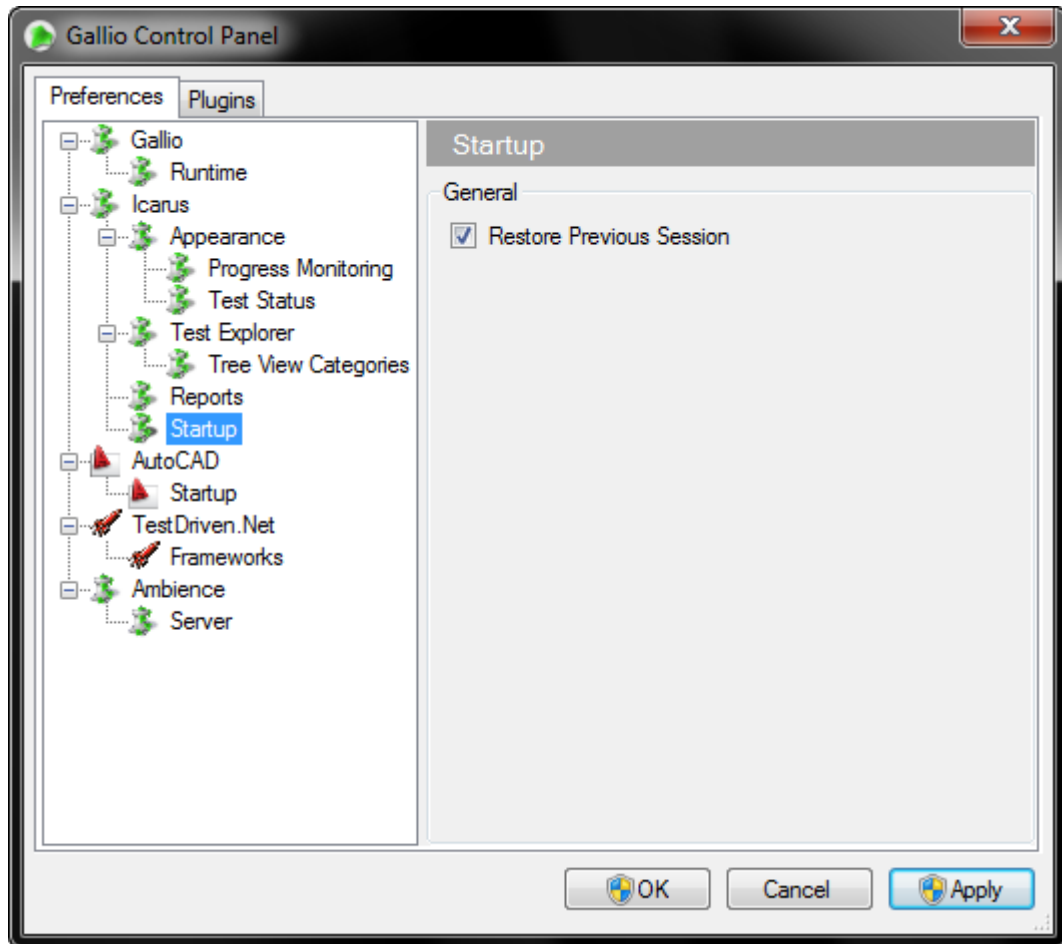
## Reports

The *Reports* tab, shown in Figure 1.19, “Configuring Test Reports”, has one option which lets you set whether or not Icarus should generate a report once a test run is complete.

**Figure 1.19. Configuring Test Reports**

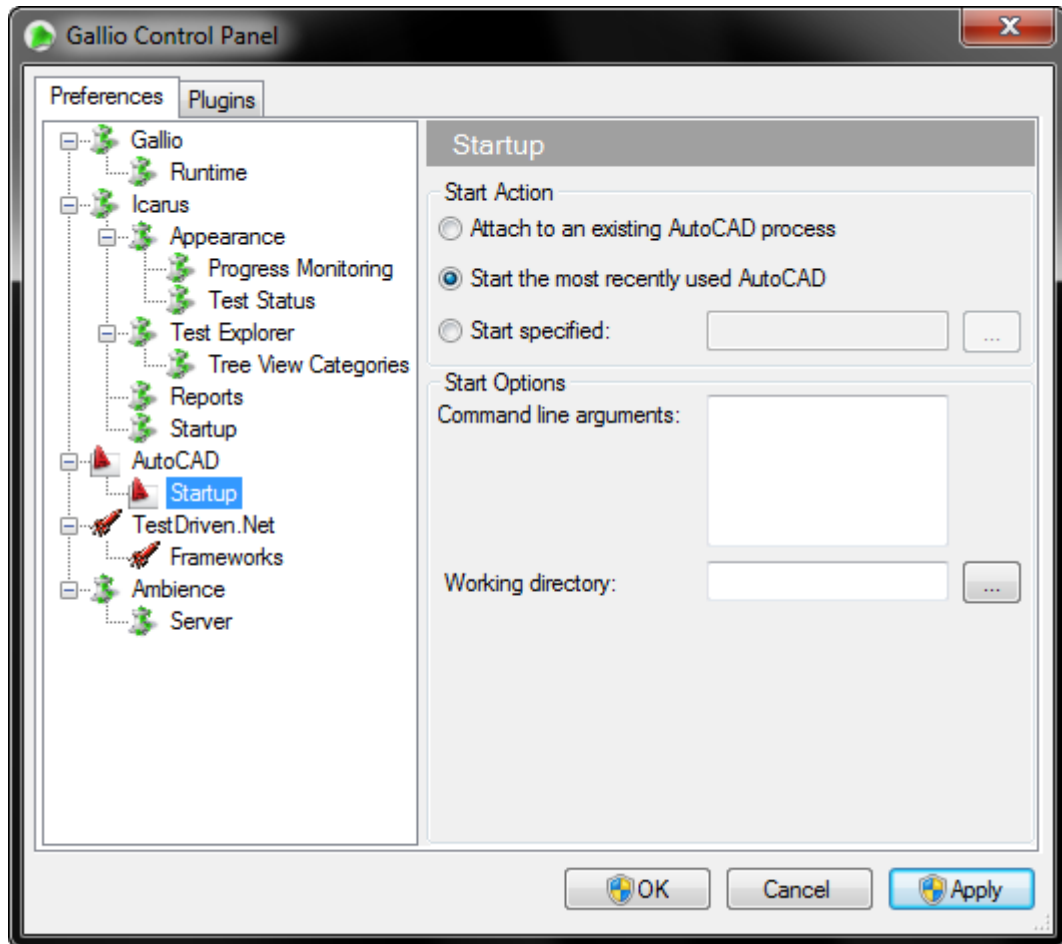
## Startup

The final dialog for Icarus, shown in Figure 1.20, “Configuring Icarus Startup Actions”, allows to set whether or not Icarus should automatically load back in the test assemblies it had loaded when it was last closed.

**Figure 1.20. Configuring Icarus Startup Actions**

## AutoCAD Settings

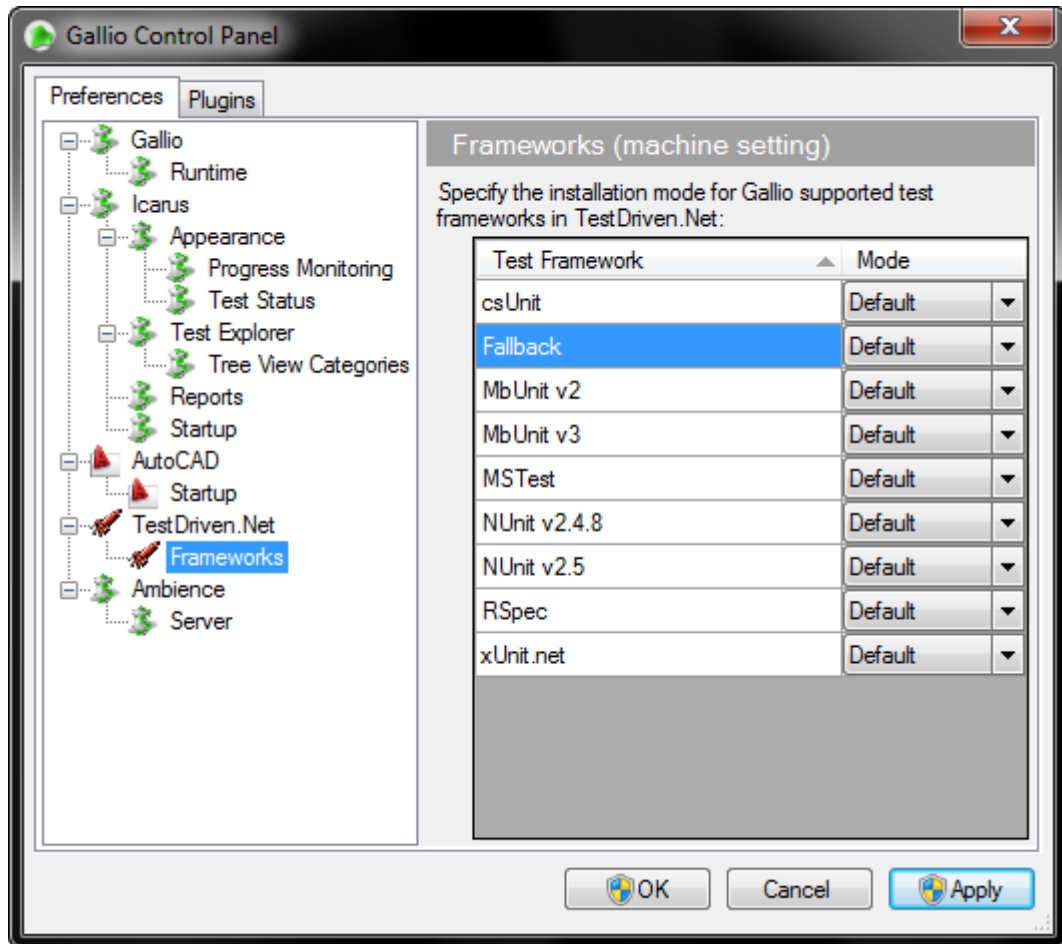
If you use Gallio to write tests that automate AutoCAD, you can use the *AutoCAD \ Startup* dialog, shown in Figure 1.21, “Configuring AutoCAD Integration”, to define how Gallio will link into AutoCAD when a test run starts.

**Figure 1.21. Configuring AutoCAD Integration**

## TestDriven.NET Settings

If you use the TestDriven.NET add-in for Visual Studio to run the tests in your solutions, adding Gallio now gives you a few more options for running tests from within Visual Studio. You can either run the tests directly within TestDriven.NET, or set TestDriven.NET to run the tests within Gallio.

Use the *TestDriven.Net \ Frameworks* dialog, shown in Figure 1.22, “Configuring TestDriven.NET Integration”, to set which tests are run by either TestDriven.NET or Gallio according to the test framework they are written with.

**Figure 1.22. Configuring TestDriven.NET Integration**

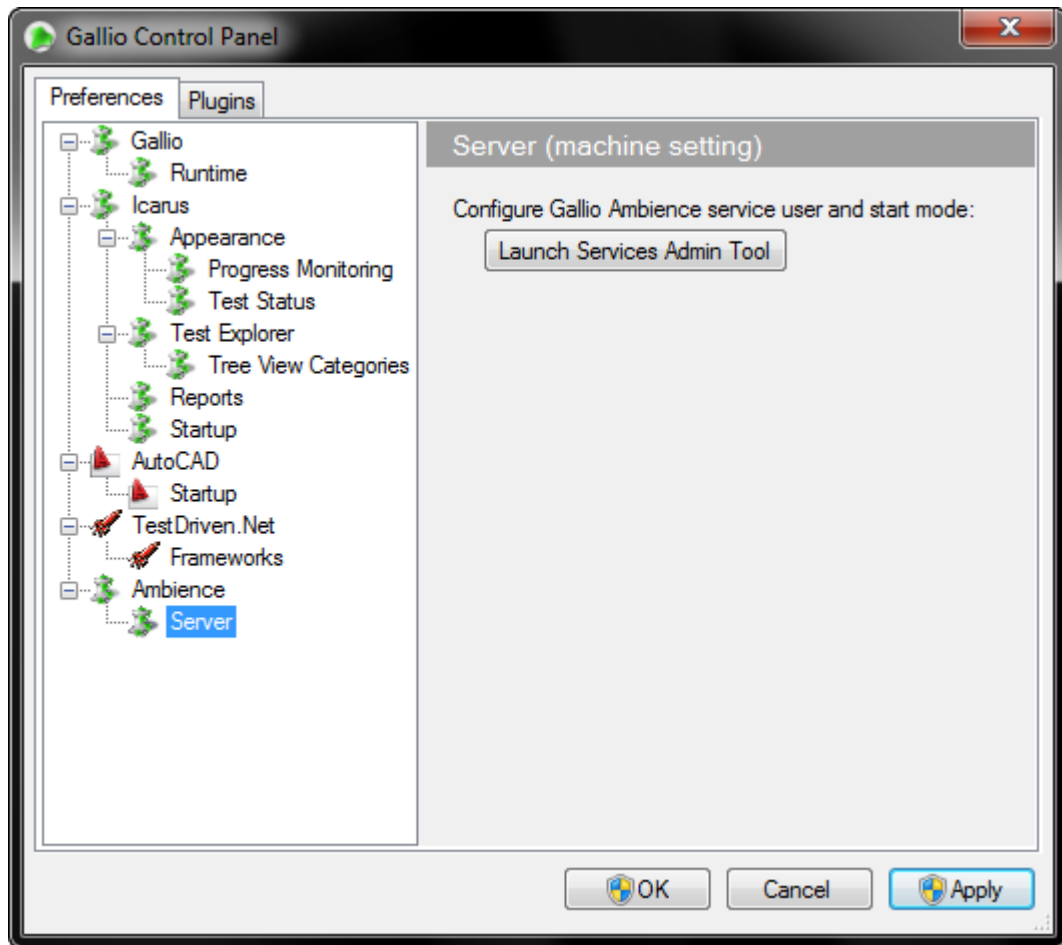
The Mode drop down list for each test framework has three options:

- Default: Use the default runner for the framework
- Preferred: Use Gallio to run the tests
- Disabled: Use TestDriven.NET to run the tests.

## Ambience Settings

*Ambience* is Gallio's own utility for storing persistent data across multiple test runs. It runs in the background as a Windows service when called upon. If you used the MSI installer to install Gallio and included *Ambience* in the option list, you'll have been asked whether the *Ambience* service should be started manually or automatically. The *Ambience \ Server* dialog, shown in Figure 1.23, "Accessing The *Ambience* Service", provides a shortcut to the Windows Services administration tool to let you change that setting.

**Figure 1.23. Accessing The Ambience Service**



## Conclusion

In this chapter, we've seen how to install and configure Gallio and MbUnit v3. In Chapter 2, *My First Tests* we'll introduce the idea of unit testing.

---

## Chapter 2. My First Tests

---

# Chapter 3. MbUnit Fundamentals



---

# Chapter 4. Testing Recipes 1

---

# Chapter 5. Testing Recipes 2

---

# Chapter 6. Domain Testing

---

# Chapter 7. Test Reports

---

# Chapter 8. Extending MbUnit

---

# Chapter 9. Automated Testing With Gallio

---

## **Part II. Developing MbUnit and Gallio**

---

---

## Table of Contents

|                                     |    |
|-------------------------------------|----|
| 10. Gallio Development Basics ..... | 38 |
| 11. Creating Gallio Plugins .....   | 39 |
| 12. Creating Test Frameworks .....  | 40 |



---

# Chapter 10. Gallio Development Basics

---

# Chapter 11. Creating Gallio Plugins

---

# Chapter 12. Creating Test Frameworks

---

## **Part III. Appendices**

---

---

## Table of Contents

|                                   |    |
|-----------------------------------|----|
| A. Migration Guide .....          | 43 |
| B. Creative Commons License ..... | 44 |
| Creative Commons Notice .....     | 48 |
| Glossary .....                    | 49 |

---

# Appendix A. Migration Guide

---

# Appendix B. Creative Commons License

The Gallio Book is licensed under a Creative Commons v3 Attribution-ShareAlike (United States) license. We present it here for your reference.

<http://creativecommons.org/licenses/by-sa/3.0/us/legalcode>

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

## 1. Definitions

- a. *Collective Work* means a work, such as a periodical issue, anthology or encyclopedia, in which the Work in its entirety in unmodified form, along with one or more other contributions, constituting separate and independent works in themselves, are assembled into a collective whole. A work that constitutes a Collective Work will not be considered a Derivative Work (as defined below) for the purposes of this License.
- b. *Creative Commons Compatible License* means a license that is listed at <http://creativecommons.org/compatiblelicenses> that has been approved by Creative Commons as being essentially equivalent to this License, including, at a minimum, because that license: (i) contains terms that have the same purpose, meaning and effect as the License Elements of this License; and, (ii) explicitly permits the relicensing of derivatives of works made available under that license under this License or either a Creative Commons unported license or a Creative Commons jurisdiction license with the same License Elements as this License.
- c. *Derivative Work* means a work based upon the Work or upon the Work and other pre-existing works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which the Work may be recast, transformed, or adapted, except that a work that constitutes a Collective Work will not be considered a Derivative Work for the purpose of this License. For the avoidance of doubt, where the Work is a musical composition or sound recording, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered a Derivative Work for the purpose of this License.
- d. *License Elements* means the following high-level license attributes as selected by Licensor and indicated in the title of this License: Attribution, ShareAlike.
- e. *Licensor* means the individual, individuals, entity or entities that offers the Work under the terms of this License.
- f. *Original Author* means the individual, individuals, entity or entities who created the Work.
- g. *Work* means the copyrightable work of authorship offered under the terms of this License.

- h. *You* means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

## 2. *Fair Use Rights*

Nothing in this license is intended to reduce, limit, or restrict any rights arising from fair use, first sale or other limitations on the exclusive rights of the copyright owner under copyright law or other applicable laws.

## 3. *License Grant*

Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

- a. to reproduce the Work, to incorporate the Work into one or more Collective Works, and to reproduce the Work as incorporated in the Collective Works;
- b. to create and reproduce Derivative Works provided that any such Derivative Work, including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked "The original work was translated from English to Spanish," or a modification could indicate "The original work has been modified.";
- c. to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission the Work including as incorporated in Collective Works;
- d. to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission Derivative Works.
- e. For the avoidance of doubt, where the Work is a musical composition:
  - i. *Performance Royalties Under Blanket Licenses*. Licensor waives the exclusive right to collect, whether individually or, in the event that Licensor is a member of a performance rights society (e.g. ASCAP, BMI, SESAC), via that society, royalties for the public performance or public digital performance (e.g. webcast) of the Work.
  - ii. *Mechanical Rights and Statutory Royalties*. Licensor waives the exclusive right to collect, whether individually or via a music rights agency or designated agent (e.g. Harry Fox Agency), royalties for any phonorecord You create from the Work ("cover version") and distribute, subject to the compulsory license created by 17 USC Section 115 of the US Copyright Act (or the equivalent in other jurisdictions).
- f. *Webcasting Rights and Statutory Royalties*. For the avoidance of doubt, where the Work is a sound recording, Licensor waives the exclusive right to collect, whether individually or via a performance-rights society (e.g. SoundExchange), royalties for the public digital performance (e.g. webcast) of the Work, subject to the compulsory license created by 17 USC Section 114 of the US Copyright Act (or the equivalent in other jurisdictions).

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. All rights not expressly granted by Licensor are hereby reserved.

## 4. *Restrictions*



The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

- a. You may distribute, publicly display, publicly perform, or publicly digitally perform the Work only under the terms of this License, and You must include a copy of, or the Uniform Resource Identifier for, this License with every copy or phonorecord of the Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of a recipient of the Work to exercise of the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties. When You distribute, publicly display, publicly perform, or publicly digitally perform the Work, You may not impose any technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise of the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Work itself to be made subject to the terms of this License. If You create a Collective Work, upon notice from any Licensor You must, to the extent practicable, remove from the Collective Work any credit as required by Section 4(c), as requested. If You create a Derivative Work, upon notice from any Licensor You must, to the extent practicable, remove from the Derivative Work any credit as required by Section 4(c), as requested.
- b. You may distribute, publicly display, publicly perform, or publicly digitally perform a Derivative Work only under: (i) the terms of this License; (ii) a later version of this License with the same License Elements as this License; (iii) either the Creative Commons (Unported) license or a Creative Commons jurisdiction license (either this or a later license version) that contains the same License Elements as this License (e.g. Attribution-ShareAlike 3.0 (Unported)); (iv) a Creative Commons Compatible License. If you license the Derivative Work under one of the licenses mentioned in (iv), you must comply with the terms of that license. If you license the Derivative Work under the terms of any of the licenses mentioned in (i), (ii) or (iii) (the "Applicable License"), you must comply with the terms of the Applicable License generally and with the following provisions: (I) You must include a copy of, or the Uniform Resource Identifier for, the Applicable License with every copy or phonorecord of each Derivative Work You distribute, publicly display, publicly perform, or publicly digitally perform; (II) You may not offer or impose any terms on the Derivative Works that restrict the terms of the Applicable License or the ability of a recipient of the Work to exercise the rights granted to that recipient under the terms of the Applicable License; (III) You must keep intact all notices that refer to the Applicable License and to the disclaimer of warranties; and, (IV) when You distribute, publicly display, publicly perform, or publicly digitally perform the Work, You may not impose any technological measures on the Derivative Work that restrict the ability of a recipient of the Derivative Work from You to exercise the rights granted to that recipient under the terms of the Applicable License. This Section 4(b) applies to the Derivative Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Derivative Work itself to be made subject to the terms of the Applicable License.
- c. If You distribute, publicly display, publicly perform, or publicly digitally perform the Work (as defined in Section 1 above) or any Derivative Works (as defined in Section 1 above) or Collective Works (as defined in Section 1 above), You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or (ii) if the Original Author and/or Licensor designate another party or parties (e.g. a sponsor institute, publishing entity, journal) for attribution ("Attribution Parties") in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; the title of the Work if supplied; to the extent reasonably practicable, the Uniform Resource Identifier, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and, consistent with Section 3(b) in the case

of a Derivative Work, a credit identifying the use of the Work in the Derivative Work (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). The credit required by this Section 4(c) may be implemented in any reasonable manner; provided, however, that in the case of a Derivative Work or Collective Work, at a minimum such credit will appear, if a credit for all contributing authors of the Derivative Work or Collective Work appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.

5. *Representations, Warranties and Disclaimer*

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND ONLY TO THE EXTENT OF ANY RIGHTS HELD IN THE LICENSED WORK BY THE LICENSOR. THE LICENSOR MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MARKETABILITY, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. *Limitation on Liability*

EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. *Termination*

- a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Derivative Works or Collective Works from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.
- b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. *Miscellaneous*

- a. Each time You distribute or publicly digitally perform the Work (as defined in Section 1 above) or a Collective Work (as defined in Section 1 above), the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

- b. Each time You distribute or publicly digitally perform a Derivative Work, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.
- c. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.
- d. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.
- e. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

## Creative Commons Notice

Creative Commons is not a party to this License, and makes no warranty whatsoever in connection with the Work. Creative Commons will not be liable to You or any party on any legal theory for any damages whatsoever, including without limitation any general, special, incidental or consequential damages arising in connection to this license. Notwithstanding the foregoing two (2) sentences, if Creative Commons has expressly identified itself as the Licensor hereunder, it shall have all rights and obligations of Licensor.

Except for the limited purpose of indicating to the public that the Work is licensed under the CCPL, Creative Commons does not authorize the use by either party of the trademark "Creative Commons" or any related trademark or logo of Creative Commons without the prior written consent of Creative Commons. Any permitted use will be in compliance with Creative Commons' then-current trademark usage guidelines, as may be published on its website or otherwise made available upon request from time to time. For the avoidance of doubt, this trademark restriction does not form part of this License.

Creative Commons may be contacted at <http://creativecommons.org/>.

---

# Glossary

## A

|          |   |
|----------|---|
| Alt.Net  | A community of .NET developers who embrace tools, techniques and knowledge from all development communities, and not just Microsoft's own tooling and guidance. |
| Ambience | Gallio's own utility for storing persistent data across multiple test runs.   |

## C

|                   |   |
|-------------------|---|
| CruiseControl.NET | A free and open source build management and continuous integration server written in C#.    |
| csUnit            | A unit-testing framework written in C++. Tests written against csUnit can be run by Gallio. |

## E

|                     |  |
|---------------------|--|
| Echo                | The Gallio project command-line based test runner  |
| Extreme Programming | A type of agile software development,advocating frequent "releases" in short development cycles. The idea is to improve developer productivity and create points in the process where new customer requirements can be included into the software specification. |

## G

|        |  |
|--------|--|
| Gallio | The framework-neutral Automation Platform half of the Gallio project, written in C#. |
| gUnit  | The original name for MbUnit.  |

## I

|                |  |
|----------------|--|
| Icarus         | The Gallio project's GUI-based test runner   |
| IronPython.NET | An implementation of the Python programming language running under .NET and Silverlight. |
| IronRuby.NET   | An implementation of the Ruby programming language running under .NET and Silverlight.   |

## J

|       |   |
|-------|---|
| jUnit | A unit-testing framework written in Java. |
|-------|---|

## M

|         |   |
|---------|---|
| MbUnit  | The unit-testing framework half of the Gallio project, written in C#. |
| MSBuild | A build platform with an XML syntax built into Visual Studio.         |
| MSTest  | The unit-testing framework built into Visual Studio.                  |

## T

|          |  |
|----------|--|
| TeamCity | TeamCity is a Java-based build management and continuous integration server. |
|----------|--|

## N

|         |  |
|---------|--|
| nAnt    | A free and open source build platform with an XML syntax.  |
| nBehave | One of the first behavior-driven development frameworks written in C#. Tests written against nBehave can be run by Gallio. |
| nCover  | A code coverage tool measuring how much of your code is run by the tests you have written.                                 |
| nUnit   | One of the first unit-testing frameworks written in C#. Tests written against nUnit can be run by Gallio.                  |

## O

|           |  |
|-----------|--|
| ObjectARX | ObjectARX (AutoCAD Runtime Extension) is an API for customizing and extending AutoCAD. |
|-----------|--|

## R

|           |   |
|-----------|---|
| Resharper | A refactoring plug-in for Visual Studio. It has a test runner which can use Gallio to run tests written against any framework that Gallio supports. |
| RSpec     | The original Behaviour Driven Development framework for Ruby  |

## S

|                                       |   |
|---------------------------------------|---|
| Software Development Engineer in Test | A development role at Microsoft that sees a product through two phases of the software lifecycle - Development and Testing. |
|---------------------------------------|---|

## T

|                         |  |
|-------------------------|--|
| TeamCity                | A build management and continuous integration server. It has a test runner which can use Gallio to run tests written against any framework that Gallio supports. |
| Test-Driven Development | A style of software development with four simple steps.<br><ol style="list-style-type: none"><li>1. Create a test case for code that fails.</li></ol>            |

2. Write code so the test case and all other test cases pass.
3. Refactor all the code so there are no code smells and the test cases still pass
4. Repeat

TestDriven.NET      TestDriven.NET is a plug-in for Visual Studio that allows you to run or debug their (non-MSTest) tests from within Visual Studio

Test Runner      A test runner is a program or plug-in for other programs that provide different ways to run tests through Gallio

TypeMock      A mock object framework written in C#.

## X

xUnit      One of the newer unit-testing frameworks written in C#, written from scratch by some members of the NUnit team based on their experience with that. Tests written against xUnit can be run by Gallio.