Serenity – The Complete Guide

Documentation for Serenity – Version 0.4.2.0

Contents

[Introduction 3](#_Toc175399215)

[What’s New in 0.4.2.0? 3](#_Toc175399216)

[Overview 4](#_Toc175399217)

[Modules 4](#_Toc175399218)

[Pages 4](#_Toc175399219)

[Installing and Configuring Serenity 5](#_Toc175399220)

[Downloading 5](#_Toc175399221)

[Installing 5](#_Toc175399222)

[Testing Your New Installation 5](#_Toc175399223)

[Adding Domains 6](#_Toc175399224)

[List of Options 6](#_Toc175399225)

[Frequently Asked Questions 7](#_Toc175399226)

[General 7](#_Toc175399227)

[User Questions 7](#_Toc175399228)

[Development Questions 8](#_Toc175399229)

[Design/Architecture 8](#_Toc175399230)

[History and Background 8](#_Toc175399231)

[Further Adventures 9](#_Toc175399232)

[Change Log 10](#_Toc175399233)

[0.3.2.0 10](#_Toc175399234)

[0.3.1.0 10](#_Toc175399235)

[0.3.0.0 10](#_Toc175399236)

[0.27.0.0 11](#_Toc175399237)

[Credits 13](#_Toc175399238)

# Introduction

Welcome to the documentation for the Serenity Web Server! In this document you will find pertinent information about installing, running, administering, and developing for this exciting new web platform.

You should read through this guide, including the FAQ section located at the end, before posting your question to the forums or IRC channel.

## What’s New in 0.4.2.0?

This release features some regressions introduced with the previous release as well as an improvement of the Theme system. Additionally, there are some new features:

* Resource can now be served from modules.

# Overview

Serenity is an open source web server, written entirely in C# and running on the .NET Framework 2.0. The design of Serenity has been carefully considered, and consistent with our motto “speed through simplicity”, we have tried to make the software with as few moving parts as possible, as some would say. There are still some concepts that should be understood before you get involved with Serenity.

## Modules

A module is essentially a web application. Modules are a collection of related Pages. Modules can be loaded into the server at start-up time, or later while the server is running, and they can provide services to users, such as powering a forum, blog, etc.

## Pages

A page in Serenity is a smaller part of a Module that performs a specific task. The job of a page is to generate dynamic content. Pages can generate either textual content (like HTML, XML, etc), or binary content (like images, sounds, videos, etc.). Each Page generates a specific set of content, or handles a specific kind of action from the user.

# Installing and Configuring Serenity

The deployment process for Serenity has been smoothed as much as possible, but as with any software there are still some things you need to know before you attempt to set up Serenity. This section will help you through the steps of acquiring a suitable release of Serenity and installing it.

## Downloading

The best available version of Serenity is always located on the Serenity homepage at:

* <http://serenityproject.net/>

Find the section of the site called “Releases” and navigate to it. There may be several available versions of Serenity present, and it is important to read the release notes of each one before choosing a version. Usually when more than one version is available, it means that the newer version(s) are not as stable as the older one(s).

It is important that you determine the environment that you will be deploying Serenity into before downloading a release. If you intend to run a production site (users will be visiting the server and relying on its continued operation on a daily basis), you should choose the most stable version available. However, if you simply wish to experiment with the software, you should pick the latest version.

Download the release file marked as “Installer”, for example, “Serenity-0.4.2.0-Installer.exe”.

## Installing

Once you’ve downloaded the release you want, simply double click the downloaded installer. The wizard will guide you through the installation steps.

## Testing Your New Installation

To verify that your new installation of Serenity is working properly, go to the Start Menu -> All Programs -> Serenity Project -> Serenity and then click on the Serenity shortcut.

A command line window should pop up and display information about the server as it loads.

Note: If the server fails to start up properly, there may be additional steps required for your operating system, or there may be problems with the software that are causing it to fail to start. In either case, you can report your problem to the Serenity team and we will do our best to provide a solution.

Once the server has finished loading, you can verify it is working correctly by opening a web browser and typing “<http://localhost/>” in the address bar and pressing enter. The default page of a new Serenity installation should then come up.

## Server Configuration

Some configuration options affect the entire server, such as which modules to load, how the server should communicate with clients, etc.

A sample server configuration file looks like this:

[General]

LogToConsole = true

LogToFile = true

OnDemandLoading = false

[Modules]

Serenity = Serenity.dll

[Network]

Ports = 80 8080 8081 8082 8083 8084 8085 8086 8087 8088 8089

BlockingIO = true

### [General] Options

#### LogToConsole

This option determines if log entries will be written (displayed) in the console window, if available.

#### LogToFile

This option determines if log entries will be written to the log file.

#### OnDemandLoading

Determines if resources or other data can be loaded the first time they are requested, or if they should all be loaded when the server starts up.

### [Modules] Options

Each line in the Modules section is one module that will be loaded into the server at startup, and takes the following format:

<name of module> = <path to module assembly file>

## Adding and Configuring Domains

Another major feature of Serenity is its native support for configuring options on a per-domain basis. You can define options that affect a specific domain and its sub-domains.

To configure settings for a specific domain, create an INI file with the name of the desired domain. Note that the domain must be in reverse order. For example, to configure settings for “www.domain.com”, you would need to reverse the parts of the domain name, resulting in “com.domain.www”. This file must be placed in the “Domains” folder within the Serenity installation directory, usually “C:\Program Files\Serenity Project\Serenity\”.

A sample Domain Settings file would look something like this:

[DomainSettings]

DefaultResourceClass = static

DefaultResourceName = default.html

OmitResourceClass = false

DocumentRoot = ./Domains/Common/

ThemeName = system

You do not need to define every option, only the ones that are different from parent Domain Settings. For example, “serenityproject.net” is the parent domain of “www.serenityproject.net”, and “net” is the parent of “serenityproject.net”. Any options defined in “net” will cascade into all child domains, but will be overridden if redefined in a child domain’s settings file. This allows a great deal of flexibility and ease of use.

### List of DomainSettings Options

#### DefaultResourceClass

This is the name of the ResourceClass that will be used if the URL does not contain one.

#### DefaultResourceName

This is the name of the resource that will be used if the URL does not contain one.

#### OmitResourceClass

Allows requests to the domain to be sent without including a ResourceClass, and instead uses the DefaultResourceClass for all requests.

#### DocumentRoot

Specifies the local file system location where static files will be served from.

#### ThemeName

Specifies the name of the Theme that will be used for resources requested from the domain.

# Development

To develop modules for Serenity you should have solid knowledge of the following:

* C# or VB.NET.
* HTML
* Web Design/Development.

## Creating Dynamic Pages

For starters, the most basic and common task, creating dynamic web pages will be explained.

### Example One: Basic Dynamic Page

We’ll start with a code sample:

using System;

using System.Collections.Generic;

using System.Text;

using Serenity;

using Serenity.Web;

using Serenity.Web.Drivers;

namespace Serenity.Pages

{

public sealed class DefaultPage : ContentPage

{

public override ContentPage CreateInstance()

{

return new DefaultPage();

}

public override void OnRequest(CommonContext context)

{

context.Response.WriteLine("This Page has not yet been implemented!");

}

public override string Name

{

get

{

return "Default";

}

}

}

}

Dynamic pages must inherit from the Serenity.ContentPage class, which provides the necessary set of methods and properties to make web development possible. Most of the work will happen in the OnRequest method, which is an abstract method of Serenity.ContentPage.

In the above example, the only thing we do when a request is received is write back to the client, saying “This Page has not yet been implemented!”. This is the lowest-level form of creating output; writing text directly to the client. You build your own higher-level framework on top of this low-level base, or use someone else’s framework, or just write out your content. Serenity gives you flexibility in this area.

The CreateInstance method is simply a utility method that the current version of Serenity uses internally. In order for your Page to work properly, you need to override CreateInstance as shown, and return a new instance of the type of your page. For example, if your page is “ForumPage”, you would need to override CreateInstance like so:

public override ContentPage CreateInstance()

{

return new ForumPage();

}

The Name property must simply return the name of the page, so it can be accessed properly. It should be read-only (only “get”, no “set”), and return a constant string which is the name of the page, as is it accessed from the web.

### Example Two: Fibbonaci Sequence Page

Now we’ll make a simple page that outputs the fibbonaci sequence.

# Frequently Asked Questions

Note: Please read the rest of the documentation,.

## General

#### What is Serenity?

Serenity is a web server. If you are unfamiliar with this term, you should read the Serenity Overview document, which contains more information about how Serenity works and how web servers work in general.

#### Is Serenity really free?

Yes. Serenity is developed as Open-Source Software. This means that the work which is put into Serenity is done so for the benefit of everyone. Being open source also means that you are free to download and share Serenity as much as you like, and even make changes to the software if you want to.

#### Why should I use Serenity?

If you are a developer and plan on creating dynamic websites, Serenity provides a great deal of valuable features that make your task easier. If you are an end user, Serenity is extremely easy to configure and maintain.

#### How can I contribute to Serenity?

If you wish to help the Serenity Project by contributing in any way, please feel free to drop in the Serenity IRC channel or make a post on the project forums. The Serenity Team always appreciates donations of any sort (code, money, time, etc.)

#### what if I find a bug or want to suggest a feature?

You can use the Issue Tracker located on the project’s homepage to submit both of these types of items.

## User Questions

#### Why did Serenity stop working?

In the rare case where the software suddenly stops working, the cause is usually an environmental effect, such as another application or user somehow deleting files that Serenity requires in order to run properly.

Note: The more information about a bug or issue that you provide, the more effectively the Serenity team can fix it.

If this is not the case, the cause could also be a bug in the software. Be sure to check the Issue Tracker on the project website to see if this bug or problem has been reported yet. If it has, you can add your feedback and information. Otherwise, you can report it as a new work item.

Additionally, you may want to check the Releases page on the project website for a version that is newer than the one you are using.

## Development Questions

This section covers questions most frequently asked by developers.

#### Where can I find help with writing a Module?

Join the Serenity IRC channel, #serenityproject on irc.freenode.net or make a post on the project’s forums. Be sure to search the forums and read the Developer documentation before asking your question, though.

#### What if my Module suddenly breaks?

Sorry. The current phase of development means that breaking changes sometimes have to be implemented in order to improve overall stability, performance, or security, or to implement a feature. We will try and keep these breaking changes to a minimum, however.

## Design/Architecture

This section covers common technical questions about the development or operation of Serenity.

#### What language is Serenity written with?

All the code for Serenity is written using the C# language.

#### What other software is required to run Serenity?

Note: The windows installer for Serenity can download and install the .NET Framework 2.0 if it does not detect it on your computer.

The only thing you need to run Serenity is the Microsoft .NET Framework 2.0 (or compatible).

#### What platforms are supported?

Currently, support is only offered for Windows XP SP2, Windows Server 2003, and Windows Vista. Both the 32-bit (x86) and the 64-bit (x64) versions of these operating systems should work. Serenity may also run on Linux via Mono, but this is not guaranteed and unexpected behavior may result.

## History and Background

#### Why Serenity?

Serenity was developed primarily due to the perceived weaknesses within Apache and IIS, especially in relation to management and administration. We saw a need for a web server that was both powerful and easy to use, and didn’t require a lot of tinkering to maintain a secure and stable production web site.

#### How old is Serenity?

Development on Serenity began in November of 2006. Active development has continued since then.

# Further Adventures

Well, now you’re at the end of the guide. If you still have questions that are unanswered, you can either post them to the Serenity discussion board, or hop into the Serenity IRC channel at #serenityproject on irc.freenode.org. We (the Serenity Team), or another user, will do our best to answer your question or address your issue.

Also, this guide will be updated (hopefully) with every new release of Serenity. So check back frequently to see if a new version is available.

# Change Log

## 0.3.2.0

This release fixes a fatal flaw with GET variables and makes some other minor tweaks and improvements.

#### Additions

* Setup application now adds a shortcut to the application data directory for easy administration in a shell session.

#### Fixes

* GET variables with no value no longer cause the server to crash.

## 0.3.1.0

This is mostly a stability release; more exciting new features will be introduced next release (hopefully).

#### Additions

* Included a public key keyfile allowing anyone to compile Serenity for debugging or testing purposes by delay-signing the assembly with the provided public key.
* Started implementing basic localization support.
* Installation program now executes ngen to pre-compile native images of Serenity and its dependencies. This should improve performance a little.

#### Changes

* Changed the way Serenity is installed and how it starts up so that system resources, like the icons that are distributed with the software, are copied to the current user’s profile folder if they don’t exist.
* Merged the previously separate Serenity.Server.Control, Serenity.Service and Serenity.Service.Control projects into the Serenity.Server project, and renamed it to just “Server” to reduce complexity even further.

#### Fixes

* Fixed the build process so that assemblies are properly delay-signed.
* “Fixed” solution so that it is now compatible with Microsoft Visual C# Express 2005 (by eliminating the use of solution folders and making a few other minor changes).

## 0.3.0.0

This release marks a major milestone in the development of Serenity. The introduction of some new features, most notably the usage of the profile directory, will help to demonstrate that Serenity is more than just a toy project and might start gaining serious usage. Additionally, new releases after 0.3.0.0 will be farther between, but each one will add significant stability, functionality, and performance improvements.

#### Additions

* Added Serenity.ProductInfo class, containing some constant values which hold information about the product (Version, Company, Copyright, Product Name, etc.).
* Added Serenity.SPath class to manipulate path information for Serenity-related files and directories.
* Added DomainSettings class to support new DomainSettings system.
* Added ResourceClass and ResourceClassRegistry to support new Resource Class system.

#### changes

* Reverted back to old versioning scheme (e.g. 0.3.0.0 instead of 0.30.0.0).
* Completely reworked server to support usage of %APPDATA% to store application data (Domain Settings, Environments, Modules, etc.).
* Shared SolutionInfo.cs now pulls from the new Serenity.ProductInfo class.
* Updated build script to provide a better installation experience. Installer now auto-detects presence of .NET framework 2.0 and can download and install it automatically if the user chooses.
* Updated documentation.
* API: Serenity.SerenityEnvironment.StaticRoot property renamed to StaticFilesDirectory.
* Complete rewrite of context handling to improve just about every aspect of it.

#### Fixes

* Resolved “8575: System.UnauthorizedAccess Exception when starting Serenity on Windows Vista” issue.
* Repaired some minor XML comments and typos throughout the code.

#### Removals

* Got rid of some outdated solution clutter.
* Removed the “Tranquility” project because it was unused and unnecessary.

## 0.27.0.0

#### Additions

* Added documentation.
* Added a generic abstract Singleton<T> class, to be inherited by objects that follow the singleton design pattern.
* Started recording changes in the Serenity Change Log document.

#### Changes

* Updated SolutionInfo.cs to reflect current copyright information.
* Made Multiton<TKey, TValue> class abstract.
* Changed Serenity.Dynamic.IndexPage class to be internal, to reduce clutter of the Serenity API.

# Credits

Many thanks to all the people who have helped out by contributing their time or money to the project!

#### Team Members

* Will ‘AnarkiNet’ Shelley – Project Coordinator, Programming Lead, and Design Lead.
* Smellyhippy –Programming and Design.

#### Other Contributors

* SeveredCross – Extensive Linux testing/reporting.