

Namespaces in the PDSA Framework

The PDSA .NET Productivity Framework has many Namespaces with hundreds of classes to help you develop your application. All of the namespaces and all of the class are prefixed with “PDSA”. This will help you distinguish the classes within the PDSA Framework from other frameworks that you might use.

For the most up-to-date list of namespaces, please see our PDSAFramework5.chm file contained in the installation folder of our PDSA .NET Productivity Framework.

Common Assemblies

There are a few

DLL Name	Description
Desaware.MachineLicense40	This DLL is used for licensing our DLLs to ensure they are used only by authorized users of our Framework
PDSA.Common	Our most generic classes that are used by the rest of our DLLs. This DLL is required by all other DLLs.
PDSA.Common.Extensions	A set of generic classes that are in addition to the ones in the PDSA.Common DLL.
PDSA.DataLayer	The PDSA Data Class Layer. This data access layer has been in use since .NET 1.0. Our newer data access layer is located in the PDSA.Common DLL and should be used for new applications.
PDSA.Framework	This DLL contains all the data classes for interacting with all of the PDSA tables in the PDSAFramework500 database. This DLL also has our security system, caching, logging and other systems used by our

	Framework applications.
PDSA.Web	A set of classes for working with ASP.NET web applications.
PDSA.WPF	A set of classes for working with WPF applications.
PDSA.WPF.Extensions	A set of classes for working with WPF applications.
System.Linq.Dynamic	A Microsoft DLL that extends the LINQ classes to allow for dynamic building of sorting and filtering expressions.

.NET Framework DLLs needed

Depending on some of the different ways you use our DLLs, you may be required to add some additional .NET Framework DLLs, namely:

- System.ComponentModel.DataAnnotations
- System.Configuration
- System.Runtime.Serialization

PDSA.ActiveDirectory

This namespace contains a set of classes for working with Active Directory. Using the classes in this namespace allow you to retrieve a list of domains in your AD. You can retrieve a list of organizational units, groups and users as well. On the user classes you also have a way to authenticate a user against AD.

Sample Folder

C:\Program Files (x86)\PDSA Framework
5\Samples\WPF\PDSAActiveDirectorySample-CS

Assemblies

PDSA.Common
PDSA.Common.Extensions

PDSA.Cache

This namespace contains a set of provider classes to help you cache objects or other values in your application. What is nice about using these classes is you have one consistent API that you can use in your Web and Desktop applications. Other classes in our PDSA Framework make use of these classes to perform caching.

Sample Folder

C:\Program Files (x86)\PDSA Framework 5\Samples\ASPNET-WebForms\PDSACacheProviderSample-CS

Assemblies

PDSA.Framework

PDSA.Common

This namespace has a set of miscellaneous classes that will give you a lot of functionality for any type of .NET application you may create. Examples of classes in this namespace include string manipulation, date manipulation, number manipulation, registry reading/writing, XML processing, File IO wrappers, object serialization helper and many others.

Sample Folder

C:\Program Files (x86)\PDSA Framework 5\Samples\ASPNET-WebForms\PDSACommonSample-CS

Assemblies

PDSA.Common

PDSA.Common.Extensions

PDSA.Configuration

This namespace has a set of provider classes for helping you retrieve and save configuration information for your application. You can retrieve and save configuration settings from your App.Config or Web.Config file, from an XML file, the pdsaApplicationSetting table, or the registry. The point of this configuration class is it gives you one method of retrieving configuration data regardless of where that data is stored.

Sample Folder

C:\Program Files (x86)\PDSA Framework 5\Samples\ASPNET-WebForms\PDSAConfigurationProviderSample-CS

Assemblies

PDSA.Common.Extensions

PDSA.Framework

PDSA.Cryptography

This namespace contains a set of classes to help you encrypt and decrypt or hash any data in your application. Instead of writing 10 lines of fairly cryptic code, you will only need to write a couple.

Sample Folders

C:\Program Files (x86)\PDSA Framework
5\Samples\WPF\PDSACryptographyLibrarySample-CS

and

C:\Program Files (x86)\PDSA Framework
5\Samples\WPF\PDSACryptographyProviderSample-CS

Assemblies

PDSA.Common.Extensions

PDSA.Framework

PDSA.Cryptography.Hash

This namespace has a set of provider classes that help you with hashing using any of the available hashing algorithms in the .NET Framework. The classes in this namespace will make it easy to switch from one hashing algorithm to another and simplifies the code you need to write compared to using the .NET Framework.

PDSA.Cryptography.KeyManagement

This namespace has a set of provider classes that help you with storing keys for your encryption/decryption in a .config file, an xml file or the registry.

PDSA.Cryptography.Symmetric

This namespace has a set of provider classes that will help you use any of the .NET framework symmetric algorithms to encrypt and decrypt data quickly and easily. The classes in this namespace will make it easy to switch from one symmetric algorithm to another and simplifies the code you need to write compared to using the .NET Framework.

PDSA.DataAccess

The classes in this namespace are an updated data access layer as of August of 2016. It fully supports nullable types and data annotations. It is recommended that you use this data layer for your projects.

Sample Folders

The samples for this data layer are located in Haystack.

C:\Program Files (x86)\Haystack2\Samples\DataAccessLayer

Assemblies

PDSA.Common

PDSA.DataLayer

This namespace has a set of provider classes that allow you to have one consistent interface to the ADO.NET data classes. This allows you to use SQL Server, Oracle and eventually other providers using a single interface. No matter how much Microsoft changes things in the future, this will help you not have to re-write your applications each time.

<p>NOTE: This data layer has been in existence since .NET 1.0 and has been superseded by the classes in the PDSA.DataAccess namespace.</p>

Sample Folders

C:\Program Files (x86)\PDSA Framework
5\Samples\WPF\PDSADataLayerProviderSample-CS

Assemblies

PDSA.DataLayer

PDSA.DataLayer.DataClasses

This namespace has a set of base classes that are used by the data classes generated from Haystack. These classes allow us to generate less code for handling standard CRUD (Create, Read, Update, Delete) operations against your tables, views and stored procedures.

PDSA.DataLayer.Schema

The Haystack code generator needs to read schema data from various database systems in order to generate CRUD classes. This namespace contains a set of provider classes that understand each database system and how to read that schema data.

Sample Folders

C:\Program Files (x86)\PDSA Framework
5\Samples\WPF\PDSADBSchemaProviderSample-CS

Assemblies

PDSA.DataLayer

PDSA.FileIO

If you need to work with sets of files and folders on a hard drive, this namespace has a set of classes that will help you do so.

Sample Folders

C:\Program Files (x86)\PDSA Framework 5\Samples\ WPF\PDSAFileIOSample-CS

Assemblies

PDSA.Common.Extensions

PDSA.Framework

This namespace has a set of classes that work with the application settings you add to your .config file. It also contains enumerations that are used in various classes in the PDSA Framework.

Sample Folders

C:\Program Files (x86)\PDSA Framework 5\Samples\
WPF\PDSAFrameworkSample-CS

Assemblies

PDSA.Framework

PDSA.Framework.BusinessLayer

This namespace contains business validation classes generated by Haystack for the PDSA tables in PDSA Framework database.

Assemblies

PDSA.Framework

PDSA.Framework.DataLayer

This namespace contains data access classes generated by Haystack for the PDSA tables in PDSA Framework database.

Assemblies

PDSA.Framework

PDSA.Framework.EntityLayer

This namespace contains the entity classes generated by Haystack for the PDSA tables in PDSA Framework database.

Assemblies

PDSA.Framework

PDSA.Framework.ViewModelLayer

This namespace contains the view model classes for working with the various views used to administer the tables in the PDSA Framework database.

Assemblies

PDSA.Framework

PDSA.Logging

This namespace contains provider classes to help you log information from your application. Logs can be informational, debug, exceptions, warning, audit tracking, user tracking, or any custom data you wish to log. There are a few different providers to help you log data to a table, a file, the event log or send an email with the log data.

Sample Folder

C:\Program Files (x86)\PDSA Framework
5\Samples\WPF\PDSALoggingProviderSample-CS

Exception Logging: General Usage

To log exceptions from within your “catch” block simply call the following:

```
try
{
    // Code that might cause an error
}
catch(Exception ex)
{
    AppLogExceptionHandler.Log(ex);
}
```

If you wish to add additional information to your log you use a `NameValueCollection` class:

```
try
{
    // Code that might cause an error
}
catch(Exception ex)
{
    NameValueCollection nv = new NameValueCollection();
    nv.Add("Key", "Value");
    AppLogExceptionHandler.Log(ex, nv);
}
```

If you have standard items you wish to add to every exception, open the `AppLogCommon` and locate the `BuildCommonNVC()` method. Add any “global” items from `Session` or other cache areas into this `NameValueCollection` object and these items will always be logged with your exceptions.

Configuration Settings

Be sure to setup the `<PDSALoggingProviders>` section with the appropriate information prior to using the exception logging. You may specify where you wish the exception log information to be stored such as in a database table (`pdsa.pdsaLog`), in a file, sent in an email or to the Event Log.

Assemblies

PDSA.Common.Extensions

PDSA.Framework

PDSA.Menus

This namespace contains classes that work with the `pdsaMenu` table in the PDSA Framework database. This allows you to create data-driven menu systems for your application.

Assemblies

PDSA.Common.Extensions

PDSA.Framework

PDSA.MessageBroker

Communication between different parts of your XAML applications is essential. Creating a system that allows you to communicate without any coupling between those different parts creates a flexible system. This namespace contains classes that will help you send messages from one component to another within your XAML applications without creating a coupling between those components.

Sample Folders

C:\Program Files (x86)\PDSA Framework
5\Samples\WPF\PDSAMessageBrokerSample-CS

Assemblies

PDSA.WPF

PDSA.Provider

This namespace contains some base classes that are used by all of the different providers in the PDSA Framework.

Assemblies

PDSA.Common

PDSA.ReferenceTable

This namespace contains the classes to work with the pdsaReference tables within the PDSA Framework database. Reference tables are also commonly called lookup tables or validation tables. Examples would be US States, Countries, Customer types, Address types, etc. Any table that generally contains a code and a description and is commonly used as a foreign key reference within your database can be a candidate for a reference table.

Assemblies

PDSA.Common.Extensions
PDSA.Framework

PDSA.ReferenceTable.Web

This namespace contains classes to help you work with reference tables within an ASP.NET web application.

Assemblies

PDSA.Common.Extensions
PDSA.Framework

PDSA.Resource

In order to localize an application you will need to store all messages, labels, grid headers, and even the data in your tables in that user's language. Microsoft has made storing labels and messages easy to do using resource files that are built-in to Visual Studio and .NET. But, to get a consistent system that is easy to maintain and use across your whole application this system falls way short. This is why we created our PDSA Resource system. This method provides a consistent approach to all your data localization needs. In addition, we provide a web interface to the messages to allow remote translators to do their job. We also give you tools to help identify which messages and labels have not yet been localized.

Sample Folder

C:\Program Files (x86)\PDSA Framework
5\Samples\WPF\PDSAResourceProviderSample-CS

General Usage

To use this system with the pdsa.pdsaResource table, make sure the .Config file is set to point to the database where this table is located.

Now, when you want to retrieve a value to replace into a Label on your screen, for example, you make the following call:

```
lblFirstName.Text = AppResources.GetResource("FirstName",  
                                             "Not Found");
```

The default language is set in the .config file, however you can change this at runtime by making the following call:

```
AppResources.SetLanguage("en-US");
```

Assemblies

PDSA.Common.Extensions

PDSA.Framework

PDSA.Security

Almost any application built today needs a security system. You need to keep track of users, the roles users play within a system, and most often you need permissions for individual items on a form. Microsoft's security system works fairly well on web applications, but does almost nothing for desktop applications. In addition, they do not provide a permission system. The PDSA Security system gives you users, roles and permissions and is easily used across both web and desktop and mobile applications. Under this namespace is where you will find a set of classes and providers to help you create a security system for your application.

We provide a few different modes for working with security in your application.

- Forms based authentication
 - Users, roles and permissions are stored using our tables
 - We provide the maintenance screens for all of these tables
- Forms based authentication using Single Sign-on
 - Same as Forms based authentication, but we use Windows authentication to grab the user identification from the operating system
 - Users, roles and permissions are stored using our tables
 - We provide the maintenance screens for all of these tables
- Windows authentication

- Windows authentication and authorization is used for your complete application
- We provide classes to help you secure your menus and controls on forms.

PDSA.ServiceModel

Creating a Service-Oriented Architecture (SOA) is almost a requirement for applications built today. This means all business, entity and data access is done in the cloud (on a web server) and each application you build will access those classes from a service. This namespace contains classes that help you build an SOA based application. Haystack will generate your entity, business and data classes and a WCF service layer for you.

Assemblies

PDSA.Common

PDSA.SqlServer

If you need to work with SQL Servers such as getting a list of servers and detecting the type of server, then this namespace has a set of classes that will do this for you.

Sample Folder

C:\Program Files (x86)\PDSA Framework
5\Samples\WPF\PDSASqlServerSample-CS

Assemblies

PDSA.Common.Extensions

PDSA.SqlServer.Picker

If you are building a WPF application and wish to have your user select from a list of SQL Servers, this namespace has some classes and even a WPF User Control that will give you this ability to select a SQL Server.

Sample Folder

C:\Program Files (x86)\PDSA Framework
5\Samples\WPF\PDSASqlServerSample-CS

Assemblies

PDSA.WPF.Extensions

PDSA.Tips

This namespace contains a set of classes that will help you present a set of tips or help screens to your user in your application.

Sample Folder

C:\Program Files (x86)\PDSA Framework 5\Samples\WPF\PDSATipsSample-CS

Assemblies

PDSA.Common.Extensions

PDSA.UI

This namespace contains a set of base classes that are used by many of our other classes in our Framework. All of these help in presenting data on the user interface of your application.

Assemblies

PDSA.Common

PDSA.Validation

This namespace has a set of classes that are used by all the validation classes generated by Haystack. We provide an easy to use system for validating entity classes that is independent of where the data comes from that populates those entity classes. In this way you can have entity classes that come from a database, an xml file, an SAP system, or anywhere else. Your entity classes can then be validated using our classes in this namespace.

Sample Folder

C:\Program Files (x86)\Haystack\Samples\WPF\Validation_Sample_CS

Assemblies

PDSA.Common

PDSA.Web

When developing an ASP.NET web application you will find you need a common set of functionality over and over again. This namespace contains classes that will help you with a lot of this common functionality.

Sample Folder

C:\Program Files (x86)\PDSA Framework 5\Samples\ASPNET-WebForms\PDSASWebLibrarySample-CS

Assemblies

PDSA.Web

PDSA.WindowsService

This namespace contains a set of classes to help you build Windows Service applications.

Assemblies

PDSA.Common.Extensions

PDSA.WPF

When developing a WPF applications you will find you need a common set of functionality over and over again. This namespace contains classes that will help you with a lot of this common functionality.

Sample Folder

C:\Program Files (x86)\PDSA Framework 5\Samples\WPF\PDSAWPFSample-CS

Assemblies

PDSA.WPF

PDSA.WPF.Extensions

Summary

In this chapter you were presented with an overview of the various namespaces available within the PDSA Framework.