

# Overview of the PDSA .NET Productivity Framework

The PDSA .NET Productivity Framework is made up of many re-useable classes, providers, components, utilities, sample applications, database tables, business objects and data access objects. The classes that connect to the "pdsa" tables in the database were created using the Haystack Code Generation product, which is also included with the PDSA Framework. The PDSA .NET Productivity Framework extends the .NET Framework, adding additional classes and complete sub-systems that you will not find in the .NET Framework. This gives you additional functionality that you would otherwise have to build yourself. This will save you hundreds of hours of time and frustration.

## Why you need a Framework

When you start with a solid foundation of pre-built sample applications and templates, all you need to do is to start adding your own unique windows, user controls, and web pages to this foundation. You can look at the existing samples, follow our guidelines, use our utilities, and generate code to make you much more productive than starting from scratch.

When you begin building any .NET application, you should never start with File | New Project. If you do, you are already behind in your schedule as you will just have too much to add in and build yourself. You should already have built a project template that consists of such re-usable components as base classes, base pages

classes, methods to handle configuration data, a data handling layer, exception handling, caching, and many other design patterns. By having these templates already built you will have a good foundation for adding the custom pieces that make up the application you are trying to build. This is exactly what the PDSA .NET Productivity Framework gives you.

## Goals of an Application Framework

Any well-run programming shop **always** has a framework, an architecture, and standards to follow. Included in your framework should be things like the following:

- A set of template applications
- A set of re-usable components
- A design pattern for common types of pages/windows
- Naming and coding standards

Having the above pieces in place prior to development has proven to lower the total cost of ownership of a software application. In addition you will save a great deal of time and money developing each application.

## Application Templates

Instead of starting with File | New project you should always have a set of template projects with all of your configuration handling, exception handling, logging, caching and other reusable classes already included. In addition you will most likely have a set of maintenance forms that give you the ability to maintain your application. You should have templates for any application you might create such as:

- ASP.NET MVC applications
  - Can be used for Web API, jQuery, and Angular applications too
- ASP.NET Web Form applications
- WPF applications

## Guidelines & Standards

Standards and guidelines for development are essential to ensure a consistent use of the tools and a consistent programming style is followed by all programmers on your project. You should typically have standards and guidelines for the following:

- Programming Standards

- Database Design Standards
- N-Tier Design Patterns
- Stored Procedure Standards
- Visual Studio Configuration Standards
- Source Code Control Usage

The intent of the PDSA .NET Productivity Framework is to give you a good starting point when developing a brand new .NET application that includes all of the above and more. We fully intend that you will modify the sample code and extend the basic functionality as you create your applications. But we provide all of the above architectural guidelines.

## Overview of the PDSA Framework Namespaces

There are many components to the PDSA .NET Productivity Framework. Each of the components is broken up into a namespace. The top level namespace is named **PDSA**. Underneath this namespace there are several other namespaces (see Figure 1). Each namespace contains classes that will help you become more productive when building applications with .NET.

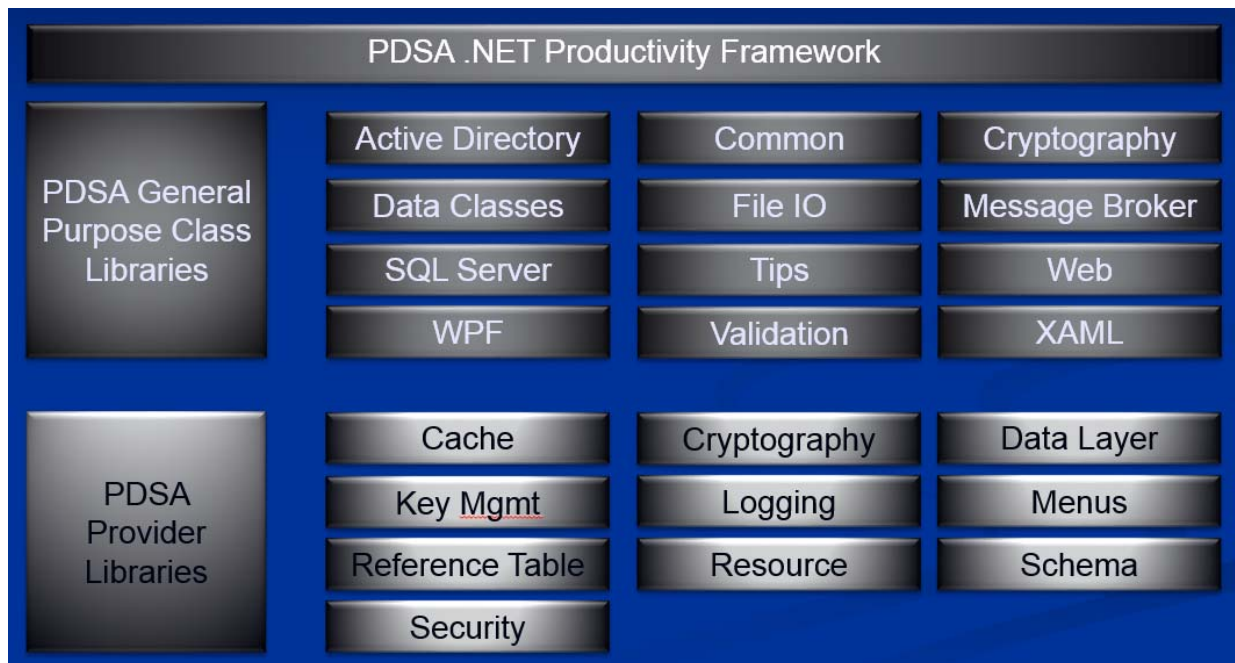


Figure 1. The PDSA Namespace Architecture

The namespaces on the top of Figure 1 are generic namespaces that can be used in any type of .NET application you create. These will be covered in more detail in a later chapter. The namespaces on the bottom of Figure 1 are provider classes. Within the top portion are namespaces that are specific to the type of application you are creating. When you are creating a Web application using ASP.NET you can use any of the generic namespaces combined with any of the namespaces under the PDSA.Web namespace. In most cases you will find a namespace such as **PDSA.Web.Security** or **PDSA.Web.Menus** under which are classes that are specific for web applications. There is also a **PDSA.WPF** namespace and a **PDSA.XAML** namespace for working with XAML applications.

## PDSA Framework Utilities

There are many utilities included with the PDSA .NET Productivity Framework. For example, the Application Builder will help you create a new application from one of our standard template applications. There is also a utility to help you reset passwords for users.

## PDSA Database Tables

In order to have a logging system, security system, a reference table system, localization, etc., you need a set of tables for these components to store their data within. The PDSA Framework comes with a complete set of scripts to create these tables and stored procedures for SQL Server. A SQL Server is necessary for the PDSA tables. However, the database for your application can be any database supported by ADO.NET and our PDSA Database Provider system.



Figure 2. Some of the Tables in the "PDSA" schema

## Summary

The PDSA .NET Productivity Framework contains a complete set of reusable components, a solid architecture on which to build your applications and several utilities that will make your development experience much more productive.