**ASP.NET MVC FAQs**

**What is ASP.NET MVC?**

ASP.NET MVC helps Web developers build compelling standards-based Web solutions that are easy to maintain by leveraging the Model-View-Controller (MVC) pattern to decrease the dependencies between layers. It also provides complete control over the page markup and improves testability which enables Test Driven Development (TDD).

**What is MVC? Was it invented at Microsoft?**

MVC is an acronym for Model-View-Controller. It is a design pattern that decreases the coupling between the different layers of an application (presentation, business and data) making it more robust and easy to maintain. ASP.NET MVC is a component of the ASP.NET Web application framework that allows Web developers to use the MVC design pattern when creating Web solutions. The MVC design pattern was not invented at Microsoft.

You can read more about the MVC design pattern on Wikipedia (<http://en.wikipedia.org/wiki/Model-view-controller> ). Stephen Walther has a nice blog entry about the evolution of the MVC design pattern over the years leading up to ASP.NET MVC (<http://www.stephenwalther.com/blog/archive/2008/08/24/the-evolution-of-mvc.aspx>).

**What is MVP? How is it different from MVC?**

MVP (Model-View-Presenter) and MVC (Model-View-Controller) are both design patterns that reduce the coupling between the different layers of an application (i.e. separation of concerns).

The MVP pattern is similar to the MVC pattern. The MVP pattern is more suited to component based GUI applications where the GUI components themselves handle the user input whereas the MVC pattern is well suited for applications where a central controller handles user input. This works especially well for Web applications where the user input is in the form of HTTP requests that are handled by the controller.

**What are some of the benefits of using ASP.NET MVC?**

* **Web solutions created using ASP.NET MVC are easy to maintain**Modular applications isolate the presentation, user interaction logic and data/domain layers from each other. This helps reduce the dependency between these layers and makes maintenance easier. ASP.NET MVC achieves this modularity by breaking the application into the Model, View and Controller layers as outlined in the Model-View-Controller design pattern.
* **Create powerful Web solutions using ASP.NET MVC**  
  ASP.NET MVC allows developers to gain complete control over the page markup. This empowers developers to customize pages to suit their requirements.
* **ASP.NET MVC is a part of ASP.NET**  
  The ASP.NET Web application framework helps Web developers by providing ready access to rich features and functionality such as Membership, Roles, Caching, etc. ASP.NET MVC is a part of the ASP.NET Web application framework and Web solutions built using ASP.NET MVC can leverage the built-in features and functionality provided by ASP.NET.
* **ASP.NET MVC provides ways to improve collaboration amongst teams**Web solutions created using ASP.NET MVC have modular architecture. This allows members of a team to work independently on the various modules and can be used to improve collaboration. For example: Developers can work on the Model and Controller layers (domain logic and user interaction logic) while the designers work on the View (presentation).
* **Improve reliability and robustness of Web solutions by using ASP.NET MVC and Test Driven Development (TDD)**ASP.NET MVC provides better support for Test Driven Development (TDD). This makes it easier for developers to incorporate tests early on in the project and improve the reliability and robustness of Web solutions.
* **ASP.NET MVC can help make upgrades and fixes easier**  
  Web applications created using ASP.NET MVC are modular in nature. These modules are largely independent of each other making it easy to upgrade one module at a time without having to rewrite the entire application together. It can also make debugging and fixing bugs easier by limiting the effects to a single component.

**When should I use ASP.NET Web Forms vs. ASP.NET MVC?**

ASP.NET Web Forms and ASP.NET MVC offer developers the choice to build Web solutions in different ways. Developers should pick one over the other based on the needs of the project and their personal preference.  
  
You should use ASP.NET Web Forms when:

* Creating a website faster is more important to you than complete control over markup.
* You like the abstraction offered by controls.
* You want to avoid writing inline code.
* You are not interested in pattern-based software development and are not familiar with the MVC pattern.

You should use ASP.NET MVC when:

* Complete control over markup is important to you and you are willing to put in the extra effort.
* You don’t like the abstraction offered by controls.
* You don’t mind writing inline code.
* You are interested in pattern-based software development and like the separation between layers offered by the MVC pattern.

**Where can I learn more about ASP.NET MVC?**

You can learn more about ASP.NET MVC on the ASP.NET Website <http://www.asp.net/mvc>. There are tutorials and videos to help you get started.   
  
Many Microsoft folks regularly post blog entries on ASP.NET MVC. Some of them are:

Phil Haack - <http://www.haacked.com/>   
Scott Guthrie - <http://weblogs.asp.net/scottgu/>   
Stephen Walther - <http://stephenwalther.com/blog/default.aspx>   
Scott Hanselman - <http://www.hanselman.com/blog/>

**Are there any examples of sites that are using ASP.NET MVC?**

StackOverflow (<http://www.stackoverflow.com>) is a great example of a site that is built using ASP.NET MVC. Jeff Atwood one of the founders of StackOverflow was a co-presenter for the ASP.NET MVC session at PDC 2008. He outlined some of the statistics of StackOverflow which include about 300,000+ page views per day. You can view the entire session here: <http://channel9.msdn.com/pdc2008/PC21/> .  
  
Some other sites using ASP.NET MVC are:

* <http://www.cruvee.com/>
* <http://www.ideavine.net/>
* <http://www.theloungenet.com/>
* <http://www.retaggr.com/>
* <http://www.visitmix.com/>
* <http://www.dimecasts.net/>
* <http://www.39x27.com/home.mvc>
* <http://www.codeplex.com/>
* <http://www.richpictures.co.uk/>
* <https://www.netidme.com/Personal>
* <http://www.jumpthegun.co.uk/>
* <http://www.devtacular.com/>

**Where can I download ASP.NET MVC? Are there any pre-requisites?**

When ASP.NET MVC is released you can download it from the download section on the ASP.NET Website: <http://www.asp.net/downloads/>.   
  
Before you install ASP.NET MVC you must have Microsoft .NET Framework 3.5 SP1 installed which is available at the same location: <http://www.asp.net/downloads/>.

**Where can I get support for ASP.NET MVC?**

You can ask questions or report issues in our ASP.NET MVC forum on the ASP.NET site: <http://forums.asp.net/1146.aspx>. We have an active community of Web developers including the Microsoft team that owns ASP.NET who actively participate in the forums and help each other.  
[What other options are there for support?]

**How can I participate in the design process for future releases of ASP.NET MVC?**

The ASP.NET team treats Web developer satisfaction as one of the most important requirement. To facilitate this process the ASP.NET team publishes its source code (read-only) on the Codeplex website ([www.codeplex.com/aspnet](http://www.codeplex.com/aspnet)). This includes the source code for ASP.NET MVC. Web developers can download the source code of in-progress upcoming releases and submit feedback on things they like/don’t like. The team uses this feedback to determine the features to include in each release.