

SharePoint 2010 Developer Track

Duration: 5 days

Thorough education is key to working with SharePoint 2010 productively. This course guides you through essential 2010 elements, from pre-requisites to system integration, giving you the skills to work confidently and leverage full value from new technology.

Student Prerequisites

Ideally, students will already possess some hands on SharePoint end user experience, be familiar with Visual Studio, and possess some knowledge of .NET programming language (like C# or VS.NET) as well as object oriented concepts. Some ASP.NET experience including User Controls, Server Controls, and Web Services is helpful. Web oriented development and Web technologies like CSS, ECMAScript, and HTML are not required but will help with some course labs.

Audience

The primary audience for this course is .NET developers that need to learn SharePoint.

COURSE COMPONENTS:

- Visual studio 2010 SharePoint Tools
- Application Lifecycle Management (ALM)
- Farm and Sandboxed Solutions
- Server and Client Object Models (.NET, Silverlight and JavaScript)
- Authentication and Membership Providers
- Master Pages, Branding, Themes, CSS, and JavaScript
- The Fluent User Interface (AKA The Ribbon) and Custom Navigation
- OOB Webparts, XLV Webpart, and Custom Web Parts
- Business Connectivity Services (BCS)
- LINQ to SharePoint
- ADO.NET Data Services (REST)
- PowerShell for SharePoint Foundation 2010
- Web Services and WCF
- Field Types and Field Controls
- Timer Jobs
- Event Receivers and Feature Callouts
- Site and List Workflow
- Mobile Applications
- SharePoint Foundation 2010 Developer Dashboard
- Change Log, Usage Log, Auditing and Unified Logging Service (ULS)

SharePoint Foundation 2010 Primer

- It is essential for developers to master the SharePoint vernacular that is pivotal to the successful implementation of an out of the box SharePoint Farm before setting out to alter that experience. So, this module is an attempt at level-setting the participants understanding of the SharePoint platform, essential SharePoint terminology, core SharePoint developer concepts, the SharePoint developer environment, and along the way we debunk several SharePoint myths for good measure.

SharePoint Tools in Visual Studio 2010

- It's been said that craftsman are only as good as their tools. So, alongside the release of SharePoint 2010 is Visual Studio 2010 which includes a plethora of new tools for building SharePoint platform extensions and packaging them for deployment. This module explores the numerous new capabilities Visual Studio provides to enrich the SharePoint development experience and enhance SharePoint development productivity. Further, the labs explore how to extend Visual Studio itself to enhance the development experience to achieve even greater developer productivity.

Features and SharePoint Solutions

- A SharePoint Feature is a bundle of functionality that can be activated at a given scope. A SharePoint Solution is a collection of assets packaged for deployment to a given SharePoint context. These two facilities are fundamental to every SharePoint platform extension. No matter how big or small, everything deployed to SharePoint must be managed using Features and Solutions. Fortunately, Visual Studio 2010 provides significant assistance in configuring and generating both Features and Solutions.

Server and Client Object Models

- The Share Object Model comes in two flavours. The Server Object Model is used to query and update the configuration and content databases. This provides a layer of abstraction that ensures that code created today will work tomorrow even significant underlying architectural changes are made to the SharePoint platform. The Client Object Model provides methods techniques of interacting with the Server Object Model from contexts other than the SharePoint server. The SharePoint Ribbon is dependent on the Client Object Model.

PowerShell for SPF 2010

- SharePoint administration is now done from PowerShell, like all the other Microsoft server products. Yet PowerShell represents so much more than just a new administrative interface. This module first explains how PowerShell is like/unlike other environments that developers have used in the past. The labs then demonstrate how to create custom PowerShell cmdlets to supplement and enhance the SharePoint administrator's experience and productivity by encapsulating complex code into a DLL for easy and less error-prone use within a SharePoint environment. This encapsulation can include the ability to pipe objects directly into the custom cmdlet. Very "powerful".

SharePoint Web Parts

- Web Parts are the basic building blocks that fulfill SharePoint's prime directive: Empower the end user to self-sufficiency. A self-reliant user is a happy user. Web Parts allow the end user, within some constraints, to assemble, configure, and customize their own applications. Visual Studio 2010 introduces a SharePoint Project Item that facilitates creating Web Parts with a design surface called Visual Web Parts. The labs for this module also explore the concept of Replaceable Parameters in Visual Studio 2010 SharePoint Projects.

Branding

- When directing corporate consistency (content placement or look and feel - AKA brand) becomes more important than empowering individualistic flexibility, a Web Content Management solution is probably in order. While this is not a Web Content Management course (we have one of those too), Master Pages are fundamental of the core platform. SharePoint 2010 also introduces a new theming framework called THMX which provides a flavor for a brand. The labs for this module not only address Master Pages but also annotating custom CSS with theme colors and fonts including recoloring images.

The Ribbon and Custom Navigation

- One pivotal aspect to any Web development effort is navigation. One of the most visible changes in SharePoint 2010 is the user interface element called the Ribbon. The Ribbon along with several other the context-sensitive elements: the Dialog Framework, the Notification Area, the Status Area, and the AJAX framework make up with is known as the fluent UI. Creating Custom Actions and orchestrating these elements requires a few JavaScript programming skills. Since SharePoint leverages the ASP.NET Navigation Provider model for providing a Site Map of links to Sites and Lists in close proximity to the current Site, the labs explore creating a Custom Site Map Provider too.

Business Connectivity Services (BCS)

- Showing data from external applications has long been the job of the Data View Web Part, but it required the end user have some technical expertise and knowledge of the external applications content organization; updating that external data was also quite difficult. In SharePoint Foundation 2010, the Business Data Catalog (BDC) has been included in the platform and surfaced via Business Connectivity Services. So, creating a BDC entity model that users can use to interact with using Virtual Lists from data stores external to SharePoint doesn't require a SharePoint Server client access license (CAL). The labs in this module show the tools included in SharePoint Designer and Visual Studio 2010 to create models and generate Virtual Lists.

WCF Data Services (REST) for SPF 2010

- Representational State Transfer (REST) is such a poor name. Think query by URL to retrieve JSON object response. Also known as ADO.NET Data Services, WCF Data Services, and oData, SharePoint 2010 now fully supports the REST protocols for querying data in SharePoint Lists.

LINQ to SharePoint

- First there was LINQ to Objects, then LINQ to SQL and LINQ to XML. Now there is LINQ to SharePoint. Integrating query into the language provides many benefits not the least of which is a strongly-typed (think object properties rather than arrays with string keys) design-time validation of a SharePoint List query. No more CAML queries in managed code! This module demonstrates not only how to query SharePoint Lists using LINQ to SharePoint but also how to update that content.

Event Receivers and Feature Receivers

- Extending SharePoint imperatively can easily be done by hooking into the events surfaced by the SharePoint platform. Events are fired when users interact with List Items, Lists, Webs, and Features. This module explores what events are available, how to configure them declaratively and imperatively, and how to package them so the end user can decide what contexts they will run in. The end user in two of the four contexts is the SharePoint Farm administrator. :)

Timer Jobs

- There are only two ways to run code: 1. someone did something (see Event Receivers and Feature Receivers) or 2. Time has passed. Timer Jobs can be scheduled to run only once or on a given schedule. They can run on a single server or every Web Front End (WFE) on the SharePoint Farm. They always initially run in a God-mode thread so they make a wonderful catch-all for anything that hasn't been accommodated by the SharePoint platform extension framework.

Authentication and Membership Providers

- Who is that trying to access to the SharePoint Farm? Evaluating the user's ability to identify themselves by providing some predetermined evidence that they are who they claim to be is not something that SharePoint tries to do. Instead, SharePoint relies on the pluggable ASP.NET membership provider model. So, this could be as simple as collecting the users username and password or could include additional factors like the presence of a smart card, a digital certificate, a USB dongle, or even a biometric scan. This modules lab demonstrates how even SharePoint could be used as a repository of members for authenticating to SharePoint.

Role Providers and Claims-based Access

- While many developers would be in favour of just granting everyone administrator access and be done with security. The reality is that businesses need to make sure only certain people have access to certain information. SharePoint provides four securable objects: Site, List, Folder, and Item. By default, all objects inherit their permissions from the top-level Site. Permissions can be granted on an individual by individual basis or a collection of people can be granted access (AKA role-based security). There are the traditional ways of naming collections of people like Active Directory Groups, Forms Based Authentication (FBA) Roles, and even SharePoint Groups. However, there is a new way of identifying a collection of people via some common characteristic provided as a Claim made by the identify provider. This modules lab explores how to augment claims for an existing identify.

Site and List Workflow

- Windows Workflow Foundation (WF) workflows help people know what steps remain to complete a business process. While SharePoint doesn't _do_ workflow, WF is part of the .NET Framework, SharePoint makes an excellent workflow host. SharePoint 2010 now includes Site-level workflow and reusable declarative workflow. The labs for this module focus on the facilities in SharePoint Designer (SPD) to generate reusable declarative workflows as well as more challenging imperative workflows using Visual Studio 2010.

Application Lifecycle Management (ALM)

- SharePoint may maintain the fully qualified reference to imperative code in dozens of web.config files scattered across one or more Web Front End (WFE) servers, in XML files through the {SharePoint Root}, and in hundreds of rows in the configuration and content databases. Upgrading from one assembly version to another is a daunting undertaking and typically hasn't been done. Most 1.0.0.0 assemblies running in SharePoint have been upgraded without revving the version number. SharePoint Foundation 2010 introduces facilities that will begin to see that change. However, they are very much in the first release stage. This module's labs not only explore FeatureUpgrade,

BindingRedirect and the like but also demonstrates the use of a WebConfigModification when it is required.

Mobile Applications

- More and more people have mobile devices from which they can easily interact with SharePoint. So, SharePoint 2010 has invested in making the mobile experience slightly more aesthetic. This module's lab explores extending the out of the box mobile application with custom functionality.

Field Types and Field Controls

- All SharePoint Lists and Libraries consist of a collection of columns which derive their display-mode and edit-mode characteristics from one of the out of the box Field Types. The presentation of the data managed by the Field Type is governed by a Field Control. This module's lab examines how custom Field Types can be derived from an appropriate out of the box Field Type and how a custom user interface can be presented in the context of a List Item being displayed in a list or on its own, as well as its behaviour and presentation when the screen is in edit-mode.