



Working with SharePoint Data

Joseph Ackerman
Senior Consultant, Allin Corp.
November 8-9, 2008



The Ever-Popular Intro Slide



- **Joseph Ackerman**
(joseph.ackerman@allincal.com)
- **20 + years of software development for clients of all sizes and shapes**
 - Wells Fargo
 - Hewlett-Packard
 - Aetna
 - Visa
 - Stanford University
- **Currently consulting for Allin Corp.**
 - Microsoft's Northwest Region Partner of the Year 2008
 - Most recent project: internal systems for **eBay**

- What it's not
- The big (overall) “why”
- SharePoint storage
- The SharePoint data object model
- The custom ListInfo web part
- The little (specific) “why”
- Finding what's hidden
- Queries
- Summary and wrap-up

What this talk is not

- **It's not about how to build web parts**
 - However, we will build some
- **It's not a CAML tutorial**
 - But we will discuss it and use it a bit
 - It's pretty much unavoidable
- **It's not a solution to all your SharePoint data woes**
 - We could spend days talking about this stuff and not cover everything
 - Things are getting better, but any experienced developer will tell you that SharePoint is woefully under-documented for developers
 - Your best bet is just to dig in and start working with it

Why am I giving this talk?



- **SharePoint is built on top of**
 - ASP.Net 2.0
 - SQL Server
 - Easy stuff, Right?
- **Yes...but!**
- **There are many differences between working with ASP.Net and SharePoint!**
- **One of the biggest differences is working with data stored in SharePoint.**
- **There is no ADO.Net Provider for SharePoint**
- **There is no (official) LINQ to SharePoint (yet!)**

Why am I giving this talk?



- **Until the day Microsoft sees fit to provide us with simpler, easier to use tools, our best bet is to access data in SharePoint via**
 - The SharePoint Object Model
 - CAML (more on this later)
- **There's Web Services too, but that's another talk**

- **SharePoint stores all of its data in a construct called a List**
- **Conceptually, Lists are like Tables in a Database**
- **Columns are referred to as Fields**
- **Rows are referred to as ListItems**
- **However, Lists have a lot more meta data**
- **They are less flexible in some ways - some things we take for granted in a database (like joins, for example) are either difficult or impossible**

Where do we begin? In Context!



- You have to reference the SharePoint Libraries

```
using Microsoft.SharePoint;  
using Microsoft.SharePoint.Administration;  
using Microsoft.Office.Server;
```

- To communicate with SharePoint data, you have to have the proper context
- The SPContext object provides that for you
- If your code is in a web part you can use the “current” context

```
SPSite site = SPContext.Current.Site;
```

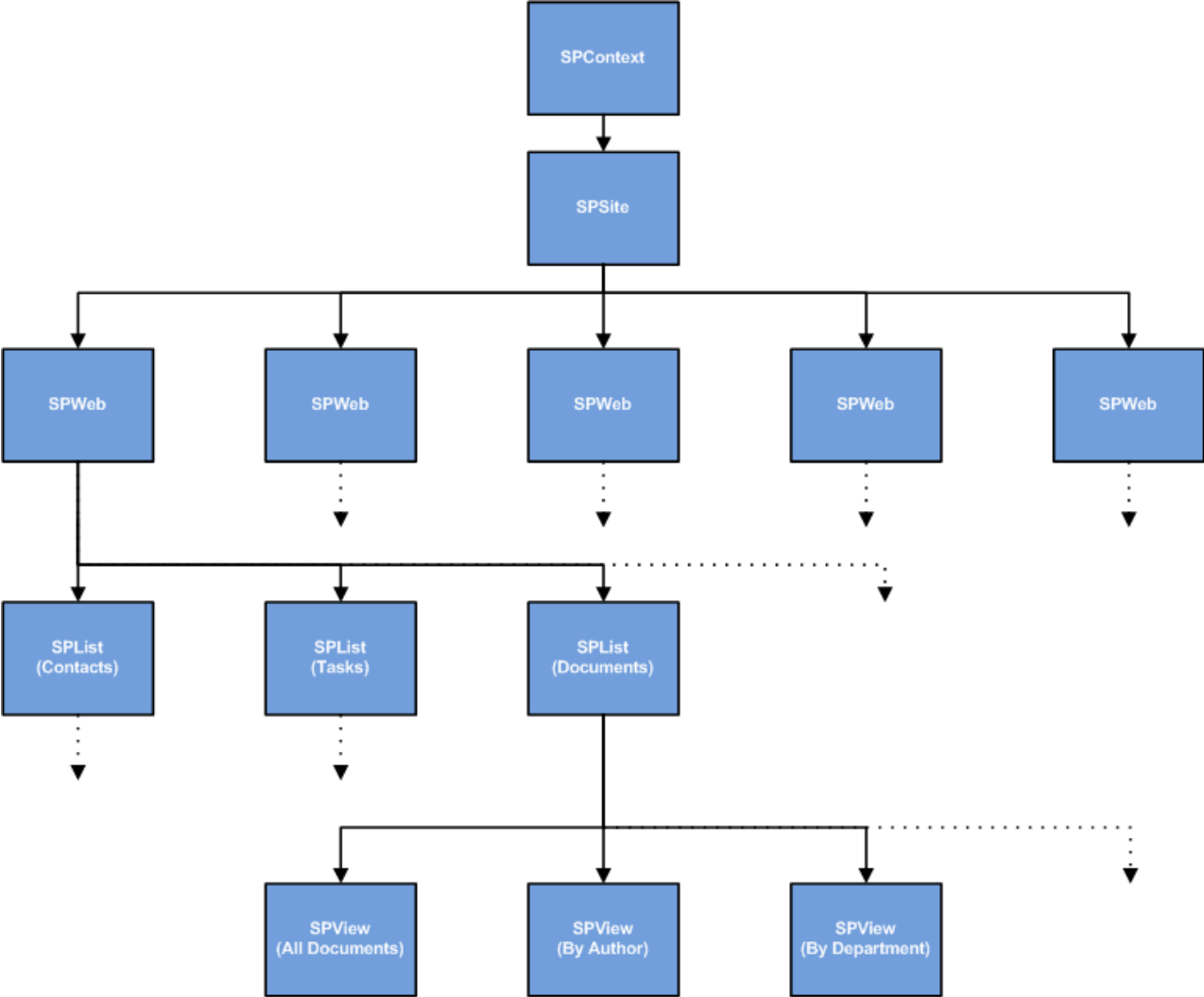
- The context object allows you to get the SPSite object (actually a “site collection”)

Sites have webs



- The Site Collection has one or more webs, accessible through the AllWebs collection
- The default web for the collection is accessed with the OpenWeb() method
- Each web has a collection of lists of varying types (Custom Lists, Document Libraries, Picture Libraries, Task Lists, Announcements etc.)
- Each List has 1-to-n Views

Object Model Map





ListInfo Web Part Demo



So, why do we need all this?



- **First reason:**
- **SharePoint hides a bunch of stuff**
 - There are hidden lists
 - There are hidden fields in the various list types
 - Some built in methods don't return everything
 - For example `SPList.Items.GetDataTable()`
 - Where are the rest of my fields?!



HiddenList Web Part Demo



So, why do we need all this?



- **Second Reason:**
- **If you want to write queries to retrieve list items that meet specific criteria, you have to use CAML**
 - CAML queries use an XML syntax in place of SQL
 - The field list in your query requires the use of the internal field names and their SharePoint Types
 - There isn't really a clean, simple, obvious place to find this information

- **CAML, while useful, has some serious shortcomings:**
 - There's a limited number of conditions and functions
 - There's a lot of details about fields you **MUST** know or you'll spend a lot of time debugging:
 - Internal field names
 - SharePoint data types
 - Data formats for things like dates and lookup values
 - There's no joins - you can get values from lookup fields but that's it
 - CAML queries are a real pain to use in code - you can write your own helper classes or you can use some that others have written, but the ultimate result will always be a string

- You can write your query as SQL and convert it by hand using the method found here:
 - http://www.codeproject.com/KB/sharepoint/Sharepoint_CAML_query.aspx
- Or you can save yourself a lot of work by using other tools
 - The U2U CAML Query Builder is VERY helpful



CAML Query Web Part Demo



- Learn and know the SharePoint data object model
- You may also want to learn the web services
- Familiarize yourself with CAML
- Use the available tools, and when necessary, build your own
- Learn to search the SharePoint blogs because the official documentation from Microsoft (even the SDK!) leaves much to be desired
- With apologies to Nike, “Just do it”

- **Tools**

- U2U CAML Query Builder
- LINQ to SharePoint (CodePlex)
- Stramit CAML Viewer 2007 (CodePlex)

- **Blogs**

- Andrew Connell
- Heather Solomon
- <http://www.sharepointblogs.com/>

- **Books**

- *Professional SharePoint 2007 Web Content Management Development* by Andrew Connell
- *Microsoft SharePoint 2007 Development Unleashed* by Kevin Hoffman and Robert Foster
- *Developer's Guide to the Windows SharePoint Services v3 Platform* by Todd Bleeker