

SharePoint Development Standards

This is only meant as application specific standards. You should always review these standards along with regular development standards which identify things like naming conventions and approaches.

General Principals:

- All new functionality and customizations must be documented.
- Do not edit out of the box files.
 - For a few well defined files such as the Web.config or docicon.xml files, the built-in files included with SharePoint Products and Technologies should never be modified except through official supported software updates, service packs, or product upgrades.
- Do not modify the Database Schema.
 - Any change made to the structure or object types associated with a database or to the tables included in it. This includes changes to the SQL processing of data such as triggers and adding new User Defined Functions.
 - A schema change could be performed by a SQL script, by manual change, or by code that has appropriate permissions to access the SharePoint databases. Any custom code or installation script should always be scrutinized for the possibility that it modifies the SharePoint database.
- Do not directly access the databases.
 - Any addition, modification, or deletion of the data within any SharePoint database using database access commands. This would include bulk loading of data into a database, exporting data, or directly querying or modifying data.
 - Directly querying or modifying the database could place extra load on a server, or could expose information to users in a way that violates security policies or personal information management policies. If server- side code must query data, then the process for acquiring that data should be through the built-in SharePoint object model, and not by using any type of query to the database. Client-side code that needs to modify or query data in SharePoint Products and Technologies can do this by using calls to the built-in SharePoint Web services that in turn call the object model.
 - Exception: In SharePoint 2010 the Logging database can be queried directly as this database was designed for that purpose.
- In SharePoint 2007 site and list templates must be created through code and features (site and list definitions). STP files are not allowed as they are not updatable.
- Use best practices when accessing data using the SharePoint object model. See [Working with Large Lists](#), [Common Coding Issues When Using the SharePoint Object Model](#), and [Using Disposable Windows SharePoint Services Objects](#)

Quality Assurance:

- Custom code must be checked for memory leaks using [SPDisposeCheck](#).
 - False positives should be identified and commented.

- Code should be carefully reviewed and checked. As a starting point use this [code review checklist](#) (and provide additional review as needed).
- Provide an Installation Guide which contains the following items (Note this relates to [SharePoint Deployment Standards](#)):
 - Solution name and version number.
 - Targeted environments for installation.
 - Software and hardware Prerequisites: explicitly describes what is all needed updates, activities, configurations, packages, etc. that should be installed or performed before the package installation.
 - Deployment steps: Detailed steps to deploy or retract the package.
 - Deployment validation: How to validate that the package is deployed successfully.
 - Describe all impacted scopes in the deployment environment and the type of impact.

Branding:

- A consistent user interface should be leveraged throughout the site. If a custom application is created it should leverage the same master page as the site.
- Editing out of the box master pages is not allowed. Instead, duplicate an existing master page; make edits, then ensure you add it to a solution package for feature deployment.
- When possible you should avoid removing SharePoint controls from any design as this may impact system behavior, or impair SharePoint functionality.
- All Master Pages should have a title, a description, and a preview image.
- All Page Layouts should have a title, a description, and a preview image.

Deployment:

- All custom SharePoint work should be deployed through SharePoint Solution (.wsp) files.
- Do not deploy directly into the SharePointRoot (12-Hive, 14-Hive) Folders. Instead deploy into a folder identified by Project Name.

Features:

- Features must have a unique GUID within the farm.
- Features with event receivers should clean up all changes created in the activation as part of the deactivation routine.
 - The exception to this is if the feature creates a list or library that contains user supplied data. Do not delete the list/library in this instance.
- Features deployed at the Farm or Web Application level should never be hidden.
- Site Collection and Site Features may be hidden if necessary.
- Ensure that all features you develop have an appropriate name, description, updated version number and icon.

SharePoint Designer:

- SharePoint Designer 2007 updates are generally not allowed.
 - The only exception to this rule is for creating DataForm Web Parts.

- The following is a recommended way of managing this aspect:
Create a temporary web part page (for managing the manipulation of a data view web part). Once the web part is ready for release and all modifications have been made export the .webpart and then delete the page. You can now import it onto a page elsewhere or place it in the gallery. This way none of your production pages are unghosted. The other advantage is that you can place the DVWP on a publishing page (as long as there are web part zones to accept them).
- DataForm Web Parts should be exported through the SharePoint GUI and solution packaged for deployment as a feature.
- This does not mean that SharePoint Designer should not be used for creating and testing branding artifacts such as master pages and page layouts.
 - It is important for these artifacts to be deployed through solution files (WSPs) and typical build and deployment processes and not by manual methods.
- SharePoint Designer 2010 updates are generally only allowed by a trained individual.
 - The following is a recommended way of managing the creation of DataForm Web Parts:
Create a temporary web part page (for managing the manipulation of a data view web part). Once the web part is ready for release and all modifications have been made export the .webpart and then delete the page. You can now import it onto a page elsewhere or place it in the gallery. This way none of your production pages are unghosted. The other advantage is that you can place the DVWP on a publishing page (as long as there are web part zones to accept them).
 - DataForm Web Parts should be exported through the SharePoint GUI and solution packaged for deployment as a feature.
- SharePoint Designer workflows should not be used for Business Critical Processes.
 - They are not portable and cannot be packaged for solution deployment.
 - Exception Note: Based on the design and approach being used it may be viable in SharePoint 2010 for you to design a workflow that has more portability. This should be determined on a case by case basis as to whether it is worth the investment and is supportable in your organization.

Site Definitions:

- In SharePoint 2007 site and list templates must be created through code and features (site and list definitions).
 - STP files are not allowed as they are not updatable.
- Site definitions should use a minimal site definition with feature stapling.

Solutions:

- Solutions must have a unique GUID within the farm.
- Ensure that the new solution version number is incremented (format V#.#.#).
- The solution package should not contain any of the files deployed with SharePoint.
- Referenced assemblies should not be set to "Local Copy = true"
- All pre-requisites must be communicated and pre-installed as separate solution(s) for easier administration.

Source Control:

- All source code must be under a proper source control (like TFS or SVN).
- All internal builds must have proper labels on source control.
- All releases have proper labels on source control.

InfoPath:

- If an InfoPath Form has a code behind file or requires full trust then it must be packaged as a solution and deployed through Central Administration.
- If an InfoPath form does not have code behind and does not need full trust the form can be manually published to a document library, but the process and location of the document library must be documented inside the form.
 - Just add the documentation text into a section control at the top of the form and set conditional formatting on that section to always hide the section, that way users will never see it.

WebParts

- All WebParts should have a title, a description, and an icon.

Application Configuration

- There are only a few methods in which application configuration data can be stored. When selecting the option that is right for the situation any decision maker must review this article [Managing Custom Configuration Options for a SharePoint Application](#) before making the decision.
 - Web.config
 - APIs such as the ConfigurationSection class and SPWebConfigModification class should always be used when making modifications to the Web.config file.
 - HTTPModules, FBA membership and Role provider configuration must be made to the Web.config.
 - Property Bags
 - It is recommended that you create your own _layouts page for your own settings.
 - It is also recommended that you use this codeplex tool to support this method <http://pbs2010.codeplex.com/>
 - Lists
 - This is not a recommended option for Farm or Web Application level configuration data.
 - It is also recommended that you use this codeplex tool to support this method <http://spconfigstore.codeplex.com/>
 - Hierarchical Object Store (HOS) or SPPersistedObject
 - Ensure that any trees or relationships you create are clearly documented.

- It is also recommended that you use the Manage Hierarchical Object Store feature at <http://features.codeplex.com/>. This feature only stores values in the Web Application. You can build a hierarchy of persisted objects but these objects don't necessarily map to SPSites and SPWebs.

References:

- [SharePoint Products and Technologies Customization Policies](#)
- [Developing Applications for SharePoint 2010](#)
- [Creative SharePoint's SharePoint Development Standards](#)
- [Johan's Suggested SharePoint Development Standards](#)
- [Working with Large Lists](#)
- [Common Coding Issues When Using the SharePoint Object Model](#)
- [Using Disposable Windows SharePoint Service Objects](#)
- [Approaches to Creating MasterPages and Page Layouts in SharePoint 2007](#)
- [Implementing SharePoint 2007 and WSS 3 Solutions](#)