Modifier L Apparence Des Sites Sharepoint 2013 En Academic PDF

modifier l apparence des sites sharepoint 2013 en est une étape essentielle pour personnaliser l?expérience utilisateur et renforcer l?identité visuelle de votre organisation. SharePoint 2013 est une plateforme puissante qui permet la gestion collaborative, le partage de documents et la création de sites d?entreprise. Cependant, par défaut, ses thèmes et ses modèles de pages peuvent ne pas correspondre à l?image de marque de votre société ou aux besoins spécifiques de votre équipe. La capacité à modifier l?apparence des sites SharePoint 2013 permet ainsi d?améliorer l?ergonomie, la navigation, et l?esthétique globale, rendant votre environnement plus attrayant et adapté à vos objectifs.

Dans cet article, nous explorerons en détail comment modifier l?apparence des sites SharePoint 2013, en abordant les différentes méthodes, outils, et meilleures pratiques pour une personnalisation efficace. Que vous soyez administrateur SharePoint, développeur ou utilisateur avancé, vous trouverez ici des conseils pratiques pour transformer vos sites SharePoint 2013 en espaces modernes, cohérents avec votre charte graphique.

Comprendre la personnalisation de l?apparence dans SharePoint 2013

Avant d?entrer dans le vif du sujet, il est important de saisir les différentes options qui s?offrent pour modifier l?apparence de vos sites SharePoint 2013.

Les thèmes et styles prédéfinis

SharePoint 2013 propose plusieurs thèmes et styles prédéfinis permettant de changer rapidement l?aspect visuel d?un site. Ces thèmes incluent des variations de couleurs, polices et arrière-plans qui peuvent être appliqués directement via l?interface d?administration.

Les pages maîtres (Master Pages)

Les pages maîtres constituent l?ossature de toutes les pages SharePoint. En modifiant la page maître, vous pouvez changer la structure, la navigation, et le design global du site.

Les feuilles de style CSS personnalisées

L?utilisation de CSS permet d?affiner le design en appliquant des styles spécifiques à certains

éléments du site. Cela requiert une connaissance en développement web, mais offre une flexibilité maximale.

Les web parts et modèles de page

Les web parts sont des composants que vous pouvez ajouter ou configurer pour personnaliser une page. Modifier leur apparence ou leur disposition contribue également à modifier l?aspect global du site.

Méthodes pour modifier l?apparence des sites SharePoint 2013

Plusieurs méthodes existent pour personnaliser l?aspect de vos sites SharePoint 2013. Voici une présentation détaillée des principales techniques.

1. Utiliser les thèmes intégrés

Les thèmes intégrés sont la méthode la plus simple pour une personnalisation rapide.

- Accéder à la galerie de thèmes via l?interface d?administration SharePoint.
- Choisir un thème existant ou créer un nouveau thème personnalisé.
- Appliquer le thème au site pour voir instantanément le changement.

Cette méthode convient pour des modifications légères et rapides, sans nécessiter de compétences en développement.

2. Modifier la page maître (Master Page)

Pour une personnalisation avancée, la modification de la page maître est incontournable.

- Exporter la page maître existante : utiliser SharePoint Designer ou PowerShell.
- Modifier le fichier HTML de la page maître pour changer la structure et le style.
- Réimporter la nouvelle page maître dans SharePoint et l?associer au site.

Conseils importants :

- Toujours faire une copie de sauvegarde avant modification.
- Utiliser SharePoint Designer pour une édition visuelle ou en mode code.
- Respecter les bonnes pratiques pour éviter les erreurs de rendu.

3. Appliquer des CSS personnalisés

Pour un contrôle précis du design, l?ajout de CSS personnalisé est idéal.

- Créer une feuille de style CSS contenant les règles de design souhaitées.
- Intégrer cette feuille de style dans la page maître ou via la personnalisation des pages.
- Utiliser la balise pour référencer votre CSS personnalisé.

Note importante : Il est conseillé de réduire l?impact sur la maintenance en utilisant des fichiers CSS séparés et en documentant vos modifications.

4. Modifier les web parts et leur apparence

Les web parts offrent une flexibilité pour modifier la présentation du contenu.

- Personnaliser le style via leurs propriétés.
- Utiliser des web parts de contenu WebPart pour ajouter des éléments visuels.
- Créer des modèles de web parts pour réutiliser des configurations spécifiques.

Utiliser SharePoint Designer pour une personnalisation approfondie

SharePoint Designer est un outil puissant pour modifier la structure, le style, et le comportement des sites SharePoint 2013.

Étapes pour personnaliser avec SharePoint Designer

- Ouvrir SharePoint Designer et se connecter à votre site.
- Naviguer jusqu?à la section ?Pages Maîtres? pour modifier la page maître.
- Modifier le fichier HTML ou ASPX pour changer la mise en page ou ajouter du CSS.
- Personnaliser les pages de site, les éléments WebPart, et les styles.
- Sauvegarder et publier les modifications.

Avantages :

- Contrôle total sur la mise en page.
- Possibilité d?intégrer du code HTML, CSS et JavaScript.
- Gestion centralisée des modifications.

Inconvénients :

- Nécessite une certaine maîtrise technique.
- Risque de casser la mise en page si mal configuré.

Meilleures pratiques pour une personnalisation réussie

Pour assurer une personnalisation efficace et maintenable, voici quelques conseils clés.

1. Planifier avant de modifier

- Définir clairement vos objectifs de design.
- Identifier si une simple application de thème suffira ou si une personnalisation avancée est nécessaire.

2. Toujours sauvegarder et documenter

- Créer des sauvegardes avant toute modification.
- Documenter chaque étape pour faciliter la maintenance.

3. Utiliser des solutions évolutives

- Préférer la personnalisation via des fichiers CSS ou des pages maîtres plutôt que des modifications directes dans SharePoint Designer.
- Créer des modèles réutilisables pour uniformiser l?apparence.

4. Tester dans un environnement de développement

- Avant de déployer en production, tester toutes les modifications dans un environnement de test pour éviter les interruptions.

5. Respecter la charte graphique

- S?assurer que les couleurs, polices et éléments visuels respectent l?identité de votre marque.

Conclusion

Modifier l?apparence des sites SharePoint 2013 est une étape essentielle pour créer un environnement à la fois esthétique, fonctionnel et cohérent avec l?image de votre organisation. Que vous choisissiez des méthodes simples comme l?application de thèmes ou des techniques avancées telles que la modification de pages maîtres et l?ajout de CSS personnalisé, chaque approche offre des avantages adaptés à différents niveaux de compétence et besoins.

En respectant les bonnes pratiques, en planifiant soigneusement vos modifications, et en utilisant les outils appropriés comme SharePoint Designer, vous pourrez transformer vos sites SharePoint 2013 pour mieux répondre à vos attentes en termes de design, d?ergonomie, et d?expérience utilisateur. La personnalisation de l?apparence n?est pas seulement une question d?esthétique, c?est aussi une stratégie pour améliorer l?engagement et la productivité de vos collaborateurs.

N?hésitez pas à explorer toutes ces options pour faire de votre site SharePoint 2013 un espace unique, professionnel et aligné avec votre identité visuelle.

Modifier l'apparence des sites SharePoint 2013 en est une démarche essentielle pour les entreprises et les organisations souhaitant personnaliser leur environnement collaboratif afin qu'il reflète leur identité visuelle, facilite la navigation, et améliore l'expérience utilisateur. SharePoint 2013, bien qu'étant une plateforme robuste pour la gestion de contenu, la collaboration et la création de portails d'entreprise, offre également une flexibilité considérable en matière de design et de personnalisation. Dans cet article, nous explorons en détail les différentes méthodes et outils permettant de modifier l?apparence des sites SharePoint 2013, en fournissant une analyse approfondie pour aider les administrateurs et développeurs à tirer le meilleur parti de cette plateforme.

Introduction à la personnalisation de l?apparence dans SharePoint 2013

SharePoint 2013 est une plateforme qui, dès sa sortie, a mis l?accent sur une meilleure expérience utilisateur et une personnalisation accrue. La possibilité de modifier l?apparence des sites est devenue une nécessité pour aligner l?interface utilisateur avec la charte graphique de l?organisation, mais aussi pour améliorer la convivialité et l?engagement des utilisateurs. La personnalisation ne se limite pas à un simple changement de couleurs ou de logos ; elle englobe aussi la structuration de la navigation, le design des pages, et même la modification du comportement des composants.

Les principales stratégies pour modifier l'apparence des sites SharePoint 2013 incluent :

- La personnalisation via les thèmes et master pages
- L?utilisation de CSS et JavaScript
- La modification des pages via SharePoint Designer
- La création de composants Web Part customisés
- La conception responsive pour une adaptation mobile

Chacune de ces approches offre différents niveaux de contrôle, de complexité, et de maintenabilité, que nous allons explorer en détail.

Comprendre la structure de l?apparence dans SharePoint 2013

Avant de plonger dans les méthodes de personnalisation, il est crucial de comprendre la structure de l?apparence dans SharePoint 2013.

Les master pages

Les master pages constituent la base de l?apparence de tout site SharePoint. Elles définissent la structure globale, notamment l?en-tête, le pied de page, la zone de navigation, et le rendu général. En modifiant la master page, on peut changer l?aspect visuel de l?ensemble du site ou de ses sous-sites.

Les thèmes

Les thèmes permettent de modifier rapidement la palette de couleurs, les styles de texte, et les images de fond. SharePoint 2013 offre une gestion intégrée des thèmes, facilitant ainsi une personnalisation cohérente et simple.

Les CSS et JavaScript

Les feuilles de style CSS et les scripts JavaScript permettent d?affiner le design et d?ajouter des fonctionnalités interactives. Leur utilisation offre une flexibilité importante, mais nécessite des compétences en développement web.

Les pages et composants Web Part

Les pages de site contiennent des composants Web Part, qui peuvent être configurés ou remplacés pour modifier la présentation du contenu.

Personnalisation via les thèmes et master pages

L?approche la plus fondamentale pour modifier l?apparence consiste à travailler avec les master

pages et thèmes. Voici un aperçu détaillé de ces techniques.

Utilisation des master pages

Les master pages dans SharePoint 2013 sont des fichiers ASP.NET qui contrôlent la structure globale. La master page par défaut, appelée `seattle.master`, peut être copiée, modifiée ou remplacée pour créer une apparence personnalisée.

Étapes clés :

- Création d?une copie de la master page existante : en utilisant SharePoint Designer ou Visual Studio.
- Modification du fichier : en ajustant le HTML, le CSS intégré, ou en incorporant des contrôles ASP.NET.
- Déploiement personnalisé : en enregistrant la nouvelle master page dans la galerie de styles ou en la définissant comme master page par défaut dans les paramètres du site.

Points importants :

- La modification directe de la master page doit être faite avec précaution pour éviter de briser la structure.
- Il est conseillé de créer une copie de sauvegarde.
- La compatibilité avec les futures mises à jour doit être prise en compte.

Gestion des thèmes

Les thèmes dans SharePoint 2013 permettent une personnalisation rapide sans toucher à la structure globale. SharePoint propose un générateur de thèmes intégré :

- Accédez à la galerie de thèmes via `Paramètres du site > Apparence`.
- Créez ou modifiez un thème existant en utilisant l?outil de colorisation.
- Appliquez le thème à un site ou à une collection de sites.

Les thèmes peuvent aussi être créés manuellement en utilisant l?outil `PowerPoint` ou en éditant directement le fichier `.spcolor` et `.spfont`.

Personnalisation avancée avec CSS et JavaScript

Pour des modifications plus fines ou spécifiques, l?utilisation de CSS et de JavaScript est essentielle.

Intégration de CSS personnalisé

- Méthodes d?intégration :
- Ajout dans la master page : insérer une balise `` ou `

Sharepoint 2010 Development With Visual Studio 201

SharePoint 2010 Development with Visual Studio 2010

SharePoint 2010 development with Visual Studio 2010 is a powerful combination that enables developers to create customized solutions, automate business processes, and extend the capabilities of SharePoint environments. This integration provides a robust platform for building, deploying, and managing SharePoint applications efficiently, leveraging the familiar development tools and features of Visual Studio 2010.

In this comprehensive guide, we'll explore the essentials of SharePoint 2010 development with Visual Studio 2010, covering setup, key development approaches, best practices, and tips to optimize your development workflow.

Understanding SharePoint 2010 and Visual Studio 2010 Integration

SharePoint 2010 is a feature-rich collaboration platform that offers document management, enterprise content management, business process automation, and social computing features. Visual Studio 2010 is a versatile integrated development environment (IDE) that supports SharePoint development through specialized project templates and tools.

This integration allows developers to:

- Create SharePoint solutions such as Web Parts, Event Receivers, Workflow extensions, and Sandboxed Solutions.
- Automate deployment and configuration tasks.
- Debug SharePoint components directly within Visual Studio.
- Manage SharePoint artifacts more effectively.

Setting Up the Development Environment

Before diving into development, ensure your environment is properly configured.

Prerequisites

- SharePoint 2010 installed on your development machine or a dedicated development environment.
- Visual Studio 2010 with the SharePoint development tools installed.
- Microsoft SharePoint SDK compatible with SharePoint 2010, which includes project templates and libraries.
- Administrative privileges on the development machine.

Installing SharePoint 2010 Development Tools

- Install Visual Studio 2010 if not already installed.
- Run the SharePoint 2010 SDK setup to add SharePoint-specific project templates.
- Ensure SharePoint is configured for development, including setting up a developer site or sandbox environment.

Creating Your First SharePoint 2010 Project in Visual Studio 2010

To develop SharePoint solutions, follow these steps:

- Open Visual Studio 2010.
- Go to **File > New > Project**.
- Under **Installed Templates**, select **SharePoint**.
- Choose a project template such as **Blank SharePoint Solution** or a specific component like **Web Part**.
- Name your project and specify the location.
- Click **Create**.

This setup creates a solution structure with predefined folders and project files, ready for customization.

Developing SharePoint Components with Visual Studio 2010

SharePoint 2010 supports various development artifacts. Here are some common components you

can build:

Web Parts

Web Parts are modular units of information that users can add to SharePoint pages.

- To create a Web Part, add a new item to your project: **Web Part**.
- Customize the Web Part code by overriding the *Render* or *CreateChildControls* methods.
- Use Visual Studio's designer tools for layout and properties.

Event Receivers

Event Receivers respond to specific SharePoint list or library events, such as item added or deleted.

- Add a new item: **Event Receiver**.
- Select the list or library type.
- Write code to handle events, such as validation, logging, or custom workflows.

Workflow Extensions

Extend SharePoint workflows with custom code.

- Use the Workflow project template.
- Implement activity logic and integrate with SharePoint workflows.

Sandboxed Solutions

Deploy lightweight solutions within SharePoint's sandbox environment.

- Add a new sandboxed solution project.
- Package features and deploy them via Visual Studio or SharePoint Central Administration.

Debugging SharePoint 2010 Solutions

Debugging is critical for efficient development. Visual Studio 2010 offers seamless debugging capabilities.

- Attach the debugger to the SharePoint process (usually *OWSTIMER.EXE* or *W3WP.EXE*).
- Set breakpoints in your code.
- Deploy your solution in debug mode.
- Test functionality directly within SharePoint pages.

Deploying SharePoint 2010 Solutions

Deployment involves packaging your solutions and deploying them to the SharePoint farm.

- Package your solution as a WSP file (SharePoint Solution Package).
- Deploy using Visual Studio?s deployment tools, PowerShell scripts, or SharePoint Central Administration.
- Activate features or solutions as needed.

Best practices include testing in a development environment before moving to production and documenting deployment steps.

Best Practices for SharePoint 2010 Development

- Design for maintainability: Use clear naming conventions and modular designs.
- Follow SharePoint development guidelines: Avoid direct database modifications; use SharePoint APIs.
- Leverage features and solutions: Use SharePoint features to enable easy deployment and management.
- Test thoroughly: Use multiple environments to test solutions before production.
- Document your code and deployment process.

Common Challenges and Troubleshooting Tips

- Deployment issues: Ensure your solution is properly packaged and permissions are correct.
- Compatibility problems: Verify your development environment matches the SharePoint farm version.
- Debugging difficulties: Attach Visual Studio debugger to the correct SharePoint process and check for conflicts.
- Performance concerns: Optimize code for efficiency and minimize server load.

Conclusion

SharePoint 2010 development with Visual Studio 2010 empowers developers to create robust, scalable, and customized solutions tailored to organizational needs. By understanding the integration, setting up the environment correctly, and following best practices, you can streamline your development process and deliver high-quality SharePoint applications.

Whether building Web Parts, event receivers, workflows, or sandboxed solutions, Visual Studio 2010 provides the tools necessary for efficient development. Remember to keep abreast of SharePoint development standards and continuously test and optimize your solutions for best results.

Embark on your SharePoint development journey today by leveraging the powerful synergy of SharePoint 2010 and Visual Studio 2010, transforming your collaboration platform into a customized enterprise solution.

SharePoint 2010 Development with Visual Studio 2010 is a powerful combination that empowers developers to create robust, scalable, and customized solutions for enterprise collaboration, document management, and intranet portals. Leveraging the capabilities of Visual Studio 2010, developers can streamline their development process, leverage rich tooling, and adhere to best practices for SharePoint customization and extension. This guide provides a comprehensive overview of how to approach SharePoint 2010 development using Visual Studio 2010, exploring key concepts, tools, and best practices to help you build effective SharePoint solutions.

Introduction to SharePoint 2010 Development

SharePoint 2010 is a mature platform that enables organizations to build collaborative environments. Its architecture allows for extensive customization through development of custom features, web parts, workflows, event receivers, and more. Visual Studio 2010, as the primary development environment for SharePoint 2010, offers a specialized set of tools and templates designed specifically for SharePoint development tasks.

Why Use Visual Studio 2010 for SharePoint 2010 Development?

- Rich Development Environment: IntelliSense, debugging, and project management tailored for SharePoint solutions.
- Template-Based Projects: Easy creation of SharePoint-specific projects like farm solutions and sandboxed solutions.
- Deployment and Packaging: Built-in support for packaging solutions into SharePoint solution packages (.wsp files).
- Integrated Debugging: Debug SharePoint code directly within Visual Studio, streamlining testing and troubleshooting.

- Version Control: Compatibility with source control systems to manage complex SharePoint projects.

Setting Up Your Development Environment

Before diving into development, ensure your environment is properly configured.

Prerequisites

- Visual Studio 2010: The primary IDE for SharePoint 2010 development.
- SharePoint 2010 SDK: Provides templates, libraries, and documentation.
- SharePoint 2010 Server or Developer Farm: A development environment with SharePoint installed.
- Microsoft Office SharePoint Designer 2010 (optional): For rapid prototyping and design tasks.
- Additional Tools: SharePoint Solution Generator, PowerShell, and other command-line tools as needed.

Installing Necessary Components

- Install Visual Studio 2010 Professional, Premium, or Ultimate.
- Install the SharePoint 2010 SDK, available from Microsoft.
- Configure your environment to develop on a SharePoint development farm (recommended) or sandboxed environment.
- Set up necessary permissions for deploying solutions.

Understanding SharePoint Solution Types

SharePoint development primarily involves creating solutions that can be deployed to a SharePoint farm or site.

Farm Solutions

- Scope: Entire farm; can include features, Web Parts, event receivers.
- Deployment: Fully trusted; requires farm administrator privileges.
- Use Cases: Customizations that require full trust, such as custom server-side code.

Sandboxed Solutions

- Scope: Site collection; limited to the site collection.
- Deployment: Limited trust; safer for shared hosting environments.
- Use Cases: Lightweight customizations, such as simple Web Parts or list definitions.

Developing SharePoint Solutions with Visual Studio 2010

Creating a New SharePoint Project

- Launch Visual Studio 2010.
- Select File > New > Project.
- Under Installed Templates, choose SharePoint.
- Pick either Empty SharePoint Project or use predefined templates for Web Parts, Event Receivers, etc.
- Specify the target SharePoint version (2010) and the deployment location.

Configuring the Project

- Solution Name: Name your solution meaningfully.
- Deployment Type: Decide between farm solution or sandboxed solution.
- Local SharePoint Site: Specify the site collection where the solution will be deployed during development.

Adding Features and Components

Visual Studio provides templates and wizards for adding various components:

- Web Parts: Custom UI components for pages.
- Event Receivers: Handle list or site events.
- Workflow Activities: Automate business processes.
- Content Types & List Definitions: Extend SharePoint content models.
- Custom Actions & Ribbon Extenders: Enhance UI.

Building Common SharePoint Components

Web Parts Development

Web Parts are the building blocks for customizing SharePoint pages.

- Use the Web Part template.
- Implement the `WebPart` class and override `CreateChildControls()`.
- Use SharePoint's server-side object model to interact with lists, libraries, and data.

Event Receivers

Event receivers allow you to respond to list or site events.

- Choose Event Receiver template.
- Specify the event type (e.g., ItemAdding, ItemUpdated).
- Use the SharePoint object model to implement custom logic.

Workflow Integration

- Use Visual Studio to create SharePoint Designer workflows or custom workflow activities.
- Automate approval processes, notifications, or data processing.

Deploying and Testing Your Solutions

Packaging the Solution

- Visual Studio generates a `.wsp` file, the SharePoint solution package.
- Use the Solution Explorer to manage features, assemblies, and dependencies.

Deploying the Solution

- Debugging in Visual Studio: Attach the debugger to the SharePoint process (`OWSTIMER.EXE` or `STSADM.EXE`).
- Use SharePoint Solution Installer within Visual Studio or PowerShell commands like `Add-SPSolution` and `Install-SPSolution`.

Testing

- Deploy to your development farm.
- Activate features at the site or site collection level.
- Access the deployed web parts or features via SharePoint UI.

- Use Visual Studio's debugging tools to troubleshoot.

Best Practices and Tips for SharePoint 2010 Development

Code Organization

- Modularize your code into reusable components.
- Use namespaces and proper folder structures.

Security and Trust

- Understand the difference between farm and sandboxed solutions.
- Minimize server-side code in sandboxed solutions to maintain safety.

Performance Considerations

- Cache data where appropriate.
- Avoid excessive server calls.
- Use asynchronous processing for heavy tasks.

Versioning and Deployment

- Version your solutions properly.
- Use SharePoint's deployment pipeline to manage updates.

Documentation and Maintenance

- Comment your code thoroughly.
- Maintain a changelog.
- Document custom features and configurations.

Conclusion

SharePoint 2010 Development with Visual Studio 2010 offers a comprehensive environment for creating tailored enterprise solutions. By understanding the architecture, leveraging Visual Studio?s specialized tooling, and following best practices, developers can build powerful, maintainable, and

scalable SharePoint customizations. Whether creating custom web parts, event receivers, workflows, or content types, the combination of SharePoint 2010 and Visual Studio 2010 remains a potent platform for enterprise collaboration solutions. As you grow more familiar with the development lifecycle, from project creation to deployment and maintenance, you'll be well-equipped to harness the full potential of SharePoint 2010.

QuestionAnswer How can I create a SharePoint 2010 solution using Visual Studio 2010? To create a SharePoint 2010 solution in Visual Studio 2010, install the SharePoint Developer Tools, then select 'SharePoint 2010 - Empty SharePoint Project' from the project templates. Configure the project settings, add necessary features or web parts, and deploy directly from Visual Studio. What are the best practices for developing SharePoint 2010 web parts in Visual Studio? Best practices include modular design, using SharePoint's server-side object model correctly, leveraging feature-based deployment, maintaining code cleanliness, and testing web parts in a local SharePoint environment before deployment. How do I deploy SharePoint 2010 solutions from Visual Studio 2010? Set your deployment options in the project properties, then right-click the project and select 'Deploy'. Visual Studio will package the solution, deploy it to the SharePoint farm, and activate the features as configured. Can I develop SharePoint 2010 workflows using Visual Studio 2010? Yes, Visual Studio 2010 supports workflow development for SharePoint 2010 through Workflow Foundation. You can create declarative or code-behind workflows, design them visually, and deploy them directly to SharePoint 2010. What are common troubleshooting tips when developing SharePoint 2010 solutions in Visual Studio? Common tips include ensuring the correct SharePoint SDK is installed, verifying that the solution is properly deployed and activated, checking for compatibility issues, reviewing ULS logs for errors, and testing in a clean SharePoint environment to isolate problems.

Related keywords: SharePoint 2010, Visual Studio 2010, SharePoint development, SharePoint solutions, SharePoint web parts, SharePoint features, SharePoint deployment, SharePoint workflow, SharePoint API, SharePoint customizations

Read the full article online: SHAREPOINT 2010 DEVELOPMENT WITH VISUAL STUDIO 201