Disclaimer

THE INFORMATION CONTAINED IN THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT ANY EXPRESS REPRESENTATIONS OF WARRANTIES. IN ADDITION, INFRAGISTICS, INC. DISCLAIMS ALL IMPLIED REPRESENTATIONS AND WARRANTIES, INCLUDING ANY WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

SharePlus™ 4.0 - Administrator Guide 1.2.1

All text and figures included in this publication are the exclusive property of Infragistics, Inc., and may not be copied, reproduced, or used in any way without the express permission in writing of Infragistics, Inc. Information in this document is subject to change without notice and does not represent a commitment on the part of Infragistics, Inc. may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents except as expressly provided in any written license agreement from Infragistics, Inc.

Infragistics, Inc. and SharePlus are trademarks of Infragistics in the United States and/or other countries.

This document also contains registered trademarks, trademarks and service marks that are owned by their respective owners. Infragistics, Inc. disclaims any responsibility for specifying marks that are owned by their respective companies or organizations.

©2014 Infragistics, Inc. All rights reserved.
# Table of Contents

Disclaimer ................................................................................................................................................. 2
Table of Contents ...................................................................................................................................... 3
Introduction ................................................................................................................................................ 4
How to Use this Guide ................................................................................................................................ 5
What's in this Guide ...................................................................................................................................... 6
Chapter 1 Introducing SharePlus .............................................................................................................. 7
  Welcome to SharePlus! .............................................................................................................................. 8
  Functional Overview .................................................................................................................................. 9
Chapter 2 Planning SharePlus .................................................................................................................. 34
  Solution Architecture .............................................................................................................................. 35
  Planning User Experience ......................................................................................................................... 37
  Planning SharePlus Licensing ................................................................................................................... 39
  Planning Configuration ............................................................................................................................... 40
  Planning Deployment ................................................................................................................................. 48
  Planning Connectivity ............................................................................................................................... 50
  Planning Security ...................................................................................................................................... 54
Chapter 3 Configuration .......................................................................................................................... 63
  Central Application Configuration ............................................................................................................ 64
  Site Configuration a.k.a “MobileNavigation” .......................................................................................... 86
  Server Configuration .................................................................................................................................. 93
Chapter 4 Deployment .............................................................................................................................. 96
  Enterprise Deployment ............................................................................................................................ 97
  Application Re-signing ............................................................................................................................. 114
Chapter 5 Advanced Configuration ......................................................................................................... 135
  User Experience Customization ............................................................................................................. 136
  Offline Support ......................................................................................................................................... 155
  Support Enablement .................................................................................................................................. 160
Chapter 6 Advanced Features ................................................................................................................ 165
  Social Networking ..................................................................................................................................... 166
  ReportPlus Integration ............................................................................................................................. 172
  Search ....................................................................................................................................................... 186
  Application Integration .............................................................................................................................. 189
Appendices .................................................................................................................................................. 199
  Appendix 1: Configuration File Reference .............................................................................................. 200
  Appendix 2: Document Changelog .......................................................................................................... 225
Introduction

Welcome to the SharePlus Administrator’s Guide. This guide's purpose is to assist you with every aspect of a SharePlus Enterprise solution, empowering you to do the planning, deployment, and complete administration of your solution. SharePlus has been crafted towards both small and large scale organizations, therefore, administrators need to comprehend the wide number of scenarios available that build upon all SharePlus functionality.

Audience

Despite the title, this documentation is not directed to administrators only, it is recommended for project managers, business analysts, developers, and also power users who want to understand more about SharePlus administration, design, and configuration.

**Power users** – Will learn more about planning SharePlus functionalities, configuration details, and also some advanced features like Search, Social Networking, and Offline Support.

**Project managers** – Will find key information from the Planning SharePlus chapter very useful when working with SharePlus projects.

**Developers** – Will have a detailed view of all SharePlus functionality, implementation architecture, configuration and deployment.

**Business Analysts** – As project managers, they will find the Planning SharePlus chapter very handy.

About SharePlus

SharePlus is a native mobile application that provides online and offline, read/write access to SharePoint content like libraries and lists. SharePlus Free and Subscription, available in the AppStore, are consumer-oriented applications that meet the needs and interests of individuals. SharePlus Enterprise, on the contrary, can be used as a base to build complex Business Enterprise Solutions that assemble aspects like connectivity, security, deployment, and configuration.
How to Use this Guide

This guide was formatted with the reader in mind, including not only illustrative images and diagrams but also elements like notes and links, in order to highlight/redirect to relevant information.

**Note:** Notes include information that needs to be highlighted, and sometimes tips for the reader.

<table>
<thead>
<tr>
<th>About Tables</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Importance</td>
<td>Tables add value for the user by presenting complex data in a user-friendly and more readable format.</td>
</tr>
</tbody>
</table>

Gesture icons provide a close-to-reality representation for applications with touch-based UI.
What’s in this Guide
SharePlus Administrator’s Guide is divided into 4 different chapters:

**Chapter 1 – Introducing SharePlus**
This chapter introduces SharePlus and gives a general overview of the application functionality.

**Chapter 2 – Planning SharePlus**
This chapter helps administrators understand the architecture of a SharePlus solution and plan the different aspects of it, including connectivity, security, user experience, configuration and deployment.

**Chapter 3 – Configuration**
The Configuration chapter gives detailed information about the configuration components available on SharePlus and how to adjust them to tailor the application’s behavior.

**Chapter 4 – Deployment**
This chapter details the available methods used to distribute SharePlus Enterprise, including the Web, Ad-hoc, and MDM-based deployment. The application re-signing process is also included in this chapter, as a reference to be used when the application is distributed through the Web or Ad-hoc deployment methods.

**Chapter 5 – Advanced Configuration**
The “Advanced Features” chapter covers advanced configuration aspects including UX Customization, Offline Support, and Support Enablement.

**Chapter 6 – Advanced Features**
This chapter covers advanced application features including ReportPlus Integration, Social Networking, Search capabilities, and MDM Integration.

**Appendices**
The appendices included in this final chapter provide further information about how to create your IG Account and request priority support. A complete reference to the Configuration File’s contents is also included here.
Chapter 1
Introducing SharePlus

Section 1: Welcome to SharePlus
Section 2: Functional Overview
Welcome to SharePlus!

SharePlus is a native mobile application that provides online and offline, read/write access to SharePoint content like libraries, lists, and social features. SharePlus offline capabilities allows you to continue working with your SharePoint content even while offline. Changes introduced while offline are synchronized automatically with the server when the devices goes back online.

Create amazing charts and pivot tables from SharePoint Lists and Excel spreadsheets, or find the experts on your network using People Search and discuss your insights by sharing the charts. Be in sync with your team with by sharing Calendars, Task Lists, Contacts, and Issues, Discussion Boards and Announcements and storing content in MySite/OneDrive for instant sharing.

SharePlus is available for iOS and Android platforms and is offered in three different license models:

- Free
- Subscription
- Enterprise

Note: This document refers to the SharePlus for iOS only. For other mobile platforms, please refer to the corresponding documentation.
# Functional Overview

## Mobile Collaboration

<table>
<thead>
<tr>
<th>Feature</th>
<th>FREE</th>
<th>SUBSCRIPTION</th>
<th>ENTERPRISE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Browse SharePoint Content (online &amp; offline)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Navigate SharePoint sites, lists/libraries and its contents</td>
<td>Restricted</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Embedded Document Previewer</td>
<td>Restricted</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>View in Web</td>
<td>Blocked</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>SharePoint List Views Support</td>
<td>Restricted</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>External Lists/Content Types Support</td>
<td>Restricted</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Quick access to Content</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sidebar with access to Content Hubs</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Recent Content</td>
<td>Restricted</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Favorites</td>
<td>Blocked</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Recommended Content (SP 2013 only)</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Application Launchpads</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Custom Site Homes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Edit Content</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add &amp; delete list items and documents</td>
<td>Blocked</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Edit list items and document properties</td>
<td>Restricted</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>File editing via native 3rd party apps and Office Web Apps</td>
<td>Blocked</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Check In/Check Out and Approve/Reject Support</td>
<td>Restricted</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>“Save as Draft” (save to device) Support</td>
<td>Restricted</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>In-App PDF Annotation and PDF Forms (read only)</td>
<td>Restricted</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>SharePoint Document Versioning</td>
<td>Restricted</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Rich Text Editor for Rich Text Fields</td>
<td>Restricted</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Offline Synchronization with collision resolution</td>
<td>Blocked</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>InfoPath Forms</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Collaborate</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Send files by Mail (attachment and URL)</td>
<td>Restricted</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Add User Profiles and Contact Items to device Contacts</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Wi-Fi Sharing</td>
<td>Blocked</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Analytics on the Go</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Visualize SharePoint Content</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rich Visualizations (Grid, Calendar, Picture Thumbnails)</td>
<td>Restricted</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Optimized PDF Navigator</td>
<td>Restricted</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Search Hub</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integration with SharePoint Server Search Engine</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Unified Search for content in all connected portals</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>ReportPlus Integration</strong></td>
<td>No</td>
<td>Restricted</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Enterprise Ready</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Social Hub</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Newsfeeds (SP 2013) and change notifications (Alerts)</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>MySite/OneDrive Support</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>People navigation and search</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Browse Colleagues (SP 2010), Followers and Followees (SP 2013)</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Security</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data-at-Rest Encryption</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>HTTPS and VPN Support</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Feature</td>
<td>FREE</td>
<td>SUBSCRIPTION</td>
<td>ENTERPRISE</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>------</td>
<td>--------------</td>
<td>------------</td>
</tr>
<tr>
<td><strong>Enterprise Ready</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Authentication</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>App Passcode Lock</td>
<td>Blocked</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Native Support for Windows, Forms, Office 365, and Web Login</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Integrated Client-Side Certificates</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Custom Authentication methods</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Advanced Security</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Editor Whitelisting</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Passcode Fail handle (including Data Wipe)</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Authentication Time-Bomb</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Kiosk Mode</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Deployment</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>In-House Deployment (OTA w/Remote Update)</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>MDM Server Deployment</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Advanced Configuration</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Centralized Configuration</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Per-site Configuration (MobileNavigation)</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>In-App Support Tools (Feedback Email and Help URL)</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Language Support</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>English, Japanese, French, Spanish, Chinese, German, Russian, Italian</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>MDM Integration</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MobileIron Integration – AppConnect &amp; AppTunnel (Deployment, Central Configuration and Policies, Per-app VPN)</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Rebranding</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Corporate Branding</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Custom Development</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTML/JS SDK</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Native custom development</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Table Reference**

- **Yes** – Available
- **No** – Not available
- **Restricted** – Limited application behavior, e.g., only the first list items are shown.
- **Blocked** – Not available without Subscription or Enterprise.
**Browse SharePoint Content**

SharePlus provides an intuitive navigation of SharePoint content including sites, webs, lists, document libraries, documents and items.

**Navigating Sites**

The Sites Module gives you access to your SharePoint content. Pre-configured **Portals, Recent** sites and, when available, **Following** and **Recommended** sites are displayed.

The Sites that you are member of (Membership) and the ones that you have tagged for quick access (Favorites) can be accessed through the bottom bar.
Navigating Lists and Libraries

Tapping on a Portal or site, you can browse its sub-sites, webs, libraries, and lists. The content, displayed on the right pane, depends on the user permissions on the SharePoint server.

SharePoint items are displayed in a table, grid, or calendar view (according to the list type and settings). Items are paged to optimize network use and you can scroll up and down to look through the list.
Navigating Documents
SharePlus fully supports SharePoint content types, including SharePoint default types and custom content types you may create.
The properties from an item can be opened and they will be displayed organized in columns.
When the item is, or has attached, a document, tapping the item’s icon will bring up a document preview (the most common slideshow, text, spreadsheet, PDF, and multimedia formats are supported).

In case of SharePlus previewer does not supports a specific file type, it may be opened with any 3rd party app that supports it and uses Open In.

Previewing Files
SharePlus uses the native iOS File previewer to display files within the app. All file formats that can be previewed on iOS Safari, or the iOS Mail Client, can be displayed in SharePlus. Please refer to the Appendix for a detailed list of Supported File Formats.

“View in Web”
Besides SharePlus’ document previewer, the app includes a “view in web” feature that displays the resource as it would be shown in a web browser. This is especially useful to view a site or the home page of a sub-site, with its web parts, for example. Broadly speaking, any pure-web content will be better appreciated this way.

---

1 Visio files not supported. PowerPoint slides might not be fully respected.
SharePoint List Views Support

List Views are pre-defined configurations created on the server to organize and filter the items of a list, depending on the needs of different SharePoint groups of users. You are allowed to change the list view by selecting among the Views you have access to.

Quick Access to Content

SharePlus helps you access your content with little effort, by showing recent content, allowing the creation of favorites, and including a SideBar menu with access to content hubs.

Favorites

SharePlus contains a “Favorites” Module that gives you quick access to all your favorite content displayed in an organized list view by category.
To mark an item as Favorite, just tap the “Star” icon on its properties.

**Recommended Content**

When working with SharePoint 2013 server you can get recommendations on sites and documents that may be relevant to you. Suggestions are included on the SharePlus list view under the *Recommended* section.
Application Launchpads

By using web technologies like HTML, CSS and JavaScript, SharePlus can display rich launchpads to enhance the user experience. In addition, high quality interactions can be achieved by using CSS and JavaScript frameworks like JQuery. Besides that, SharePlus provides a JavaScript API that gives access to SharePlus data, giving you the ability to present dynamic content from the server, even while offline.

For further information about Application Launchpads refer to SharePlus Launchpads Developer Guide.
Custom Site Homes
Portals and sites can both display home content in SharePlus, presenting the user with a customized view for a given site. Site Homes can be pre-configured and shared across sites, displaying different content depending on the SharePoint site’s context.
Every site in SharePlus has a Site Home by default, which can be changed manually (through the UI) or by configuration settings.
You can configure different types of content to be displayed as Site Home, i.e., Application Launchpads, ReportPlus dashboards, and PDF files.

For further details about Site Homes configuration refer to Home Customization

Edit Content
On top of reading SharePoint data, SharePlus allows users to:

- Add, edit & delete list items, documents and its properties, stored in lists, document libraries and attached to list items.
- Send/receive documents to 3rd party apps for editing. Those apps must support incoming and outgoing Open In.

SharePlus centralizes all possible actions to be performed over any content, including sites, webs, lists, and items. Once the menu is displayed, scroll the list up and down to navigate through all the available actions for that type of content.
Perform a tap & hold to display the menu with all available actions.

**Editing Items**
SharePlus enables users to view and edit SharePoint List Items using native item forms.

Item forms are built dynamically, presenting both native and custom properties defined on the Item’s content type or columns definition. All SharePoint property types are supported and rendered using the corresponding iOS visual control.

**Editing Files**
SharePlus relies on invoking 3rd party apps to edit documents. Any 3rd party app that supports incoming and outgoing Open In can be used.

When working with Office file formats, Microsoft Office for iPad is strongly recommended as you can save your work straight to the SharePoint server. Office for iPad will ask you for SharePoint credentials to access the document. Once you provide them the document is changed directly on the server and it will be automatically updated in SharePlus.
Check In/Check Out and Approve/Reject support

Check-In and Check-Out actions are supported on SharePlus. The actions appear on the document actions menu when available to the current user. Approving and rejecting documents is only available on content-approval-enabled libraries for users with the “Approver” role on SharePoint.

“Save as Draft”

SharePlus features two save options: Save and Save as Draft. The Save as Draft option is useful when you want to work on a local copy without uploading changes to the SharePoint server.

Save as Draft  Save

In-App PDF Annotation and Form Filling

SharePlus provides a PDF annotation tool suite that lets the user annotate PDF documents and fill PDF forms without leaving the application. This maximizes productivity and keeps sensitive documents inside SharePlus.

The available tools are:

- Shapes: Square, circle, line, arrow and free drawing
- Markups: Highlighter, underline and strike
- Text box and Sticky Notes
- Stamps

Microsoft SharePoint® Online for Enterprises

Service Description
Document Versioning

SharePlus supports lists and document libraries with version control enabled. You can keep a history, track changes, and restore previous versions if needed.

Rich Text Editor

SharePlus features a fully capable Rich Text Editor which lets the user create rich content for Rich Text columns. The most common tools are available: font type face, font color, highlight color, bold, italic and underline, font size, paragraph alignment and insert picture. The editor also incorporates Undo and Redo support.

Besides being able to edit rich content, the user may also edit the HTML source of the content.
Managed Metadata Services
SharePlus also supports Managed Metadata Fields. Native iOS controls are used for these field types as well. When editing a managed metadata field, SharePlus will present the user with a pop-up dialog with search capabilities, as well as adding new tags and multiple selection.

Offline Synchronization with Conflict Resolution
Offline support is one of the most critical aspects of a mobile solution. SharePlus allows users to synchronize SharePoint content with the device, enabling offline reading and editing when not connected. Items can be added, edited and deleted while working offline. Changes are synchronized back with the server when the connection is restored.
Synchronization Monitoring
The “Synchronization” SideBar component allows you to access and monitor information when synchronizing your data. All errors and recent activity is displayed on the popover, as shown in the image below.

Collision Resolution
When synchronizing changes back to the server you may encounter conflicts, for example, when two or more users edit an item at the same time. SharePlus resolution engine helps you decide how you want to proceed in those cases.
**InfoPath Forms**
SharePlus supports displaying InfoPath forms and also adding new forms based on templates. Viewing and adding new forms require SharePlus to be online. No offline support is provided for form libraries. Your SharePoint server needs to have InfoPath Forms Services enabled and correctly configured for SharePlus to be able to render forms. For more information on how to configure it refer to the [Configure InfoPath Forms](#) TechNet article.

**Collaborate**
SharePlus enables users to be in sync with their teams by supporting all SharePoint native collaboration types, including Calendars, Task lists, Contact lists, Issue lists, Discussion Boards and Announcement lists.

**Sending Files by Mail**
SharePlus users can share documents by mail either attaching the file to the email or sharing the link to the document on SharePoint.

**Email Format Support**
MSG and EML file formats are fully supported in SharePlus. As long as the file is attached to an item or saved into a document library, SharePlus will be able to read the message along with its attachments.

**Adding to device Contacts**
Any phone number, email, and SharePoint contact can be imported into the iOS contacts app from SharePlus with any SharePoint contact list.

**Local Files**
The Documents module includes the “Local Files” section, which gives you access to an isolated and secure storage on your device. This storage is not synchronized with the server and can be used to store personal files. From Local Files you are able to upload files to the SharePoint server or share them via the Wi-Fi network.

**Wi-Fi Sharing**
Share files from Local Files with others through the Wi-Fi network. It is as simple as connecting your device to the network and enabling File Sharing in SharePlus. Once the address is displayed on screen, input that address in a web browser connected to the same network.
Visualize SharePoint Content

You can control every aspect on how SharePoint lists and document libraries are displayed by configuring the following settings:

- SharePlus Visualizations
- SharePoint Views
- Ad-Hoc Sorting & Grouping
Rich Data Visualizations

The layout of Lists and Document Libraries can be adjusted using Visualizations, e.g. table, grid, calendar view. The default and available Visualizations depend on the list type and settings, e.g., the Calendar compact Visualization is only available for Calendar lists.

Available visualizations:

- **Table view based**: displays the items with their icon and basic information based on the list view, in a vertical list.
- **Table standard**: items are displayed showing only their title, in a vertical list.
- **Grid view**: displays the items in a spreadsheet fashion, one line per item and one column per metadata field.
- **Calendar view**: this is a calendar-specific view (only supported for calendar lists) which supports daily, weekly and monthly views.
- **Picture Board**: this is a picture-specific view (only supported for picture libraries and asset libraries) that displays picture thumbnails over a dark background to provide a better image visualization.
Ad-Hoc Sorting & Grouping

Once you enter the sort and group settings, you will find there are different buttons and actions available. You can specify group by criteria by dragging List columns to the **Group By** section or sort criteria by dragging columns to the **Sort By** section.
Search Hub

Search is today an integral part of any SharePoint-related solution, enabling easy access to larger and larger volumes of information.

SharePlus Enterprise Search

The Search module allows you to search server content by keyword and retrieve results from all connected portals at the same time. Filters like scope, date, result type, and author help you narrow the retrieved results which are displayed in an organized list view.

You can use the search filters, refining the search by scope, date, result type, and author.

Filtering content inside lists and document libraries

SharePlus provides both basic and advanced list filters, based on complex criteria and defined by the user, using the current list’s available columns. Results will be displayed as items in the current list view and will be cleared once the search tool is dismissed.
ReportPlus Integration

Design and view dashboards with SharePoint Data
You can use data visualization to communicate insight by creating rich dashboards that consume SharePoint data.

Visualization options include: grid, text, trend line, map, gauge, chart, and financial chart views.

Working with Pivot Tables
You are able to connect to data sources like lists and documents, including calendars, contacts, images, Excel files. SharePoint lists and Excel files can apply many different visualization options to their data.
Pivot Tables are very powerful data summarization tools that help you analyze data. To create one is as simple as selecting your data and using the **Pivot Table Editor** to slice-and-dice data according to your needs.

You can summarize data, hide null values, add quick filters, and also customize the columns to be listed in your pivot table. For further details refer to the **ReportPlus Integration**

**Social Hub**

Collaboration is encouraged through SharePlus new **Social Module**, providing a centralized and feature-rich User Experience (UX) for **SharePoint 2013 social features**. The **Social Module** lets you share your ideas, as well as discover and keep track of your colleagues and their work through their profiles, posts and updates in their Timelines. Your SharePoint personal storage (My Content) is also available in this module.

**Timeline**

The **Timeline** tab includes your newsfeed, the place where you can access all your public posts, conversations, and updates. In this section, you can gain quick access to all the posts in which you’ve been mentioned, the items you have “liked”, and the posts of the people you follow.
My content and profile

The Social Module gives you access to your personal SharePoint storage (My Content), including both your public and private document libraries. Also, you can visualize or edit your personal information shared on your corporate social network.

People

The People tab displays the list of people that you are following and the ones following you. In this section you are able to connect with people, access their content and activity feed and also start conversations with them.
Security
SharePlus provides secure access to SharePoint sites from mobile devices. To fortify data security, core security features have been built into the product at each layer. The following is a list of security features that are included in SharePlus Enterprise for iOS devices:

- Storage Security (Data-at-Rest Security)
- Channel (Communication) Security
- Authentication
- Authorization
- Application-Level Security
- MDM Integration

Custom authentication methods can be implemented through the Objective-C SDK. For further details on every feature, refer to the Planning Security

Deployment
SharePlus licensing models (Free, Subscription, and Enterprise) are distributed using different methods. You can download the Free/Subscription versions directly from Apple’s App Store distribution platform. The Enterprise version can be distributed using the following methods:

- MDM Server Deployment
- Web Deployment using SharePoint
- Ad-Hoc (typically used for testing purposes)

For further details refer to the Planning Deployment

Advanced Configuration
SharePlus enables both end-users and IT Managers to adjust the application’s behavior using two main configuration components:

- Central Application Configuration
- Per-site Configuration (also known as “Mobile Navigation”)

Central Application Configuration
You can work with a Remote Configuration XML file published in a centralized location, on SharePoint or other server. The use of a remote configuration file, facilitates governance, allowing IT Managers to modify the default application settings on all devices centrally, without even having to push a new version of the product. SharePlus also supports working with multiple remote configuration files, by using Dynamic Configuration or Library-Based Configuration methods.

For further details refer to Local and Centralized Configuration
Per-site Configuration (MobileNavigation)
When browsing a site in SharePlus, all sites, lists, and libraries accessible to the user are displayed. SharePlus provides the Site Configuration method to manage the settings for sites and lists for each site independently. These settings include aspects like navigation, visualization, and offline behavior.

- **Navigation** is about the visibility and organization of sites and lists in the navigation bar.
- **Visualization** allows the configuration of the default SharePoint List View and SharePlus’ native visualization for each list. You can also configure the site Home to personalize the site gateway.
- **Offline behavior** allows the configuration of default offline settings for a list.

For further details refer to Site Configuration “a.k.a. MobileNavigation”

In-App Support Tools (Feedback Email and Help URL)
SharePlus provides several in-app tools to facilitate access to support to the user. These different tools can be customized separately and are centralized in the Help Center.

For further details refer to Support Enablement

Language Support
SharePlus will apply language and regional settings configured on the device. The available languages are English, Spanish, Chinese (Traditional and Simplified), French, German, Japanese and Russian, and Italian.
You can switch language or regional settings on the device and SharePlus will take settings into account. The localization to other languages can be custom implemented if needed.
MDM Integration
SharePlus integrates with many MDM solutions available in the market. MDM servers and SharePlus can work in conjunction to cover Application Deployment, Authentication/Authorization, Data Leakage Policies and Data Protection.

MobileIron Integration
SharePlus integrates with Apps@Work and Advanced Mobile Management functionality by supporting AppStorefront, AppTunnel, and AppConnect.

- **App Storefront** – Core module that allows the apps distribution and management.
- **AppConnect** – Protects data-at-rest through data encryption and provides an SDK for securing and managing apps.
- **AppTunnel** – Dedicated tunneling and access control module, build upon the MobileIron Sentry technology.

For further details refer to MDM Integration

Application Customization
SharePlus User Experience can be customized in several ways:

- Through application rebranding.
- Integrating web technologies (HTML, CSS, and JavaScript) into the native User Interface.
- With native custom development, implementing Objective-C code.

Corporate Branding
Enterprises often need a detailed control of the application’s look-and-feel to meet corporate policies or match concrete solution needs. SharePlus look-and-feel can be customized during implementation, by the inclusion of customized assets and also adjusting the application’s theme by configuration.

For further details refer to Application Rebranding

Web SDK
SharePlus can integrate modern standard client-side web technologies like HTML5 and jQuery, allowing the development of rich launchpads that can offer a customized view of a given site (Site Homes) or display content for a SideBar module (Application Home). These Application Launchpads can communicate with SharePlus using S+ links or through a JavaScript API.

For further information about the Web SDK refer to SharePlus Launchpads Developer Guide.

Native Custom Development
SharePlus can be further customized to adapt the product to specific business needs. With the Objective-C SDK you can integrate custom components into SharePlus, like custom actions on items, authentication providers, custom startup steps, etc.

Additional modules included on the SideBar can display custom content. These modules allow a wide range of possibilities.
Chapter 2
Planning SharePlus

Section 1: Solution Architecture
Section 2: Planning User Experience
Section 3: Planning SharePlus Licensing
Section 4: Planning Configuration
Section 5: Planning Deployment
Section 6: Planning Connectivity
Section 7: Planning Security
Solution Architecture

SharePlus Architecture Overview

1) The Device connects to the MDM Server, enrolls and downloads SharePlus.
2) SharePlus loads and downloads the centralized configuration.
3) When the user navigates a site, the native SharePoint Web Services are invoked and the content is loaded.
4) Traffic can tunneled via VPN and/or HTTP(S)
5) Authentication is relayed to AD, by the Reverse Proxy or by SharePoint itself.

SharePlus Client Application

SharePlus is a native iOS application. It manages a local database populated by a background Offline Synchronization process. Document editing is implemented invoking 3rd party editing tools via “Open In”.

Connectivity and Network Architectures

SharePlus connects to SharePoint through SharePoint’s out-of-the-box Web Services.

The network architecture needs to ensure connectivity by properly configuring:

- VPNs (or MDM-provided tunneling)
- Firewalls and Reverse Proxies
- SharePoint’s Alternate Access Mappings and IIS Hosts

You can test your network connectivity on SharePlus by trying the following:

- Open iOS Safari and browse the SharePoint site
- Download SharePlus Free
Security
SharePlus provides secure access to SharePoint sites by addressing aspects like:

- **Data Storage security** – iOS system architecture provides security features that are enabled by default.
- **Authentication** – The user authentication on a specific device can be achieved through the combination of several mechanisms like Passcode lock and Office 365 on the cloud.
- **Authorization** – Access to SharePoint resources such as lists and documents is granted through the user’s SharePoint credentials.
- **“Feature Trimming”** – Based on security rules, many restrictions over SharePlus functionality can be included.

MDM Servers
SharePlus can work in conjunction with an MDM Server to cover Application Deployment, Authentication/Authorization, Data Leakage Policies and Data Protection. The level of support provided differs for each MDM vendor. For further details refer to the MDM Integration chapter.

Customization
SharePlus can be customized to adapt the product to specific business needs. The main mechanisms include:

- **Configuration**: By adjusting the app’s configuration, you can tailor the application’s existing functionality, for instance, pre-setting the list of sites, defining default visualizations and trimming unwanted features. You can also adjust branding.
- **“HTML” SDK**: With the HTML SDK, you can create rich custom visual experiences that leverage all the back-end SharePlus functionality and run even in offline mode.
- **Objective-C SDK**: With the Objective-C SDK you can integrate custom components into the app, like custom visual modules, authentication providers or custom startup steps.

System Requirements
When working with SharePlus, you need to meet the following requirements:

<table>
<thead>
<tr>
<th>SharePoint Version</th>
<th>Mobile OS</th>
<th>iOS Devices</th>
</tr>
</thead>
<tbody>
<tr>
<td>SharePoint 2007 or above, both on-premises and Office 365 (Cloud).</td>
<td>iOS 6 or above</td>
<td>iPhone 3GS, iPad 2nd generation, iPod touch 4th generation or above.</td>
</tr>
</tbody>
</table>
Planning User Experience

Enterprises often need a detailed control of the application’s look-and-feel to meet corporate policies or match concrete solution needs. SharePlus look-and-feel can be customized by adjusting the configuration and graphic assets during implementation.

**Application Rebranding**

The following graphic elements can be customized to adjust the application’s visual design:

- Icons
- Splash Screen
- Color Palette

To include customized assets, the existing standard components need to be replaced when creating SharePlus IPA to be deployed. The application’s theme, or color palette, can also be adjusted through SharePlus configuration theme files.

**Localization**

SharePlus available languages are English, Spanish, Chinese (Traditional and Simplified), French, German, Japanese and Russian, and Italian. Localization to other languages can be custom implemented if required.
UI Customization

SharePlus User Interface can be customized to provide a more engaging, business-specific experience. You can customize the User Experience in the following aspects:

- The setup process
- The application’s Start and Resume from background
- All the application’s Visual Areas
  - **Sidebar Area**: The starting point of the application, the first area presented to the user. This is the left pane in the default layout.
  - **Main Screen Area**: Area where the main content is displayed.
  - **Detail Area**: Area where contextual information is displayed.
  - **Auxiliary Area**: Area where temporary or “out of context” content is presented.
- The available Home scenarios
  - **Sidebar custom modules**: When accessing the Application Home or other custom Sidebar modules.
  - **Portal Home**: When navigating to a portal.
  - **Site Home**: When navigating to a site.

All these customizations can be implemented in several ways:

- Creating HTML-based components, using Application Launchpads
- Creating rich components with charts and pivot tables, including ReportPlus dashboards
- Through custom Objective-C code
Planning SharePlus Licensing

Understanding SharePlus licensing
Every SharePlus application must have a license and every license file is created for a specific application’s bundle identifier\(^2\). SharePlus will not run in any device if the license file is missing or invalid. This means that even if the application bundle is modified and resigned in order to change a given configuration, the license specification cannot be modified.

The two license-related messages displayed by SharePlus application are:

![License Message](image)

**Note:** Unlike with devices, when running SharePlus in the iOS Simulator the license is not needed.

Asking for a License
License files are included in the application’s “.IPA” file that is provided by Infragistics. The generation of new licenses or any modification over an existing one requires Infragistics assistance on the matter.

License Capabilities
SharePlus license has many possibilities and allows you to block/disallow a feature in SharePlus. This capability is conceptually different from the Configuration File settings, which are used to configure the application and its features. By configuration, you can adjust feature settings and even disable the feature if needed. However, if the license is not allowing a feature to be used it will never be available in SharePlus.

License Deployment
The license file must be included in the application’s “.IPA” file provided by Infragistics. The application package may be modified and resigned to change specific configuration, however the license must remain untouched.

The application allows changing its license configuration dynamically by using the iOS Open In feature. In this case, the license file must be valid and provided by Infragistics.

\(^2\) The Application’s Bundle Identifier is the value that identifies your application within the device, e.g., `com.yourcompany.appname`. 
Planning Configuration

Understanding SharePlus Configuration

SharePlus enables both end-users and IT Managers to adjust the application’s behavior using two main configuration components: **Central Application Configuration** and **Site Configuration** (also known as “Mobile Navigation”).

**Central Application Configuration** includes the following areas:

- **Sites and User Accounts**: Define the SharePoint Sites the user will connect to and the credentials and authentication methods to be used.
- **Features**: Turn application Features On/Off and adjust their parameters.
- **Look and Feel**: Adjust the color and behavior of specific UI components.
- **Advanced Security**: Set up advanced security policies.
- **Deployment Settings**: Define and configure how the application will be distributed to users.
- **Global Settings**: Tweak low-level parameters like connection time outs, synchronization idle time, and the use of application logs.

**Site Configuration** (a.k.a “MobileNavigation”) includes all the parameters specific to a particular site or list, including:

- **Offline Behavior**: Default Offline settings for each List within the site.
- **Navigation**: Visibility and organization of sites and lists in the navigation bar.
- **Visualization**: Default SharePoint View and native Visualization for each list. For the site and its sub-sites, you can configure the Site Home.

Central Application Configuration

SharePlus Application Configuration can be adjusted using the following mechanisms:

- **Manual Configuration**: Directly from the SharePlus’ UI at run-time\(^3\)
- **Local ("In-App") Configuration File**: Using a static XML Configuration File stored within the application.
- **Remote Configuration**: Using a Remote Configuration File stored on SharePoint or any web server.
- **Library-Based Configuration**: Access different configuration files stored in a SharePoint’s document library, providing a selective configuration depending on the application scenario.
- **Dynamic Configuration**: Use web services or scripting language to generate Remote Configuration Files “on-the-fly”.
- **Configuration “Injection”**: Passing a new Configuration File to an already installed application by “injecting” the configuration via “Open In” protocol, invoking a custom URL or via an MDM policy/command.
- **MDM Administered Configuration**: SharePlus can work in conjunction with MDM servers, to set up Remote Configuration scenarios.

\(^3\) Only a subset of the Application’s Configuration can be manually adjusted.
Manual Configuration
SharePlus can be deployed without any configuration, delegating the task of configuring the application to the end user. When first opening SharePlus, the sidebar is displayed in the left side of the application. Through the sidebar component the user can access the sites, accounts, and global settings of the application.

Site Management
To access content from a SharePoint site within SharePlus you have to set up a new portal. Once your portal is configured, you will be able to browse its sub-sites, lists, and their content. Sites can be added, modified, and deleted from the UI.
Account Management

Users can add, modify, or delete User Accounts independently from Sites, as all accounts are global to the application and can be shared between sites.

You can access Accounts from Settings or from your Portals.

After tapping Accounts, all existing accounts are displayed.
Global Settings
Manual configuration settings include, among others, the auto-lock settings, connection timeouts, and synchronization idle time.

Local and Centralized Configuration
SharePlus Application Configuration can be adjusted in a local or remote file\(^4\), thus working with **Local Configuration** or **Remote Configuration** respectively.

Main advantages of Remote Configuration over Local Configuration:
- When configuration settings are adjusted in the remote file, the application does not need to be re-deployed after a configuration change.
- You are able to work with contextual information of the application, in order to provide a selective configuration depending on the application scenario. The app's contextual information include the OS version and Device ID among others.

In-App Configuration File (Local Configuration)
The Local configuration file is an iOS “Property List” XML file stored inside the application package. This file is fully configured before publishing the application, and packed inside the application’s “.IPA” file.
After editing the configuration file, the “.IPA” file has to be re-signed, to ensure that the IPA checksum will match its content. The application is later deployed and the Configuration File is stored locally in the device.

---

\(^4\) SharePlus Configuration File is an iOS “Property List” XML file used to store, organize and access SharePlus configuration data.
Centralized Configuration File (Remote Configuration)

The Remote configuration XML file is published in a centralized location, on SharePoint or other server. The use of a remote configuration file, facilitates governance, allowing IT Managers to modify the default settings of the device base centrally, without even having to push a new version of the product.

Remote Configuration Files are not pushed from the server by SharePlus, instead, they are retrieved from the server when requested (“pull” mode).

**Note:** In Remote Configuration scenarios it is recommended to work with a partial Configuration File, including only the items you want to change in the file. The application will later merge the data provided by the remote file with the local settings automatically.

In the case of using Dynamic Configuration, the remote configuration file is generated on-the-fly for every user, in that scenario, you work with multiple remote configuration files.

In the case of using Library-Based Configuration, you also work with multiple remote configuration files stored in a SharePoint’s document library. It is a fixed number of configurations, though, as every document in that library represents a different configuration for the application.

For further information about Remote Configuration see [Central Application Configuration](#)

Library-Based Configuration

This configuration mechanism consists on accessing different configuration files stored in a SharePoint’s document library. Every document in that library represents a different configuration for the application and must have the same structure as a standard Configuration File, which means that the file content needs to include configuration entries as any iOS "Property List" XML file.

To access the Remote Configuration Files stored in the SharePoint library you need to specify the URL to the library.

Dynamic Configuration

Instead of accessing a “static” file stored in a web server or SharePoint library, you can obtain a generated “dynamic” file. To generate this Remote Configuration Files “on-the-fly”, certain contextual information needs to be processed to generate selective configuration settings.

SharePlus application provides contextual information such as OS version, Device ID, or language among others.

Dynamic Configuration can be achieved using different methods, two common choices are:

- Using scripting language, obtaining a generated file from a database-driven web site.
- Using web services, retrieving selective configuration based on known context variables.
Configuration “Injection”

SharePlus configuration files can be manually loaded after the application has been deployed. The new (applied) configuration will merge with the Local Configuration and, as a result, a new SharePlus configuration will be set on the device. Below there is a brief description of the two different methods that you can use to “inject” a new configuration to SharePlus.

**Note:** Both methods described next are disabled by default to enforce the application security. For further details refer to Configuration Injection, in the Security chapter.

**SharePlus link (S+ link)**

This SharePlus feature allows users to perform certain actions within SharePlus, invoking these actions from HTML content. S+ links are basically custom URLs that start with “splus://” or “spluss://” (for secure channels) and are followed by the resource’s URL without the http or https protocol. They can be used to modify the application’s configuration when building the URL with a set of required parameters.

Syntax needed to update the Remote Configuration URL in the Configuration File:

```
 splus://?action=configurationURL&url=<URL to the new Configuration File>&useragent=<user agent>&timeout=<timeout in seconds>
```

**Note:** Sometimes you need to encode an entire URL or just some characters. For example, when sending a S+ link with a remote configuration by email, Outlook removes one “/” character from “http://” making the URL invalid.

**Open In**

This iOS protocol allows you to share files between applications on an iOS device. The Configuration File must be opened with SharePlus through Open In and the application will automatically merge the existing configuration with the new one. This capability is very helpful when working with Remote Configuration scenarios, allowing you, for example, to change the URL from where the remote Configuration File is retrieved.

The Configuration File must be identified as a SharePlus file, so the SharePlus application will be available among the Open In list of applications. To achieve this, the Configuration File extension needs to be “.spconfig”.

---

**SharePlus 4.0**
Administrator Guide
MDM Administered Configuration

SharePlus can work in conjunction with MDM servers, to set up Remote Configuration scenarios. MDM servers include a number of different policies/capabilities to be applied, a common policy is the one that enables the remote configuration capability.

Site Configuration “a.k.a. MobileNavigation”

When browsing a site in SharePlus, all the sub-sites (or webs), lists, and libraries accessible to the user are displayed. All this SharePoint content is filtered by the user’s permissions on the server, but this solution is not enough for many SharePoint solutions. SharePlus provides the Site Configuration method to manage the settings for sub-sites and lists for each site independently. These settings include aspects like navigation, visualization, and offline behavior.

- **Navigation** is about the visibility and organization of sites and lists in the navigation bar.
- **Visualization** allows the configuration of the default SharePoint List View and SharePlus’ native visualization for each list. You can also configure the Site Home to personalize the actual site and its sub-sites gateway.
- **Offline behavior** allows the configuration of default offline settings for a list in the site.

This method is configured using SharePoint Custom Lists created under each site, as it is a site-specific configuration method. The sub-site configuration list, normally referred as “MobileNavigation”, can be created by manually creating the list or using a list template.

Manually Creating the List

This method uses the SharePoint out-of-the-box template “Custom List”. It involves creating all the required columns to meet a pre-defined structure for the MobileNavigation list, following a small number of considerations.

Creating the List from a List Template

To follow this method, you need the latest Mobile Navigation list template which can be requested to Infragistics as explained in the Configuration Chapter. Once you have the template available in the list template gallery of your site, you can create the MobileNavigation list automatically. When working with SharePoint farms and using Site Templates, you can take advantage of this approach and automatically populate new sites with the Mobile Navigation list template.
Once the list is located within the site, SharePlus will find the list, identify and read the structure and data to apply the specified settings.
Understanding SharePlus Deployment

SharePlus licensing models (Free, Subscription, and Enterprise) are distributed using different methods.

- **Free/Subscription versions.** Downloaded directly from Apple’s App Store distribution platform.
- **Enterprise version.** Can be distributed using the following methods:
  - MDM Server Deployment
  - Web Deployment using SharePoint
  - Ad-Hoc (typically used for testing purposes)

MDM Server Deployment

When using an MDM Server, you need to upload the application package to the MDM Server following the instructions of the manufacturer. The application package (.IPA) is provided by Infragistics, then the whole deployment process is delegated entirely to the MDM.

For further details about MDM Servers refer to the **MDM Integration** in Chapter 4: Deployment.

Web Deployment using SharePoint

This deployment method allows you to distribute the application without any collaboration of external services like MDM servers or the AppStore.

Working with Web deployment using SharePoint, the SharePlus application package is copied into a library located in SharePoint. When the web page is opened from a mobile device, a web page contained on the application package enables the automatic installation of SharePlus application.

**Note:** SharePlus deployment can be carried out with any web server, including SharePoint.
iOS does not provide any means to force the update of an application at the Operating System level. SharePlus Enterprise, however, provides this feature and can be configured to check for a new version every certain period of time, also suggesting the user to update the application. The application’s upgrade can be postponed for a configurable period of time, after which the application will force the upgrade.

For further information refer to Enterprise Deployment
Web Deployment using SharePoint
in Chapter 4: Deployment.

**Ad-hoc deployment**

This method allows distributing SharePlus application as a standard file and installing the app on devices using iTunes. The whole process consists of three steps:

- Locate the Unique iOS Device ID(s) (belonging to the device(s) to be used for testing).
- Build the application for that specific set of Device IDs.
- Install the application using iTunes or the iPhone Configuration Utility.

**Application Resigning**

To enable in-house deployment of SharePlus, the application needs to be signed with the client’s Apple Enterprise Certificate. To obtain the Enterprise Certificate, the client must enroll in the iOS Developer Enterprise Program.

After obtaining the Enterprise Certificate, there are two alternatives to have a SharePlus build signed with the certificate:

- Send the certificate to Infragistics, so we can directly sign the build with your certificate.
- Re-sign yourself the build sent to you by Infragistics.

For further details about the process and all related concepts, refer to Application Resigning in Chapter 4: Deployment.
Planning Connectivity

Typical Network Architectures

Basic/Standard information
- The user’s device running SharePlus connects to SharePoint through the HTTP(S) protocol, within the Company Network environment.
- SharePlus connects to the SharePoint server or SharePoint farm through OOTB Web Services.

VPN

VPN-specific information
- SharePlus connects to SharePoint through a VPN (Virtual Private Network), extending the private and secure network across a public network (internet).
- Firewalls and VPNS are used in conjunction. As using a VPN is about ensuring security, Firewalls provide control over the network traffic.
Reverse Proxy

*Reverse Proxy*-specific information

- SharePlus connects to SharePoint through a Reverse Proxy server deployed in a network DMZ, protecting internal servers from direct exposure to an untrusted network (internet).
- This architecture can be used to balance the load on a SharePoint server farm, increasing security by providing a single point of access to the internal network.
- Reverse Proxy security architectures may include domain authentication through Active Directory servers.

MobileIron with AppTunnel

*MobileIron*-specific information

- SharePlus connects to SharePoint through a secure dedicated connection, built upon MobileIron’s Sentry technology.
- This solution’s architecture brings many advantages regarding application’s security, integration, and administration.

For further information refer to [MobileIron's product page](#).
Web Services Communication

SharePlus connects to the SharePoint server through SharePoint’s Out of the Box Web Services. This access is optimized in SharePlus by design, in order to reduce the number of round trips between the application and the SharePoint server. For further details about SharePoint Foundation ASP.NET Web Services refer to the Web Service Guidelines MSDN topic.

HTTP methods

For SharePlus to function properly, the SharePoint server must allow access to GET, POST, and PUT HTTP methods. The following table list these three methods:

<table>
<thead>
<tr>
<th>HTTP Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>Method needed to retrieve data, downloading documents and configuration files.</td>
</tr>
<tr>
<td>POST</td>
<td>Method used to invoke SOAP services. Can also be used to upload files, see File Upload Method below.</td>
</tr>
<tr>
<td>PUT</td>
<td>Method used by default to upload documents. POST can be used instead, see File Upload Method below.</td>
</tr>
<tr>
<td>HEAD</td>
<td>Method used to request resources information.</td>
</tr>
</tbody>
</table>

SharePlus verification to the server

An early validation is needed to confirm that the SharePoint service meets SharePlus needs. To achieve that, SharePlus often uses absolute URLs to the SharePoint Web service. These URLs include the path to the Web site that will be available and their syntax is similar to the following example:

http://<servername>/<site>/_vti_bin/Lists.asmx

File Upload Method

You can configure the method used to upload files by choosing between PUT or POST. SharePoint supports both methods, but unlike PUT, the POST method requires contiguous free space available in the server. Because of that, the POST method may have issues with files bigger in size. Also take into account that sometimes the PUT method is blocked by Firewalls. By default, SharePlus uses the PUT method.

Alternate Access Mappings

Alternate access mappings direct user requests to the appropriate URLs while interacting with SharePoint, enabling internal URLs to be mapped to public URLs. Internal URLs originate from Web requests and are received by the SharePoint Frontend Servers. Public URLs are the base address that servers return in response to the Web requests. For SharePlus to work successfully, alternate access mappings must be correctly configured. This is particularly important when working with Reverse Proxy scenarios. For further details refer to Plan alternate access mappings TechNet article.
## Network Layer

SharePlus allows you to use different networking APIs, according to your needs.

<table>
<thead>
<tr>
<th>Network stack</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Legacy (ASIHTTPRequest)</td>
<td>This method uses low-level networking APIs and is recommended for NTLM scenarios. Default method for the Appstore (Free and Subscription versions).</td>
</tr>
<tr>
<td>Standard (NSURLConnection)</td>
<td>This method uses high-level networking APIs provided by Apple and may present issues with NTLM scenarios, e.g., timeout when uploading a file. This is the default method for Enterprise versions.</td>
</tr>
<tr>
<td>Modern (NSURLSession)</td>
<td>This method was released after iOS 7 release and was created to replace NSURLConnection as the preferred method of networking.</td>
</tr>
</tbody>
</table>
Understanding SharePlus Enterprise Security

SharePlus Enterprise provides secure access to SharePoint sites from mobile devices. Keeping enterprise data secure is the highest priority of SharePlus. To fortify enterprise data security, core security features have been built into the product at each layer. The following is a list of security features that are included in SharePlus Enterprise for iOS devices. These features will be described in detail in subsequent sections of this document.

In this Chapter:

- iOS Operative System Security
- Storage Security (Data-at-Rest Security)
- Channel (Communication) Security
- Authentication
- Authorization
- Application-Level Security
- MDM Integration

iOS Operative System Security

iOS system architecture comes with a layered approach to security which is enabled by default, for information regarding the subject refer to [http://www.apple.com/ipad/business/it-center/security.html](http://www.apple.com/ipad/business/it-center/security.html)

Data-at-Rest Security

SharePlus offers two features to ensure that data is secure on a user’s device:

- iOS Data Protection
- Secure Data Wipe

iOS Data Protection

SharePlus employs Apple’s iOS Data Protection feature to keep application data secure. This native iOS feature enhances the built-in hardware AES 256-bit encryption by protecting the encryption keys with a user’s passcode. This provides an additional layer of protection for application data such as cached documents and user configuration information. The iOS Data Protection feature protects data-at-rest, and this includes locking or powering down the device.
Secure Data Wipe
A SharePlus administrator can configure a secure data wipe to activate upon failed passcode entry attempts or failed attempts to authenticate to the server. Also, SharePlus integration with MDM servers includes secure data wipe configuration. iOS achieves data wipe by securely discarding the block storage encryption key from iOS Effaceable Storage, which renders all data unreadable.

Secure Data Wipe Triggered by Failed Passcode Entry
SharePlus may be configured centrally to securely wipe all application data upon reaching a configurable amount of failed passcode entry attempts. This feature may be enforced centrally by using the Remote Configuration feature provided by SharePlus.

Secure Data Wipe Triggered by “Authentication Time Bomb”
The Authentication Time Bomb feature allows administrators to set a limit on the number of days that a user can use the application without re-authenticating against the server. All SharePlus application data will be securely wiped from the device if a user reaches the configurable threshold for failed authentication attempts. This feature is most relevant in SharePlus when the offline functionality is used. In the offline mode, it is possible to work with SharePoint data cached on the device after the user authenticates with the server. The Authentication Time Bomb allows an administrator to limit the number of days that the application may be used without re-authenticating to the server.

MDM-Based Data Wipe
An administrator can also initiate the secure data wipe on-demand by issuing a remote wipe command from a Mobile Device Management (MDM) server. MDM integration will be discussed in a subsequent section of this document.

Channel Security
SharePlus communicates with the SharePoint Server by accessing out-of-the-box SharePoint Web Services over the network. In addition to authentication and authorization requirements, SharePlus is also able to ensure secure communication through the use of HTTPS (SSL) and VPN.

Virtual Private Network (VPN)
SharePlus supports VPN connectivity either through the built-in iOS VPN support or through 3rd party VPN client applications. The native iOS VPN support is compatible with the following VPN Tunneling protocols:

- Layer 2 Tunneling Protocol (LT2P)
- Point-to-Point Tunneling Protocol (PPTP)
- Internet Protocol Security (IPSec)

Once the VPN iOS feature has been turned on and a connection is established, SharePlus will utilize the tunnel for server communication. Additionally, the VPN can be set to automatically establish a connection when a SharePlus user attempts to connect to the server. This eliminates the need for the end user to connect to the VPN prior to using the application.

For details about how to set up a VPN on your Apple device refer to: iOS: Setting up VPN

SharePlus 4.0
Administrator Guide
MDM Server VPN
SharePlus Enterprise versions can enhance their channel security through a dedicated connection provided by MDM (Mobile Device Management) solutions. A per-app VPN, securely and transparently routes all the SharePlus network traffic.

VPN On-Demand
When using certificate-based authentication, iOS provides “VPN-on-demand” which allows apps to transparently load a VPN when connection to the server is required.
For further information, refer to [iOS Security Overview](#) within Network Security

Secure Sockets Layer (SSL)
Secure Sockets Layer (SSL) is a cryptographic protocol used to facilitate secure communication over the Internet. SharePlus supports SSL, access to certificate enabled repositories, and Self-Signed Certificates.
To access a site using Secure Sockets, just use an “HTTPS” URL when configuring your site.

Authentication
SharePlus supports the following authentication mechanisms out-of-the-box:
- Passcode Lock
- Windows Integrated Authentication (Active Directory)
- Form Based Authentication (FBA)
- Office 365
- Custom Web Logins

Application Level Passcode Lock
SharePlus provides an optional Passcode Lock. When opening the application with Passcode Lock enabled, you are prompted to enter a four digit code. The lock is automatically activated when the application goes to background or after a configurable amount of idle time.
Settings can also be centrally enforced by an Administrator through the use of global configuration.
Windows Integrated Authentication
SharePlus can work with Windows authentication.

Forms-Based Authentication
In this authentication mechanism, the user’s credentials are passed to the server over HTTP as Form data. This is one of the most common user-authentication mechanisms used on the web and, essentially, is an ad hoc technique that needs configuration.

Office 365 Authentication
This mechanism is used with SharePoint Servers hosted by Office 365 on the cloud.

Web Login Authentication
This mechanism has been implemented to support customized online authentication mechanisms. SharePlus presents the user with a browser-like window to enter his credentials.

Web-based Auto Login
When using Web Login, users normally need to enter their credentials and submit the information. Through web-based auto login, users skip that part as the whole authentication process works as an integrated mechanism. They only need to select Web Login and associate an account, SharePlus then automatically populates the fields and submits the credentials information.

For the web-based auto login to work, SharePlus will search for the login name, password, and submit button within the page. The following fields are needed:
- input tag with name = “login” or class = “SPLoaderUserInput”
- input tag with name = “password” or class = “SPLoaderPasswordInput”
- button with class = “SPLoaderAutoSubmitElement”

Client Side Certificates
Client-side digital certificates can be used to request authenticated access in a SharePlus Enterprise application. You can deploy your client-side certificates by using two methods:
- **Manual configuration.** The certificate is saved to SharePlus “Local Files”, and then it is assigned to a user account.
- **Integrated.** SharePlus reads the certificate from a shared location on the keychain, where it was stored by MDM Servers or “Helper Apps” (custom apps used to deploy in the shared location).
Manual
This method is very straightforward, and, at a very high level, you just need to complete the following two steps:

1. Copy the client-side certificate to the device running SharePlus in order to make it accessible to the application (Local Files).
2. Associate the certificate to the user account that you want to use the authenticated access.

To achieve this two steps, you need to use the Wi-Fi sharing feature to send your certificate from your PC to SharePlus’ Local Files over the network. Also, you must edit the desired account to enable the use of certificates and later select the new certificate to be used. For a detailed procedure, refer to the How to use Client-side Certificates blog post.

Integrated
An MDM Server Agent (or “Helper App”) can be used to store certificates on a shared location on the keychain. The following three steps are needed:

1. The MDM Server Agent and SharePlus must be re-signed so they can share keychain access.
2. Once deployed to the device, the MDM Server Agent obtains the certificate and stores it on the keychain on a known location.
3. SharePlus must be configured to read credentials from that known location.

As Enterprises are able to re-sign all their mobile applications before deployment, both the MDM Server Agent and SharePlus must be configured to belong to the same group. In consequence, both apps will share keychain access.

Other Authentication Methods
Besides existing out-of-the-box authentication methods, SharePlus Enterprise allows other possibilities to be taken into account, either using Web-based Authentication or by developing a custom Authentication Provider.

Custom Authentication Providers
SharePlus can be extended through the SharePlus SDK, which allows the implementation of custom authentication methods according to the Enterprise needs.

Multi-Factor Authentication methods
One-time password (OTP) methods, like RSA token, are supported using Web-based Authentication or through the implementation of custom Authentication Providers.
**Authorization**

User access to SharePoint resources such as lists and documents is granted in the SharePoint Server using permissions. Since SharePlus authenticates with SharePoint Server’s web services using the end user’s credentials, server-defined authorization rules apply. Therefore, the level of resource access (read/write, etc.) will mirror that which has been set up on the server.

For additional information regarding permissions in the SharePoint Server, please refer to the [User permissions and permission levels](https://technet.microsoft.com) TechNet article.

**SharePlus User Accounts Management**

SharePlus allows you to work with as many user accounts as you need for your sites. SharePlus accounts are not restricted to a specific site, instead they are global to the application and can be shared between sites. When creating a new account you need to select the set of credentials to be used to authenticate against the site’s server. You can select one account to be the one used by default when creating a new site.

**Kiosk Mode**

The “Kiosk mode” allows users to share a device without compromising security. In this mode SharePlus behaves as a session-based application, saving data of the current session while active and wiping all user data as soon as the session ends.

**End of session**

Sessions can end in two different ways, by specifying a session timeout (in minutes) or when the user manually logs out. The Kiosk mode can be configured to add a “Log Out” button to SharePlus UI (User Interface).

Depending on the session timeout setting, the user’s session can expire after a certain period of time. An informative dialog is displayed to notify the user that the session is over.
For further details about the Kiosk mode configuration, refer to “Kiosk” Mode Overview
Application Level Security Policies

All Enterprises frequently need to adjust application functionality based on their security rules. These rules may include restrictions over application functionality such as saving a SharePoint document locally or sharing files via the Wi-Fi Share feature. SharePlus natively supports this administrative requirements through mechanisms that will be discussed in this section.

Feature Trimming

Feature trimming allows Enterprise administrators to disable and adjust SharePlus features on a global level. This may be necessary to comply with security policies or, in other cases, just to simplify the User Experience.

Features can be disabled by modifying SharePlus global remote configuration file. This Enterprise-owned and hosted file is used to set application configuration at a global level. When this file is modified and a feature is disabled, the change will take effect for all Enterprise SharePlus users. The following lists includes some of the functional aspects of the application that are often disabled via the Feature Trimming mechanism:

Site Administration
- Adding Sites
- Deleting Connections
- Updates to Connections
- Credential Storage
- Remembering Last User Name
- My Site Support
- My Profile support

Local Files
- Copy to Local Files
- File browser
- Emailing Documents
- Swiping Documents
- Tabbed Previewing of Documents

List Management
- Advanced Search
- Offline Mode

Favorites
- Displaying Items Count

Items Management
- Add
- Edit
- Delete
- CheckOut
- Approve/Reject
- Copy URL
- Email URL

WiFi Sharing
- Allow WiFi Sharing
- Uploading Documents
- Downloading Documents

Open In Functionality
- Allow
- Restrict specific third-party application use

Printing
- Allow Printing

Enterprise Search Hub
- Enable the Search Hub
- Include Search Scopes
- Exclude Search Scopes

Global Settings
- Preview Documents On Tap
- Remove Local Files After Upload
- Help URL
- Enable Logging
- Disable Auto Lock on Preview
- Disable Device’s Auto Lock on Sync
- Connection Timeout
- Sync Idle Time
- User Agent

Location Services
- Enable Location Services
- Auto Start
Editor Whitelisting
Companies normally have a number of trusted (or preferred) apps suited for common tasks. They want to limit the sharing of information within a restricted number of applications. The Editor Whitelisting feature can be configured in SharePlus to filter which applications can receive files via the “Open In” protocol.

Trim Copy/Paste
SharePlus can be customized to restrict the copy and paste functionality prior to application deployment.

Block Screenshot Capture (iOS feature)
An administrator can modify the user profile settings within the device to disable the iOS screenshot feature.

Blank the App Screen before suspending
When an application is suspended, a screenshot is saved by iOS to quickly present the app in the last state when the app is resumed. This feature can be blocked in SharePlus, ensuring that the in-app information is never displayed without a user logged in.

MDM Servers
Mobile Device Management (MDM) software allows the secure management of mobile devices deployed across enterprises. As the “bring your own device” (BYOD) policy is spreading, security-related issues need to be addressed to avoid data breaches and other potential risks. MDM Servers capabilities are frequently used by Enterprises and, because of that, SharePlus offers integration with MobileIron, one of the most commonly used MDMs.
Chapter 3
Configuration

Section 1: Central Application Configuration
Section 2: Site Configuration a.k.a. “MobileNavigation”
Section 3: Server Configuration
Central Application Configuration

The Configuration File Format

SharePlus configuration file is an iOS “Property List” XML file containing six main sections:

- **ConfigManager** – Attributes related to the remote configuration setup.
- **Features** – Attributes related to the application’s features.
- **Pages**: Attributes that define pages (and view containers) that can be used as home content for pre-defined portals or the Application Home module.
- **Accounts** – Attributes that define site’s accounts.
- **Sites** – Attributes that define portal configurations.
- **Variables** – Attributes for custom features. Empty by default.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<plist version="1.0">
  <dict>
    <key>ConfigManager</key>
    <dict>...
    <key>Features</key>
    <dict>...
    <key>Pages</key>
    <dict>...
    <key>Accounts</key>
    <dict>...
    <key>Sites</key>
    <dict>...
    <key>Variables</key>
    <dict>...
  </dict>
</plist>
```

There are several options when choosing an XML editor. In case you don’t have a preferred one, good options are **Xcode** for Mac OS and **Notepad++** for Windows OS.

---

5 The Configuration File is a property list, a structured data representation used to store, organize and access data. Internally, it defines a dictionary (key-value pair) using XML syntax. This pre-defined structure is required by the application in order to correctly recognize and process the information. Property lists are used extensively by applications on OSX and iOS, for further details refer to Introduction to Property Lists in the Mac Developer Library.
Remote Configuration
SharePlus application configuration can be adjusted using a remote file with settings published in a centralized location, on SharePoint or other web server. The ConfigManager section contains a dictionary used to configure the remote configuration settings. It is not mandatory to provide information for all the keys every time. When specifying a URL and setting the MustLoadRemoteConfig to <true/>, the application will merge a remote Configuration File with the local settings automatically.

```
<key>ConfigManager</key>
<dict>
    <key>ConfigurationURL</key>
    <string>http://pathToRepository/SharePlus_Config_file.plist</string>
    <key>ConfigurationProviderID</key>
    <integer>0</integer>
    <key>ConfigurationVersion</key>
    <string>4.0</string>
    <key>ForceResetSettings</key>
    <false/>
    <key>MustLoadRemoteConfig</key>
    <false/>
    <key>MustUseLastAppVersion</key>
    <false/>
    <key>ConfigurationAccountName</key>
    <string></string>
</dict>
```

- **ConfigurationURL** – Possible values include HTTP or HTTPS based URLs.
- **ConfigurationProviderID**: Indicates the ID of the configuration provider to be used. This value references the app.plist configuration providers section. Possible integer values are 0 (default) used for the Endpoint provider and 1 for the Library-Based provider.
- **ConfigurationVersion** – Administrators can use this number to keep track of the different versions of the Configuration File.
- **ForceResetSettings** – All application settings are restored to default values.
- **MustLoadRemoteConfig** – When set to true and the application cannot access the remote URL, the user won’t be able to use the application.
- **MustUseLastAppVersion** – To determine if the last version must be installed, SharePlus checks against the LastVersion field from the VersionHandle feature.
- **ConfigurationAccountName** – Indicates the account to be used when retrieving the remote Configuration File. Should reference by name an account defined in the accounts section.

For further details about Remote Configuration refer to [Working with Remote Configuration](#)
Configuring Sites and Accounts

The “Accounts” Section
The Accounts section of the Configuration File contains the accounts that will be pre-loaded by SharePlus. The section is structured as an array with dictionaries, each dictionary representing an account. Pre-configured accounts must include the authentication information excluding passwords, which shouldn’t be included for security reasons\(^6\).

All accounts are global to the SharePlus application and can be shared between sites.

```
<key>Accounts</key>
<array>
  <dict>
    <key>Name</key>
    <string>Account’s name</string>
    <key>AllowDelete</key>
    <true/>
    <key>IsDefault</key>
    <true/>
    <key>Domain</key>
    <string>Domain’s name</string>
    <key>Username</key>
    <string>User’s username</string>
    <key>UseCertificate</key>
    <true/>
    <key>CertificateName</key>
    <string/>
    <key>CertificatePassword</key>
    <string/>
  </dict>
</array>
```

- **AllowDelete** – Define whether the account can be modified/deleted by the user or not.
- **IsDefault** – Turn this account into the application’s default account.
- **Domain, Username** – No password here. SharePlus will prompt the user to introduce credentials when needed.
- **Client-side certificates information** – Used to authenticate the client to the server, most of the times is not provided as it depends on the MDM configuration.

---

\(^6\) When a connection requires a password, the application will prompt the user to introduce his credentials on the first attempt to access the site.
The “Sites” Section

The *Sites* section of the Configuration File contains a collection of sites that will be pre-loaded by SharePlus application. The section is structured as an array with dictionaries, each dictionary representing a site. SharePlus sites or portals included here are the top-level connection to SharePoint content, and must specify an account and the authentication mode to be used.

```
<key>Sites</key>
<array>
  <dict>
    <key>Name</key>
    <string>Site's name</string>
    <key>URL</key>
    <string>http://mydomain.com/Sites/TheSite</string>
    <key>AuthMode</key>
    <integer>0</integer>
    <key>AccountName</key>
    <string>Account’s name</string>
  </dict>
</array>
```

- **Account Name** (Optional) – Must be an existent account (if empty the default account is used).
- **URL** – String value with the site’s location.
- **AuthMode** – Authentication mode accepted by the server. Possible values for this attribute are:

<table>
<thead>
<tr>
<th>Value</th>
<th>Authentication Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Windows Based</td>
</tr>
<tr>
<td>1</td>
<td>Form Based</td>
</tr>
<tr>
<td>2</td>
<td>Office 365</td>
</tr>
<tr>
<td>3</td>
<td>Web Login</td>
</tr>
</tbody>
</table>
Configuring Features

SharePlus allows you to configure default behaviors and turn features on/off by using a centralized Configuration File with a specific structure. Having a centralized file, allows to set the application’s configuration at a global level. When this configuration file is modified, the changes will affect all Enterprise SharePlus users. Features disabled or trimmed may respond to security policies or, sometimes, just to simplify the UX (User Experience).

Configuring a Feature

The Features section is related to the application’s behavior and feature trimming. Every feature is represented by a dictionary structure and most of these entries display the following structure:

```xml
<key>FeatureName</key>
<dict>
  <key>Enabled</key>
  <true/>
  <key>Settings</key>
  <dict>...
</dict>
</dict>
```

Each feature has the following common child nodes:

<table>
<thead>
<tr>
<th>Node</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabled</td>
<td>Boolean</td>
<td>Specifies whether the feature is enabled or not.</td>
</tr>
<tr>
<td>Settings</td>
<td>Dictionary (key-value pair)</td>
<td>Represents a dictionary of the needed parameters to configure the specified feature. Keys are always Strings and values for a specific key could be String, Integer, or Boolean.</td>
</tr>
</tbody>
</table>
# List of Configurable Features

The following table lists some features that are commonly disabled or adjusted by Enterprise administrators.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alerts</td>
<td>Enable and configure alerts to be displayed in Timeline (Social Hub).</td>
</tr>
<tr>
<td>AppMeasurement</td>
<td>Adobe Analytics specific settings.</td>
</tr>
<tr>
<td>DocumentPrint</td>
<td>Allows the user to print documents using iOS native AirPrint service.</td>
</tr>
<tr>
<td>EmailClient</td>
<td>Configure settings for a 3rd party email client, which replaces iOS email client.</td>
</tr>
<tr>
<td>EmailDocument</td>
<td>Allows the user to send documents via e-mail.</td>
</tr>
<tr>
<td>EmailURL</td>
<td>Allows the user to send the URL of a document via e-mail.</td>
</tr>
<tr>
<td>Favorites</td>
<td>Enables the use and access to Favorites in SharePlus.</td>
</tr>
<tr>
<td>Feedback</td>
<td>Allows the user to send feedback from inside the application.</td>
</tr>
<tr>
<td>GlobalSearch</td>
<td>Refine the Enterprise Search settings by including/excluding scopes.</td>
</tr>
<tr>
<td>Help</td>
<td>Configure the in-app help by replacing or hiding it.</td>
</tr>
<tr>
<td>HomeManagement</td>
<td>Configure the Application Home page and the Default Site Home page.</td>
</tr>
<tr>
<td>ImageManagement</td>
<td>Enables the user to manage the images before being uploaded.</td>
</tr>
<tr>
<td>ItemsManagement</td>
<td>Enables the user to manage list settings via UI.</td>
</tr>
<tr>
<td>ListManagement</td>
<td>Enables the user to manage the list settings via UI (User Interface).</td>
</tr>
<tr>
<td>Local Files</td>
<td>Manage the Local Files section settings.</td>
</tr>
<tr>
<td>Location Services</td>
<td>Allows the user to automatically retrieve geo-location data.</td>
</tr>
<tr>
<td>OfficeWeb</td>
<td>Customize the URL used to view/edit office documents in Office Web Apps.</td>
</tr>
<tr>
<td>OpenIn</td>
<td>Share documents with other apps through Apple’s Open In protocol.</td>
</tr>
<tr>
<td>Pages</td>
<td>Allows the definition of pages and view containers to be used as home content.</td>
</tr>
<tr>
<td>Pasteboard</td>
<td>Enable/Disable the copy &amp; cut functionality from sensible places in the app.</td>
</tr>
<tr>
<td>PDFViewer</td>
<td>Manage the PDF Viewer settings.</td>
</tr>
<tr>
<td>SitesAdmin</td>
<td>Allows the user to manage sites within SharePlus.</td>
</tr>
<tr>
<td>URLSchemes</td>
<td>Enables the use of custom URL Schemes and configures their behavior.</td>
</tr>
<tr>
<td>VersionHandle</td>
<td>Configure the automatic updates settings</td>
</tr>
<tr>
<td>ViewInWeb</td>
<td>Users can view a web version of SharePoint content in a Safari-like window.</td>
</tr>
<tr>
<td>WebManagement</td>
<td>Reorder website links in the Navigation pane.</td>
</tr>
<tr>
<td>Wi-Fi Sharing</td>
<td>Share data from the Local Files section via Wi-Fi.</td>
</tr>
</tbody>
</table>

For further information regarding these features’ configuration, see [Appendix 1: Configuration File Reference](#)
The “Global Settings” Section
This section includes the predefined configuration for a subset of SharePlus settings, the “global settings”. These are the application’s factory settings for all users and most of them can be manually adjusted through the UI. Below there is a list of these settings:

```xml
<key>GlobalSettings</key>
<dict>
    <key>Enabled</key>
    <true/>
    <key>Settings</key>
    <dict>
        <key>ConnectionTimeout</key>
        <integer>50</integer>
        <key>DisableAutoLockOnPreview</key>
        <false/>
        <key>DisableAutoLockOnSync</key>
        <false/>
        <key>DisableAutoLockMustBePluggedIn</key>
        <false/>
        <key>PreviewDocumentsOnTap</key>
        <false/>
        <key>RemoveLocalFilesAfterUpload</key>
        <false/>
        <key>UserAgent</key>
        <string></string>
        <key>SyncIdleTime</key>
        <string>60</string>
        <key>EnableLog</key>
        <integer>0</integer>
        <key>StartupModule</key>
        <string></string>
        <key>ExternalBrowser</key>
        <string>chrome</string>
        <key>NetworkStack</key>
        <integer>0</integer>
        <key>UploadMethod</key>
        <integer>0</integer>
    </dict>
</dict>
```

- **DisableAutoLockOnPreview, DisableAutoLockOnSync** – Prevent the device from locking itself while previewing docs and/or during the sync process.
- **DisableAutoLockMustBePluggedIn** – Enables the other two auto-lock options when the device is plugged in.
- **PreviewDocumentsOnTap** – Enables document preview as the default action when tapping an item. Item details are still available on a different view.
- **UserAgent** – SharePlus’ user agent. If empty the standard agent is used.
- **SyncIdleTime** – Frequency in which the sync process is triggered, in seconds.
• **StartupModule** – Specifies the application module to be loaded by SharePlus when opening the application from scratch. Possible values include all Row Identifiers specified in the NavigatorTableDefinition.plist file. Possible OOTB values for the attribute are:

<table>
<thead>
<tr>
<th>Attribute Value</th>
<th>Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>DocumentsHub</td>
<td>Documents</td>
</tr>
<tr>
<td>Favorites</td>
<td>Favorites</td>
</tr>
<tr>
<td>Home</td>
<td>Application Home</td>
</tr>
<tr>
<td>Search</td>
<td>Search</td>
</tr>
<tr>
<td>SitesHub</td>
<td>Sites</td>
</tr>
<tr>
<td>Social</td>
<td>Social</td>
</tr>
</tbody>
</table>

• **ExternalBrowser** – Specifies the external browser that will be used. Possible values are *chrome* or *safari*, and Safari is used by default.

• **EnableLog** – Enable/Disable the generation of application logs. Possible values for the attribute are:

<table>
<thead>
<tr>
<th>Integer Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>False (Default)</td>
</tr>
<tr>
<td>1</td>
<td>True</td>
</tr>
</tbody>
</table>

• **NetworkStack** – Specifies the method used in the networking layer. Legacy is recommended for NTLM scenarios. Possible values for the attribute are:

<table>
<thead>
<tr>
<th>Integer Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Legacy (ASIHTTPRequest). Default for Appstore versions (Free and Subs).</td>
</tr>
<tr>
<td>1</td>
<td>Standard (NSURLConnection). Default for Enterprise versions.</td>
</tr>
<tr>
<td>2</td>
<td>Modern (NSURLSession)</td>
</tr>
</tbody>
</table>

• **UploadMethod** – Specifies the method used to upload files, PUT or POST. SharePoint supports both methods, but unlike PUT, the POST method requires free space available in the server. Because of that, the POST method may have issues with files bigger in size. Also note that sometimes the PUT method is blocked by Firewalls. Possible values for the attribute are:

<table>
<thead>
<tr>
<th>Integer Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>WebDav (PUT) (Default)</td>
</tr>
<tr>
<td>1</td>
<td>WebService (POST)</td>
</tr>
</tbody>
</table>
Configuring Security

SharePlus enforces security in many different levels, providing a series of built-in security features for iOS devices. The security features detailed below need to be configured in the Configuration File depending on your specific needs.

Passcode Lock

The Passcode Lock feature implicates a number of related configuration settings like forcing the user to have a passcode lock or configuring the auto lock options. These configuration settings are included in the Configuration File under two different feature sections, Passcode and GlobalSettings.

Passcode

<table>
<thead>
<tr>
<th>Configuration Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EnablePasscode</td>
<td>Enable/disable passcode lock feature.</td>
</tr>
<tr>
<td>DefaultPasscode</td>
<td>Passcode lock code by default (4 digits Integer).</td>
</tr>
<tr>
<td>MustUsePasscode</td>
<td>Force the use of passcode lock feature (set up is on the first run).</td>
</tr>
<tr>
<td>PasscodeLockInactivityTime</td>
<td>Idle time before the application locks (in minutes).</td>
</tr>
</tbody>
</table>

GlobalSettings

<table>
<thead>
<tr>
<th>Configuration Keu</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DisableAutoLockOnPreview,</td>
<td>Prevent the device from locking itself while previewing docs and/or during the sync process.</td>
</tr>
<tr>
<td>DisableAutoLockOnSync</td>
<td></td>
</tr>
<tr>
<td>DisableAutoLockMustBePluggedIn</td>
<td>Enables the other two auto-lock options when the device is plugged in.</td>
</tr>
</tbody>
</table>
**Data Wipe**

Data Storage Security includes secure Data Wipe, which can be activated upon failed passcode entry attempts or failed attempts to authenticate to the server.

**Data Wipe Triggered by “Authentication Time Bomb”**

A SharePlus administrator can configure a secure Data Wipe to be performed after a number of days in which a user didn’t re-authenticate against the server. This feature can be configured through the “Authentication Time Bomb” feature.

```xml
<key>AuthenticationTimeBomb</key>
<dict>
    <key>Enabled</key>
    <true/>
    <key>Settings</key>
    <dict>
        <key>AllowedTime</key>
        <integer>10</integer>
        <key>MaxAttempts</key>
        <integer>3</integer>
        <key>AttemptFailedMessage</key>
        <string>You have [AttemptsLeft] attempt(s) left</string>
        <key>LimitReachedMessage</key>
        <string></string>
    </dict>
</dict>
```

- **AllowedTime** – Maximum period in time (hours) that the user can work with the app without login in again.
- **MaxAttempts** – Maximum number of failed authentication attempts before all the site’s data is wiped.
- **AttemptFailedMessage** – If empty, no message will be displayed for failed attempts.
- **LimitReachedMessage** – Message displayed when there are no attempts left. If empty, a default message is shown to the user.
Data Wipe Triggered by Failed Passcode Entry

A secure Data Wipe can be activated upon failed passcode entry attempts. The PasscodeFailsHandle feature section also includes the maximum authentication attempts, and the message displayed to the user when there are no attempts left.

```
<key>PasscodeFailsHandle</key>
<dict>
  <key>Enabled</key>
  <true/>
  <key>Settings</key>
  <dict>
    <key>BlockApp</key>
    <false/>
    <key>MaxAttempts</key>
    <integer>3</integer>
    <key>Message</key>
    <string>You have no attempts left</string>
    <key>WipeData</key>
    <false/>
  </dict>
</dict>
```

- **BlockApp** – Enable the application to be blocked on authentication fails.
- **MaxAttempts** – Maximum authentication attempts.
- **Message** – Message to be shown when the user has no attempts left.
- **WipeData** – When true, clears internal app data when the user has no attempts left (Local Files, Offline cache, internal DB and other user data).
Editor Whitelisting

Companies normally have a number of trusted (or preferred) apps suited for common tasks. They want to limit the sharing of information within a restricted number of applications. The “Editor Whitelisting” feature is configured within the “OpenIn” feature under the “Filters” key.

```
<key>OpenIn</key>
<dict>
    <key>Enabled</key> <true/>
    <key>Settings</key> <dict>
        <key>Annotation</key> <dict/>
        <key>Filters</key> <dict>
            <key>pdf</key> <string>customUTI, customExtension</string>
        </dict>
        <key>ShowButtonInViewer</key> <true/>
        <key>ShowButtonInViewer-iPad</key> <true/>
        <key>ShowButtonInViewer-iPhone</key> <true/>
        <key>AllowOtherExtensions</key> <true/>
        <key>AllowConfigurationUpdate</key> <true/>
    </dict>
</dict>
```

- **ShowButtonInViewer** – Shows/Hides the “Edit” button in the document preview top bar.
- **AllowOtherExtensions** – Allows to open documents with extensions not specified in the filters.
- **AllowConfigurationUpdate** – Enables/Disables the ability to update the app’s configuration using the Open In feature by sending a new Configuration File with .spconfig extension.
- **Filters** – Dictionary (key-value pairs) used to filter apps Open In feature based on file extension.
  - Key – contains the file extension, e.g., pdf.
  - Value – There are 3 different options available for this String field:
<table>
<thead>
<tr>
<th>Option</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;editor’s UTI&gt;, &lt;custom extension by the editor’s vendor&gt;</td>
<td>com.adobe.pdfreader, pdfex</td>
</tr>
<tr>
<td>&lt;app_id_1&gt;</td>
<td>&lt;app_id_2&gt;</td>
</tr>
<tr>
<td>&lt;officeWeb&gt;, &lt;action&gt;(^7)</td>
<td>officeWeb, edit</td>
</tr>
</tbody>
</table>

**Note:** When no custom UTIs are available, the app can be configured to “close” the editor app if the editor selected by the user doesn’t match a specific APP id.

**Copy & Paste Trimming**

SharePlus can disable the copy & cut functionality from sensible places in the application. This feature is configured through the *Pasteboard* feature in the Configuration File.

```xml
<key>Pasteboard</key>
<dict>
  <key>Enabled</key>
  <true/>
  <key>Settings</key>
  <dict/>
</dict>
```

\(^7\) When specifying the use of Office Web App, actions available are: edit or view.
**Configuration Injection**

Configuration files can be manually loaded or “injected” to an already installed application via the “Open In” protocol. In addition, the Remote Configuration URL can be updated by invoking a custom URL Schema (S+ link) or through a JavaScript API method (Configuration.setRemoteFileSource). All three methods can be disabled to enforce the application security.

**SharePlus links and JavaScript API**

Both SharePlus URL Schemes and the Configuration.setRemoteFileSource JavaScript API method can be completely disabled through the URLSchemes feature in the Configuration File.

```xml
<key>URLSchemes</key>
<dict>
    <key>Enabled</key>
    <true/>
    <key>Settings</key>
    <dict>
        <key>AllowConfigurationUpdate</key>
        <false/>
    </dict>
</dict>
```

- **AllowConfigurationUpdate** – Enables/Disables the ability to update the Configuration URL by using any of the two following mechanisms:
  - URL Schemes

**Open In**

This iOS protocol that allows you to share files between other applications and SharePlus can be used to load a new SharePlus configuration. You can disable this method if needed.

```xml
<key>OpenIn</key>
<dict>
    <key>Enabled</key>
    <true/>
    <key>Settings</key>
    <dict>
        <key>AllowConfigurationUpdate</key>
        <true/>
    </dict>
</dict>
```

- **AllowConfigurationUpdate** – Enables/Disables the ability to update the app’s configuration using the Open In feature with a new Configuration File with .spconfig extension.
"Kiosk" Mode Overview

SharePlus can behave as a session-based application, in which each session works as a temporal interactive dialogue between SharePlus and the user. Once the session ends, all user data can be wiped and the app can be reset to its initial state. This specific behavior is sometimes known as “Kiosk Mode”. This feature is configured through the SessionManagement feature in the Configuration File.

```xml
<key>SessionManagement</key>
<dict>
    <key>Enabled</key>
    <true/>
    <key>Settings</key>
    <dict>
        <key>AllowLogOut</key>
        <true/>
        <key>ApplyFromBackground</key>
        <false/>
        <key>ApplyToBackground</key>
        <false/>
        <key>ApplyFromScratch</key>
        <false/>
        <key>ApplyFromTimer</key>
        <false/>
        <key>AllowFromInactivity</key>
        <false/>
        <key>InactivityExpirationTime</key>
        <integer>0</integer>
        <key>ResetPasswords</key>
        <false/>
        <key>WipeData</key>
        <false/>
        <key>WipeLocalFiles</key>
        <false/>
        <key>WipeSitesCache</key>
        <false/>
        <key>WipeAccounts</key>
        <false/>
        <key>WipeAlerts</key>
        <false/>
        <key>WipeFavorites</key>
        <false/>
        <key>Timeout</key>
        <integer>10</integer>
    </dict>
</dict>
```

The SessionManagement configuration settings always apply all the enabled actions.
- **AllowLogOut** – Enables the log out button in the interface to manually end the session.
- **ApplyFromBackground** – Applies all actions when returning from an inactive scenario.
- **ApplyToBackground** – Applies all actions when the app is entering a background state.
- **ApplyFromScratch** – Applies all actions when the app starts.
- **ApplyFromTimer** – Applies all actions after a session timeout.
- **AllowFromInactivity** – Specifies whether the session will be handled by the InactivityExpirationTime or not.
- **InactivityExpirationTime** – Amount of hours the application will keep a session alive without being active.
- **ResetPasswords** – Action to reset the passwords of all the existing accounts.
- **WipeData** – Action to reset all the existing content in the device. Pre-defined sites in the Configuration file are automatically reloaded in the interface.
- **WipeSitesCache** – Action to delete all cached data for all existing sites (documents, images, thumbnails).
- **WipeAccounts** – This action deletes all existing accounts, leaving the existing portals with no account associated.
- **WipeAlerts** – This action deletes all created alerts and their definitions in the app.
- **WipeFavorites** – This action deletes all Favorites in the app.
- **WipeLocalFiles** – Action to delete all local files available in the Local Files section.
- **Timeout** – Session duration expressed in minutes, i.e., session timeout.

**Working with Remote Configuration**

Remote Configuration scenarios are very powerful and include many possibilities. It is important to fully understand all aspects of this SharePlus configuration capability. This section will help you:

- Achieve a general understanding of the scenario.
- Work with contextual information, such as the Device ID, to provide personalized configurations.
- Learn how to configure a Library-based configuration.
- Learn how to work with Dynamic Configuration.

**Understanding Remote Configuration**

SharePlus application configuration can be adjusted by merging a remote Configuration File with a local file in the device. When working with Remote Configuration, two Configuration Files are always used, one local to the device and another one stored in a centralized remote location. Note that Remote Configuration Files are not pushed from the server by SharePlus, instead, they are retrieved from the server when requested (“pull” mode).

Remote Configuration scenarios include many possibilities that include:

- Using one common file for all company devices or multiple personalized configuration files.
- Accessing static pre-defined files or dynamically generated configuration files.

In the case of working with **Dynamic Configuration**, the remote configuration file is generated on-the-fly for every user, in that scenario, you work with multiple remote configuration files.
In the case of working with **Library-Based Configuration**, you also work with multiple remote configuration files stored in a SharePoint's document library. It is a fixed number of configurations, though, as every document in that library represents a different configuration for the application.

**Advantages of Remote Configuration**

There are two main advantages derived from working with Remote Configuration scenarios:

- When configuration settings are adjusted in the remote file, the application does not need to be re-deployed after a configuration change.
- You can use contextual information of the application, e.g., OS version or device ID, in order to provide a selective configuration depending on the application scenario.

**Recommendation**

When working with Remote Configuration scenarios, it is recommended to work with partial configurations, including only the items you want to change in the Configuration File. The pre-defined configuration is complete.

**Providing contextual information for selective configurations**

**Understanding the Application Context Information**

SharePlus application’s context can be used to provide a selective configuration depending on the application scenario. You can add contextual information by replacing arbitrary tokens in the specified URL or even modifying the outgoing request. 3rd

Custom information to the application context can be added by implementing custom code for your SharePlus application. That can be achieved through the SharePlus Bridge when using Application Launchpads or working with Objective-C and SharePlus SDK.

**Examples using existing keys**

The following are examples of URL structures that can be specified as Remote Configuration URLs. Both examples describe a selective configuration based on known identifiers used in a custom implementation.

```
http://myDomain.com/Configuration_Api/GetConfigurationForInfo?osversion=[osversion]&osname=[osname]&oslanguage=[oslanguage]

http://mydomain.com/Configuration_Api/GetConfigurationByLocationIdentifier?locationidentifier=[locationidentifier]
```

**Note:** These examples use keys that already exist in the application context. When using your own keys, make sure there is a value in the context to replace a given token.
List of Application Context Key-Values

The available and ready-to-use application context information is listed below:

- **OSVersion** – The OS running version, e.g., 6.1.3.
- **OSName** – The OS name, e.g., iPhone OS.
- **OSLanguage** – The OS language preference (IETF language tag), e.g., en.
- **DeviceVendorIdentifier**: Identifier that iOS creates based on the device identifier and the bundle identifier of the application. Available from iOS 6 and later.
- **DeviceName**: Name of the device provided by the user, e.g., John’s iPad.
- **DeviceModel**: The device model, e.g., iPhone.
- **AppName**: The name of the application that appears in the springboard of the device, e.g., SharePlus.
- **AppMajorVersion**: First component of the version number, e.g., 3.
- **AppShortVersion**: Short version of the version number, e.g., 3.8.
- **AppVersion**: Current application’s full version number, e.g., 3.8.2.
- **AppType**: Type of application, provided by the application’s app delegate, e.g., Demo.
- **AppBundleIdentifier**: The bundle identifier of the application defined by the profile used to sign the application, e.g., com.infragistics.SharePlus.

Working with Library-Based Configuration

SharePlus allows you to distribute different configuration files stored in a SharePoint’s document library, where every document in the library represents a different configuration for the application.

The application’s contextual information can be used here to assign permissions, deciding which configuration applies for a given scenario by matching different list’s columns with context information.

Building the URL to the library

You need to specify the URL to a SharePoint library in order to get the configuration elements. This value can be included in the Configuration File in two different ways:

- Using the absolute URL to the configuration library in a SharePoint installation.
  
  E.g., http://mySharepoint.com/ConfigurationSite/Remote Configuration/Forms/AllItems.aspx

- Using the URL to the Sub-site where the configuration library was created along with the library title as it appears in SharePoint, both separated by a comma: [site/subsite URL],[list name].

  You can also use the list’s GUID instead of the list name. E.g.:
  
  - o http://mySharepoint.com/ConfigurationSite,Remote Configuration
  - o http://mySharepoint.com/ConfigurationSite,26534EF9-AB3A-46E0-AE56-EFF168BE562F

**Note:** If the library was renamed after being created (the display name changed), you must use the display name or list GUID in order to reference the list correctly.
The image on the right shows an example of how the library settings should look like after being created for configuration purposes in SharePoint.

**To take into account:**
- When creating the library, specify only the columns you need.
- SharePlus expects the list to be a document library.
- The content of every document must be compatible with the Configuration File's Property List structure, containing a subset or all configuration entries.

**Specify the provider in the Application Configuration Settings**

In the *ConfigManager* section, you need to specify the URL and the provider to be used (List-Based).

```xml
<key>ConfigManager</key>
<dict>
  <key>ConfigurationURL</key>
  <string>http://mySharePoint.com/ConfigurationSite,RemoteConfiguration</string>
  <key>ConfigurationProviderID</key>
  <integer>1</integer>
</dict>
```

**Working with Dynamic Configuration**

Instead of accessing a “static” file stored in a web server or SharePoint library, you can obtain a generated “dynamic” file. To generate Remote Configuration Files “on-the-fly”, specific contextual information needs to be processed to generate selective configuration settings. SharePlus application provides contextual information such as OS version, Device ID, or language among others.

Dynamic Configuration can be achieved using different methods, two common choices are:
- Using a scripting language, obtaining a generated file from a database-driven web site.
- Using web services, retrieving selective configuration based on known context variables.
**Using a Scripting Language**

You can generate a personalized Remote Configuration File “on-the-fly” for every user’s device that requests configuration. This powerful configuration scenario can be used for companies that require selective configuration depending on arbitrary variables, avoiding fixed configuration scenarios.

For example, an International Sales Company may want to provide their salesmen with personalized SharePlus configurations depending on their sales territory, sales team colleagues, position within the organization, etc.

**Example with a CRM system in a web server**

When generating the remote Configuration Files, specific queries to the site’s database can determine the configuration settings retrieved in the remote file. For example, a web server with a CRM system that generates dynamic content can be used in conjunction with Remote Configuration. In this scenario, all users’ devices have SharePlus configured to use a Dynamic URL to get the remote file from the server and load the configuration settings. The remote file is generated “on-the-fly”, depending on each specific user configuration defined in the CRM system.
Below you can find an illustrative pseudocode for this example’s Dynamic Configuration File:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
.plist version="1.0">
<dict>
<!-- Static Config Ends Here -->
<key>Sites</key>
<% Obtain user from HTTP context %>
<% Query LOB system and obtain list of sites for user %>
<Array>
<% For each site on list %>
<dict>
<key>Name</key><string><% Value %></string>
<key>URL</key><string><% Value %></string>
</dict>
<% end for %>
</Array>
</dict>
</plist>
```

**Using a Web Service**

By implementing a Web Service, you can retrieve a selective configuration for your application. This scenario is very useful when there is a fixed number of different application configurations. Below you can find a Web Service example built using .NET C#. The technology used to build the web services may vary, what really matters here is how the configuration is retrieved from the service.

**Example with a location identifier**

The URL specified in the Configuration File must be:

```
http://mydomain.com/Configuration_Api/GetConfigurationByLocationIdentifier?
locationidentifier=[locationidentifier]
```

The Service implementation will return a personalized Configuration File depending on the device’s location, and a default file when the location is not in the list of known locations.

---

8 Web Services or Application Services share business logic, data and processes across networks, allowing different sources to communicate with each other through the service implementation.
namespace EndpointSamples.Controllers
{
    public class EndpointController : ApiController
    {
        public HttpResponseMessage GetConfigurationByLocationIdentifier(string locationIdentifier = "")
        {
            DirectoryInfo resourcesDirectoryInfo = new DirectoryInfo(directoryPath);
            string configFileName = locationIdentifier + ".plist";
            FileInfo configInfo = resourcesDirectoryInfo.GetFiles().ToList().Where(f => f.Name.Equals(configFileName)).FirstOrDefault();
            if (configInfo == null)
            {
                configInfo = resourcesDirectoryInfo.GetFiles().ToList().Where(f => f.Name.Equals("defaultConfig.plist")).FirstOrDefault();
            }
            return this(ReturnTextXml(configInfo);
        }
        private HttpResponseMessage ReturnTextXml(FileInfo fileInfo) {
            return new HttpResponseMessage()
            {
                Content = new StringContent(File.ReadAllText(fileInfo.FullName), Encoding.UTF8, "text/xml")
            }
        }
    }
}

An HttpResponseMessage object is returned by the service. This object represents the content of the Configuration File as a text-based file with UTF-8 encoding. SharePlus application must be able to map this response into an XML file with a structure similar to Property List. As a recommendation, avoid using JSON and escaped responses.
Site Configuration
a.k.a “MobileNavigation”

SharePlus provides this configuration method to customize SharePlus application’s behavior for each site independently. Site Configuration includes settings to customize the sub-sites and lists belonging to the site, and also the Site Home (customized view of the site). These settings manage aspects like navigation, visualization, offline behavior.

- **Navigation** is about the visibility and organization of sites and lists in the navigation bar.
- **Visualization** allows the configuration of the default SharePoint List View and SharePlus’ native visualization for each list. For the site and its sub-sites, you can configure the Site Home.
- **Offline behavior** allows the configuration of the default offline settings for a list within the site.

Creating the Site Configuration List(s)

Site Configuration is implemented using a SharePoint custom list with a pre-defined structure. A custom list must be created on every site of the solution, as this is a site-specific configuration method. SharePlus will find the list within the site, will identify and read the pre-defined structure and later apply the specified settings.

Every item added to this configuration list will includes settings to configure a sub-site or a list belonging to the site. In addition, you can add one item to configure the Site Home for the site itself.

Manually Creating the List

First you need to create the SharePoint custom list, then you will adjust navigation, visualization, and/or offline behavior as desired. When creating the list, there are a three important details to take into account:

- The list must be created based on the out-of-the-box template “Custom List”.
- The list must be named “MobileNavigation” exactly when first created, later it can be renamed as desired. The following URL structure is required so SharePlus can locate the list:
  
  ![http://myDomain.com/MySite/Lists/MobileNavigation/AllItems.aspx](http://myDomain.com/MySite/Lists/MobileNavigation/AllItems.aspx)

- The list must keep the “Title” column provided by the custom list template, that being said, this is the only column name that can be changed if desired. Others column names described below are case sensitive.
Creating the List from a List Template

First you must upload the Mobile Navigation list template to the list template gallery in your site. Then you can create the Site Configuration List based on the new list template. Finally, adjust navigation, visualization, and/or offline behavior as desired.

When creating the list through this process, take the following details into account:

- You need administrator permissions on the site to upload a new list template.
- You need to locate the site’s list template gallery. The following URL is an example of the List Template Gallery location:

  http://myDomain.com/_catalogs/lt/Forms/AllItems.aspx

When working with SharePoint farms and using Site Templates, you can take advantage of this approach and automatically populate new sites with the Mobile Navigation list template. Get the latest Mobile Navigation list template here.

The List Structure

There are a number of required columns for the list, other columns might be added in order to add value for administrators or contributors. These “extra” columns will not affect SharePlus application behavior. The required columns’ names are case sensitive and must be the same as specified in the table.

The following table shows the whole structure of columns required for the list, including details about the expected information on each column:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Expected information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Single line of text</td>
<td>The title of the list or site to be configured. Use the “ThisWeb” value to configure the site itself.</td>
</tr>
<tr>
<td>Is Web</td>
<td>Yes/No</td>
<td>Yes to configure a site and No for a list.</td>
</tr>
<tr>
<td>Hidden</td>
<td>Yes/No</td>
<td>Yes when the site or list should not be visible.</td>
</tr>
<tr>
<td>Default Mobile View</td>
<td>Single line of text.</td>
<td>Specifies the default SharePoint list view. Must match a list view, e.g., “All Documents”.</td>
</tr>
<tr>
<td>Default Visualization</td>
<td>Single line of text.</td>
<td>Must match a SharePlus list visualization, e.g., “Table”, “TableViewBased”, “Grid”, etc.</td>
</tr>
<tr>
<td>Available Visualizations</td>
<td>Choice (Checkboxes)</td>
<td>Must be a multiple selection choice and values are the same as Default Visualization above.</td>
</tr>
<tr>
<td>Layout Mode</td>
<td>Choice (Drop-down menu)</td>
<td>Must be a choice between two values: “Normal” and “Extended”.</td>
</tr>
<tr>
<td>Sort And Group</td>
<td>Single line of text.</td>
<td>Names of the fields available in the Sort and Group functionality, separated by commas.</td>
</tr>
<tr>
<td>Offline Support</td>
<td>Yes/No</td>
<td>Yes to allow the user to modify the offline settings for the list</td>
</tr>
<tr>
<td>Offline</td>
<td>Yes/No</td>
<td>Yes to enable the offline settings for the list</td>
</tr>
<tr>
<td>Offline Background</td>
<td>Yes/No</td>
<td>Yes to set the sync mode to “Background” and No for “Browsed”.</td>
</tr>
<tr>
<td>Offline View</td>
<td>Single line of text</td>
<td>Must match a default SharePoint list view, e.g., “All Documents”.</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td>Offline Only Wifi</td>
<td>Yes/No</td>
<td>Yes to synchronize over Wi-Fi networks only, excluding 3G</td>
</tr>
<tr>
<td>Offline Download Documents</td>
<td>Yes/No</td>
<td>Yes to download attached documents and not only the item’s properties.</td>
</tr>
<tr>
<td>Offline Max Document Size</td>
<td>Number</td>
<td>Maximum size (MB) allowed for the documents to be downloaded.</td>
</tr>
<tr>
<td>Query Page Size</td>
<td>Number</td>
<td>Number of items per block that are requested from the server when synchronizing data.</td>
</tr>
<tr>
<td>Swipe Documents</td>
<td>Yes/No</td>
<td>Yes to enable swipe through documents from a document library.</td>
</tr>
<tr>
<td>Preview On Tap</td>
<td>Yes/No</td>
<td>Yes to enable document preview on tap.</td>
</tr>
<tr>
<td>Tabs</td>
<td>Yes/No</td>
<td>Yes to enable the opening of documents in different tabs.</td>
</tr>
<tr>
<td>Order</td>
<td>Number</td>
<td>The position in the sequence of webs or lists displayed in the site view.</td>
</tr>
<tr>
<td>Home</td>
<td>Single line of text</td>
<td>URL to any content that can be set as Site Home for a site, e.g., PDF file or an HTML page.</td>
</tr>
<tr>
<td>HomeTitle</td>
<td>Single line of text</td>
<td>A title to be displayed for the Site Home.</td>
</tr>
<tr>
<td>Search Fields</td>
<td>Single line of text</td>
<td>Names of the fields to be used in the search, separated by commas.</td>
</tr>
<tr>
<td>ShowInSaveToOtherLocation</td>
<td>Yes/No</td>
<td>Allow/Block the inclusion of the list as a possible destination when using the Copy to/Move to functionality.</td>
</tr>
<tr>
<td>Picture Library</td>
<td>Single line of text</td>
<td>Specify the default picture library used to store images when working with Rich text fields.</td>
</tr>
</tbody>
</table>

To **configure a site** you need:
- Only five fields: Title, Is Web, Hidden, Home, and HomeTitle.
- The “Is Web” field should be set to Yes.

To **configure the current site itself**, you need:
- Only five fields: Title, Is Web, Hidden, Home, and HomeTitle.
- The “Is Web” field should be set to Yes.
- The Title field should be set to “ThisWeb” or “ThisSite”.

Adjusting Common/Basic List Properties

The first three fields are used for both sites and lists. The following table shows the three columns and a brief description of the purpose of each column:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Single line of text</td>
<td>Should specify the title of the list or site to be configured. When configuring the site itself should be set to “ThisWeb” or “ThisSite”.</td>
</tr>
<tr>
<td>Is Web</td>
<td>Yes/No</td>
<td>Indicates whether the group of settings to be configured apply for a site (web) or a list.</td>
</tr>
<tr>
<td>Hidden</td>
<td>Yes/No</td>
<td>Specifies visibility, whether the site or list will be hidden or not when browsing the parent site.</td>
</tr>
</tbody>
</table>

Note: If the Title of the list or site was changed, you must specify the new title.

Adjusting Site Home

The Home property allows you to specify home content for a portal or one of its sites. In addition, you are able to configure one or more contents for the same site.

Your Site Home can contain an Application Launchpad, a PDF document, a ReportPlus dashboard or any content that can be rendered in a web browser.

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| Home     | Single line of text | A list of values separated by comma. Possible values are:  
• An URL - to reference a resource to be loaded.  
• A page name - referencing a PageName key value defined in the Pages section of the Configuration File.  
• “default” - to reference the default SharePlus site view, which is the native view that displays sub-sites, libraries, and lists. |
| HomeTitle| Single line of text | A list of titles separated by comma, for each of the Home values specified in the Home field respectively. |

Note: The “default” value in the Home column will reference the DefaultSitePageName key value if this value was manually changed in the Configuration File.
Adjusting Navigation

Sometimes you need another layer of content filtering that is independent from the user’s server privileges. For those scenarios, SharePlus provides a customizable hierarchical navigation model, which must be customized for every site.

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hidden</td>
<td>Yes/No</td>
<td>Specifies visibility, whether the site or list will be hidden or not when browsing the site.</td>
</tr>
<tr>
<td>Swipe Documents</td>
<td>Yes/No</td>
<td>Enable/disable the ability to swipe through documents from a document library.</td>
</tr>
<tr>
<td>Preview On Tap</td>
<td>Yes/No</td>
<td>Enables/Disables the document preview as the default action when tapping a document. Shows its properties instead.</td>
</tr>
<tr>
<td>Tabs</td>
<td>Yes/No</td>
<td>Enables/Disables the tabbed previewing of docs. When opening a single document, all docs from a list are opened in different tabs.</td>
</tr>
<tr>
<td>Order</td>
<td>Number</td>
<td>Specifies the position in the ordered sequence of webs or lists displayed in the site view.</td>
</tr>
</tbody>
</table>

Adjusting Visualizations

By configuring SharePlus Visualizations, you can control how list data is presented to the user by default and which visualizations are available through the UI. Also, the layout mode can be configured for the item’s details. Two layout modes are available, “Normal” and “Extended”. The “Normal” layout shows labels and values on the same line. The “Extended” layout includes labels on the first line and values on the second line, below.

SharePoint List Views are different from SharePlus Visualizations, but both are used to organize, filter, and display the items of a list. List Views are configured in SharePoint, e.g., “All Documents” and are supported by SharePlus, while Visualizations only apply within SharePlus, e.g., Table, Grid, Picture.

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Mobile View</td>
<td>Single line of text</td>
<td>Specifies the default SharePoint list view. Must match a list view, e.g., “All Documents”.</td>
</tr>
<tr>
<td>Default Visualization</td>
<td>Single line of text</td>
<td>Specifies the default SharePlus visualization for the list, i.e., “Table”, “TableViewBased”, “Grid”, “Calendar”, and “Picture”.</td>
</tr>
<tr>
<td>Available Visualizations</td>
<td>Choice</td>
<td>Visualizations shown in the list. Must be a multiple selection choice (using checkboxes). Values are the same as Default Visualization.</td>
</tr>
<tr>
<td>Layout Mode</td>
<td>Choice</td>
<td>Indicates whether to use “Normal” or “Extended” layout when opening an item’s details (or item’s metadata) in SharePlus.</td>
</tr>
<tr>
<td>Sort And Group</td>
<td>Single line of text</td>
<td>List of the field names to be displayed in the Sort and Group functionality, separated by commas.</td>
</tr>
</tbody>
</table>
**Note:** Through the Configuration List you can select a default SharePoint List View, other views will be displayed though.

“Normal” and “Extended” layout:
**Adjusting Offline Behavior**

The Offline Support capability is enabled at the List level and each list is configured separately. A number of offline settings are available for every list, including whether the offline settings are enabled or not. When offline behavior is disabled, all other settings are ignored.

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offline Support</td>
<td>Yes/No</td>
<td>Specifies whether the user can modify the offline settings for the list or not.</td>
</tr>
<tr>
<td>Offline</td>
<td>Yes/No</td>
<td>Enables/disables the configuration of offline settings for the list.</td>
</tr>
<tr>
<td>Offline Background</td>
<td>Yes/No</td>
<td>When enabled, sets the synchronization mode to “Background”. Otherwise is set to “Browsed”.</td>
</tr>
<tr>
<td>Offline View</td>
<td>Single line of text</td>
<td>Specifies the SharePoint list view used for synchronization. When no value is specified, the list’s default view is used.</td>
</tr>
<tr>
<td>Offline Only Wifi</td>
<td>Yes/No</td>
<td>Specifies whether the synchronization process will be done over Wi-Fi networks, and not 3G.</td>
</tr>
<tr>
<td>Offline Download Documents</td>
<td>Yes/No</td>
<td>Indicates whether attached documents will be downloaded during the sync process or only metadata (item’s properties).</td>
</tr>
<tr>
<td>Offline Max Document Size</td>
<td>Number</td>
<td>Specifies the maximum size of documents (in MB) to be downloaded during the sync process. For example, when this value is 10, only docs smaller than 10MB will be downloaded.</td>
</tr>
<tr>
<td>Query Page Size</td>
<td>Number</td>
<td>Number of items per block that are requested from the server when synchronizing data.</td>
</tr>
</tbody>
</table>

**Adjusting List Search**

The keyword “standard” list search allows the user to perform a default search based on the name or title of the document. This search can be configured to search by other fields, like “% Complete” for example.

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search Fields</td>
<td>Single line of text</td>
<td>Specifies the names of the fields to be used in the search separated by commas. E.g., “Title, % Complete, Assigned To”.</td>
</tr>
</tbody>
</table>
Server Configuration

Adjusting SharePoint “Read” permission level

You must grant “Browse Directories” permissions to accounts with read-only privileges. This can be achieved with one simple modification on the Read permission level, on the SharePoint server site.

1. Go to Site settings, and click on Site permissions.

2. Select Permission levels.
3. Click on the *Read* Permission levels link.

4. Select the *Browse Directories* option, and save.
Adding Pictures to Contact Lists

When working with Contact Lists, you need to properly configure the picture fields in the list to make them available in SharePlus. This is a very straightforward task, just include any of the field names that SharePlus will look for in the list definition, which are:

- Picture
- Photo
- Foto
Chapter 4
Deployment

Section 1: Enterprise Deployment
Section 2: Application Re-signing
Enterprise Deployment

Web Deployment using SharePoint

SharePlus Web deployment allows the distribution of the application without any collaboration of external services like MDM servers or the AppStore. The application’s IPA is copied to a SharePoint library to make it available and later SharePlus can be installed by opening a web page from any device. It is a very straightforward method, you just need to copy the required information to the server and it will be working. In addition to this, the folder holding the content is completely static and will always comply with security restrictions from the server.

**Note:** SharePlus deployment can be carried out with any web server, including SharePoint.

Understanding Web Deployment

Web Deployment lifecycle

SharePlus deployment package is distributed to the user within the context of the over-the-air (OTA) mechanism. The process illustrated with the diagram is further explained below:

1. In the device and through the wireless network, the User opens a web browser with the Download Page.
2. The Download Page contains a download link URL that includes a prefix with the “itms-services” protocol and the XML manifest file location.
3. Once the User follows the download link, iOS Launchpad proceeds to read the information on the Manifest.plist file and the required resources (application IPA and icons) are retrieved from their respective network locations.
4. The user receives a confirmation request to install the application and SharePlus is installed on the User’s device.

---

9 This mechanism consists on using the wireless network to transfer the new application to a device, and it can be configured in any operating system (e.g.: Mac OS X, Windows, Linux).
Preparing the deployment package

There are several requirements that need to be ready to follow this deployment method. Most of these components need configuration (XML manifest, download page, SharePlus IPA). The application icons are, however, simple resources to be included.

- SharePlus IPA file
- Application icons
- XML manifest file
- Download page

SharePlus IPA file

SharePlus in-house iOS application that will be provided by Infragistics in .ipa file format.

Application icons

Two icons are needed for your application, a small icon and a large one. The icons used here should be the ones used for application rebranding which, normally, are provided by a Visual Designer.

<table>
<thead>
<tr>
<th>PNG Icon</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small</td>
<td>57 x 57 px</td>
<td>Image displayed during download and installation.</td>
</tr>
<tr>
<td>Large</td>
<td>512 x 512 px</td>
<td>Image displayed in iTunes.</td>
</tr>
</tbody>
</table>

Fully qualified URLs must be specified for the images.

XML manifest file

The manifest file is an XML plist\(^{10}\) format file. It is used by iOS devices to find, download, and install applications from your web server. It contains a set of attributes required by the device to correctly install the application.

Assets (app package and images) and metadata (bundle identifier and title) must be correctly configured using any XML editor. The application’s bundle identifier value must be included in the XML manifest to identify SharePlus application in the device.

**Getting the Application’s Bundle Identifier**

This value identifies your application within the device, e.g., com.yourcompany.appname. The application’s bundle identifier to be used must match the value set on your Provisioning Profile.

You can get the application’s bundle identifier from three different places:

- iOS Provisioning Portal
- Provisioning Profile file
- IPA file

---

\(^{10}\) Property list files are used to store, organize and access data. Internally this files define dictionaries (key-value pair) using XML syntax. Property lists are used extensively by applications on OSX and iOS.
Below there is a quick guide on how to get the bundle identifier through any of these methods.

From iOS Provisioning Portal
1. Login to the [iOS Provisioning Portal](https://accounts.developer.apple.com).
2. Go to `App IDs` on the left pane.
3. Look for the `application's ID` and click `configure`.
4. The ID will be presented on a frame like the following:

   ![com.infragistics.businessapps.shareplus](image)

From Provisioning Profile file
1. Open the Distribution Provisioning Profile (`.mobileprovision`) used to sign the build in any text editor (e.g., TextEdit).
2. Search for the `Entitlements` key.
3. Search for the `application-identifier` key within the `Entitlements` key.
4. Copy the value within this key without the first part, e.g.: `com.yourcompany.appname`.

From IPA file
1. Open the iPhone Configuration Utility on your computer ([Windows](https://www.microsoft.com) | [Mac OS](https://www.apple.com/)).
2. Go to the `Applications section` on the left pane.
3. Drag the IPA file to the `iPhone Configuration Utility window`.
4. Copy the bundle identifier of the IPA from the `identifier column`.


Code included in the XML manifest file should be similar to the one below:

```xml
<dict>
  <key>items</key>
  <array>
  
  <dict>
    <key>assets</key>
    <array>
      
      <dict>
        <key>kind</key>
        <string>software-package</string>
        <key>URL</key>
        <string>http://<portal>/yourfolder/app.ipa</string>
      </dict>
      
      <dict>
        <key>kind</key>
        <string>full-size-image</string>
        <key>needs-shine</key>
        <true/>
        <key>url</key>
        <string>http://<portal>/LargeImage.png</string>
      </dict>
      
      <dict>
        <key>kind</key>
        <string>display-image</string>
        <key>needs-shine</key>
        <true/>
        <key>url</key>
        <string>http://<portal>/smallImage.png</string>
      </dict>
    </array>
  </dict>
  
  <key>metadata</key>
  <dict>
    <key>bundle-identifier</key>
    <string>com.yourcompany.appname</string>
    <key>kind</key>
    <string>software</string>
    <key>title</key>
    <string>appname</string>
  </dict>
</dict>
```

- **url (software-package)** – URL to SharePlus IPA file.
- **url (full-size-image, display-image)** – URL to image resources (large and small application icons).
- **bundle-identifier** – Application’s Bundle Identifier, the value that identifies your application within the device.
- **title** – The application’s name, the last part of the bundle identifier.
**Download page**
To distribute SharePlus application, a simple web page with a download button must be created. This download page will be opened from a user’s device to install the application. The download button must include a link URL with a specific prefix and the location of the manifest.plist file.

**Distribution link**
To install the application, OTA deployment uses URL schemas\(^\text{11}\). In this case, you need the following prefix: “itms-services://?action=download-manifest&url=”. The URL to the manifest.plist is also needed at the end of the distribution link, as shown in the example below:

```
itms-services://?action=downloadmanifest&url=http://<portal>/manifest.plist
```

**Publishing the Deployment Package**
The installation assets must be accessible from the device, but can be spread across the network in many places. A centralized location to hold all the site’s assets is, however, recommended. In addition, that location must be accessible using HTTP or HTTPS without any type of authentication.

**Note:** Since iOS 7.1 the URL to the manifest.plist must use the HTTPS protocol.

The download page can be customized according to your needs. For example, you could request for credentials when accessing the page, restricting the users allowed to download the application.

**Promoting SharePlus installation**
Once you have the distribution link set up, you can make the deployment package available in many ways like sending the link by SMS or email, using banners, through JavaScript pop-ups, or by embedding the link in another enterprise applications created by you. The key element here is to develop traffic towards the distribution link location, to promote SharePlus installation.

**Setting server MIME types**
Sometimes you need to configure your server so that the manifest file and application IPA file are transmitted correctly.

**MAC OS X Server**
Using Server administrator credentials, add the following MIME types to the MIME types settings:

```
Application/octet-stream ipa
text/xml plist
```

\(^\text{11}\) These URL types have their own format and are frequently used to initiate specific requests. For further information, refer to [Apple URL Scheme Reference](https://developer.apple.com/library/ios/documentation/userexperience/conceptual/machereadyiphone/urlscm.html) in the iOS Developer Library.
Use the IIS Manager to add the MIME types in the Properties page of the server:

```
.ipa application/octet-stream
.plist text/xml
```

For more information on the entire process refer to Distributing Enterprise Apps for iOS Devices.

### Configuring Automatic Updates

SharePlus allows your company to set up automatic updates. OTA (Over-the-air) mechanism is normally used to distribute updates to all the devices that connect to your application. The `VersionHandle` feature allows you to configure automatic updates settings in the Configuration File.

```xml
<key>VersionHandle</key>
<dict>
    <key>Enabled</key>
    <true/>
    <key>Settings</key>
    <dict>
        <key>DeferBlockToTimeSpan</key>
        <string>10/19/2013</string>
        <key>ImmediateBlock</key>
        <false/>
        <key>LastVersion</key>
        <string>3.9.8</string>
        <key>URL</key>
        <string>http://<companysite>/download.html</string>
    </dict>
</dict>
```

- **DeferBlockToTimeSpan** – Limit date for users to update the app before it becomes blocked. Date format expected is mm/dd/yyyy, e.g., 10/19/2013.
- **ImmediateBlock** – When a new version of the app becomes available, the application is blocked until updated.
- **LastVersion** – Last version number of the app.
- **URL** – The URL where the updated app version is hosted for downloading; download.html file that was previously set up in the Over-the-air configuration process.

**Note:** The DeferBlockToTimeSpan and ImmediateBlock settings are optional and mutually exclusive.
Automatic Update Process

1. The app is launched
2. Attempt to fetch a Remote Configuration File
3. Comparison of versions
4. The User accepts the update
5. The website opens

- The ConfigurationURL specified in the app’s local Configuration File is used.
- If the LastVersion number on both Configuration Files (Remote and local) is different, the user is prompted to update the application.
- If the user accepts, the app is sent to background and Safari opens the website previously configured with OTA configuration.
- The website that was previously configured through OTA configuration opens. The user must tap the install/download button and confirm the update.

Automatic Update Messages

Non-Blocking message:

<table>
<thead>
<tr>
<th>New Version (4.5)</th>
<th>There is a new version available to download.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Later</td>
</tr>
</tbody>
</table>

Blocking message:

<table>
<thead>
<tr>
<th>New Version (4.5)</th>
<th>There is a new version available to download. Until you upgrade, the application will remain blocked.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Update</td>
</tr>
</tbody>
</table>

Deferred message:

<table>
<thead>
<tr>
<th>New Version (4.5)</th>
<th>There is a new version available to download. The application will be blocked if you do not upgrade the application before: 01/05/2014</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Later</td>
</tr>
</tbody>
</table>
MDM Integration

Mobile Device Management solutions (MDM) allow Enterprises to secure, monitor, and manage mobile devices. For additional information regarding Mobile Device Management, please refer to the following article: [http://www.apple.com/ipad/business/it/management.html](http://www.apple.com/ipad/business/it/management.html)

Introduction

SharePlus can integrate with many 3rd party MDM solutions available in the market. MDM solutions include many possibilities when it comes to the integration with SharePlus. These possibilities translate in a number of policies/capabilities can be applied to SharePlus. When working in conjunction with a MobileIron solution, the following SharePlus-related capabilities can be achieved:

- Deployment
- Remote Configuration
- Secure Data Wipe
- Authentication through client-side certificates
- Enforce security policies

Deployment

MDM Servers handle application deployment by themselves. The SharePlus application package (.IPA) must be upload to the MDM Server, and it will be provided by Infragistics.

Remote Configuration

Through this capability, the MDM Server can broadcast and replicate the Remote Configuration to all devices installed with SharePlus. The MDM server manages the remote URL, and sends it to SharePlus to load a Remote Configuration File.

Secure Data Wipe

To enforce security, MDM Servers can include Data Wipe policies. Important data can be irreversibly erased through a secure procedure.

Authentication through client-side certificates

MDM Servers can deploy client-side certificates to SharePlus, in order to use them to request authenticated access. SharePlus will read the certificate from a shared location in the keychain, where the MDM stored it.

Enforce Security Policies

A set of rules defined in the MDM server can be used to regulate the behavior of mobile applications. Features like Copy/Paste, Print, or Open In can be managed by the MDM in conjunction with SharePlus to enforce security.

Introduction to MobileIron

MobileIron is one of the leading vendors of Mobile Device Management (MDM) and Enterprise Mobility Management (EMM) solutions. Its offering includes “Docs@Work”, “Web@Work” and “Apps@Work”, solutions focused on the secure distribution of files, secure web access and secure management of native mobile applications respectively.
The three offerings are built upon MobileIron’s Advanced Mobile Management framework, which provides a complete solution for enterprise mobile apps, securing both the app data on the device through AppConnect functionality, and data in motion, by leveraging AppTunnel.

Apps@Work includes

- **App Storefront** – Core module that allows the apps distribution and management.
- **AppConnect** – Protects data-at-rest through data encryption and provides an SDK for securing and managing apps.
- **AppTunnel** – The dedicated tunneling and access control module, build upon the MobileIron Sentry technology.

MobileIron is available as both an on-premises system and a cloud service through the MobileIron Connected Cloud.
MobileIron Integration with SharePlus

SharePlus integrates with [Apps@Work](#) and [Advanced Mobile Management](#) functionality by supporting AppStorefront, AppTunnel, and AppConnect.

**Note:** AppConnect Data-at-rest encryption is currently not supported.

Deployment (App Storefront)

MobileIron AppStorefront can be used to distribute SharePlus by using the following steps:

1. Obtain the application file (.IPA) from Infragistics Services.
2. Configure the info.plist file by updating the following keys:

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFBundleName</td>
<td>shareplus</td>
</tr>
<tr>
<td>CFBundleIdentifier</td>
<td>com.infragistics.businessapps.shareplus</td>
</tr>
<tr>
<td>CFBundleURLName</td>
<td>com.infragistics.businessapps.shareplus</td>
</tr>
<tr>
<td>CFBundleURLSchemes</td>
<td>accom.infragistics.businessapps.shareplus</td>
</tr>
</tbody>
</table>

3. Configure the App identifier in AppConnect.plist.
   Update the bundle identifier key value to: “com.infragistics.businessapps.shareplus”

   **Note:** As shown above, you can initialize the policies’ configuration if desired. Otherwise, default values will be used until remote information is downloaded for the policies.

4. Configure and Re-sign the .IPA file following the guidelines described on the “Deployment” chapter.

5. Upload the .IPA file to MobileIron’s App Distribution Library as an “In-house app”
Authorization
When working in conjunction with MobileIron, SharePlus will check with the MDM for execution permissions. On the server, execution can be granted both by application and by user. iOS jailbreak detection is also provided to keep your data secure, protecting contacts and corporate data. MobileIron can be configured to block compromised devices from accessing corporate resources.

Policies & Configs
MobileIron’s “Policies & Configs” are a set of rules included in the Admin Portal, used to secure, manage, and regulate the behavior of mobile applications. By using Policies and Configurations, MobileIron can regulate the following SharePlus behaviors:

- **Policies**
  - Copy/Paste – Disables the Copy feature.
  - Print – Disables the Print feature.
  - Open In – Allows the configuration of this feature. See below for further details.

- **Configurations**
  - Assign a Remote Configuration File
  - Distribute Client-side Certificates

Copy/Paste
SharePlus blocks the “Copy” command on key locations, preventing the user to take sensible information out from the application using the clipboard. Key locations are:

- Document and Content viewers
- Item’s metadata details (view/add/edit element properties)

Even though the feature is called Copy/Paste, the user is able to paste content from outside to the application in the specified locations.

---

12 The Copy/Paste policy applies only to components managed by SharePlus. For external modules, like Mail and Address Book, it is not possible to apply this configuration. Instead, you should evaluate each scenario and disable this feature using the configuration file.
Open In
As mentioned above, this policy allows you to enable/disable the Open In feature, which is the common protocol to communicate between different apps.
Mobile Iron’s available options for this policy are:

- **Allow** – Notify the application if the feature is enabled/disabled.
- When enabled, you can choose between the following options:
  - **All apps** – Allows the Open In feature for all applications installed in the device.
  - **AppConnect apps** – Allows the Open In feature only for those applications that integrate AppConnect. SharePlus is one of those applications.
  - **Whitelist** – When this option is selected, you must specify the bundle identifier of those applications you allow SharePlus to communicate with.

Assign a Remote Configuration File
To push a configuration to SharePlus, administrators need to add the following configuration keys in MobileIron:

- **“ConfigurationURL” (String)** – Specifies the Configuration File’s location used when working with Remote Configuration. List-based Configuration is supported, allowing you to provide selective configuration depending on the application scenario.
- **“ConfigurationAuthenticationProviderID” (Numeric)** – Specifies the authentication provider to be used by the configuration module. This scenario applies when the remote configuration file’s location requires authentication.
- **“ConfigurationProviderID” (Numeric)** – Specifies the configuration provider to be used to retrieve the application’s configuration from the remote location.
- **“UserAgent” (String)** – Specifies the user agent to be used by the application.
This feature is optional and you can use none, some, or all of the specified keys depending on your needs. MobileIron’s settings in SharePlus have precedence over all configuration settings, including remote configuration and manually “injected” configurations (Open In, URL Schemas).

**Note on Authentication Providers:** SharePlus Authentication Providers resolve any authentication that is required in SharePlus. Standard providers are Windows-Based, Office 365, Web Login, etc. SharePlus SDK allows you to include new authentication methods.

**Note on Configuration Providers:** SharePlus Configuration Providers provide extensibility over the remote configuration feature, allowing you to extend existing providers and also to create new ones. Configuration Providers can use contextual information of the application (such as OS version and language between others) in order to provide a selective configuration depending on the app scenario.
Distribute Certificates
The distribution of digital documents can be achieved with a few steps:

1. Create a new configuration with “Certificates” type.

2. Add the following key to the “App-specific Configurations” section on the SharePlus Configuration policy.
   - **ClientCertificate** (digital document) – This key has a Mobile Iron’s client certificate reference as value.

3. In SharePlus, once the certificate is imported to the device it becomes available in the keychain and must be manually associated to an account.

AppTunnel Integration
AppTunnel can be used to securely route all SharePlus’ network traffic. AppTunnel capabilities are transparently applied at the application level by AppConnect’s configuration.
Ad-Hoc Deployment

To avoid the need of complex deployment systems to install the application, the Ad-hoc distribution method is used during the stabilization phase. The method allows distributing the application as a standard file and installing it on a set of devices using iTunes or the iPhone Configuration Utility. The process consists of three steps:

1. Locating the Unique Device ID for every device that will be used for testing.
2. Building the application for those Device IDs.
3. Installing the application.

Locating a Unique Device ID with an App

1. Download the app "UDID Pro" from the App Store.
2. Install the app.
3. Follow the instructions to send the UDID by mail.

Locating a Unique Device ID with iTunes 7.7 or later

1. Connect your device to your Mac or PC and launch iTunes.
2. In iTunes, select your device in the ‘Devices’ section and navigate to the Summary tab.
3. Click on the Serial Number label to reveal the Identifier field and the 40 character UDID.
4. Copy the UDID identifier to your clipboard.
Installing SharePlus using iTunes
After providing the UDID, Infragistics will prepare a SharePlus build specifically targeted to your device and send you the application file (.ipa). In order to successfully create the build, Infragistics Engineers need your Distribution Certificate.

See the following instructions to install the application:

1. If you have a previous SharePlus version installed:
   - Remove the old app from the device
   - Remove the old app from iTunes App Catalog
2. Install the beta build:
   - Drag the .ipa file to iTunes.
   - Select the new app in the Apps Section from the device
   - Apply changes to synchronize the app to the device

Installing SharePlus with iPhone Configuration Utility
The iPhone Configuration Utility is a very straightforward tool that allows you to manage the device’s content without using iTunes. This tool is available for both, Mac and Win OS in Apple’s Downloads site.

Follow these steps:

1. Download the application and install it.

2. Open the app and connect your device.
   You should see the application displayed and your device recognized like the image below:

   Under the Library category you will find the Devices and Applications sections. The Devices section shows the history of connected devices.
3. Click the **Application** section and drop the **SharePlus.ipa** file in the right pane as shown below:

4. Select the connected device under the **Devices** category and click the **Applications** tab. A list of installed applications is displayed.

5. Find the SharePlus application you just dropped in the **Applications** section and install it by clicking the **Install** button.
Application Re-signing

In order to deploy a SharePlus IPA file, it has to be re-signed using your company’s Enterprise Distribution Certificate. In the following chapters you will review how to re-sign your application along with how to get the required resources for the process.

To successfully re-sign a SharePlus application you need the following resources:

- Apple’s Enterprise Distribution Certificate
- Distribution Provisioning Profile
- App IPA file: This should be the application to be re-signed

The first two resource files are owned by companies enrolled in Apple’s Enterprise Program and can be retrieved from apple’s portal.

Understanding iOS App Re-signing Concepts

**The Enterprise Distribution Certificate**

The distribution certificate is a private certificate that is provided by Apple. Certificates are created to help Apple identify your company and, once retrieved, a certificate must be installed to be used in the application signing process.

**The Distribution Provisioning Profile**

The provisioning profile is targeted to one or more apps and can be retrieved from Apple’s portal. Profiles are created to authorize specific apps to run in iOS devices. A specific app can be uniquely identified by its bundle ID.

There are two types of provisioning profiles:

- *Ad-Hoc* - It is commonly used for internal testing and is restricted to specific devices.
- *In-House* - It depends only on the App ID and is not limited by device IDs.
About the App ID and Bundle ID

App IDs are used by Apple to identify applications. An App ID is a string composed with two parts, a Team ID prefix and a bundle ID search string. The Bundle Identifier (bundle ID) is a unique string used to identify a specific application’s bundle. The application bundle contains the executable code and all resources required to run the application. The Team ID prefix is automatically provided by Apple’s portal and identifies a development team.

The Certificate-Profile Relation

Provisioning Profiles are closely related to the Distribution Certificate. Because of that, if the certificate expires the distributed application will no longer run.

A Distribution Certificate can have more than one Provisioning Profile associated and a Provisioning Profile can target more than one bundle ID. Profiles are created for a specific App ID, but you can include a wildcard (.*), in the App ID specification.

For example, by entering A1B2C3D4E5.com.infragistics.* as the App ID, the same Provisioning Profile can be used for com.infragistics.shareplus and com.infragistics.reportplus bundle IDs.

Certificate-Profile expiration

Both Distribution Certificates and Provisioning Profiles must be renewed periodically.

- Distribution Certificate – valid for 3 years
- Provisioning Profile – valid for 1 year

SharePlus License

SharePlus app must have a license associated for each application’s bundle identifier (Bundle ID). SharePlus will not run in a device if the license file (license.splic) is missing or invalid.

The Contoso Company example

Contoso company wants to distribute SharePlus Enterprise app internally. To enable app distribution to iOS devices, Apple requires Contoso to be a part of the iOS Developer Enterprise Program. A Distribution Certificate (ContosoCertificate.p12) is needed to legally register an app under Contoso’s name.

Next, Contoso must define a bundle ID (com.contoso.shareplus) in order to identify SharePlus Contoso application. To register the bundle ID, Contoso needs to create a new App ID (A1B2C3D4E5.com.contoso.shareplus) in Apple’s Provisioning portal. The Team ID prefix (A1B2C3D4E5) included in the App ID is provided by Apple.
After that, a **Provisioning Profile** *(SharePlusContoso.mobileprovision)* can be created and downloaded from Apple’s portal to use the previously defined **bundle ID** *(com.contoso.shareplus)*.

Finally, a **SharePlus license** *(license.splic)* provided by Infragistics must be included in the IPA to be signed. Note that a SharePlus application will not run without a license file. After gathering all the required resources, the SharePlus Contoso IPA can be signed and distributed by Contoso company.

For further details about the whole process refer to the *Start-up Package document* and Application Re-Signing Process section in the *SharePlus Administrator Guide*.

### The iOS Developer Enterprise Program

This document assumes that your company is already enrolled in the iOS Developer Enterprise Program. For information about the program, visit *Apple’s Enterprise Developer Home*.

**Note:** This process usually takes, at least, two weeks. So, if you are not part of this program already, is important that you start this process as soon as possible.

### D-U-N-S Numbers

Apple requires a D-U-N-S number for organizations and can’t be bypassed. The Apple Support Center has a **D-U-N-S Numbers** page with additional information about their importance and how to get one. The support page also mentions that while getting a D&B number is free, you can expedite its creation (5 days) for an additional fee.

[https://www.dandb.com/establish-business-credit/](https://www.dandb.com/establish-business-credit/)

### Preparing the Certificate and Profile

Once you enroll your company in the Apple’s Enterprise Program, you need to get a couple of files from the apple’s portal to sign the application for enterprise distribution. These files are:

- Distribution Certificate
- Distribution Provisioning Profile

### Getting the Distribution Certificate

To get the distribution certificate from the portal, first you need to create a client request, which will be uploaded to the portal in order to generate the certificate. The following are the complete steps in order to get the Certificate correctly:

1. Launch the Keychain Access app.
2. In the Preferences menu, set Online Certificate Status Protocol (OSCP) and Certificate Revocation List (CRL) to **Off**.

![Preferences Menu](image)


**Note:** If you have a non-compliant private key highlighted in the Keychain during this process, the resulting Certificate Request will not be accepted by the Provisioning Portal. Please confirm that you are selecting “Request a Certificate From a Certificate Authority...” and not selecting “Request a Certificate From a Certificate Authority with <Private Key>...”

![Keychain Access](image)

   a. In the User Email Address field, enter your email address. The email address entered must match the same information submitted when you registered as an iOS Developer.
   b. In the Common Name field enter your name. The name entered must match the information that was submitted when you registered as an iOS Developer.
   c. No CA (Certificate Authority) Email Address is required. The “Required” message will be removed after completing the following step.
   d. Select the “Saved to Disk” radio button and, if prompted, select “Let me specify key pair information” and click “Continue”.

![Certificate Assistant](image)
e. If “Let me specify key pair” option was selected, specify a file name and click “Save”. In the following screen, select “2048 bits” for the Key Size and “RSA” for the Algorithm.

f. Click “Continue”.

The Certificate Assistant will create a CSR (Certificate Signing Request) file on your desktop.
5. After creating a CSR file, log in to the iOS Provisioning Portal and navigate to Certificates > Production and click “Add Certificate”.

6. Select “In-House and Ad-Hoc” option under “Production” and then click “Continue”.
As shown below, an informative screen about how to create a SCS file is displayed.

7. Click “Continue”.

8. Click the “Choose file” button, select your CSR and click “Generate”.

Note that if the Key Size was not set to 2048 bits during the CSR creation process, the Portal will reject the CSR.
Once generated, you should be able to locate the certificate in “iOS Certificates Production”.

![Certificate Image]

**Installing the Certificate**
After downloading it, you have to install the certificate locally:

1. In the *Certificates > Production* section of the Portal, click “Download”.
   The iOS Development Certificate is downloaded as a .cer file.

2. On your local machine, double-click the certificate to launch Keychain Access and install.

![Keychain Access Image]
Exporting the Certificate
Once you have the distribution/development certificate, you can export it from your computer and save it as backup or use it with another computer. Steps to export the certificate from your computer are:

1. Open the keychain access application.

2. Locate the certificate under login keychain for My Certificates category, and check that it has an identity key associated.

After that, you need to send the exported p12 file (Personal Information Exchange) to Infragistics engineers, this file should have embedded the identity key along with the certificate.

3. Ctrl-click over the certificate and then select “Export certificate”.

**Note:** The certificate must have an identity key associated, otherwise the certificate won’t be able to sign the application.

Getting the Distribution Provisioning Profile
To get the distribution profile, you need to have the Application ID first. If you already have an Application ID for the application you will distribute, skip the procedure below and go to the Creating your Provisioning Profile section.

1. Go to the Developer Center Home.

2. Login to the Developer Center and enter the iOS Provisioning Portal.
3. Navigate to the iOS App IDs section.

4. Click the plus (+) button to register an App ID. A form is displayed:
5. Fill the blanks with the corresponding data.
   Using an explicit App ID is recommended, since it will allow you to uniquely identify the app.
   If possible, use the standard notation based on a reverse-domain name style string, i.e.,
   `com.domainname.appName`. It cannot contain special characters like “*”.

6. Finally, click the submit button.

Creating your Profile

1. Under **Provisioning Profiles**, go to the **Distribution** section to add a new Profile.

2. Select the profile type.
   Available Distribution options are:
   - **Ad Hoc** - For internal distribution and testing purposes. Specific device IDs are needed.
   - **In House** - Lets you create a profile that helps you distribute the release build of the application internally and no device IDs are needed.

3. Click Continue.
4. Once you have created the profile, select All under Provisioning Profiles. There you will find a list of your profiles.

5. Locate the profile you just created and download it.

**Note:** Sometimes the profile takes just a minute to be available for download. In case you are not able to download the profile, just refresh the page after a few seconds and the download button should appear next to the profile name.
The Application Re-signing Process

Once you have the required Distribution Certificate and Distribution Provisioning Profile from the Apple portal, it’s time to re-sign the application. To re-sign the application we are going to follow these steps:

1. Gather the needed resources for the re-sign process
2. Open the IPA package and perform modifications (when needed)
3. Re-sign the application
4. Close the IPA package

After these steps the application is re-signed with the appropriate certificate and ready for distribution. This method applies for Web Deployment and also when working with MDM servers.

To test the re-signed application, you can use iTunes or the iPhone Configuration Utility. This way you can install the application locally in a device before updating all the distribution environment.

Gathering the required resources for the re-sign process

The required resources are:

- Distribution Certificate and Distribution Provisioning Profile
- ResourcesRules.plist
- Entitlements.plist
- App.IPA file

Distribution Certificate and Distribution Provisioning Profile

To proceed with the re-sign process, you must have:

- The Distribution Certificate installed.
- Access to the Provisioning Profile file location.
ResourcesRules.plist

You need to create a plain text file, an iOS “Property List” XML file with certain configuration on it. Make sure there are no additional characters, as it is important not to break the Property List structure.

The file content should be:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
    <dict>
        <key>rules</key>
        <dict>
            <key>*</key>
            <true/>
            <key>Info.plist</key>
            <dict>
                <key>omit</key>
                <true/>
                <key>weight</key>
                <real>10</real>
            </dict>
            <key>ResourceRules.plist</key>
            <dict>
                <key>omit</key>
                <true/>
                <key>weight</key>
                <real>100</real>
            </dict>
        </dict>
    </dict>
</plist>
```

Save the file as “ResourcesRules” and change its extension to .plist.
Entitlements.plist

You need to create another plain text file as you did with the ResourcesRules.plist. This file content should be:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
  <dict>
    <key>application-identifier</key>
    <string>PREFIX.bundle.id</string>
    <key>get-task-allow</key>
    <false/>
    <key>keychain-access-groups</key>
    <array>
      <string>PREFIX.*/</string>
    </array>
  </dict>
</plist>
```

- **application-identifier** – Specifies your app ID, i.e.: ET77GUVU7D.com.myCompany.appName.
- **get-task-allow** – Specifies whether the application should let external tools to be attached to its processes, must be false for non-testing scenarios.
- **keychain-access-groups** – Refers to the keychain information group that the application can access. Following the example above, the standard value for this key should be similar to ET77GUVU7D.* which refers to the certificate’s code.

Save the file as “Entitlements” and change its extension to .plist.

**Note:** The keychain-access-groups key is not needed in most cases. If the application didn’t implement access groups (or any keychain access), it is recommended to remove the section from the Entitlements.plist file.

App.IPA

This is the Application file to modify and then re-sign. The name of this file may change depending on the application.

At this point you should have gathered the following files in a folder in your file system:

- Distribution Provisioning Profile
- ResourcesRules.plist
- Entitlements.plist
- App.IPA file
Opening an IPA and performing modifications

To modify the application bundle the application IPA must be decompressed. The IPA format is just a particular extension which can be renamed to “zip” to be decompressed with any tool.

To open the IPA file follow this steps:

1. Locate the target file.
2. Rename the file by changing the file extension to .zip. E.G., a target file named app.ipa should be changed to app.zip.
3. Decompress the new ZIP file.
4. Open the “Payload” folder located in the ZIP file. myApp.app is inside of the Payload folder.

Once the .app package is unveiled, you can make the needed modifications to match the application IPA with the certificate and profile or your company.

About the Provisioning Profile

The embedded Provisioning Profile is just a copy of the provisioning profile. In most distribution and re-signing scenarios, in which the signature must be changed, this file needs to be either removed or replaced.

Removing the file – Once the file is deleted, you have to manually install the profile in the device, in order to install the application from a remote location. This allows the control of which users have access to the application.

Replacing the file – The most common scenario, when re-signing for internal distribution, is to replace the profile with a different one. The new profile is the one related to the certificate used to re-sign de build.
Locating and removing the Provisioning Profile

To remove or replace the embedded Provisioning Profile file, you need to locate and delete the file. Follow the steps:

1. Once the target IPA has been decompressed, locate `myApp.app` file should be inside the Payload folder.

2. Control-click (or right-click) over the .app file, and select “Show Package Contents”.

3. Locate the `embedded.mobileprovision` file and delete it.
Replacing an Embedded Provisioning Profile
When replacing the profile file, there are two possible scenarios. You can re-sign the application with a new certificate or using the same one. Using a new certificate implies a number of extra steps detailed below.

1. After deleting the file, rename your Provisioning profile with the exact same name “embedded.mobileprovision”.
2. Place the file in the same location.

When re-signing with a new Certificate, proceed to:
3. Locate the Info.plist file inside the target’s bundle (this is the same place as the embedded.mobileprovision file).
4. Open the Info.plist file with a Property List Editor application.
5. Change the bundle identifier’s value, CFBundleIdentifier, to the new app ID’s bundle identifier. In this case it would be com.mycompany.myapp.
6. Change the bundle name, CFBundleName, to the last component of the bundle identifier. In this case myapp.
7. Save the plist file and close the Property List Editor.
Optionally, you can change two other values:

- **CFBundleDisplayName** – to change the name of the application (the text that will be displayed under the icon in the device’s home screen, which can be different than the app name).
- **CFBundleIconFile** – to change the icon file name.

**Note:** It is very important not to perform changes over the Info.plist file without understanding the keys being changed.

### Re-sign the Application

Now, after gathering all the required resources and modifying the IPA file, it’s time to re-sign the package. To re-sign the build, you will use the codesign command line tool executed as follows:

```
/usr/bin/codesign -f -s "{SIGNER IDENTITY}" --resource-rules={RESOURCE RULES PATH} --entitlements {ENTITLEMENTS FILE PATH} {APPLICATION BUNDLE PATH}
```

Parameters to be replaced:

- **{SIGNER IDENTITY}:** The signer identity of the Certificate. E.G., “iPhone Distribution: My-Company”. To get this value, you can launch the Keychain Access app, and find the correct Certificate. The value you need to specify in the command is the name of the Certificate.

![](image)

- **{RESOURCES RULES PATH}:** Path to the ResourcesRules.plist you created previously.
- **{ENTITLEMENTS FILE PATH}:** Path of the Entitlements.plist file you created previously.
- **{APPLICATION BUNDLE PATH}:** Path to the target application’s bundle, E.G., `myapp.app`. 
For the previous example, the command should look as follows:

```
```

Once you have all the parameters in place, just open the **Terminal** application and paste the command. If the app was previously signed with a different certificate you should get the following message:

```
replacing invalid existing signature
```

Or the following message in case the app was previously signed with the same certificate:

```
replacing existing signature
```

The certificate used for the re-signing process must be installed in Mac OS X and must have an identity key associated. You can check about this using the Keychain Access application under the Keychains login in the My Certificates category, as shown below:

```
Note: The certificate must have an identity key associated or the re-signing process will fail.
```
**Closing the IPA package**

To finish the application re-signing process, you need to re-pack the .app file into an IPA file. Follow these instructions:

1. Locate the target application’s Payload folder.
2. Compress the Payload folder.
   a. In Mac OS X, this can be accomplished by secondary clicking (control-click or right-click) on the Payload folder and choosing “Compress ‘Payload’”.
3. Rename Payload.zip to app.ipa again.

**Note:** Before zipping the Payload folder, make sure there are no additional files inside the folder. Any additional file in the folder, even hidden files like the common .DS_store iOS file will make the application return errors when trying to install the generated IPA file.
Chapter 5
Advanced Configuration

Section 1: User Experience Customization
Section 2: Offline Support
Section 3: Support Enablement
User Experience Customization

The User Experience can be customized in several ways, through application re-branding, configuring the behavior of the user interface, configuring different Home content, changing the application launch module, or even including further customization by using SharePlus Web SDK or Native custom Development.
For further information about the Web SDK refer to SharePlus Launchpads Developer Guide.

Application Re-Branding

Re-branding can be achieved through the inclusion of customized assets and theme files that configure the application color settings. Image resources must be included on the application package to be distributed. You can either include theme files in the app package or download them from a remote URL.

1. **Download** sample graphic assets from [here](#). The sample graphic package includes the required images to be customized, also detailed in the *List of Re-Branding PNG Images* section below. Optional images are not included in the download.

2. **Create** the customized images. The images used for application rebranding are normally provided by a Visual Designer.

3. **Replace** the images in the application package to be distributed. In the file system folder where the application package is located, replace sample images with rebranded images.

4. **Replace or reference** the theme file according to the *Changing the Application’s Theme* section below. Themes can be loaded from the app package or from a remote location.

5. **Re-sign** the application.

Adding Re-Branding Assets

The Branding folder in your file system must contain all re-branding resources. You need to store the customized images in the folder before SharePlus build is created, so that they will be included in the application package to be distributed.

List of Re-Branding PNG Images

Re-branding images include:

- **Application Icons**, used in the Springboard, iTunes Artwork, device settings, Spotlight.
- **Splash screens**, images displayed when first launching the app while the interface loads.
- **Passcode Lock headers**, you can optionally change the Passcode Lock header.
App Icons
Springboard / iOS home screen

iTunes Artwork

Device settings

Spotlight search

Splash screens
When adding re-branding assets, some application icons are required and others are optional.

**Required** – Not including all the required app icons will result in a blank/black space within the app, affecting the application’s user experience.

**Optional** – These app icons may not be included, in which case the closest icons in size will be used instead. Note that these icons will be automatically stretched to match the required size.

In the tables below you will find the application icons, splash screens, and passcode lock images.

<table>
<thead>
<tr>
<th>Required Images</th>
<th>IOS VERSION</th>
<th>IMAGE LOCATION</th>
<th>DEVICE</th>
<th>SIZE (px)</th>
<th>TENTATIVE NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>iTunes</td>
<td>iTunes Artwork</td>
<td>-</td>
<td>512 x 512</td>
<td>iTunesArtwork</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1024 x 1024</td>
<td>iTunesArtwork@2x</td>
</tr>
<tr>
<td></td>
<td>iOS7</td>
<td>Springboard / iOS Home Screen</td>
<td>iPad</td>
<td>76 x 76</td>
<td>apilcon-76x76-ipad</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>152 x 152</td>
<td>apilcon-76x76@2x-ipad</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>120 x 120</td>
<td>apilcon-60x60@2x-iphone</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>iPhone</td>
<td>768 x 1024</td>
<td>default-700-Portrait-ipad</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1536 x 2048</td>
<td>default-700-Portrait@2x-ipad</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1024 x 768</td>
<td>default-700-Landscape-ipad</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2048 x 1536</td>
<td>default-700-Landscape@2x-ipad</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>iPad</td>
<td>640 x 960</td>
<td>default-700@2x-iphone</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>640 x 1136</td>
<td>default-700-568h@2x-iphone</td>
</tr>
<tr>
<td></td>
<td>iOS6</td>
<td>Springboard / iOS Home Screen</td>
<td>iPad</td>
<td>72 x 72</td>
<td>apilcon-72x72-ipad</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>144 x 144</td>
<td>apilcon-72x72@2x-ipad</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>57 x 57</td>
<td>apilcon-57x57-iphone</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>114 x 114</td>
<td>apilcon-57x57@2x-iphone</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>iPhone</td>
<td>768 x 1024</td>
<td>default-Portrait-ipad</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1536 x 2048</td>
<td>default-Portrait@2x-ipad</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1024 x 768</td>
<td>default-Landscape-ipad</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2048 x 1536</td>
<td>default-Landscape@2x-ipad</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>iPhone</td>
<td>320 x 480</td>
<td>default-iphone</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>640 x 960</td>
<td>default@2x-iphone</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>640 x 1136</td>
<td>default-568h@2x-iphone</td>
</tr>
</tbody>
</table>
### Passcode Images (Recommended)

<table>
<thead>
<tr>
<th>IOS VERSION</th>
<th>IMAGE LOCATION</th>
<th>DEVICE</th>
<th>SIZE (px)</th>
<th>TENTATIVE NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>iOS6 &amp; iOS7</td>
<td>Passcode lock header</td>
<td>iPad</td>
<td>474 x 146</td>
<td>header-ipad</td>
</tr>
<tr>
<td></td>
<td></td>
<td>iPhone</td>
<td>948 x 292</td>
<td>header@2x-ipad</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>292 x 77</td>
<td>header-iphone</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>584 x 154</td>
<td>header@2x-iphone</td>
</tr>
</tbody>
</table>

### Optional Images

<table>
<thead>
<tr>
<th>IOS VERSION</th>
<th>IMAGE LOCATION</th>
<th>DEVICE</th>
<th>SIZE (px)</th>
<th>TENTATIVE NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>iOS6 &amp; iOS7</td>
<td>Device Settings</td>
<td>iPad &amp; iPhone</td>
<td>29 x 29</td>
<td>appIcon-29x29</td>
</tr>
<tr>
<td></td>
<td></td>
<td>iPad</td>
<td>58 x 58</td>
<td>appIcon-29x29@2x</td>
</tr>
<tr>
<td></td>
<td></td>
<td>iPhone</td>
<td>58 x 58</td>
<td>appIcon-29x29@2x</td>
</tr>
<tr>
<td>iOS7</td>
<td>Spotlight Search</td>
<td>iPad</td>
<td>40 x 40</td>
<td>appIcon-40x40-ipad</td>
</tr>
<tr>
<td></td>
<td></td>
<td>iPad &amp; iPhone</td>
<td>80 x 80</td>
<td>appIcon-40x40@2x</td>
</tr>
<tr>
<td>iOS6</td>
<td>Spotlight Search</td>
<td>iPad</td>
<td>50 x 50</td>
<td>appIcon-50x50-ipad</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>100 x 100</td>
<td>appIcon-50x50@2x-ipad</td>
</tr>
</tbody>
</table>

#### Important considerations on graphic assets:

- All images must follow these general requirements:
  - No transparency, except the Passcode Lock header
  - 72 dpi
  - PNG format
- iTunes Artwork images must be named without the .png extension.
- Required images are mandatory.
- iOS6-related images are relevant only if you are using devices running iOS6.

### Changing the Application’s Theme (Color Palette)

You can adjust the application’s theme according to your needs by loading a configuration theme file. There are two different theme files, one for iOS7 and another for iOS6. The theme file is a JSON object that allows tweaking appearance properties for iOS controls.

### Overview of the iOS7 theme file

With the default.theme file, you can adjust the UI appearance in iOS7 environments.
The tintColor property can be used to define a key color in the application.

```
{
    "UIWindow": {
        "tintColor": [0, 145, 200]
    }
}
```

For further details, refer to Using Tint Color in Apple’s iOS7 UI Transition Guide.

**Overview of the iOS6 theme file**

With the default-ios6.theme file, you can adjust the color palette and also apply individual changes to some UI components like the top and bottom bars.

```
{
    "Variables": {
        "UINavigationBar": {
            "tintColor": [0, 0, 0]
        },
        "UISearchBar": {
            "Phone": {
                "tintColor": [0, 0, 0]
            },
            "Pad": {
                "tintColor": [0, 0, 0]
            }
        },
        "UIToolbar": {
            "tintColor": [0, 0, 0]
        },
        "UITabBar": {
            "selectedImageTintColor": [245, 130, 32]
        }
    }
}
```
Including the app’s theme file inside the Application package
To change SharePlus application’s theme for iOS6 and/or iOS7 you need to replace the default theme so it can be loaded by SharePlus.

Configuring a Remote application theme file
Using the Configuration File, you can specify the location of the iOS6 and/or iOS7 theme files.

```
<key>ThemeManagement</key>
<dict>
    <key>Enabled</key>
    <true/>
    <key>Settings</key>
    <dict>
        <key>DefaultTheme</key>
        <string>http://pathToRepository/default.theme</string>
        <key>DefaultTheme-ios6</key>
        <string>http://pathToRepository/default-ios6.theme</string>
    </dict>
</dict>
```

- **DefaultTheme, DefaultTheme-ios6** – Source location of the iOS7 and iOS6 theme files respectively. The source can be a remote URL or a local path inside the application.

These configuration settings can be empty, in that case the app will use the default theme files stored in the application package.
User Interface Behavior

In SharePlus you can configure the UI (User Interface) to better suit your needs. The whole configurations set is grouped into 5 different areas which are included under several features in the Configuration File.

- Swipe through documents
- Disable Auto-Lock
- Tabbed preview of documents
- Office documents preview
- Picture Board
- Document Preview on Tap

Swipe Through Documents

```
<key>DocumentViewer</key>
<array>
    <dict>
        <key>SwipeDocumentsEnable</key>
        <true/>
        <key>SwipeListTemplates</key>
        <string>109, 851</string>
    </dict>
</array>
```

- **SwipeDocumentsEnable** – Enable/Disable swipe through list documents when opening a single document from a library or list.
- **SwipeListTemplates** – Specify the list templates that will have the Swipe Documents feature enabled or disabled. Those SharePoint list templates are represented by numbers, e.g., 109 is a Picture Library and 851 is an Assets Library. Default values are: 109, 851.

Disable Auto-Lock

```
<key>GlobalSettings</key>
<dict>
    <key>Enabled</key>
    <true/>
    <key>Settings</key>
    <dict>
        <key>DisableAutoLockOnPreview</key>
        <false/>
        <key>DisableAutoLockOnSync</key>
        <false/>
        <key>DisableAutoLockMustBePluggedIn</key>
        <true/>
    </dict>
</dict>
```
- **DisableAutoLockOnPreview** – Prevents the device from locking itself while previewing documents.
- **DisableAutoLockOnSync** – Prevents the device from locking itself during the synchronization process.
- **DisableAutoLockMustBePluggedIn** – Enables the **DisableAutoLockOnSync** and **DisableAutoLockOnPreview** options only when the device is plugged in.

### Tabbed preview of documents

```xml
<key>DocumentViewer</key>
 <key>Settings</key>
 <dict>
   <key>TabsEnable</key>
   <false/>
   <key>TabsListTemplates</key>
   <string></string>
 </dict>
```

- **TabsEnable** – Enables/Disables the tabbed previewing for documents – Have all documents from a list opened in different tabs when opening a single document.
- **TabsListTemplates** – Enables/Disables the tabbed previewing feature only for the specified list templates. SharePoint list templates are represented by numbers, e.g., 109 is a Picture Library and 851 is an Assets Library. Default values: BLANK.

### Office documents preview

```xml
<key>DocumentViewer</key>
 <key>Settings</key>
 <dict>
   <key>ForceWebViewForOffice</key>
   <false/>
 </dict>
```

- **ForceWebViewForOffice** – When true, disables the use of *quick look*, a native view that can display any type of document but does not support swipe between documents (useful when opening several documents at the same time). The alternative, *web view*, can’t display Office Macros but it does support swipe. This setting is false by default.
**Picture Board**

This feature changes the Picture Board visualization settings, and is only available for Picture and Assets libraries.

```
<key>PictureBoard</key>
<dict>
    <key>Settings</key>
    <dict>
        <key>HideInfoButton</key>
        <true/>
    </dict>
</dict>
```

- **HideInfoButton** – When enabled, hides the button on the bottom right corner of the pictures.

**Document preview on Tap**

```
<key>GlobalSettings</key>
<dict>
    <key>Enabled</key>
    <true/>
    <key>Settings</key>
    <dict>
        <key>PreviewDocumentsOnTap</key>
        <false/>
    </dict>
</dict>
```

- **PreviewDocumentsOnTap** – Enables document preview as the default action when tapping an item. Item details are still available on a different view.
Home Customization

“Home” refers to the content displayed when accessing a Site Home or the Application Home within SharePlus application. You can configure different types of content to be displayed as Home, i.e., Application Launchpads, ReportPlus dashboards, and PDF files.


In addition, a Home can have more than one content assigned, in those cases you are able to switch between the different content by using a selector.

Home content

You can customize Site Homes or the Application Home to displayed customized content in SharePlus.

Application Home

Home content can be displayed in SharePlus when accessing the Application Home module in the SideBar. This module can be loaded by default when opening the application from scratch.

Site Homes

Portals and sites can both display home content in SharePlus, presenting the user with a customized view for a given site. Site Homes can be pre-configured and shared across sites, displaying different content depending on the SharePoint site’s context. Every site in SharePlus has a Site Home by default, which can be changed by configuration.
Understanding Homes, Pages, and View Parts

Home content can be specified manually (through the UI) or in the Configuration File by creating and configuring Pages, which include one or more elements called View Parts. View Parts are in charge of showing content on the screen when the Home is loaded.

Pages

In SharePlus, pages are not HTML web pages from a web site. A SharePlus Page is a different matter, it has a name for identification and it is used as a container of elements (View Parts). Pages are created in the Configuration File and they display Home content by including one or more View Parts. Each View Part can display an Application Launchpad, ReportPlus Dashboard, or a PDF file. When having more than one View Part, the user can swap between them through the User Interface.

View Parts

They are in charge of displaying content inside a Page, which is displayed on screen when a Home is loaded. There are two types of View Parts OOTB, Application Launchpads and ReportPlus Dashboards.

New View Part controllers can be added and configured for a custom application using the SharePlus Native SDK.

Pages and their View Parts are configured in the Configuration File (config.plist) at the same level as Sites, Accounts, and Features. Pages can also be pre-loaded by the application.
Home Configuration

The following table lists the available scenarios when configuring Homes.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Details</th>
<th>Configuration Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Home (In-App)</td>
<td>Configure a Site Home or Application Home through the UI.</td>
<td>Actions menu – Add as Home</td>
</tr>
<tr>
<td>Application Home</td>
<td>Create a Page holding home content and reference the PageName in the HomeManagement feature.</td>
<td>Configuration File – Pages, HomeManagement</td>
</tr>
<tr>
<td>Site Home</td>
<td>Add an entry to the MobileNavigation list and include the URL to the home content.</td>
<td>Site Configuration a.k.a. “MobileNavigation”</td>
</tr>
<tr>
<td>Default Home - Portal and sites</td>
<td>Configure the default page settings to be used across all sites of a portal.</td>
<td>Configuration File – Pages, Sites, HomeManagement</td>
</tr>
</tbody>
</table>

Application Home (In-app)

You are able to manually configure home content for a Site Home or the Application Home module through the User Interface.

**STEPS**

1. **Access the Actions menu**
   - Tap & hold over the item/content to open the Actions menu and select the *Add as Home* action.

2. **Then choose where to display your dashboard**
   - You can choose between two targets, Application (Application Home module) or This Site (the site where the item is located).
Application Home
The Application Home will be displayed when tapping the module in the SideBar. You can pre-
configure this Home in the Configuration File using an already created Page.

STEPS OVERVIEW
1. Creating a Page that includes one or more View Parts
2. Configuring your Page as the Application Home module

STEPS
1. **Create a Page that includes one or more View Parts**
   You need to define a new page in the Configuration File or use an already existing one.
   a. Open the Configuration File.
      Navigate to the Pages section to create a new Page.
   b. Define a Page.

```xml
<key>Pages</key>
<array>
  <dict>
    <key>PageName</key>
    <string>customPageName</string>
    <key>ShowSelector</key>
    <false/>
    <key>ViewParts</key>
    <array>
      ...
    </array>
  </dict>
</array>
```

The three elements are:

- **PageName** – The name used to reference the page when configuring a Home.
- **ShowSelector** – Enables/Disables the possibility to select between the different View Parts from a page/view container. Does nothing when there is only one View Part specified.
- **ViewParts** – Array of the View Part elements defined for the page.
c. Specify a View Part containing home content.

Each page has an array of view Part elements that you must specify:

```xml
...<key>ViewParts</key><array>
    <dict>
        <key>ViewControllerID</key><integer>0</integer>
        <key>Title</key><string>customWebDashboard</string>
        <key>Settings</key><dict>
            <key>Source</key><string>Documents/WebDashboardSample.html</string>
            <key>SourceType</key><integer>1</integer>
        </dict>
    </dict>
</array>...
```

The elements are:

- **ViewControllerID** – Specifies the View Controller to be used (ID defined in the app.plist). Possible OOTB values are:
  - 0 – to reference Application Launchpads or PDF files
  - 1 – to reference ReportPlus Dashboards
- **Title** – Specifies the title to be displayed on the top toolbar for this View Part.
- **Source** – Specifies the source location of the home content. The source can be an URL or a local path inside the application.
- **SourceType** – Sets the source type of the home content. Possible values are:
  - 0 – to reference an URL to download the file
  - 1 – to reference a path inside the application, in the application’s resource bundle or Local Files if the path starts with “Documents/”.

**Note on Sources:**

- When loading home content from local files, the path must start with “Documents/”. This is a useful approach for debugging and testing new home content.
- When downloading the content from a static URL, the URL location must be public or otherwise located under the same Domain than the Configuration File to share its credentials.
2. Configure your Page for the Application Home
   a. Open the Configuration File.
      Navigate to the HomeManagement feature in the Features section.
   b. Reference the Page
      You need to reference the Page you created by name (AppPageName key value).

   The Configuration File should be similar to:

   ```
   <key>HomeManagement</key>
   <dict>
      <key>Enabled</key>
      <true/>
      <key>Settings</key>
      <dict>
         ...
         <key>AppPageName</key>
         <string>customPageName</string>
         <key>ShowMainAreaButton</key>
         <false/>
         <key>ShowNavigatorButton</key>
         <true/>
         ...
      </dict>
   </dict>
   ```

   The relevant elements are:
   - **AppPageName** – Defines the page name to be used as Application Home. The page name must match the name of an existing page in the Pages section.
   - **ShowMainAreaButton** – When true, shows the Application Home button permanently at the top of the Main Screen or “Working Surface”.
   - **ShowNavigatorButton** – When false, hides the Application Home button in the SideBar.
Site Home
When working with Site Configuration (a.k.a. “MobileNavigation”), the Home property allows you to set home content for a portal or one of its sites. In addition, you are able to configure one or more contents at the same time.
Your Site Home can contain an Application Launchpad, a PDF document, a ReportPlus dashboard or any content that can be rendered in a web browser.

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home</td>
<td>Single line of text</td>
<td>A list of values separated by comma. Possible values are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- An URL - to reference a resource to be loaded.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- A page name - referencing a PageName key value defined in the Pages section of the Configuration File.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- “default” - to reference the default SharePlus site view, which is the native view that displays sub-sites, libraries, and lists.</td>
</tr>
<tr>
<td>HomeTitle</td>
<td>Single line of text</td>
<td>A list of titles separated by comma, for each of the Home values specified in the Home field respectively.</td>
</tr>
</tbody>
</table>

**Note:** The “default” value in the Home column will reference the DefaultSitePageName key value if this value was manually changed in the Configuration File.

For further details about the MobileNavigation list refer to [Site Configuration a.k.a “MobileNavigation”](#).

Default Home – Portal and sites
In SharePlus, Portals and sites can present a pre-configured customized view that can be shared with other sites. In this scenario, different content is displayed depending on the site’s context. You can configure a Site Home to be used by default for a portal and all its sites. To achieve this, Portals’ Site Homes need to be pre-configured using Pages and Sites in the Configuration File instead of using the MobileNavigation list.

**STEPS OVERVIEW**
1. Creating a Page that includes one or more View Parts
2. *(Optional)* Enabling the default list view for the Portal
3. Configuring your Page for the Portal’s Home
4. Defining the Site Home by default (DefaultSitePageName)
5. Adjusting inheritance settings (InheritParentPage)

**STEPS**
1. **Create a Page that includes one or more View Parts**
   You need to define a new page in the Configuration File or use an already existing one. For details on how to specify a Page and a View Part including home content see the previous scenario [Application Home](#).

---

**SharePlus 4.0**
**Administrator Guide**
2. **Optional** Enable the default list view for the Portal

When pre-configuring custom home content as Portal Home, the default content is replaced.

A View Part with SharePlus default list view needs to be added, so you can switch between content (default list view and the new home content) by using the selector.

**Note:** SharePlus default list view shows all the sub-sites, libraries, and lists for portals or sites. The bottom bar also includes the site’s Favorites.

In the Configuration File, the Page should include the following View Part:

```xml
<key>ViewParts</key>
<array>
  <dict>
    ...
    <key>Settings</key>
    <dict/>
    <key>ViewControllerID</key>
    <integer>2</integer>
    <key>Title</key>
    <string>View Part Title</string>
    ...
  </dict>
</array>
```

The elements are:

- **ViewControllerID** – Specifies the ID defined in the app.plist for the View Controller to be used. For the default list view the value specified must be 2.
- **Title** – Specifies the title to be displayed on the top toolbar for this View Part.

3. **Configure your Page for the Portal’s Home**

   a. Open the Configuration File.
      Navigate to your Portal configuration in the Sites section.

   b. Reference the Page in the pre-configured Portal
      You need to reference the page you created by name (PageName key value).
The Configuration File should be similar to:

```xml
<key>Sites</key>
<array>
    <dict>
        ... 
        <key>PageName</key>
        <string>yourCustomPage</string>
        ... 
    </dict>
</array>
```

4. **Define the Site Home by default (DefaultSitePageName)**
   b. Reference an existing page by name (PageName key value).

5. **Adjust inheritance settings (InheritParentPage).**
   Go to *HomeManagement* to enable inheritance between the parent site and its sites.
   The *HomeManagement* feature in the Configuration File should be similar to:

```xml
<key>HomeManagement</key>
<dict>
    <key>Enabled</key>
    <true/>
    <key>Settings</key>
    <dict>
        ... 
        <key>DefaultSitePageName</key>
        <string>yourCustomPage</string>
        <key>InheritParentPage</key>
        <false/>
        ... 
    </dict>
</dict>
```

The existing elements are:

- **DefaultSitePageName** – References the page to be used as Site Home by default. The page name must match the name of a page in the Pages section.
- **InheritParentPage** – When true, sub-sites inherit pre-configured pages from their parent site.
**Startup Module Customization**

In SharePlus you can change the module to be displayed when the application starts. The possible module choices are the Application Home, the Sites Hub, or any other module specified by configuration.

**Starting with a pre-configured module**

The module to be displayed when the application starts can be pre-configured in the Configuration File. To specify this module, you need to configure the `StartupModule` setting from Global Settings feature.

```xml
<key>GlobalSettings</key>
<dict>
    <key>Enabled</key>
    <true/>
    <key>Settings</key>
    <dict>
        ...  
        <key>StartupModule</key>
        <string>FavoritesRow</string>
        ...
    </dict>
</dict>
```

This setting can be empty and the possible values are the Row Identifiers specified in the `NavigatorTableDefinition.plist` file.

Current OOTB values are:

- **DocumentsHub** – to reference the Documents Module.
- **Favorites** – to reference the Favorites Module.
- **Home** – to reference the default Application Home Module.
- **Search** – to reference the Search Module.
- **SitesHub** – to reference the Sites Module.
- **Social** – to reference the Social Module.

**Starting with the Application Home**

When no module was specified by configuration and the Application Home module is assigned, SharePlus will display this module when opening the app from scratch. The module can be pre-configured in the Configuration File or manually assigned through the UI.

**Starting with the Sites Hub**

SharePlus will display the Sites Hub when the application starts only when:

- No module was specified by configuration
- The Application Home module was not assigned.
Offline Support

Understanding Offline Support
SharePlus provides the ability to cache SharePoint content locally in the device, allowing you to access information even when not connected. This significant advantage comes with the obvious consequence of having to synchronize your content later with the server. The Offline Support capability is enabled at the List level, and each list or document library is configured separately. Changes made offline are synchronized back to the server. Items with conflicts that need to be resolved will be displayed as Pending Updates for the list.

Offline Settings
When enabling the Offline Support for a list, several synchronization settings can be configured, like choosing the Synchronization Mode and the SharePoint List View to be used.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offline Support</td>
<td>Enables offline browsing for a list, when turned on other settings will be enabled.</td>
</tr>
<tr>
<td>Sync Mode</td>
<td>Choose between Browsed and Background modes.</td>
</tr>
<tr>
<td>Sync List View</td>
<td>Select the SharePoint List View to be used.</td>
</tr>
<tr>
<td>Only Wifi</td>
<td>Enable the background process to run only when a wireless network is present.</td>
</tr>
<tr>
<td>Download Documents</td>
<td>If disabled, only the item properties (metadata) are saved and not the actual files.</td>
</tr>
<tr>
<td>Max Document size</td>
<td>Filters the downloaded documents by size, only documents with less size than the specified value will be synchronized.</td>
</tr>
<tr>
<td>Query Page Size</td>
<td>Specifies the number of items per block that are requested from the server.</td>
</tr>
</tbody>
</table>
Configuring Offline Support

SharePoint List Views
SharePoint List Views are pre-defined configurations created on the server to organize and filter the items of a list, depending on the needs of different SharePoint groups of users. Selecting the SharePoint list view is mandatory to configure the Offline Support in SharePlus.

SharePlus Synchronization Modes
SharePlus synchronization process can be done following two different Modes, Browsed and Background.

Background mode
This mode synchronizes the list content while SharePlus is active and the device is online. The sync process runs in the background and is triggered when SharePlus is idle for more than 30 seconds.

This process requests a block of list items to the SharePoint server to store them locally. After storing the block of items, if SharePlus remains idle the process continues to synchronize the rest of the list items. The frequency in which the synchronization process is triggered can be specified in the Configuration File (SyncIdleTime setting), or directly in the device through the UI (Idle time)

When SharePlus is moved to the background (the app state changes and is no longer active), the synchronization process is triggered. The process runs for about 7 minutes and then stops due to platform limitations.

**Note:** The background sync mode is suspended when:
- The user interacts with the device, to free bandwidth for navigation.
- SharePlus is sent to the background.
- The device is suspended.

“Browsed” mode
With this mode, only the items, documents, and files that you open while online will be available offline. You have to manually synchronize changes by re-opening the list contents.

Changes made outside SharePlus, directly on the SharePoint server, are not reflected in the application until you reopen the documents with changes.

**Note:** When using the Browsed mode, items deleted from the SharePoint server are not detected in SharePlus. Instead, these items remain on the application and take up storage space. In this cases, using the Erase Downloaded Content option is recommended to reset all the locally stored content.
Understanding Synchronization

The Synchronization process is activated when there are lists with offline support enabled and the device is idle for a fixed number of seconds. The frequency in which the process is triggered can be modified through an advanced configuration setting (*SyncIdleTime*).

Once the Synchronization process is finished, all the list contents will be cached locally.

You can open *Synchronization* from the SideBar menu to start and also monitor the synchronization process. Recent activity and errors are displayed, as shown in the image above.

Lists go through the synchronization process while the sync status icon is red.

Once the process is successfully completed, the list is synchronized and the sync status icon turns green.

**Note:** A document bigger in size than the Max Document Size setting (15 Mb by default) won’t be synchronized. As a consequence, the sync status icon will remain in color red to notify the document’s status.
**The Synchronization status icons**
The sync status icons are used to illustrate different states when having Offline Support enabled.

- **Grey icon** - A list or item is not yet synchronized.
- **Red icon** - A list is being synchronized or has pending changes.
- **Green icon** - A list or item is fully synchronized with the SharePoint server.
- **Blue icon** - Browsed Mode is selected as the synchronization method.

**Conflict Resolution**
When synchronizing changes back to the server you may encounter conflicts, for example, when two or more users edit an item at the same time. SharePlus resolution engine helps you decide how you want to proceed in those cases.
Permission Changes on the server while Offline

While working Offline, permission changes in SharePoint can’t be replicated to SharePlus at that time. However, you will get those changes later when going Online again and content is synchronized. Synchronizing your work can lead to several conflicts which need to be addressed, like having SharePoint permissions changed for a site, a list or even a specific document.

Structural changes on the server

The synchronization process updates all content changes from the SharePoint server. Changes in the structure, however, are not updated. This is true for either online or offline scenarios. Structural changes in the SharePoint server include, for example, changing the default list View or adding/deleting a column from a list.

By using list Views, you can filter or manage the visibility of a list. Because of that, it is important to have list Views up-to-date. SharePlus allows you to retrieve structural changes for a list by using the Refresh Cache option. Alternatively, using the Erase Downloaded Content option you can delete all the locally stored content for the list. After performing this operation, all the server information for the list will be downloaded again to SharePlus.
Support Enablement

SharePlus provides several in-app tools to facilitate access to support to the user. These different tools can be customized separately and are centralized in the Help Center.

These tools are:

- In-App Help Documentation
- Mail Feedback Tools
- UserVoice
In-App Help Documentation
SharePlus displays a help documentation to the user within the application. This help documentation consists of a few relevant topics which can be embedded on the application or accessed through external links.
Mail Feedback Tools

By default, SharePlus sends user feedback to mobileapps@infragistics.com. SharePlus Enterprise allows the redirection of user inquiries to an internal help desk, customizing the Email Subject and recipient.

UserVoice

UserVoice is a helpdesk system used to provide customer support and it can be integrated to SharePlus requiring a custom implementation.

SharePlus Free and Subscription versions integrate UserVoice, allowing users to send feedback and request support right from within the application.

The Enterprise version can be integrated with a custom UserVoice account, so that an internal help desk can receive notifications and respond to the company users. A custom development is required to integrate UserVoice following the company requirements.

The image below shows UserVoice’s helpdesk interface.
Configuring Support Tools
SharePlus Enterprise allows you to redirect user inquiries by customizing the feedback email and the help’s URL.

In-App Help with External URLs
SharePlus’ help is configured through the Help feature in the Configuration File. HTTP or HTTPS based URLs are accepted in all cases.

```xml
<key>Help</key>
<dict>
    <key>Enabled</key>
    <true/>
    <key>Settings</key>
    <dict>
        <key>HelpURL</key>
        <string></string>
        <key>HelpURL-iPad</key>
        <string></string>
        <key>HelpURL-iPhone</key>
        <string></string>
        <key>ShowAboutUs</key>
        <true/>
        <key>ShowFAQ</key>
        <true/>
        <key>ShowHelp</key>
        <true/>
    </dict>
</dict>
```

- **HelpURL** - URL to an online help page that replaces the app’s help (only if no per-device URLs are defined).
- **HelpURL-iPad/HelpURL-iPhone** - Replaces the app’s help when SharePlus is running in an iPad/iPhone.
- **ShowAboutUs/ShowFAQ/ShowHelp** – Shows/Hides the help/FAQ/About US on the app’s global settings.
Mail Feedback Support

The Feedback feature is used to customize feedback emails. The accepted format for this feature is to specify addresses separated by commas, e.g., example1@domain.com, example2@domain.com.

```
<key>Feedback</key>
<dict>
    <key>Enabled</key>
    <true/>
    <key>Settings</key>
    <dict>
        <key>BCC</key>
        <string></string>
        <key>Body</key>
        <string></string>
        <key>CC</key>
        <string>Sent via [ProductSignature] </string>
        <key>Subject</key>
        <string>[DocumentName] - [FeedbackType]</string>
        <key>To</key>
        <string>mobileapps@infragistics.com</string>
    </dict>
</dict>
```

- **To** - Default email addresses to be included in the TO field when composing a new message. This setting must contain at least one valid email address, as it is used to report errors.
- **CC/BCC** – Default email addresses to be included in the CC/BCC field.
- **Subject** – Default Subject when composing a new message.
  Available Tokens are: [DocumentName] and [FeedbackType].
- **Body** – Text to be used as the default Body in the message.
  Available Tokens are: [DocumentName], [DocumentURL], [ProductSignature], [NewLine], and [FeedbackType].
Chapter 6
Advanced Features

Section 1: Social Networking
Section 2: ReportPlus Integration
Section 3: Search
Section 4: Application Integration
Understanding Social Networking

Social collaboration has been reorganized and greatly enhanced through SharePlus new Social Module, providing a centralized and feature-rich User Experience (UX) for SharePoint 2013 social features. The Social Module is available for SharePlus Enterprise versions.

With SharePlus Enterprise, the Social Module lets you share your ideas, as well as discover and keep track of your colleagues and their work through their profiles, posts, and updates in the Timeline. You can connect with people, start conversations, and also access their content and activity feed. Your SharePoint personal storage (My Content) is available in this module, including both your public and private document libraries.

The Module is divided in four sections: Timeline, People, My Content and My Profile.
Timeline

The Timeline tab provides you access to the following sections:

- **Newsfeed** – Includes public posts and other people’s activities. Besides creating new posts, or mark items you like, you can also start conversations with colleagues here.

- **Alerts** – Displays all the recent notifications on resources that were modified. SharePlus allows you to keep close track of important resources by marking them with the Alert Me option.

- **Mentions** – Gives quick access to all the posts in which you’ve been mentioned.

- **Following** – Displays posts from the people and content you are following.

- **Likes** – The Likes tab includes all the posts you have “liked”.

**Note:** Timeline sections require SharePoint 2013 to be displayed in SharePlus. Alerts is the only exception and is always available in the Timeline.
People
The People tab displays the list of people that you are following and the ones following you. In this section you are able to connect with people, access their content and activity feed and also start conversations with them.

My Content and Profile
The Social Module gives you access to the content of your SharePoint’s personal site, including both your public and private document libraries. Also, you can visualize or edit your personal information shared on your corporate social network.
Recommended and Followed content

When working with SharePlus Enterprise and SharePoint 2013 server you can get recommendations on sites and documents that may be relevant to you. The portal suggestions are included on the SharePlus list view under the Recommended section. The sites, documents, and people that you follow is also listed in the Sites, Documents, and Social Modules respectively.

Through the SharePlus User Interface you can choose to Follow/Unfollow content from a SharePoint 2013 server.
Configuring Social Networking

Configuring your Personal SharePoint site
The integration with My Site (SharePoint 2007, 2010) or OneDrive (SharePoint 2013) provides users with a personal site, a central location to manage and store documents, content, links, and contacts.
To configure the feature’s integration with SharePlus, you need to configure attributes from an existing site/connection in the Configuration File.

MySite feature Overview

```xml
<key>Sites</key>
<array>
  <dict>
    <key>SupportMySite</key>
    <true/>
    <key>MySiteHost</key>
    <string>https://rootUrl/personal/[accountName]</string>
  </dict>
</array>
```

- **SupportMySite** – This setting can be enabled for only one site at a time.
- **MySiteHost** – Specify the URL where the personal site’s root is hosted. You can use the `[accountName]` token and it will be replaced with the user’s account name.

**Note:** When configuring the My Site feature for SharePoint 2007/2010, you don’t need to specify the URL. My Site is automatically configured once the user enters the site. For this to work, MySite feature must be enabled and the current user needs to have My Site associated with his credentials.

Configuring My Profile
The integration with My Profile (SharePoint 2007, 2010, 2013) is closely-related to My Site and, basically, it consists on a public profile page located in SharePoint.
To configure the feature’s integration with SharePlus, you need to configure attributes from an existing site/connection in the Configuration File.
MyProfile feature Overview

- **SupportMyProfile** – This setting can be enabled for only one site at a time.
- **ProfileHost** – The URL where the person’s profile pages are hosted. When empty, the root URL will be concatenated with “/my”.

```xml
<key>Sites</key>
<array>
  <dict>
    <key>SupportMyProfile</key>
    <true/>
    <key>ProfileHost</key>
    <string/>
  </dict>
</array>
```
ReportPlus Integration

Data visualization can be used to communicate insight, but sometimes the tool gets in the way and it is really difficult to achieve both a useful and also visually compelling result. The user experience (UX) is crucial because it’s all about the user needs, that’s why SharePlus invested on capitalizing the ReportPlus interactive, easy-to-use and fast data visualization experience. As a result from the integration with ReportPlus, SharePlus 4.0 introduces the possibility to design and create data visualizations from data stored in SharePoint.

Introduction to ReportPlus

You can design, create, or modify ReportPlus dashboards and achieve different data visualizations. The large number of visualization options, grid, text, trend line, map, gauge, chart, and financial chart views, help you maximize business insights.
Dashboards are highly interactive, you can navigate to any of the composing widgets as shown below.

Creating Dashboards
Creating new dashboards is as simple as adding a new dashboard, selecting your data, and customizing the visualization. That being said, let’s dig a little deeper and create a dashboard from scratch:

1. **Add a new ReportPlus Dashboard**
   As other documents, you can add dashboards and share them with other users by uploading them to the SharePoint server.

   ![Dashboard Templates](image)

   **Note:** Unlike other documents, you can add dashboards from Local Files tab in the Documents module.

2. **Select your data**
   You can connect to data sources like lists and documents, including calendars, contacts, images, Excel files. SharePoint lists and Excel files are particularly interesting, as you can apply many different visualization options to their data.
Perform a tap & hold gesture over your dataset and then drag it to a widget placeholder.

Dashboards connect to SharePoint data using the current user’s credentials.

3. **Customize the visualization**
   Once a dataset was dragged to an empty widget, the Chart Wizard is displayed.

The Wizard helps you create a chart visualization, but you can, alternatively, cancel the wizard to gain full control of the visualization. Following this path, you can dig deeper and apply filters or conditional formatting to further tweak the information displayed.
You can get help by tapping the question mark icon (?) in the editor screen.
Working with Pivot Tables

Pivot Tables are very powerful data summarization tools that help you analyze data. Let’s create one using a custom SharePoint list as an example. The custom list (Business Opportunities from SharePlus Demo site) contains information about accounts and their value:

<table>
<thead>
<tr>
<th>Account</th>
<th>Category</th>
<th>Estimated Value</th>
<th>Executive</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agoos/Lovera Architects</td>
<td>SMB</td>
<td>180000,00</td>
<td>Zaruba Evans</td>
<td>United Kingdom</td>
</tr>
<tr>
<td>Clarke Caton Hintz, A Professional Corporation</td>
<td>SMB</td>
<td>9000000,00</td>
<td>Yancoh Hauzen</td>
<td>Japan</td>
</tr>
<tr>
<td>CS Davidson, Inc.</td>
<td>Corporate</td>
<td>20000,00</td>
<td>Nicolas Igarte</td>
<td>Brazil</td>
</tr>
<tr>
<td>Boles, Smyth Associates, Inc.</td>
<td>Retail</td>
<td></td>
<td></td>
<td>Brazil</td>
</tr>
<tr>
<td>W2A Design Group</td>
<td>Retail</td>
<td>3500000,00</td>
<td>Nicolas Igarte</td>
<td>Australia</td>
</tr>
<tr>
<td>Boles, Smyth Associates, Inc.</td>
<td>Corporate</td>
<td>150000,00</td>
<td>Zaruba Evans</td>
<td>United Kingdom</td>
</tr>
<tr>
<td>Macintosh Engineering</td>
<td>Retail</td>
<td>150000,00</td>
<td></td>
<td>Germany</td>
</tr>
<tr>
<td>Paragon Engineering, PC</td>
<td>SMB</td>
<td>3540000,00</td>
<td>Nicolas Igarte</td>
<td></td>
</tr>
<tr>
<td>Urbitran Associates, Inc.</td>
<td>Retail</td>
<td></td>
<td>Zaruba Evans</td>
<td></td>
</tr>
<tr>
<td>Agoos/Lovera Architects</td>
<td>Retail</td>
<td></td>
<td>Zaruba Evans</td>
<td></td>
</tr>
<tr>
<td>ABHA Architects, Inc.</td>
<td>Retail</td>
<td>1750000,00</td>
<td>Zaruba Evans</td>
<td></td>
</tr>
<tr>
<td>Bower Lewis Thrower Architects, Ltd.</td>
<td>Corporate</td>
<td>120000,00</td>
<td>Zaruba Evans</td>
<td></td>
</tr>
</tbody>
</table>

1. **Select your data**
   Drag your dataset to one of the dashboard widget placeholders. Then, cancel the wizard to manually customize the dashboard.

2. **Start using the Pivot Table Editor**
   List countries as rows using the Pivot Table Editor located at the bottom of the Widget Editor.
   The Pivot Table editor gives you the ability to slice-and-dice data according to your needs.
Just drag & drop fields in the corresponding placeholders like shown with Country below:

If you want to hide <null> values, tap over the Country field and enable Filter Empty Values.
3. **Summarize data**

Drag the Estimated Worth field to the Values placeholder, the Sum aggregation function is selected by default for numeric fields.

4. **List the columns**

Now add the Category field to the Columns placeholder, in order to dynamically create a column for every existing category.

5. **Add a quick filter**

Define a quick filter to dynamically change the dashboard information at run-time. This means that you can later open the dashboard and filter the information by the *Executive* field values without editing the dashboard.
The Pivot table is ready, you are now able to quickly summarize and analyze your data.

Still, dashboards can be customized in many different ways, including the Visualization type. The **grid view** used on this example is the visualization type selected by default, but you can choose between several visualization types.

**Customizing the Visualization**

When customizing your dashboard you can style a dashboard or the individual widgets separately. In addition, the visualization type (e.g., grid, gauge) can also be styled depending on the visualization type.

Following the same example (Pivot Table based on a custom SharePoint list), let’s start by changing the visualization type from the default Grid to a Linear Gauge View. After that, let’s review style customizations for the chart first, then for the widget, and finally for the entire dashboard.

Linear Gauges are particularly useful to compare values of different rows side by side.

1. **Customize the Grid view**
   Before switching to the Gauge view, let’s take a look to the previous Pivot Table and how it can be customized.

   *Through the Grid Settings you can configure the table borders, text alignment, and alternating background colors among other settings.*
2. **Switch to the Gauge view**

After making the switch, clear the Pivot Table Editor except by the Estimated Worth aggregated function. To clear a field just drag it outside the editor.

3. **Adjust/Tweak the Gauge view**

The Linear Gauge can be used to display the Estimated Worth totals for each employee. Add the *Executive* field to *Rows*, so that the employees assigned are listed as rows.
4. **Customize the Gauge view**

Through *Bounds* in the Gauge Settings you can configure the minimum and maximum values, and the bands range or color among other settings.

5. **Switch to other views**

Change how your data is presented by switching between the different data visualizations.
6. **Customize the widget**

Once the Visualization is ready tap Done to go back to the dashboard general view.
Access the widget style properties to configure the margins, borders, color, and padding values.

7. **Customize the dashboard**

The dashboard settings include many different customizations, including:

- **Layout** – A dashboard is composed of widgets, this setting allows you to specify the amount and disposition of these widgets
- **Style** – Specify general dashboard properties, select a background image, or choose between predefined dashboard themes.
- **Password** – You can define a password that will be required to edit the dashboard.

Open the dashboard settings and customize your dashboard:
Creating new dashboards can be a fast and simple task when you need it to be. Nevertheless, there are a large amount of possible configurations and customizations. Dashboard tweaking may take more time if you who want to consider every little detail to achieve a tailor-made visualization.

**SharePlus Enterprise Features**

ReportPlus dashboards display information contained in one or more data sets. Each widget in a dashboard displays information from a specific data set, retrieved from a data source e.g.: Excel spreadsheets, SharePoint files and lists, SQL databases.

With SharePlus Enterprise versions you are not limited to SharePoint content, instead you can connect to more than 20 different enterprise data sources, including:

- Excel spreadsheets
- SharePoint files and lists
- Analysis Services
- SQL databases
- Reporting Services.
- Dropbox
- Google Drive
The New Data Source wizard guides you to configure a new connection, requesting a set of credentials with permissions to read the data.

Disabling ReportPlus Integration
You can configure ReportPlus Integration through the ReportPlus feature in the Configuration File. The feature can be completely disabled through configuration.

```
<key>ReportPlus</key>
<dict>
   <key>Enabled</key>
   <true/>
</dict>
```
Nowadays, search can be considered an integral part of any successful SharePoint-related application. As site and folder structures continue to grow, information needs to be easy to find. That is why SharePlus has included a Search module in the SideBar (Enterprise versions) and also an advanced filtering tool for SharePoint lists that allows you to locate relevant items.

**Understanding SharePlus Search**

**List Filtering Support**
SharePlus allows users to filter items within lists and libraries by keyword (name or title) or using the *Advanced Search* tool. These two methods are suited for different situations, the *Advanced List Search* is actually a richer and more complex criteria-based filtering method.

**“Standard” Filtering**
This filtering method allows the user to locate items based on their name or title. The filtering bar used for this method is displayed on top of any list browsed in SharePlus, and it can be used to quickly found an item within a list filtering by its name or title.

**Advanced List Filtering**
The *Advanced Search* tool allows users to define complex and powerful filtering criteria within a list. This type of filtering complements the “standard” filtering for those cases that the user needs more control or detailed criteria. To open the *Advanced List Search* tap the *Advanced* button on the UI, a popover panel is displayed below the filtering bar as shown in the image below.
When using the Advanced Search, there are a number of elements to take into account:

- **The Advanced button** displays the popover panel.
- **Fields** and also **Conditions** add relevant information to be used by the filtering criteria, e.g., “Equals”, “Is null”.
- Extra fields can be included with **And/OR operators** to expand the filtering criteria.
- **Sub-folders** can be included.

**SharePlus Enterprise Search**

The Search module allows you to search server content by keyword and retrieve results from all connected portals at the same time. Content is indexed by the SharePoint Search Engine, leveraging the entire SharePoint Search power, including full-text search of all kinds of file formats and even external resources.

The Search Hub provides full control when searching for SharePoint content, you must specify a number of filters that help you narrow the retrieved results.

<table>
<thead>
<tr>
<th>Filter</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scope</td>
<td>Define the relevant area for the search.</td>
<td>This Site, All Sites, People</td>
</tr>
<tr>
<td>Result Type</td>
<td>Specify the expected result type.</td>
<td>All Results, Word Documents, PDF</td>
</tr>
<tr>
<td>Modified Date</td>
<td>Refine the search criteria by date</td>
<td>Any Modified Date, Past Week, Earlier</td>
</tr>
<tr>
<td>Author</td>
<td>Filter the search by author name</td>
<td>Any potential author name</td>
</tr>
</tbody>
</table>
To open unauthorized content you need to provide the required credentials.

**FAST Search**

FAST Search APIs are compatible with SharePoint Search APIs, therefore, the basic core of functionalities provided by FAST Search are supported by SharePlus.

**Customizing the Search Module**

Search settings can be customized to include/exclude scopes, allowing a better control of the relevant area to be included on the actual search. The GlobalSearch feature from the Configuration File includes three attributes that may be configured to refine the search experience.

**GlobalSearch feature Overview**

```
<key>GlobalSearch</key>
<dict>
  <key>Enabled</key>
  <true/>
  <key>Settings</key>
  <dict>
    <key>IncludeScopes</key>
    <string></string>
    <key>ExcludeScopes</key>
    <string>Global Query Exclusion, Rank Demoted Sites</string>
    <key>ProfileDocumentsScopes</key>
    <string>All Sites</string>
  </dict>
</dict>
```

- **IncludeScopes** – Search scopes to be displayed. Must be based on the server and can be empty.
- **ExcludeScopes** – Filtered search scopes. SharePlus OOTB has internal scopes that may not be relevant to the user. Value by default: Global Query Exclusion, Rank Demoted Sites.
- **ProfileDocumentsScopes** – Scopes to be included when searching documents associated to the user’s profile. Assumes all scopes when not specified.
Application Integration

Web-to-App Integration

“Web-to-App” Integration feature allows users to perform certain actions within SharePlus, invoking these actions from HTML content. SharePlus is automatically opened after one of the available actions was called.

SharePlus links or S+ links are basically custom URL Schemes¹³ that start with $\text{splus://}$ instead of $\text{http://}$.

The image below shows a web page about to instruct the browser to invoke SharePlus and navigate to the targeted element natively.

Basic SharePlus Links

Opening a site, list or item in SharePlus is as simple as replacing the $\text{HTTP}$ protocol by $\text{splus}$ URL scheme.

Just take the original URL:

\[
\text{http://<portal>/site/list/item.docx}
\]

And replace:

\[
\text{splus://<portal>/site/list/item.docx}
\]

¹³ URL Schemes are used by iOS applications to initiate specific requests, for further details refer to Apple URL Scheme Reference in the iOS App Programming Guide.
SharePlus Custom URL Schemes (SharePlus Links)

Custom URL Schemes can include the URL to the SharePoint’s site or resource (web, list, item or document) and the action to be performed (if any). In some cases, only application actions can be used, like displaying Favorites or Local Files.

Structure of SharePlus custom URL Schemes

Invoking actions over SharePoint resources:

```
“splus://” + <SharePoint’s URL> [+ <action>]
```

Invoking SharePlus application actions:

```
“splus://” + <action>
```

It is also supported to use “spluss://” for HTTPS channels, as shown in the example below:

```
spluss://<portal>/site/list
```

URL Encoding

When working with SharePlus links, if the URL or the action’s parameters include special characters that are not part of the ASCII character-set you need to encode them. For example, “=” must be replaced with “%3d”.

You can encode/decode URLs using free tools like: [http://meyerweb.com/eric/tools/dencoder/](http://meyerweb.com/eric/tools/dencoder/)

SharePlus Custom URL Schemes

A set of custom URL Schemes can be used to call different actions within SharePlus, enabling a considerable number of application-related behaviors.

<table>
<thead>
<tr>
<th>Actions Group</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SharePoint Navigation</td>
<td>Navigation to Webs, Lists, Items or Documents</td>
</tr>
<tr>
<td>Edition</td>
<td>Add/Edit Items or Documents</td>
</tr>
<tr>
<td>SharePlus Basic</td>
<td>Navigation within SharePlus components (e.g., Local Files, Favorites, Help)</td>
</tr>
<tr>
<td>Configuration Change</td>
<td>Change the Remote Configuration URL or Application Launchpad URL.</td>
</tr>
<tr>
<td>Search</td>
<td>Open a query through the Advanced List Search or Enterprise Search (Web).</td>
</tr>
</tbody>
</table>

SharePoint Navigation Actions

When invoking navigation actions to SharePoint Webs or Lists, there are three different modes in which you can access the content:

- **InWeb** – A native view is opened within SharePlus displaying the navigation action.
- **InSafari** – iOS Safari browser is invoked by SharePlus, opening with the navigation action.
- **Native** (Default mode) – The navigation action is displayed within SharePlus
Web Samples

- Navigating to a SharePoint Web, opening Safari and requesting the mode to the user.

```splus://<portal>/site?action=view&mode=InSafari
splus://<portal>/site/multimedia?action=view&mode=AskUser```

The user is prompted to select the mode to be used (SharePlus, Safari, or Browser).

List Samples

- Navigating to a SharePoint List, to the default view and also to a specific List View.

```splus://<portal>/site/calendar?action=view
splus://<portal>/site/calendar?action=view&viewName=All%20Events```

- Browsing a list's folder contents, both samples below can be used indistinctly.

```splus://<portal>/site/Shared%20Documents/RSS%20Samples
splus://<portal>/site/Shared%20Documents/RSS%20Samples?action=query```

Document Samples

- Navigating to the details of a SharePoint Document.

```splus://<portal>/site/multimedia/Far%20Away.mp3```

- Opening a SharePoint Document

```splus://<portal>/site/multimedia/Away.mp3?action=viewdocument```
Edition Actions
These actions may include initial field values to be loaded when opening the SharePoint Add/Edit view. When initial field values are included, there are three important considerations:

- The field names must match the same names used in SharePoint, including the ows_prefix.
- All field values must be URL escaped.
- All field values must be included in a specific format, as shown below.

<table>
<thead>
<tr>
<th>Field type</th>
<th>Format</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single/Multiple lines of text, Numbers, Currency, Hyperlink</td>
<td>These values need no format.</td>
<td>Text: ows_TextField=Text%20Value Number: ows_NumberField=15</td>
</tr>
<tr>
<td>Choice</td>
<td>Values must be separated by “;#” characters (%3B%23 when escaped).</td>
<td>Single: ows_ChoiceField=%3B%23Value1%3B%23 (Non-escaped: ows_ChoiceField=#Value1;#)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Multi: ows_ChoiceField=%3B%23Value1%3B%23Value2%3B%23 (Non-esc: ows_ChoiceField;#Value1;#Value2;#)</td>
</tr>
<tr>
<td>Lookup, Person or Group</td>
<td>ItemID;#Name</td>
<td>ows_LookupField=103%3B%23Test (Non-esc: ows_LookupField =103;#Test)</td>
</tr>
<tr>
<td>Date and Time</td>
<td>yyyy-MM-dd'T'HH:mm:ss'Z'</td>
<td>ows_DateField=2012-12-27T16%3A15%3A31Z (Non-esc: ows_DateField =2012-12-27T16:15:31Z)</td>
</tr>
<tr>
<td>Yes/No</td>
<td>TRUE or FALSE</td>
<td>ows_YesNoField=TRUE</td>
</tr>
</tbody>
</table>

Samples

- Adding a new SharePoint Item or Document.

splus://<portal>/site/multimedia/Away.mp3?action=additem
&contenttype=audio

splus://<portal>/site/multimedia/Home.mp3?action=additem
&contenttype=audio&ows_comments=Added%20with%20URL%20schemes

- Editing a new SharePoint Document.

splus://<portal>/site/multimedia/Away.mp3?action=edititem

splus://<portal>/site/multimedia/Away.mp3?action=edititem
&ows_title=New%20title
SharePlus Application Actions

- Invoking a *SharePlus component*.

```plaintext
splus://?action=localfiles&folder=Logs
splus://?action=favorites
splus://?action=settings
splus://?action=help
splus://?action=feedback
```

Configuration Change Actions

These actions let you modify the application’s configuration and the Application Launchpad’s source when included in the Application Home. The *URLSchemes* feature and its settings *AllowConfigurationUpdate* and *AllowWebDashboardUpdate* respectively must be enabled in the application’s configuration (Configuration File) for these actions to work.

**Samples**

- Modifying the Remote Configuration section

  ```plaintext
  splus://?action=configurationURL&url=https%3A%2F%2Fportal%2FConfigurationFiles%2FCustomConfiguration.plist
  ```

- Modifying the source of an Application Launchpad in the Application Home

  ```plaintext
  splus://?action=webdashboard&source=http%3A%2F%2Fportal%2Fsites%2FSiteAssets%2FCustomLaunchpad.webarchive&title=MyLaunchpad
  ```

<table>
<thead>
<tr>
<th>Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>source</td>
<td>The URL to the dashboard resource or the local path to an existing web resource. An empty value will disable the Application Launchpad feature.</td>
</tr>
<tr>
<td>title</td>
<td>This is an optional value and sets the new title for the dashboard.</td>
</tr>
</tbody>
</table>
Search Actions

Using these actions you can open a query to a list in the Advanced List Search, as you would from the UI. The Advanced Search feature filters items by the specified fields’ value, using a specific operator. Only one field can be filtered at a time. E.g., ows_Title:contains(SharePlus)

Format to be used:

```
fieldname:operator(value)
```

<table>
<thead>
<tr>
<th>Operators</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>equals</td>
<td>The field value must be exactly the same</td>
</tr>
<tr>
<td>notequals</td>
<td>The field value must be different</td>
</tr>
<tr>
<td>greater</td>
<td>The field value must be greater</td>
</tr>
<tr>
<td>greaterorequal</td>
<td>The field value must be greater or equal</td>
</tr>
<tr>
<td>lower</td>
<td>The field value must be lower</td>
</tr>
<tr>
<td>lowerorequal</td>
<td>The field value must be lower or equal</td>
</tr>
<tr>
<td>isnull</td>
<td>The field value must not be specified (isnull() )</td>
</tr>
<tr>
<td>isnotnull</td>
<td>The field value must not be specified (isnotnull() )</td>
</tr>
<tr>
<td>beginswith</td>
<td>The field value must start with the value specified</td>
</tr>
<tr>
<td>contains</td>
<td>The field value must contain the value specified</td>
</tr>
</tbody>
</table>

The Search Module (Enterprise) can also be opened from a search action, and search queries can be constructed using the keyword query syntax. Advanced filtering can be achieved through property restrictions, for further details refer to the [Property Restriction Keyword Queries] MDSDN article.

Samples

- Performing a search in the Search Module on a specific site.

  splus://<portal>/site?action=search

  splus://<portal>/site?action=search&query=pdf

- Performing a custom search on a SharePoint List.

  splus://<portal>/site/multimedia?action=query&filter=ows_Title:contains(Text)&filtertitle=Text%20Filter&includesubfolders=true
Configuring Web-to-App Integration

You can configure SharePlus custom URL Schemes for your application through the URLSchemes feature in the Configuration File. The feature can be completely disabled through configuration.

```xml
<key>URLSchemes</key>
<dict>
    <key>Enabled</key>
    <true/>
    <key>Settings</key>
    <dict>
        <key>AllowConfigurationUpdate</key>
        <false/>
        <key>AllowWebDashboardUpdate</key>
        <true/>
    </dict>
</dict>
```

- **AllowWebDashboardUpdate** – Enables/Disables the ability to update the Application Launchpad’s source using URL Schemes.
- **AllowConfigurationUpdate** – Enables/Disables the ability to update the Configuration URL by using URL Schemes.

**Note:** This setting disables the `Configuration.setRemoteFileSource` JavaScript API method which is also used to update the Remote Configuration URL.

Using SharePlus Links

Using this versatile feature, you can invoke SharePlus from iOS Safari, from the email client, from QR code readers, and from any other application that handles HTML content.

Invoke SharePlus from Safari

You can recommend using SharePlus to the users that access the SharePoint site in Safari, from a mobile device. There are a number of possibilities or paths that you can follow to achieve this which require a custom implementation. Below you can find two commonly used methods.

Implementing a Mobile landing page

When the user navigates to the SharePoint site, the server can identify the client technology through the HTTP user agent. For further details see [http://detectmobilebrowsers.com/](http://detectmobilebrowsers.com/).

In the case of mobile devices, the server detects iOS and the client browser and then redirects the user to the custom mobile landing page. The standard SharePoint site is displayed for PCs.

Using a mobile landing page allows you to display an easy-to-navigate page with eye-catching design and buttons leading to actionable content, thus inviting the user to install SharePlus.

The custom page could include two buttons:

- An “Install SharePlus” button, which will trigger the installation of SharePlus using Over-The-Air deployment.
• An “Open in SharePlus” button, which will launch SharePlus passing the URL of the current site.

**Note:** The HTTP User Agent can be easily tricked. Because of that, the mobile landing page solution can be a weak one when you take security into account.

---

**Customizing the SharePoint site**

The SharePoint site could include a custom button on the SharePoint Ribbon, which will include a SharePlus link to invoke SharePlus. This solution can be achieved through a small code snippet located either on the SharePoint master page (server) or on the page’s JavaScript (client). The link included on the HTML page can be static or dynamically rendered using JavaScript either by the server or the client.

**Note:** When including S+ links as hyperlinks within SharePoint, you have to use an HTML Forms Web Part. This is the only web part that doesn't check or rewrite the included HTML.
Send SharePlus Links by E-mail
Once you have the SharePlus Link ready, you can share the link by sending it through e-mail. When the user follows the link within the device, SharePlus is invoked and the specified action is called.

Invoke SharePlus from a QR Code
SharePlus links can be set as QR Code Data. After creating the QR Code, users are allowed to scan the code to invoke any SharePlus behavior specified in the SharePlus link. For example, a company that stores technical specification documents in SharePoint can enable employees to open those documents directly from SharePoint using QR Codes. The codes will be created using SharePlus Links, associating a SharePoint document to a QR Code. Employees later scan a specific QR Code in their devices, opening the associated SharePoint document.

App-to-App Integration
SharePlus can be invoked from a native iOS application, from Objective-C code, using custom URL schemes (SharePlus links). This communication between other applications and SharePlus can initiate specific requests, switching the control to SharePlus.

Communication process between other applications and SharePlus (quick description):
1. An NSURL object with SharePlus custom URL Scheme is created.
2. The object is passed to the the openURL: method of the shared UIApplication object.
3. The openURL: method launches SharePlus, which has been registered to receive URLs of that type (SharePlus custom URL Schemes).
4. SharePlus takes control.

For further details refer to Communicating with Other Apps in the iOS App Programming Guide.
Setting a Custom Email Client
SharePlus can configure the settings for 3rd party email client that overrides the iOS email client. The EmailClient feature in the Configuration File is used to configure these settings. HTTP or HTTPS based URLs are accepted in all cases.

```
<key>EmailClient</key>
<dict>
    <key>Enabled</key>
    <true/>
    <key>Settings</key>
    <dict>
        <key>AllowDefault</key>
        <true/>
        <key>ClientName</key>
        <string></string>
        <key>Schema</key>
        <string>mailto://[To]?subject=[Subject]?body=[Body]</string>
    </dict>
</dict>
```

- **AllowDefault** – Allows SharePlus to work with iOS default email client when the 3rd party email client is not available on the device.
- **ClientName** – Custom client name. This is the 3rd party email client application name.
- **Schema** – Custom URL schema with mailto protocol. The available tokens are:
  - [To]
  - [Subject]
  - [Body]
Appendices

Appendix 1: Configuration File Reference
Appendix 2: Document Changelog
Appendix 1: Configuration File Reference

SharePlus is able to centrally configure default behaviors and feature trimming using a configuration file with a specific structure. This is an XML file that can be previously established by the user or even updated remotely from a repository.

Structure and XML Details

The configuration file has a **plist** extension, which internally is just an XML with a pre-define structure that it’s needed for the application in order to correctly recognize and process the information.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
  <dict>
    <key>ConfigManager</key>
    <dict>
    <key>Features</key>
    <dict>
    <key>Pages</key>
    <dict>
    <key>Accounts</key>
    <dict>
    <key>Sites</key>
    <dict>
    <key>Variables</key>
    <dict>
  </dict>
</plist>
```

This file has five root items

- **ConfigManager**: Attributes related to the remote configuration setup.
- **Features**: Attributes related to the application’s features.
- **Pages**: Attributes that define pages (or view containers) that can be used when first accessing the application, sites, and/or sub-sites.
- **Accounts**: Attributes that define site’s accounts.
- **Sites**: Attributes that define possible connection configurations.
- **Variables**: Attributes for custom behaviors.
In this section you will find the following keys:

```
<key>ConfigManager</key>
<dict>
    <key>ConfigurationURL</key>
    <string>http://pathToRepository/SharePlus_Config.plist</string>
    <key>ConfigurationVersion</key>
    <string>3</string>
    <key>MustLoadRemoteConfig</key>
    <false/>
    <key>MustUseLastAppVersion</key>
    <false/>
    <key>ConfigurationProviderID</key>
    <integer>0</integer>
    <key>ConfigurationAccountName</key>
    <string></string>
    <key>ForceResetSettings</key>
    <false/>
</dict>
```

- **ConfigurationURL**: Represents the URL of the repository where the configuration file will be stored for future updates.
  - Possible values: Http or https based URLs are accepted (Strings).
  - Can be empty: Yes

- **ConfigurationVersion**: Indicates the current version of the configuration file. This number lets the administrators keep track of the different configuration files.
  - Possible values: any string
  - Can be empty: Yes

- **MustLoadRemoteConfig**: Indicates if loading remote configuration is mandatory. This means that if the application cannot access the given URL, the application won’t let the user use the application.
  - Possible values: <true/> or <false/>
  - Can be empty: No

- **MustUseLastAppVersion**: Indicates if the last version of SharePlus must be installed in order to use it. To check whether the current version is the last one, it’ll use the value from the Last Version field of the Version Handle feature.
  - Possible values: <true/> or <false/>
  - Can be empty: No

- **ForceResetSettings**: Forces all application settings to default values.
  - Possible values: <true/> or <false/>
  - Can be empty: No
- **ConfigurationProviderID**: Indicates the ID of the configuration provider to be used. This value references the app.plist configuration providers section.
  - Possible values:
    - 0 – Endpoint Provider (default)
    - 1 – Library-based Provider
  - Can be empty: Yes

- **ConfigurationAccountName**: Indicates the account to be used when retrieving the remote Configuration File. Should reference by name an account defined in the accounts section.
  - Possible values: any string
  - Can be empty: Yes

The Accounts section represents a collection of accounts that will be pre-loaded in the application, following the configuration given in each item.
Features

The feature section is directly related to the application's behavior and feature trimming. Every feature is represented by a dictionary structure and most of these entries follow a pattern:

```xml
<key>FeatureName</key>
<dict>
  <key>Enabled</key>
  <true/>
  <key>Settings</key>
  <dict>...
</dict>
</dict>
```

This means that for each feature we will find the following common children:

- **Enabled**: Tells the app if the feature is enabled. It is a **Boolean** field, so the possible values for this entry are `<true/>` or `<false/>`
- **Settings**: Which represents a dictionary (key-value pair) of the needed parameter to configure the specified feature. The keys in a dictionary are always strings, but the value for a specific key could be of one of the following types:
  - **String**: any string is accepted
  - **Integer**: any number value is accepted
  - **Boolean**: only `<true/>` or `<false/>` values are accepted.

As an example, the following is the **EmailDocuments** feature structure:

```xml
<key>EmailDocument</key>
<dict>
  <key>Enabled</key>
  <true/>
  <key>Settings</key>
  <dict>
    <key>BCC</key>
    <string/>
    <key>Body</key>
    <string>Sent via [ProductSignature]</string>
    <key>CC</key>
    <string/>
    <key>Subject</key>
    <string>[DocumentName]</string>
    <key>To</key>
    <string/>
  </dict>
</dict>
```

As you can see, you will find the **Enabled** key and the **Settings** key. In this case, the settings key will represent relevant data for the email action, such as the subject or recipients of the email.
Features Specifications

The following are the possible entries in the Feature section:

- **Alerts**: Allow the usage of alerts/notifications in SharePlus.
  - **CheckTime**: The interval of time (in seconds) that the application waits to check for modifications over items marked with alerts.
    - **Possible values**: Integer
    - **Can be empty**: No
  - **Providers**: Allows specifying from which sources the application will be getting the alerts.
    - **Manual**: Allows the creation of alerts when browsing lists or items.
      - **AllowFromLists**: Enable/Disable alerts for lists.
        - **Possible values**: Boolean
        - **Can be empty**: No
      - **AllowFromItems**: Enable/Disable alerts for items.
        - **Possible values**: Boolean
        - **Can be empty**: No
    - **Favorites**: Allows the creation of alerts from Favorites.
      - **AllowFromLists**: Enable/Disable alerts for lists.
        - **Possible values**: Boolean
        - **Can be empty**: No
      - **AllowFromItems**: Enable/Disable alerts for items.
        - **Possible values**: Boolean
        - **Can be empty**: No

- **AnnotationEditor**: Allows configuring multiple settings for the Annotation Editor such as Shapes, Freehand tool, Text and Markups features.

- **AuthenticationTimeBomb**: When the Authentication Time Bomb feature is enabled, it requires user authentication against the server within a specific period of time.
  - **AllowedTime**: Maximum period of time (hours) that the user can work with the app without login in again.
    - **Possible values**: Integer
    - **Can be empty**: Yes
  - **MaxAttempts**: Maximum amount of failed authentication attempts before all the site’s data is wiped.
    - **Possible values**: Integer
    - **Can be empty**: Yes
  - **AttemptFailedMessage**: Message shown every time the user fails to enter valid credentials and still has attempts left. If empty, no message will be shown.
    - **Available Tokens**: [AttemptsLeft] - number of remaining attempts. This token can be inserted into the message. (e.g. "You have [ AttemptsLeft ] attempt(s) left").
      - **Possible values**: String
      - **Can be empty**: No
LimitReachedMessage: Message shown when the user fails to input valid credentials and has no more attempts left. If empty, the app will show the default message (i.e. "You have reached the maximum attempts allowed. Existing data was deleted.")
  ▪ Possible values: String
  ▪ Can be empty: No

Contacts: Allows the user to add contacts from a List to the Device contacts.

DocumentPrint: Allows the user to print documents using iOS native AirPrint service.

  o ForceWebViewForOffice: Enables/Disables the use of Quick look, a native view that can display any type of document but does not support swipe between documents. The alternative, Web view, can’t display Office Macros.
    ▪ Possible values: Boolean
    ▪ Can be empty: Yes
  o SwipeDocumentsEnable: Enables/Disables Swipe Documents feature – Swipe through list documents when opening a single document from a library or list.
    ▪ Possible values: Boolean
    ▪ Can be empty: Yes
  o SwipeListTemplates: Enables/Disables the swipe document feature only for the specified list templates. If SwipeDocumentsEnable is enabled the specified list templates will have Swipe Documents enabled. Otherwise, the list templates will have swipe documents disabled.
    Default values: 109, 851. These numbers are linked to a list template in SharePoint. 109 is a Picture Library and 851 is an Assets Library.
    ▪ Possible values: String
    ▪ Can be empty: Yes
  o TabsEnable: Enables/Disables tabbed previewing for documents. All documents from a list are opened in different tabs when opening a single document.
    ▪ Possible values: Boolean
    ▪ Can be empty: Yes
  o TabsListTemplates: Enables/Disables tabbed previewing feature only for the specified list templates. If TabsEnable is enabled the specified list templates will have tabbed previewing enabled. If TabsEnable is disabled, the list templates will have tabbed previewing disabled. Default value is BLANK. These numbers are linked to a list template in SharePoint.
    ▪ Possible values: String
    ▪ Can be empty: Yes

EmailClient: Settings for third-party email client. Overrides iOS email client.
  o AllowDefault: Allows the app to work with iOS’ default email client if the third-party email client it is not available on the device.
    ▪ Possible values: Boolean
    ▪ Can be empty: Yes
  o ClientName: Custom client name. This is the third-party email client application name.
    ▪ Possible values: String
    ▪ Can be empty: Yes
- **Schema**: Client’s URL schema. Example: `mailto://[To]?subject=[Subject]?body=[Body]`. Available tokens: [To], [Subject], [Body].
  - Possible values: String
  - Can be empty: Yes

- **EmailDocument**: Allows the user to send documents via e-mail.
  - **To**: Default email addresses to be included in the TO field when composing a new message. Defined as comma-separated email addresses (e.g.: example1@domain.com, example2@domain.com)
    - Possible values: String
    - Can be empty: Yes
  - **CC**: Default email addresses to be included in the CC field when composing a new message. Defined as comma-separated email addresses (e.g.: example1@domain.com, example2@domain.com)
    - Possible values: String
    - Can be empty: Yes
  - **BCC**: Default email addresses to be included in the BCC field when composing a new message. Defined as comma-separated email addresses (e.g.: example1@domain.com, example2@domain.com)
    - Possible values: String
    - Can be empty: Yes
  - **Subject**: Default SUBJECT when composing a new message. Available Tokens: [DocumentName]. Adding [DocumentName] to the subject line will add the document’s title.
    - Possible values: String
    - Can be empty: Yes
  - **Body**: Text to be used as the default message body. Available Tokens: [DocumentName], [DocumentURL], [ProductSignature], [NewLine].
    - Possible values: String
    - Can be empty: Yes

- **EmailURL**: Allows the user to send the URL of a document via e-mail.
  - **To**: Default email addresses to be included in the TO field when composing a new message. Defined as comma-separated email addresses (e.g.: example1@domain.com, example2@domain.com)
    - Possible values: String
    - Can be empty: Yes
  - **CC**: Default email addresses to be included in the CC field when composing a new message. Defined as comma-separated email addresses (e.g.: example1@domain.com, example2@domain.com)
    - Possible values: String
    - Can be empty: Yes
  - **BCC**: Default email addresses to be included in the BCC field when composing a new message. Defined as comma-separated email addresses (e.g.: example1@domain.com, example2@domain.com)
    - Possible values: String
    - Can be empty: Yes
  - **Subject**: Default SUBJECT when composing a new message. Available Tokens: [DocumentName]. Adding [DocumentName] to the subject line will add the document’s title.
    - Possible values: String
    - Can be empty: Yes
- **AllowHTML**: Allows the user to send the message body in HTML format.
  - Possible values: Boolean
  - Can be empty: No

- **Body**: Text to be used as the default body in the message. Available Tokens: [DocumentName], [DocumentURL], [ProductSignature], [NewLine].
  - Possible values: String
  - Can be empty: Yes

- **Favorites**: Enables marking items, folders, lists, libraries and sub sites as favorites and accessing them from the "Favorites" folder in SharePlus home screen.
  - **AllowInItems**: Allows to favorite items and folders.
    - Possible values: Boolean
    - Can be empty: No
  - **AllowInLists**: Allows to favorite lists and libraries.
    - Possible values: Boolean
    - Can be empty: No
  - **AllowInWebs**: Allows to favorite sub-sites.
    - Possible values: Boolean
    - Can be empty: No

- **Feedback**: Allows the application to send user feedback about the application’s usage.
  - **To**: Default email addresses to be included in the TO field when composing a new message. Defined as comma-separated email addresses (e.g.: example1@domain.com, example2@domain.com). This setting cannot be blank since the address is also used for error reporting.
    - Possible values: String
    - Can be empty: Yes (No, if Feedback feature is enabled)
    - Should contain at least one valid email address to mail feedback.
  - **CC**: Default email addresses to be included in the CC field when composing a new message Defined as comma-separated email addresses (e.g.: example1@domain.com, example2@domain.com)
    - Possible values: String
    - Can be empty: Yes
  - **BCC**: Default email addresses to be included in the BCC field when composing a new message. Defined as comma-separated email addresses (e.g.: example1@domain.com, example2@domain.com)
    - Possible values: String
    - Can be empty: Yes
  - **Subject**: Default SUBJECT when composing a new message. Available Tokens: [DocumentName], [FeedbackType]
    - Possible values: String
    - Can be empty: Yes
  - **Body**: Text used as message default body. Available Tokens: [DocumentName], [DocumentURL], [ProductSignature], [NewLine], [FeedbackType]
    - Possible values: String
    - Can be empty: Yes

- **GlobalSearch**: Enterprise Search settings.
  - **IncludeScopes**: Search scopes to be displayed. All search scopes must be based on the server based. If no match is found the setting is ignored.
    - Possible values: String
    - Can be empty: Yes
- **ExcludeScopes**: Filtered search scopes. SharePlus OOTB has internal scopes that may not be relevant for the user. Excluded scopes by default: Global Query Exclusion, Rank Demoted Sites.
  - Possible values: String
  - Can be empty: Yes

- **ProfileDocumentsScopes**: Allows setting the scopes in which documents associated to the user’s profile are searched. If not specified, assumes all scopes.
  - Possible values: String
  - Can be empty: Yes

- **GlobalSettings**: Predefined configuration for the application's global settings. These are the app's factory settings for all users, which can't be changed via remote configuration.
  - **ConnectionTimeout**: Amount of seconds the app waits for a response before assuming the connection has timed out.
    - Possible values: Integer
    - Can be empty: No
  - **DisableAutoLockOnPreview**: Prevents the device from locking when previewing documents.
    - Possible values: Boolean
    - Can be empty: No
  - **DisableAutoLockOnSync**: Prevents the device from locking during the synchronization process.
    - Possible values: Boolean
    - Can be empty: No
  - **DisableAutoLockMustBePluggedIn**: Enables the DisableAutoLockOnSync and DisableAutoLockOnPreview options only if the device is plugged in.
    - Possible values: Boolean
    - Can be empty: No
  - **PreviewDocumentsOnTap**: Show document preview as the default action when tapping on it instead of displaying the item details. Items details will still be available on a different view.
    - Possible values: Boolean
    - Can be empty: No
  - **RemoveLocalFilesAfterUpload**: Remove the local copy of a file after it has been uploaded.
    - Possible values: Boolean
    - Can be empty: No
  - **UserAgent**: SharePlus' user agent. If empty the app will use the standard Mozilla user agent.
    - Possible values: String
      - Tokens: [AppName], [AppVersion], [DeviceInfo]
    - Can be empty: Yes
  - **SyncIdleTime**: Specifies the wait time value in seconds, after being idle, before triggering the synchronization process.
    - Possible values: Integer
    - Can be empty: No
- **EnableLog**: Log all the app’s requests and server’s responses. Logs will be generated in the Local Files under the “Logs” folder.
  - **Possible values**: Integer
    - 0. False (default)
    - 1. True
  - **Can be empty**: No
- **NetworkStack**: Specifies the method uses in the networking layer. Legacy is recommended for NTLM scenarios.
  - **Possible values**: Integer
    - 0. Legacy (ASIHTTPRequest). Default for Free and Subscription.
    - 2. Modern (NSURLSession)
  - **Can be empty**: Yes
- **UploadMethod**: Specifies the method used to upload files, PUT or POST.
  - **Possible values**: Integer
    - 0. WebDav (PUT) (default)
    - 1. WebService (POST)
  - **Can be empty**: Yes
- **StartupModule**: Specifies the application module to be loaded by SharePlus when opening the application from scratch. Possible values include all Row Identifiers specified in the NavigatorTableDefinition.plist file.
  - **Possible values**: String
    - DocumentsHub – to reference the Documents Module.
    - Favorites – to reference the Favorites Module.
    - Home – to reference the default Application Launchpad Module.
    - Search – to reference the Search Module
    - SitesHub – to reference the Sites Module
    - Social – to reference the Social Module.
  - **Can be empty**: Yes
- **ExternalBrowser**: Specifies the external browser that will be used. Possible values are *chrome* or *safari*, and Safari is used by default.
  - **Possible values**: String
  - **Can be empty**: Yes
- **Help**: Replace or hide SharePlus’ Help.
  - **HelpURL**: URL to an online help page to replace the app's if no per-device URLs are defined.
    - **Possible values**: Http or https based URLs are accepted (String).
    - **Can be empty**: Yes
  - **HelpURL-iPad**: URL to an online help page to replace the app's when the app is running in an iPad.
    - **Possible values**: Http or https based URLs are accepted (String).
    - **Can be empty**: Yes
  - **HelpURL-iPhone**: URL to an online help page to replace the app's when the app is running in an iPhone.
    - **Possible values**: Http or https based URLs are accepted (String).
    - **Can be empty**: Yes
- **ShowHelp**: Shows/Hides the app's Help on the app's global settings.
  - Possible values: Boolean
  - Can be empty: No
- **ShowFAQ**: Shows/Hides the app's FAQ on the app's global settings.
  - Possible values: Boolean
  - Can be empty: No
- **ShowAboutUs**: Shows/Hides the app's About Us on the app's global settings.
  - Possible values: Boolean
  - Can be empty: No

- **HomeManagement**: Allows the configuration of the Application Home page and the Default Site Home page.
  - **AllowSetAsHome**: Enables/Disables the inclusion of the “Set as Home” action among the possible actions displayed for an item.
    - Possible values: Boolean
    - Can be empty: No
  - **DefaultSitePageName**: References the page to be used as Site Home by default. The page name must match the name of an existing page in the Pages section.
    - Possible values: String
    - Can be empty: No
  - **InheritParentPage**: When true, sub-sites inherit pre-configured pages from their parent site.
    - Possible values: Boolean
    - Can be empty: No
  - **AppPageName**: Defines the page name to be used as Application Launchpad. The page name must match the name of an existing page in the Pages section.
    - Possible values: String
    - Can be empty: No
  - **ShowNavigatorButton**: If false, hides the Application Launchpad button (SideBar).
    - Possible values: Boolean
    - Can be empty: No
  - **ShowMainAreaButton**: If false, hides the Application Launchpad button (Main Area).
    - Possible values: Boolean
    - Can be empty: No

- **ImageManagement**: Enables the adjustment of the size and quality of captured images before being uploaded.
  - **DefaultNamePrefix**: Default filename prefix for pictures to be uploaded.
    - Possible values: String
    - Can be empty: YES
- **Qualities**: Available quality options (options names may vary, but values must be between 5 and 100).
  - **Low**
    - **Possible values**: Integer (from 5 – 100)
    - **Can be empty**: No
  - **Normal**
    - **Possible values**: Integer (from 5 – 100)
    - **Can be empty**: No
  - **High**
    - **Possible values**: Integer (from 5 – 100)
    - **Can be empty**: No
- **Ratios**: Available size ratios (the name of each ratio may vary, but values must be between 5 and 100)
  - **Small**
    - **Possible values**: Integer (from 5 – 100)
    - **Can be empty**: No
  - **Medium**
    - **Possible values**: Integer (from 5 – 100)
    - **Can be empty**: No
  - **Large**
    - **Possible values**: Integer (from 5 – 100)
    - **Can be empty**: No
  - **Actual Size**
    - **Possible values**: Integer (from 5 – 100)
    - **Can be empty**: No
- **ItemsManagement**: Enable/Disable list content management via User Interface.
  - **AllowAdd**: Allows users to add items.
    - **Possible values**: Boolean
    - **Can be empty**: No
  - **AllowEdit**: Allows users to edit items.
    - **Possible values**: Boolean
    - **Can be empty**: No
  - **AllowDelete**: Allows users to delete items.
    - **Possible values**: Boolean
    - **Can be empty**: No
  - **AllowCheckOut**: Allows users to check out items.
    - **Possible values**: Boolean
    - **Can be empty**: No
  - **AllowApproveReject**: Allows users to Approve/Reject items.
    - **Possible values**: Boolean
    - **Can be empty**: No
  - **AllowCopyURL**: Allows users to Copy item’s URL.
    - **Possible values**: Boolean
    - **Can be empty**: No
  - **AllowEmailURL**: Allows users to Email item’s URL.
    - **Possible values**: Boolean
    - **Can be empty**: No
  - **AllowSaveAsDraft**: 
    - **Possible values**: Boolean
    - **Can be empty**: No
- **AllowOpenInSafari**: Tells the application if the option "Open In Safari" must be shown in the WebBrowser when pre-viewing items i.e: Links
  - **Possible values**: Boolean
  - **Can be empty**: No

- **ListManagement**: Enable/Disable list settings management via User Interface.
  - **AllowAdvanceSearch**: Enables the list's Advanced Search.
    - **Possible values**: Boolean
    - **Can be empty**: No
  - **AllowOffline**: Allows setting offline support for lists.
    - **Possible values**: Boolean
    - **Can be empty**: No
  - **ShowItemsCount**: Shows/Hides items count in the navigator’s lists.
    - **Possible values**: Boolean
    - **Can be empty**: No
  - **AllowManualSorting**: Allows the user to set sorting criteria.
    - **Possible values**: Boolean
    - **Can be empty**: No

- **LocalFiles**: Local Files section settings.
  - **AllowCopyToLocalFiles**: Allows copying documents from libraries to Local Files.
    - **Possible values**: Boolean
    - **Can be empty**: No
  - **AllowEmailDocument**: Allows sending documents via email from Local Files.
    - **Possible values**: Boolean
    - **Can be empty**: No
  - **AllowSubfolderInSelector**: Displays a file browser when copying a file from a list/library to Local Files.
    - **Possible values**: Boolean
    - **Can be empty**: No
  - **SwipeDocumentsEnable**: Enable/Disable Swipe Documents feature - Swipe through Local Files documents when opening a single document.
    - **Possible values**: Boolean
    - **Can be empty**: No
  - **TabsEnable**: Enable/Disable tabbed previewing for documents - Have all documents from Local Files opened in different tabs when opening a document.
    - **Possible values**: Boolean
    - **Can be empty**: No

- **LocationServices**: Allows the user to get geo-location data.
  - **AutoStartLocation**: If enabled, tapping the Location Services button will automatically retrieve your location's coordinates. If disabled, the user will have to manually trigger the coordinates' retrieval.
    - **Possible values**: Boolean
    - **Can be empty**: No

- **LogHandler**: Saves internal logging into Local Files under the Logs folder.
  - **Services**: Defines event logging type.. Accepted values: errors (error logging only), all (all events are logged).
    - **Possible values**: String (errors / all)
    - **Can be empty**: No

- **OfficeWeb**: These entries allow you to customize the URL used to view/edit office documents in Office Web Apps.
- **Edit_Excel_Url**: Edit Excel files app URL.
  - Possible values: String
  - Can be empty: No
- **View_Excel_Url**: View Excel files app URL.
  - Possible values: String
  - Can be empty: No
- **Edit_Word_Url**: Edit Word files app URL.
  - Possible values: String
  - Can be empty: No
- **View_Word_Url**: View Word files app URL.
  - Possible values: String
  - Can be empty: No
- **Edit_PowerPoint_Url**: Edit PowerPoint files app URL.
  - Possible values: String
  - Can be empty: No
- **View_PowerPoint_Url**: View PowerPoint files app URL.
  - Possible values: String
  - Can be empty: No
- **Edit_OneNote_Url**: Edit One Note files app URL.
  - Possible values: String
  - Can be empty: No
- **View_OneNote_Url**: View One Note files app URL.
  - Possible values: String
  - Can be empty: No
- **OpenInExternalBrowser**: When true, the *Edit in Web* action is performed using the defined ExternalBrowser (Global Settings). When false, the embedded browser is used.
  - Possible values: Boolean
  - Can be empty: No
- **OpenIn**: Allows the user to use the Apple’s Open In protocol in order to share documents with other applications.
  - **AllowConfigurationUpdate**: Enable/Disable the possibility to load a Configuration File using the Open In protocol.
    - Possible values: Boolean
    - Can be empty: Yes
  - **Annotation**: Allows you to specify which annotations will be sent to a specific appId (applies to OpenIn functionality).
    - Possible values: Dictionary
    - Can be empty: No
  - **Filters**: Allows you to filter OpenIn apps based on file extension (only one by extension).
    - Possible values: Dictionary
    - Can be empty: Yes
  - **AllowOtherExtensions**: If false, only allows opening documents on predefined extensions’ apps.
    - Possible values: Boolean
    - Can be empty: No
  - **ShowButtonInViewer**: Shows/Hides the "Edit" button in document preview top bar.
    - Possible values: Boolean
    - Can be empty: No
**ShowButtonInViewer-iPad**: Show/Hide “Edit” button in document preview top bar on iPad devices.
- **Possible values**: Boolean
- **Can be empty**: No

**ShowButtonInViewer-iPhone**: Show/Hide “Edit” button in document preview top bar on iPhone devices.
- **Possible values**: Boolean
- **Can be empty**: No

- **Passcode**: Passcode Lock settings.
  - **EnablePasscode**: Enable/Disable passcode lock feature by default.
    - **Possible values**: Boolean
    - **Can be empty**: No
  - **DefaultPasscode**: Default Passcode Lock code.
    - **Possible values**: 4 digits Integer.
    - **Can be empty**: Yes (if **MustUsePasscode** is set to false).
  - **MustUsePasscode**: Forces the user to work with the Passcode Lock and requires the user to set it up on the first run.
    - **Possible values**: Boolean
    - **Can be empty**: No
  - **PasscodeLockInactivityTime**: Idle time in minutes before the app locks. Accepted values: 0 (disabled), 5, 10, 15, 30, 45, 60.
    - **Possible values**: Integer
    - **Can be empty**: No
  - **EnablePasscodeLockInactivityTime**: This is an internal key. Must always be set to YES.

- **PasscodeFailsHandle**: Actions performed and messages displayed when Passcode Lock authentication fails.
  - **BlockApp**: Blocks the application.
    - **Possible values**: Boolean
    - **Can be empty**: No
  - **MaxAttempts**: Maximum authentication attempts.
    - **Possible values**: Integer
    - **Can be empty**: No
  - **Message**: Message shown when the user fails to input the valid passcode and has no more attempts left.
    - **Possible values**: String
    - **Can be empty**: No
  - **WipeData**: When true, clears internal app data when the user has no attempts left (Local Files, Offline cache, internal DB and other user data).
    - **Possible values**: Boolean
    - **Can be empty**: No

- **Pasteboard**: Enable/Disable copy & cut functionality from sensible application places.
- **PDFViewer**: PDF Viewer settings.
  - **MaxZoom**: Lets the maximum zoom multiplier value for zooming in PDFs. Default value is 1000.
    - **Possible values**: Integer
    - **Can be empty**: No
  - **MinZoom**: Lets the minimum zoom multiplier value for zooming out PDFs. Default value is 90.
    - **Possible values**: Integer
    - **Can be empty**: No
o **PageFit**: Configures page fit feature.
  - **Possible values**: Integer
    - 0 – Fit page
    - 1 – Fit width
    - 2 – Fit height
  - **Can be empty**: No

o **ShowNavigationButtons**: Enables PDF Navigation Buttons in the bottom toolbar.
  - **Possible values**: Boolean
  - **Can be empty**: Yes

o **VerticalScroll**: Sets scroll type to vertical.
  - **Possible values**: Boolean
  - **Can be empty**: Yes

- **PictureBoard**: Picture Board visualization settings. This visualization is only available for Picture and Assets libraries.
  - **HideInfoButton**: Hides the button on the bottom right corner of the pictures.
    - **Possible values**: Boolean
    - **Can be empty**: No

- **ReportPlus**: Enables/disables ReportPlus dashboards. This includes the creation, opening, and editing of dashboards.

- **SessionManagement**: The session management feature lets SharePlus behave as a session based application. This means that the application will save data of the current session while being active and reset to its initial state when the session ends (i.e. the session expires or the user logs out). Different behaviors for the session timeout and actions to perform after the session expires may vary based on configuration.
  - **AllowLogOut**: Enables the log out button in the interface to manually reset the session.
    - **Possible values**: Boolean
    - **Can be empty**: No

  - **ApplyFromBackground**: Applies actions when returning from an inactive scenario. Possible actions are: Reset Password, Timeout, Wipe Data, Wipe Local Files, and Reset Sites Cache.
    - **Possible values**: Boolean
    - **Can be empty**: No

  - **ApplyToBackground**: Applies actions when the app is entering a background state.
    - **Possible values**: Boolean
    - **Can be empty**: No

  - **ApplyFromScratch**: Applies the actions when the application starts.
    - **Possible values**: Boolean
    - **Can be empty**: No

  - **ApplyFromTimer**: Applies the actions after a session timeout. The duration of the session can be specified using the Timeout parameter.
    - **Possible values**: Boolean
    - **Can be empty**: No

  - **ResetPasswords**: Action to reset the password of all the existing accounts.
    - **Possible values**: Boolean
    - **Can be empty**: Yes

  - **Timeout**: Duration of the session in minutes (i.e. Session timeout).
    - **Possible values**: Integer
    - **Can be empty**: Yes
o **AllowFromInactivity**: Specifies whether the session will be handled by the `InactivityExpirationTime` or not.
   - Possible values: Boolean
   - Can be empty: No

o **InactivityExpirationTime**: Amount of hours the application will keep a session alive without being active.
   - Possible values: Integer
   - Can be empty: No

o **WipeData**: Action to resets all of the existing content in the device. If one or more sites are defined in the configuration file, they will be reloaded automatically in the interface.
   - Possible values: Boolean
   - Can be empty: No

o **WipeLocalFiles**: This action deletes all local files available in the Local Files section of the app.
   - Possible values: Boolean
   - Can be empty: No

o **WipeSitesCache**: This action deletes all cached data such as documents, images or thumbnails for all existing sites. This action does not make any change on the database; this means all lists configurations will remain intact.
   - Possible values: Boolean
   - Can be empty: No

o **WipeAccounts**: This action deletes all existing accounts, leaving the existing portals with no account associated.
   - Possible values: Boolean
   - Can be empty: No

o **WipeAlerts**: This action deletes all created alerts and their definitions in the app.
   - Possible values: Boolean
   - Can be empty: No

o **WipeFavorites**: This action deletes all Favorites in the app.
   - Possible values: Boolean
   - Can be empty: No

- **SitesAdmin**: Site administration settings.
  o **AllowAdd**: Allow the user to add new connections to the application. If `SitesAdmin` feature is disabled this value is disregarded.
    - Possible values: Boolean
    - Can be empty: No

  o **AllowDelete**: Allow the user to delete existing connections in the application. If `SitesAdmin` feature is disabled this value is disregarded.
    - Possible values: Boolean
    - Can be empty: No

  o **AllowUpdate**: Allow the user to edit existing connections in the application. If `SitesAdmin` feature is disabled this value is disregarded.
    - Possible values: Boolean
    - Can be empty: No

  o **CredentialStorageClass**: Sets the class that implements the credentials storage. Possible values: `SPCredentialsStorageKeyChain` – to store in the keychain. Default value is `SPCredentialsStorageKeyChain`.
    - Possible values: String
    - Can be empty: Yes
RememberUserName: Always saves the username in the Keychain to suggest it on the login alert. If the CredentialStorageType field is set to 1, RememberUserName field is set to YES. If the username is different from the previous one, it will delete the site’s data.
  ▪ Possible values: Boolean
  ▪ Can be empty: No

SupportMyProfile: Enables “My Profile” feature. This feature will only work on preconfigured sites.
  ▪ Possible values: Boolean
  ▪ Can be empty: No

SupportMySite: Enables “My Site” feature. This feature will only work on preconfigured sites that have also enabled “My Site” support. The user’s “My Site” will be displayed only if created in the SharePoint server.
  ▪ Possible values: Boolean
  ▪ Can be empty: No

ThemeManagement: Loads the theme file to configure the application color settings. The theme file is a JSON object that allows the tweaking of appearance properties for iOS controls.
  o DefaultTheme: Specifies the source location of the iOS7 theme file. The source can be an URL or a local path inside the application.
    ▪ Possible values: String
    ▪ Can be empty: YES
  o DefaultTheme-ios6: Specifies the source location of the iOS6 theme file. The source can be an URL or a local path inside the application.
    ▪ Possible values: String
    ▪ Can be empty: YES

URLSchemes: Enable and configure custom URL Schemes in SharePlus.
  o AllowConfigurationUpdate: Enable/Disable the ability to update the Configuration URL using URL Schemes or the Configuration.setRemoteFileSource JavaScript API method.
    ▪ Possible values: Boolean
    ▪ Can be empty: Yes
  o AllowWebDashboardUpdate: Enable/Disable the ability to update Application Launchpad’s source using URL Schemes.
    ▪ Possible values: Boolean
    ▪ Can be empty: No

VersionHandle: Automatic updates settings.
  o DeferBlockToTimeSpan: Time limit for users to update the app. When this limit is reached, the app becomes blocked.
    ▪ Possible values: String
    ▪ Can be empty: Yes
  o ImmediateBlock: Blocks the application when a new version of the app becomes available until it is updated.
    ▪ Possible values: Boolean
    ▪ Can be empty: No
  o LastVersion: Last version of the app. If this value is different to the one on the remote configuration file, the app will alert there is an update available.
    ▪ Possible values: String
    ▪ Can be empty: No
- **URL**: URL where the update is hosted for downloading.
  - **Possible values**: String
  - **Can be empty**: No
- **ViewInWeb**: Enables View In Web feature to view a web version of the current resource or location in a Safari-like window.
  - **AllowInSite**: Enables View In Web feature for Sites.
    - **Possible values**: Boolean
    - **Can be empty**: No
  - **AllowInWeb**: Enables View In Web feature for sub sites.
    - **Possible values**: Boolean
    - **Can be empty**: No
  - **AllowInList**: Enables View In Web feature for Lists.
    - **Possible values**: Boolean
    - **Can be empty**: No
  - **AllowInProfile**: Enables View In Web feature for People Profile.
    - **Possible values**: Boolean
    - **Can be empty**: No
  - **AllowOpenInSafari**: Indicates if the "Open In Safari" option must be shown in the WebBrowser in "View in Web" actions.
    - **Possible values**: Boolean
    - **Can be empty**: No
- **WebDashboard**: Configures the SharePlusBridge options.
  - **BridgeEnabled**: Enable/Disable the SharePlusBridge functions.
    - **Possible values**: Boolean
    - **Can be empty**: No
- **WebManagement**: Determines if the user is able to reorder sub sites in the navigation pane of the application.
  - **AllowManualSorting**: Determines if the user is able to reorder website links in the navigation pane of the application.
    - **Possible values**: Boolean
    - **Can be empty**: No
- **WiFiSharing**: Allows the user to share data via WiFi from the Local Files section in the application.
  - **AllowDownload**: Allows the user to download content from the application via WiFi.
    - **Possible values**: Boolean
    - **Can be empty**: No
  - **AllowUpload**: Allows the user to upload content to the application via WiFi.
    - **Possible values**: Boolean
    - **Can be empty**: No
Pages

This section is dedicated to the Pages definition, which can be pre-loaded by the application.

The following code sample shows you how they should be defined.

```
<key>Pages</key>
<array>
  <dict>
    <key>PageName</key>
    <string>DefaultSitePage</string>
    <key>ShowSelector</key>
    <false/>
    <key>ViewParts</key>
    <array>
      <dict>
        <key>ViewControllerID</key>
        <integer>0</integer>
        <key>Title</key>
        <string>dashboard WEB</string>
        <key>Settings</key>
        <dict>
          <key>Source</key>
          <string>Documents/WebDashboardSample.html</string>
          <key>SourceType</key>
          <integer>1</integer>
        </dict>
      </dict>
    </array>
  </dict>
  <dict>
    <key>ViewControllerID</key>
    <integer>1</integer>
    <key>Title</key>
    <string>ReportPlus dashboard</string>
    <key>Settings</key>
    <dict>
      <key>Source</key>
      <string>http://serverPath/dashboard.rplus</string>
      <key>SourceType</key>
      <integer>0</integer>
    </dict>
  </dict>
</array>
```

The Pages section represents a collection of pages/view containers that are pre-loaded by the application.
Available key/values are as follows:

- **PageName**: A name for the page. This name will be used to reference the page when configuring Home content for the application, sites, and/or sub-sites.
  - **Possible values**: Any string
  - **Can be empty**: No

- **ShowSelector**: Enables/Disables the possibility to select between the different View Parts from a page/view container. Does nothing when there is only one View Part specified.
  - **Possible values**: Boolean
  - **Can be empty**: No

- **ViewParts**: Array of the View Part elements defined for the page.
  - **ViewControllerID**: Specifies the ID defined in the app.plist for the View Controller to be used.
    - **Possible values**: this is a integer field and the possible values are
      - 0 - for SPWebDashboardViewController
      - 1 - for SPReportPlusDashboardViewController
    - **Can be empty**: No
  - **Title**: Specifies the title to be displayed on the top toolbar for this View Part.
    - **Possible values**: String
    - **Can be empty**: Yes
  - **Settings**: Allows the configuration of any settings required for the View Part to display its content. Current View Parts use Source and SourceType, but different View Parts may use different settings as required.
    - **Source**: Specifies the source location of the Application Launchpad or ReportPlus Dashboard. The source can be a URL or a local path inside the application.
      - **Possible values**: String
      - **Can be empty**: No
    - **SourceType**: Sets the source type of the Application Launchpad or ReportPlus Dashboard.
      - **Possible values**: this is a integer field and the possible values are
        - 0 - to reference an URL to download the file
        - 1 - to reference a path inside the application
      - **Can be empty**: No
**Accounts**

This section is dedicated to the information of the accounts that can be pre-loaded on the application. Here’s how it look like:

```xml
<key>Accounts</key>
<array>
    <dict>
        <key>Name</key>
        <string>My Account</string>
        <key>CredentialStorageClass</key>
        <string/>
        <key>Domain</key>
        <string>yourDomain</string>
        <key>Username</key>
        <string>yourUserName</string>
        <key>Password</key>
        <string/>
        <key>UseCertificate</key>
        <false/>
        <key>CertificateName</key>
        <string/>
        <key>CertificatePassword</key>
        <string/>
        <key>isDefault</key>
        <false/>
        <key>AllowDelete</key>
        <true/>
    </dict>
    <dict>...another account...</dict>
</array>
```

The Accounts section represents a collection of accounts that will be pre-loaded in the application, following the configuration given in each item.

- **Name**: A name for the account.
  - Possible values: any string
  - Can be empty: No

- **CredentialStorageClass**: Sets the class that implements the storage for the credentials. Used only for this account. Possible values: SPCredentialsStorageKeyChain – to store in the keychain. The default value is SPCredentialsStorageKeyChain.
  - Possible values: any string
  - Can be empty: Yes

- **Domain**: Domain name.
  - Possible values: any string
  - Can be empty: Yes

- **Username**: User’s username.
  - Possible values: any string
  - Can be empty: Yes
• **Password**: User’s password.
  - Possible values: any string
  - Can be empty: Yes

• **UseCertificate**: Tells the app if the connection needs a client side certificate (.p12 file)
  - Possible values: Boolean
  - Can be empty: Yes

• **CertificateName**: If client side certificate is set to `<true/>`, provide here the certificate file name.
  - Possible values: any string
  - Can be empty: Yes

• **CertificatePassword**: If a password is needed for the client certificate, you can specify it here.
  - Possible values: any string
  - Can be empty: Yes

• **IsDefault**: Specifies if this account is the Default Account. When a site does not have an account set, the Default Account is used.
  - Possible values: Boolean
  - Can be empty: No

• **AllowDelete**: Specifies if this account can be modified or deleted.
  - Possible values: Boolean
  - Can be empty: No

Regarding security, we recommend **not** to specify the passwords in the configuration file. If the connection needs a password, which is not given, the application will prompt the user to introduce his credentials in the first online access to the site.
Sites

This section is dedicated to the information of the sites that can be pre-loaded on the application. Here’s is how it look like:

```
<key>Sites</key>
<array>
  <dict>
    <key>Name</key>
    <string>Share</string>
    <key>URL</key>
    <string>http://mydomain.com/Sites/MySite</string>
    <key>AuthMode</key>
    <integer>0</integer>
    <key>AccountName</key>
    <string>Share’s account</string>
    <key>SupportMySite</key>
    <true/>
    <key>MySiteHost</key>
    <string/>
    <key>SupportMyProfile</key>
    <true/>
    <key>ProfileHost</key>
    <string/>
  </dict>
  <dict>...another site...</dict>
</array>
```

The Sites section represents a collection of sites that will be pre-loaded in the application, following the configuration given in each item.

- **Name**: A name for the site.
  - **Possible values**: any string
  - **Can be empty**: No
- **URL**: The URL to the site.
  - **Possible values**: any string, http and https URLs are accepted.
  - **Can be empty**: No
- **AuthMode**: Refers to the authentication mode accepted by the server. There are three options for this key:
  - **Possible values**: this is a integer field and the possible values are
    - 0 - for Windows Based authentication
    - 1 - for Form Based authentication
    - 2 - for Office 365 authentication
    - 3 - for Web Login authentication
  - **Can be empty**: No
• **AccountName**: The name of an account in “Accounts” section used to connect to the site.
  - **Possible values**: any string
  - **Can be empty**: Yes

• **SupportMySite**: Enables “My Site” feature. Only one site can have this setting in YES.
  - **Possible values**: Boolean
  - **Can be empty**: No

• **MySiteHost**: The URL where “My Site” pages are hosted. If the token “[accountName]” is used it will be replaced with the user account name. Example: https://rootUrl/personal/[accountName]
  - **Possible values**: any string
  - **Can be empty**: Yes

• **SupportMyProfile**: Enables/Disables “My Profile” feature. Only one site can have this setting in YES.
  - **Possible values**: Boolean
  - **Can be empty**: No

• **ProfileHost**: The URL where the person profile related pages are hosted. If this is empty, the root URL will be used concatenated with “/my”.
  - **Possible values**: any string
  - **Can be empty**: Yes

### Variables
This section is empty by default and its items should be previously agreed depending on the user’s needs.

### Notes
In remote configuration scenarios, it is not mandatory to fill all of the items in the file. This means that the application will merge the data to the local settings automatically if a remote configuration URL is provided.
### Appendix 2: Document Changelog

<table>
<thead>
<tr>
<th>Version</th>
<th>Chapter</th>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.2.1</td>
<td>Configuration</td>
<td>Site Configuration a.k.a.</td>
<td>Added missing information about how to configure the current site itself.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>“MobileNavigation”</td>
<td></td>
</tr>
<tr>
<td>1.2</td>
<td>Introducing SharePlus</td>
<td>Welcome to SharePlus!</td>
<td>Removed MAC from SharePlus platforms.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Functional Overview</td>
<td>Table fix regarding Favorites and removal of QuickOffice in favor of Office for iPad functionality.</td>
</tr>
<tr>
<td></td>
<td>Planning SharePlus</td>
<td>Planning Configuration</td>
<td>Global Settings screenshots were updated.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Planning Connectivity</td>
<td>Accounts can now be managed from Settings.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Planning Security</td>
<td>The File Upload method and Network Layer configuration information were included.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Integration with QuickOffice was removed as Office for iPad saves documents straight to SharePoint by sharing credentials.</td>
</tr>
<tr>
<td></td>
<td>Configuration</td>
<td>Configuring Features</td>
<td>Added new settings to the Global Settings section.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Site Configuration a.k.a.</td>
<td>Several missing columns were added.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>“MobileNavigation”</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Advanced Configuration</td>
<td>User Interface Behavior</td>
<td>Added Office documents preview setting and included minor fixes on descriptions.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Advanced Features</td>
<td>Fixed SharePoint Navigation actions, “InBrowser” replaced by “InWeb”.</td>
</tr>
<tr>
<td></td>
<td>Appendices</td>
<td>Configuration File Reference</td>
<td>Updated with the latest changes on settings.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reference</td>
<td></td>
</tr>
<tr>
<td>1.1</td>
<td>Deployment</td>
<td>Application Re-signing</td>
<td>Added a new sub section about Certificates, app IDs, and other concepts: iOS App Re-signing Concepts.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-</td>
<td>The Web Developer Guide was renamed as Launchpads Developer Guide.</td>
</tr>
<tr>
<td></td>
<td>Configuration</td>
<td>Site Configuration a.k.a.</td>
<td>Adjusting Site Home was updated as multiple values are now allowed in the Home column.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>“MobileNavigation”</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Advanced Configuration</td>
<td>Home Customization</td>
<td>Site Home configuration was updated and now allows multiple values in the Home column.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Default Homes configuration has minor fixes and was renamed to Default Home – Portal and sites.</td>
</tr>
<tr>
<td>1.0.1</td>
<td>Introducing SharePlus</td>
<td>Functional Overview</td>
<td>Table fix regarding ReportPlus Integration.</td>
</tr>
<tr>
<td></td>
<td>Planning SharePlus</td>
<td>Configuration “Injection”</td>
<td>Texts slightly improved.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MDM Integration</td>
<td>Texts slightly improved.</td>
</tr>
<tr>
<td></td>
<td>Configuration</td>
<td>Remote Configuration</td>
<td>Minor fix on code snippet sample.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Configuration “Injection”</td>
<td>Section updated (JS API) and texts improved.</td>
</tr>
<tr>
<td>Section</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Deployment</strong></td>
<td>Site Configuration a.k.a. “MobileNavigation”</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fixes on descriptions, screenshots, and links.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Server Configuration</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Section updated to SharePoint 2010 server version.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Web Deployment using SharePoint</td>
<td>Screenshots update in Configuring Automatic updates.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MDM Integration</td>
<td>Texts slightly improved.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ad-Hoc Deployment</td>
<td>Minor screenshot fix.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Distribution Certificate and Provisioning Profile</td>
<td>Minor fixes on Getting the Distribution Certificate steps.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Advanced Configuration</strong></td>
<td>Application Re-Branding</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Section updated, fixes on graphic assets.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Advanced Features</strong></td>
<td>ReportPlus Integration</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Minor fixes on screenshots.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Application Integration</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>URL encoding information added.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Appendices</strong></td>
<td>Creating my IG Account</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Section removed.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>