



# **SMARTGUIDE® 6.5**

.NET INSTALLATION GUIDE

© 2016 Alphinat Inc. All rights reserved.

Alphinat SmartGuide® .NET Environment – Installation Guide  
November 2016

If this guide is distributed with software that includes an end user agreement, this guide, as well as the software described in it, is furnished under license and may be used or copied only in accordance with the terms of such license. Except as permitted by any such license, no part of this guide may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of Alphinat Incorporated. Please note that the content in this guide is protected under copyright law even if it is not distributed with software that includes an end user license agreement.

The content of this guide is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Alphinat Incorporated. Alphinat Incorporated assumes no responsibility or liability for any errors or inaccuracies that may appear in the informational content contained in this guide. Please remember that existing artwork or images that you may want to include in your project may be protected under copyright law. The unauthorized incorporation of such material into your new work could be a violation of the rights of the copyright owner. Please be sure to obtain any permission required from the copyright owner. Any references to company names in sample templates are for demonstration purposes only and are not intended to refer to any actual organization.

Alphinat, SmartGuide, Smartlets and the Alphinat logo are either, registered trademarks or trademarks of Alphinat Incorporated in Canada and/or other countries. All other trademarks are the property of their respective owners.

Alphinat Inc., 2000 rue Peel #680, Montréal (Qc), H3A 2W5, Canada.

# Contents

|  |           |
|--|-----------|
| <b>Description .....</b>                       | <b>5</b>  |
| About SmartGuide .....                         | 5         |
| Guide Content .....                            | 5         |
| Target audience.....                           | 5         |
| <b>Architecture.....</b>                       | <b>7</b>  |
| Organic view .....                             | 7         |
| <b>System requirements.....</b>                | <b>9</b>  |
| Outline .....                                  | 9         |
| Memory .....                                   | 9         |
| Disk Space.....                                | 9         |
| Application Servers .....                      | 9         |
| Database .....                                 | 10        |
| Communication Ports .....                      | 10        |
| <b>IIS application server .....</b>            | <b>11</b> |
| Configuring the environment .....              | 11        |
| Configuring SmartGuide application in IIS..... | 11        |
| Important notes .....                          | 13        |
| <b>Deployment on disk .....</b>                | <b>15</b> |
| Configuring the environment .....              | 15        |
| <b>Database .....</b>                          | <b>17</b> |
| Creating the database .....                    | 17        |
| Configuring the environment .....              | 17        |
| <b>License Installation .....</b>              | <b>19</b> |
| Installing the license file .....              | 19        |
| <b>Runtime Environment.....</b>                | <b>21</b> |
| SmartGuide Server directory structure .....    | 21        |
| Runtime environment installation.....          | 22        |
| <b>Configuration parameters.....</b>           | <b>23</b> |
| SmartGuide Designer .....                      | 23        |
| SmartGuide Server .....                        | 27        |



# Description

1

---

## About SmartGuide

SmartGuide® Version 6.5 is the latest creation, application management and deployment tool from the Alphinat software line. This software is a true SOA tool which allows quickly deploying new applications that maximise the value of knowledge and existing systems of an organization.

Subject matter experts and business architects finally have a tool which allows them to directly translate their know-how to benefit the users. In fact, SmartGuide Designer enables business people to manage the entire content, logic and rules of Web applications and self-service tools. Someone with no technical background can now automate a vast number of business processes and greatly simplify its organization's interaction with clients, partners and employees. These business procedures are translated in the form of Smartlets®, which are applications that facilitate the research, exchange and treatment of information 100% relevant to end-users. SmartGuide Designer allows creating and managing Smartlets through visual, user-friendly Web interfaces. This results in XML files which are then interpreted by Alphinat SmartGuide Server in order to present the applications to end-users.

---

## Guide Content

This guide includes instructions to set up SmartGuide Designer and Server under a .NET environment.

---

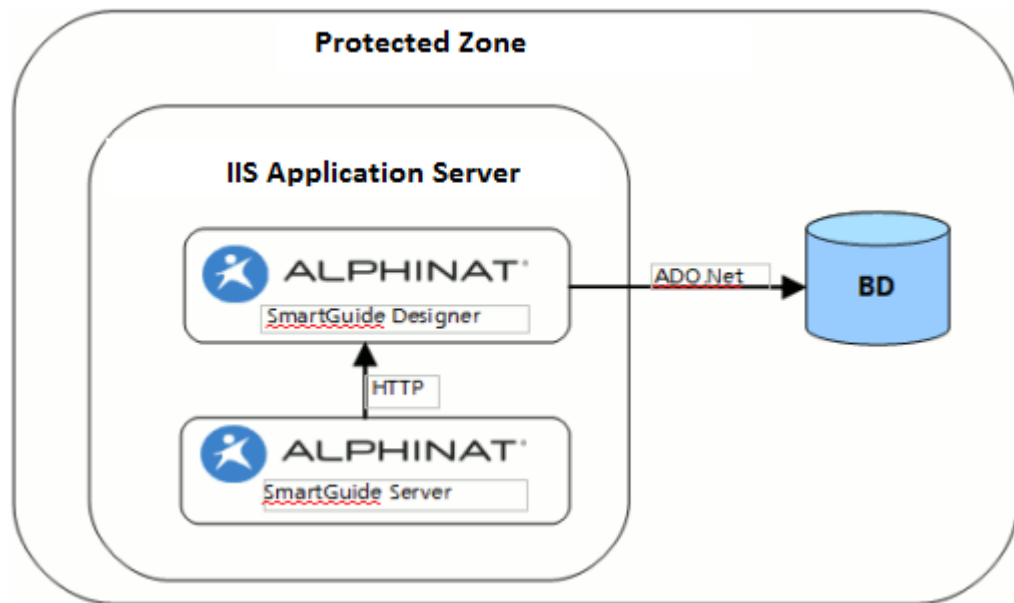
## Target audience

This guide targets the individuals who will be installing SmartGuide Designer and Server under a .NET environment.



## Organic view

The following illustration shows the organic view of the SmartGuide Designer and the runtime environment of SmartGuide server:







# System requirements

## 3

---

### Outline

The recommendations for disk and memory space vary in relation to the deployment environment, the projected usage, the number of field users, the complexity of the Smartlets and the desired response time. The following suggestions apply to an IIS development environment.

---

### Memory

A minimum of 1GB of RAM is required to power SmartGuide. Allocate more RAM for an optimized performance.

- SmartGuide Designer requires 1GB to run under Windows 2008, 2012 and 2016 and 512 MB under Windows 2003;
- SmartGuide Server requires a minimum of 1GB of RAM. Allocate more RAM for better performance.

---

### Disk Space

165 MB disk space is recommended for an initial SmartGuide installation:

- SmartGuide Designer : 115 MB
- SmartGuide Server : 50 MB

Please note that additional space will be required in order to support the storage of Smartlets and accompanying information.

---

### Application Servers

SmartGuide has successfully tested on the following application servers:

- IIS 5.0 (32-bit)

- IIS 6.0 (32-bit and 64-bit)
- IIS 7.0 (32-bit and 64-bit)
- IIS 7.5 (32-bit and 64-bit)
- IIS 8.0, 8.5 (64-bit)
- IIS 10.0 (64-bit)

---

## Database

SmartGuide can be deployed with or without a database. Smartlets and associated data can either be stored on disk or in a relational database that offers an ADO.Net connector. SmartGuide has been successfully tested on the following databases:

- HSQL
- DB2
- MSSQL
- Oracle
- MySQL
- Firebird
- PostgreSQL

---

Note: It is recommended to store the project's data on disk during development as it makes it easier to make changes to the presentation themes that are used to display Smartlets to end-users.

---

---

## Communication Ports

Access by HTTP and HTTPS is supported.

# IIS application server

## 4

This chapter will guide you, step by step, through the installation process of SmartGuide Designer on IIS.

---

### Configuring the environment

The following instructions will guide you through the configuration of your environment:

➤ **To configure your environment:**

1. Make sure **IIS** is installed.
2. Make sure that the **.NET Framework version 2.0 or higher** is installed. It is recommended to use the latest Framework available.

If IIS was installed before the Framework .NET installation, proceed with installation "restitution" in order for IIS to be configured to use .NET. Alternatively, you may execute the following command in order to register, for example, the .NET 4.0 Framework in IIS:

```
%SystemDrive%\%SystemRoot%\Microsoft.NET\Framework64\v4.0.30319\aspnet_regiis.exe -i
```

3. Download **SmartGuide** from the site provided to you by Alphinat.
4. Extract the **smartguide.zip** file in a directory that is accessible by IIS and ASP.NET.

**E.g.: %SystemDrive%\inetpub\wwwroot\smartguide\6.x.x\**

5. This file contains two folders, the "sgd" folder contains the Designer files and the "smartlets" folder contains the SmartGuide server files.

---

### Configuring SmartGuide application in IIS

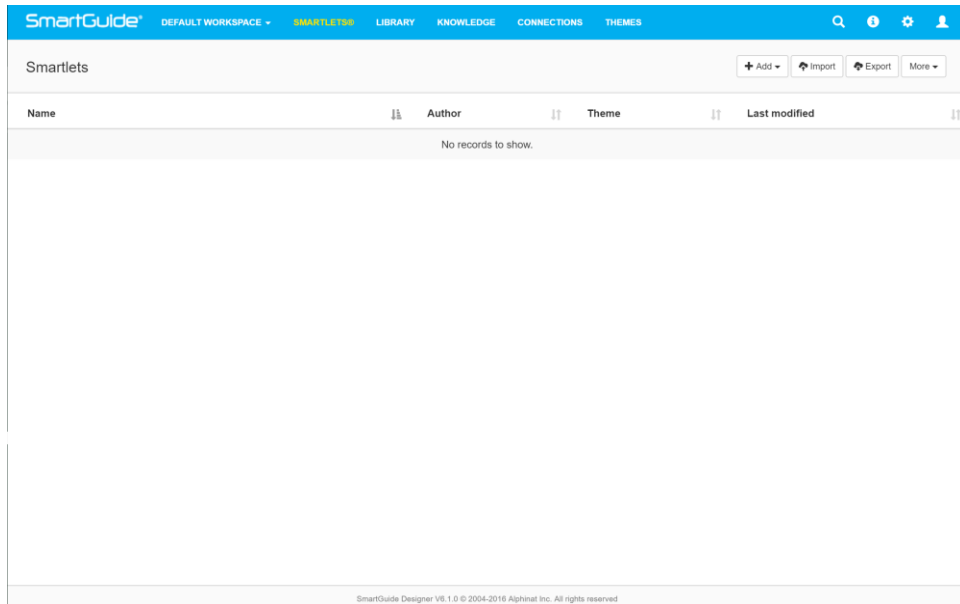
The following instructions will guide you through the SmartGuide Designer and Smartlet application installation process:

➤ **To configure SmartGuide in IIS:**

1. Add an application to the default web site (or a newly created site) from the IIS administrator:
  - alias: 'sgd'
  - specify the application content location: (...\\smartguide\\6.x.x\\sgd)
  - for the application pool click on select and choose the latest .Net framework in integrated mode (i.e.: .Net 4.5)
2. Add another application to the default web site (or a newly created site) from the IIS administrator:
  - alias: 'smartlets'
  - specify the application content location: (...\\smartguide\\6.x.x\\smartlets)
  - for the application pool click on select and choose the latest .Net framework in integrated mode (i.e.: .Net 4.5)
3. Install the license file
  - Follow the explanations from section 7 below.
4. Configure the Smartlets repository
  - For disk mode, refer to section 5
  - For database mode, refer to section 6
5. Test the configuration

The SmartGuide Designer application should be accessible from the following URL `http://localhost:<port>/sgd`

The home page will appear as follows:



You may view your Smartlets directly from SmartGuide Designer. In the *Smartlets* section, mouse over the Smartlet you wish to view and click the **View Smartlet** icon:



Alternately, you can click on the icon located to the left of the name of the Smartlet and select the **View Smartlet** option.

---

## Important notes

Both the *SGD* and *Smartlets* applications can be configured to run under any version of the .Net framework greater than or equal to 2. However, when running SGD under .Net framework 2, make sure to comment the following line in the *web.config* file found under the *sgd\ajax* directory:

```
<httpRuntime requestPathInvalidCharacters="" requestValidationMode="2.0" />
```

**Note 1**

If you must remove anonymous access to the *sgd* directory, make sure that the sub-directory *smartletsruntime* can still be accessed anonymously or the Smartlet page preview option will not work.

**Note 2**

If running under IIS 10.0 make sure to comment the following line found in the *web.config* file under the *SGD* and *Smartlets* applications:

```
<mimeMap fileExtension=".woff2" mimeType="application/font-woff2" />
```

# Deployment on disk

## 5

This chapter will guide you, step by step, through the configuration process of SmartGuide to store the Smartlets on disk.

### Configuring the environment

➤ **Enable the file system mode**

1. Access the SmartGuide Designer installation directory:  
e.g.: '..\smartguide\6.x.x\sgd\'
2. Open the **Web.config** file in a text editor.
3. Uncomment the following section:

```
<!--File system connection sample -->
<!--
<add key="jdbc-driver"
value="com.alphinat.smartguidedesigner.fileaccess"/>
<add key="jdbc-url" value="c:\sgd_smartlets"/>
-->
```

This section, commented by default, allows you to save the Smartlets and related data on the disk file system. The only parameter to configure is **jdbc-url**. The value of this parameter represents the location on the disk where the Smartlets are saved.

Make sure to remove or place in comments the database configuration settings detailed in the previous chapter when using the file mode to avoid storage conflicts.





---

This chapter will guide you, step by step, during the SmartGuide installation process, using a database support for storing the Smartlets.

---

## Creating the database

If you are configuring SmartGuide to store its data in a database, you must build the database structure by executing the SQL file corresponding to the database management system you wish to use:

| Database             | DDL file name to be executed   |
|----------------------|--------------------------------|
| Oracle               | createtables_oracle.sql        |
| Microsoft SQL Server | createtables_sqlserver2000.sql |
| IBM DB2              | createtables_db2.sql           |
| PostgreSQL           | createtables_postgre.sql       |
| MySQL                | createtables_mysql.sql         |
| Firebird             | createtables_ib.sql            |

You can obtain these files from the download site provided to you by Alphinat.

---

## Configuring the environment

- **Configure the connection to the database**
  1. Access the SmartGuide Designer installation directory:  
e.g.: '..\smartguide\6.x.x\sgd\'
  2. Open the **Web.config** file in a text editor.
  3. Configure the **ado-connection-string** key in order to define the database SmartGuide will connect to:
    - **Microsoft SQL server 2000 and up** – Uncomment the following section in the *Web.config* file:

```
<!-- MS SQL server connection sample -->
<!--
<add key="database-product-name" value="Microsoft SQL Server"/>
<add key="ado-provider" value="System.Data.SqlClient"/>
<add key="ado-connection-string" value="Server=localhost;
Database=smartguidedb;User Id=username; password=password;"/>
```

You must replace the values that are highlighted above to match your system settings. If you use a name other than *smartguidedb* for the database, make sure you modify the first line of the *createtables\_sqlserver2000.sql* script accordingly.

- **Oracle Database** – Search for the following section in the *Web.config* file:

```
<!-- Oracle connection sample -->
<add key="database-product-name" value="Oracle"/>
<add key="ado-provider" value="System.Data.OracleClient"/>
<add key="ado-connection-string" value="Data
Source=localhost:1521/smartguidedb;User
ID=username;Password=password"/>
```

You must replace the values that are highlighted above to match your system settings. If you use a name other than *smartguidedb* for the database, make sure you modify the first line of the *createtables\_oracle.sql* script accordingly.

# License Installation

## 7

---

### Installing the license file

Each SmartGuide application, whether it is the Designer or the Server, requires a license file in order to function. This includes the « smartguideserver.zip » shell deployment in the entirety of your SmartGuide Server installations. When you acquired SmartGuide, you should have received a *smartguide.lic* license file to deploy on your installation.

The *smartguide.lic* file must be copied under `..\smartguide\6.x.x\sgd\data` and under `..\smartguide\6.x.x\smartlets\data`.

Alternately, you can specify the location of the licence file using the following configuration key in the *Web.config* file of SmartGuide Designer and SmartGuide Server:

```
<!-- Configuration parameters -->
<!--
<add key="com.alphinat.sg.license.Path"
value="c:\path_to_license\smartguide.lic"/>
-->
```



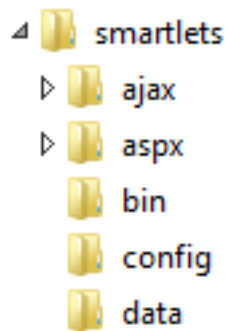
# Runtime Environment

## 8

When deploying to a production environment, only the Server portion of SmartGuide is required. This chapter gives additional information about the structure of SmartGuide Server.

### SmartGuide Server directory structure

The following illustration represents the directory structure:



The *aspx* directory contains the themes as well as the templates used for all the Smartlets.

**Note:** A theme is defined in SmartGuide Designer and can be used by one or multiple Smartlets. By default, a single template is used by all the pages of a Smartlet unless a specific template has been assigned to a page.

In the *aspx* directory you will find a sub-directory named *default*. This directory represents the default presentation theme delivered with SmartGuide.

For each additional theme, a directory using the name of this theme will be added in the *aspx* directory or in the workspace sub-directory if that functionality has been enabled in the *Web.config* file.

Each theme (e.g. default, test, etc.) must contain a sub-directory named *default*. This directory corresponds to the default template that is used to display the Smartlet pages.

Please refer to the ASPX Template Author Guide for detailed information on how to create and manage presentation themes.

---

## Runtime environment installation

Follow the same steps outlined in section 4, but create only the 'smartlets' IIS application.

It is recommended to configure the *com.alphinat.sgs.repository.Path* registry key, detailed in section 9, to point to a copy of the Smartlets you want to run on disk. Once the Smartlets are deployed on the server you can access them through a URL of the following form:

[http://localhost:<port>/smartlets/do.aspx?interviewID=\[code\]](http://localhost:<port>/smartlets/do.aspx?interviewID=[code])

where [code] of the Smartlet code (directory name containing the Smartlet on disk).

# Configuration parameters

## 9

This chapter details the configuration parameters that can be modified for SmartGuide Designer as well as SmartGuide Server. These configuration parameters are located in the *Web.config* files under `..\smartguide\6.x.x\sgd\` and `..\smartguide\6.x.x\smartlets\`.

### SmartGuide Designer

#### File access mode

```
<add key="jdbc-driver"
value="com.alphinat.smartguidedesigner.fileaccess"/>
<add key="jdbc-url" value="C:\smartlets"/>
```

This section allows saving the smartlets as well as all other SGD related information on the file system. This is the preferred deployment method as it allows easy backup, and integration into source code control systems. The path can be local to the server, or it can be on the network (i.e.: [\\servername\sharename](#)). The only parameter to configure is jdbc-url.

For a configuration making use of a database please refer to the sections specific to each application server.

#### Path to presentation themes

```
<add key="com.alphinat.sgs.themes.Path"
value="C:\inetpub\wwwroot\smartguide\6.x.x\smartlets\aspx"/>
```

This section, commented by default, allows specifying the location on the disk where the themes are saved. In a standard SmartGuide installation, these are available under `..\smartguide\6.x.x\smartlets\aspx`. When this section is active, all theme manipulations in the SmartGuide Designer

interfaces (« themes » tab) are reflected on the disk file system. This includes the import of themes, their duplication, their deletion and even the editing of the .aspx file contents.

## Theme synchronization

```
<add key="com.alphinat.sgs.themes.SynchronizeMode"
value="fromdisk"/>
```

This section, commented by default, allows the activation of synchronization between the theme files on disk the ones stored in the SmartGuide database. When a user has direct access to the theme files on disk to modify the aspx files, it is recommended to activate this parameter. This parameter requires the *com.alphinat.sgs.themes.Path* above to be defined as well.

The other accepted value for this parameter is "todisk". It is usually used in more complex scenarios where Smartlets are stored in a database replicated from another location. In such cases a check must be performed by SmartGuide to validate which files on disk are no longer in sync with the database.

## Path to license file

```
<add key="com.alphinat.sg.license.Path"
value="C:/path_to_license"/>
```

This section, commented by default, allows specifying the location on the disk where the SmartGuide Designer license is installed and must include the name of the license file. If this key is commented, by default SmartGuide Designer tries to find the *smartguide.lic* license file in the *../smartguide/6.x.x/sgd/data* directory.

## Preview URL

```
<add key="com.alphinat.sgd.smartlet.PreviewUrl"
value="http://host:port/path_to_sgs"/>
```

This section, commented by default, allows to specify the URL to use to view the Smartlets from SmartGuide Designer. Please note that if a value has already been specified for a Smartlet, under the « Properties » tab, field « Configure the Smartlet URL », then the latter will prevail.



## Default Designer language

```
<add key="com.alphinat.sgd.DefaultLanguage" value="en"/>
```

This section, commented by default, defines the default language in which to load SmartGuide Designer. There are currently 2 values supported, "en" and "fr".

## Log files

```
<add key="com.alphinat.sg.log4j.properties.Path" value="apn-  
log4j.xml"/>
```

This section, commented by default, allows specifying the pathname of the configuration file used by log4j for the logging of messages from SmartGuide Designer. If it is not specified, default values will be used for logging (only errors, sent to the console).

```
<!--Log file location -->  
<add key="out-log" value="/sgd-out.txt"/>  
<add key="err-log" value="/sgd-err.txt"/>
```

This section, commented by default, allows specifying the location on disk where you can find the SmartGuide Designer log files. The file defined by the keyword *err-log* contains the SmartGuide execution errors while the file defined by the keyword *out-log* contains SmartGuide descriptive execution information. It is recommended to uncomment the previous key to allow proper configuration of log4j. The "console" as defined in the log4j config file will actually output to the out-log and err-log files specified.

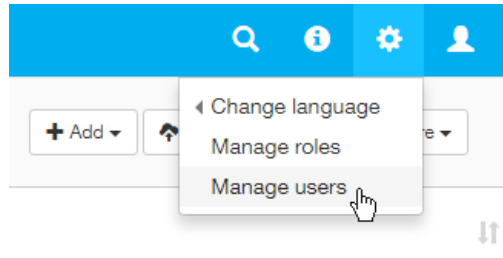
## Authentication

```
<add key="com.alphinat.sgd.auth.Type" value="apn"/>  
<add key="com.alphinat.sgd.auth.Administrator" value="admin"/>
```

This section, commented by default, allows defining the authentication type used as well as the username of the system administrator.

There are 3 possible values for the authentication type : *apn*, *jaas*, and *iis*. The value *apn* correspond to an internal user base in SmartGuide. The value *jaas* is for j2ee using the JaaS security model (for example Jboss). The value *iis* is for Windows installations running under IIS with the windows integrated security.

When connecting to a new secure SmartGuide installation, the password for the administrative username is « admin ». It can be changed by selecting the *Manage users* options from the upper right portion of the screen:



The following key allows you to constrain the format of usernames and passwords for the internal users of SmartGuide through the use of regular expressions:

```
<add key="com.alphinat.sgd.auth.UsernameRegex" value="([A-Za-z0-9]+)"/>
<add key="com.alphinat.sgd.auth.PasswordRegex"
value="(=.{9,})(?=.*?[^\w\s])(?=.*?[0-9])(?=.*?[A-Z]).*[a-z].*" />
```

## Workspaces

```
<add key="com.alphinat.sg.workspace.Enabled" value="true"/>
```

This section, commented by default, allows specifying whether or not to enable workspaces. The concept of workspaces is present in SmartGuide since version 5.3. Please note that if you go from an installation without workspaces to one which has them, you will need (if storing your Smartlets on disk) to first empty the contents of the directory specified by *jdbc-url* above, start SmartGuide Designer, and finally recopy the content under *jdbc-url/main*. You will need to do the same if you had uncommented the *com.alphinat.sgs.themes.Path* parameter.

## Mail server

```
<!-- Mail Server Configuration. You can specify the mail server
name, port, username and password, and whether or not is uses SSL.
-->
<add key="com.alphinat.sg.mail.From" value="noreply@domain.com"/>
<add key="com.alphinat.sg.mail.Server" value="smtp.gmail.com"/>
<add key="com.alphinat.sg.mail.Port" value="587"/>
<add key="com.alphinat.sg.mail.Username" value="user"/>
<add key="com.alphinat.sg.mail.Password" value="pass"/>
<add key="com.alphinat.sg.mail.UseSSL" value="true"/>
```

This section, commented by default, allows specifying the parameters for a mail server. This will enable functionalities like password recovery in the Designer interface where an email with a password reset link can be sent to users who forgot their password.

The other *Web.config* sections are of internal nature and should not be modified without the support of Alphinat.

---

## SmartGuide Server

### Upload directory

```
<!-- Directory-based file upload settings -->  
<add key="com.alphinat.sgs.UploadedFileDir" value="c:\temp"/>
```

This section, commented by default, allows specifying a directory on the disk where files uploaded by end-users are saved. By default, SmartGuide Server saves files to memory which limits the size of the files that can be uploaded. You can therefore uncomment this section to increase the file upload limit.

Please note that in memory storage mode, the uploaded files are cleared from memory when the session ends. However, when storage of uploaded files on disk is selected, files will be kept on disk even when the session ends. You must therefore anticipate the purging mechanisms of this directory. Also, please take note that the disk storage format is in the directory/<some numeric id>/name form of the file (e.g. c:\upload\1246367608106\test.pdf).

### Parameter encoding

```
<add key="com.alphinat.sgs.parameter.Encoding" value="UTF-8"/>
```

This section, commented by default, allows specifying the encoding used when posting parameters to the server. By default, SmartGuide uses the UTF-8 encoding.

### Localization

```
<add key="com.alphinat.sgs.localization.LanguageParameter"  
value="lang"/>
```

This section, commented by default, specifies the name of the parameter that, if present in the URL (or posted) will ask the Smartlet to change its current locale to the one specified. If not specified then the default name *com.alphinat.sgs.lang* is used.

## Error handling

```
<add key="com.alphinat.sgs.error.handling.Type" value="custom"/>
<add key="com.alphinat.sgs.error.handling.CustomPageUrl"
value="customerror.aspx"/>
```

This section, commented by default, specifies how errors are processed by SmartGuide. The type can be *silent*, where no actions are performed, *custom*, where a custom page parameter must be provided to serve as a redirect page in case of errors, or *throw* which simply throws the error.

## Action processing

```
<add key="com.alphinat.sgs.actions.StopProcessingActionTypes"
value="all"/>
```

This section, commented by default, specifies the behavior of SmartGuide when processing a sequence of actions. When processing different type of actions, like sending mail, calling a service, executing BeanShell scripts, SmartGuide can decide to either stop processing or continue. By default it continues processing. You can specify which type of actions will stop processing. Possible values are: *runbsh*, *sendmail*, *callservice* and *all*. You can specify more than one value, separated by commas.

## Timeout settings

```
<add key="com.alphinat.sgs.http.ConnectTimeout" value="10"/>
<add key="com.alphinat.sgs.http.SocketTimeout" value="10"/>
```

This section, commented by default, specifies the connection timeouts for web service calls, both SOAP and REST. The value is the number of seconds to wait before aborting. The *ConnectTimeout* is for the time to wait before getting a connection on the server. The *SocketTimeout* is for the time to wait after a connection, before getting data from the server.

## Retries

```
<add key="com.alphinat.sgs.http.ConnectRetries" value="3"/>
```

This section, commented by default, specifies the number of retries SmartGuide Server should perform on web service calls both SOAP and

REST. By default, SmartGuide uses 1 which means that if the web service call fails it is immediately reported back and no retry is performed.

## Stateless deployment settings

```
<add key="com.alphinat.sgs.session.storage.Key" value=""/>
<add key="com.alphinat.sgs.session.storage.Scope" value="request"/>
<add key="com.alphinat.sgs.session.retrieval.Key" value=""/>
<add key="com.alphinat.sgs.session.retrieval.Scope"
value="parameter"/>
```

This section, commented by default, allows you to configure SGS such that session information (state of the Smartlets, value of fields, etc.) can be saved in a different location than the IIS server session (in case session is not allowed on IIS, for example.)

There are two configuration blocks, one for session information storage and the other for its recovery.

The `com.alphinat.sgs.session.storage.Key` and `com.alphinat.sgs.session.retrieval.Key` parameters define the key names used to access the information.

The `com.alphinat.sgs.session.storage.Scope` and `com.alphinat.sgs.session.retrieval.Scope` parameters allow defining the context within which the information will be saved. The values allow defining the context within which the information will be saved. Possible values are:

- **Application:** use the application context.
- **Session:** uses the session context (default method when this section is in discussion).
- **Request:** use the page context.
- **Parameter:** use a hidden field in the page that will be transmitted during the submission of the « form » in which it is located.

For example, in an environment that possesses its own information storage mechanism, it would be possible to use *request* for the storage and retrieval scope. You would then recuperate the SG information from this external system and put it in a page variable specified by the name: `com.alphinat.sgs.session.retrieval.Key`.

This should be done before the SmartGuide control implementation. Following this action, you can retrieve the current SmartGuide session information from the page variable named `com.alphinat.sgs.session.storage.Key` and save it back to the external information storage mechanism.

### Check Smartlet version

```
<!-- Disable checking new version of smartlet each time executing
Smartlet, it will improve performance under production environment
In order to deploy new version of smartlet, sgs application has to
be restarted. -->
<add key="com.alphinat.sgs.DisableTimestampCheck" value="true"/>
```

This section, commented by default, allows disabling the reference time verification of the Smartlet. Take note that when this parameter is active, all Smartlet updates will require an SGS restart in order to take effect.

### Path to Smartlet repository

```
<add key="com.alphinat.sgs.repository.Path"
value="http://localhost/sgd/smartletsruntime/do.aspx"/>
```

This section, commented by default, allows specifying where the Smartlets are to be retrieved from. By default the parameter has no value and SmartGuide Server will perform an http call to SmartGuide Designer in order to get the Smartlets. The value of this parameter can be a path to a directory on disk or a relative path to the web application. For example with the value « `/mysmartlets` », SmartGuide Server will look for a « `mysmartlets` » directory under your smartlets application installation (for example `c:\www\smartlets`).

Please note that for each new instance of the Smartlet, if the *`com.alphinat.sgs.DisableTimestampCheck`* parameter above is false or commented, there will be a timestamp check on the Smartlet.

### Data processing mode

```
<add key="com.alphinat.sgs.data.ProcessingMode" value="strict"/>
```

This section, commented by default, allows to specify whether strict data validation should be enforced or not (on the click of a button in a

Smartlet). For example, if the end-user enters in a Smartlet and decides, at page 2, to open a new browser window, both browser windows share the same session for the Smartlet, but could be in different states. If the user clicks and posts data, and this data doesn't correspond to the last known state on the server, then the data and attached actions for the click will be discarded and the user will be presented with the last known page to the server.

This parameter is strongly recommended for transactional type applications. Otherwise data loss could occur. In the preceding paragraph example, a change of decision on page 2 could affect the path taken in the application (different pages visited afterward). If the user comes back to page 1 and tries to submit data, these would then be inconsistent with the last choice made on page 2.

### Ignored field types

```
<add key="com.alphinat.sgs.data.IgnoreFieldTypes"
value="staticImg,staticText,hid"/>
```

This section, commented by default, lists the field types to ignore when processing submitted data. In the example above, static images, static text, and hidden field types are added to the list of fields to be ignored. So when trying to pass a parameter through the URL for example for a field of one of these types, it will be ignored.

### Extension functions

```
<add key="com.alphinat.sgs.extlib.CheckScope" value="application"/>
```

This section, commented by default, allows specifying when the extension functions are loaded by SmartGuide. By default, extension functions are loaded at the same time as the SmartGuide Server application. This is the recommended setting for production environments. However, in a development environment, you may want to change this setting to reload extensions functions on every request so that changes to the extension function assemblies are automatically reflected in the Smartlet execution without having to restart the application.

Possible values are:

- **Application:** at application load time
- **Session:** at session initialization time (default)

- **Request:** at every request

The following section, commented by default, allows disabling the reference time verification of the extension functions loaded by SmartGuide:

```
<add key="com.alphinat.sgs.DisableExtlibCheck" value="true"/>
```

Please note that when this parameter is active, all Smartlet updates will require an SGS restart in order to take effect. This is functionally equivalent to the *com.alphinat.sgs.extlib.CheckScope* with value *application*.

### Path to license file

```
<add key="com.alphinat.sg.license.Path"
value="C:/path_to_license_including_filename"/>
```

This section, commented by default, allows specifying the full path on disk of the license file for SmartGuide Server. If not specified, SmartGuide Server will try to read the license from *smartguide.lic* under a data directory under the smartlets application.

### Log files

```
<add key="com.alphinat.sg.log4j.properties.Path" value="apn-
log4j.xml"/>
```

This section, commented by default, allows specifying the pathname of the log4j configuration file for SmartGuide server. If not specified, then default values will be used for logging (errors only, sent to the console).

```
<!-- Log file location -->
<add key="out-log" value="/sgd-out.txt"/>
<add key="err-log" value="/sgd-err.txt"/>
```

This section, commented by default, allows specifying the location on disk where you can find the SmartGuide Server log files. The file defined by the keyword *err-log* contains the SmartGuide execution errors while the file defined by the keyword *out-log* contains SmartGuide descriptive execution information. It is recommended to uncomment the previous key to allow proper configuration of log4j. The "console" as defined in the log4j config file will actually output to the out-log and err-log files specified.



## Log message size

```
<add key="com.alphinat.sgs.logging.MessageMaxSize" value="2048"/>
```

This section, commented by default, specifies the maximum size in bytes of messages logged by web service calls. If not specified, log messages related to web service calls are not limited and can use a lot of space in your log files when in debug mode.

## Workspaces

```
<add key="com.alphinat.sg.workspace.Enabled" value="true"/>
```

This section, commented by default, allows specifying whether or not to enable workspaces. You can find more information in the SmartGuide Designer parameter section above.

## HTTP proxy

```
<add key="com.alphinat.sg.http.ProxyHost" value="host_name"/>  
<add key="com.alphinat.sg.http.ProxyPort" value="host_port"/>
```

This section, commented by default, allows specifying an http proxy to use when making web service calls.

## Template rendering

```
<add key="com.alphinat.sgs.template.Path"  
value="/test/aspx/{theme}/{template}/default.aspx"/>
```

This section, commented by default, allows to render themes from different locations than the default one under smartlets/aspx. The following placeholders are defined: {theme}, {template}, {smartlet}, {domain}, {workspace} and will get replaced by their actual value in the path above.

## Page auto-refresh

```
<add key="com.alphinat.sgs.page.AutoRefreshEnabled" value="true"/>
```

This section specifies whether or not to enable the auto-refresh mechanism in SmartGuide themes. For example, changing the value of a dropdown field can automatically modify the value of another field on the page without reloading the whole page from the server.

## Mail server

```
<!-- Mail Server Configuration. You can specify the mail server
name, port, username and password, and whether or not is uses SSL.
-->
<add key="com.alphinat.sg.mail.From" value="noreply@alphinat.com"/>
<add key="com.alphinat.sg.mail.Server" value="smtp.gmail.com"/>
<add key="com.alphinat.sg.mail.Port" value="587"/>
<add key="com.alphinat.sg.mail.Username" value="user"/>
<add key="com.alphinat.sg.mail.Password" value="pass"/>
<add key="com.alphinat.sg.mail.UseSSL" value="true"/>
```

This section, commented by default, allows specifying the parameters for a mail server. This will enable the sendmail action in Smartlets you create, or the sendmail API function in extension functions you develop.

## Base URL for email images

```
<add key="com.alphinat.sgs.resources.baseURL" value="
http://mysite.com/resources/" />
```

This section, commented by default, allows you to specify the base URL of the resources embedded in your Smartlet when using the resource manager of the Designer. This base URL will substitute the current relative URL for resources in email messages sent through the SendMail action in the Designer.

For example, suppose you embed an image in a message sent to a user. By default the URL for the image will look like:

`./resources/do.aspx?file=myimage.jpg`

However, when the message is viewed in an email client, the image would not be shown because of its relative path. A possible value for the *baseURL* would therefore be <http://myserver.com/smartlets/> for a standard SmartGuide installation.

## Multi-Smartlet mode

```
<add key="com.alphinat.sgs.MultipleSmartletsEnabled" value="true"/>
```

This section, commented by default, enables the multiple Smartlets per session mode. By default, in a given user session, only one Smartlet can be active at a time unless you make use of the sub Smartlet functionality

or GoToSmartlet actions (or API) to navigate through Smartlets. But even in such cases only one Smartlet displays information at a time.

With multiple Smartlets mode enabled you can create portal type application with many Smartlets rendering different parts of a page (please consult the ASPX template author guide for more details). Or you can allow your user to have different browser tabs open on different Smartlets without conflicts.

### **HTTP session reuse**

```
<add key="com.alphinat.sgs.ReuseHTTPSessionsEnabled" value="true"/>
```

This section, commented by default, enables the reuse of a http session across web service calls. In other words, if a first web service call receives cookies from the server, these will be passed on in subsequent requests to the same server.

### **Language change detection**

```
<add key="com.alphinat.sgs.history.LangEnabled" value="true"/>
```

This section, commented by default, enables the detection of history modification when the language changes. For example, if navigation is affected by the language in which the Smartlet is displayed, then after modifying the language, SmartGuide Server might prevent the user from going directly back to a recap page, and require the user to navigate page by page because the navigation was modified by the language change.

### **Disable all validations**

```
<add key="com.alphinat.sgs.debug.ValidationsEnabled" value="true"/>
```

This section, commented by default, will disable validations on all Smartlets. To be used for development purposes only. Do not use on production servers.

The other *Web.config* sections are of internal nature and should not be modified without the support of Alphinat.