Presto Advanced Administrator's Guide

Presto 3.5

Document date: September 29, 2010

Contents

Introduction

Inmagic® *Presto* is a social knowledge management platform that provides the ability to organize, search, access, and contribute to information within organizations. The *Presto* interface, which provides a portal to information resources, can be modified for individual users and organizations.

About this Guide

Purpose

This document explains how administrators can perform advanced configuration of *Presto*.

Audience

This document is intended for Inmagic Professional Services and authorized Inmagic personnel or Inmagic Partners who have been trained to perform these operations. This document assumes a working knowledge of the *Presto* application and architecture, and familiarity with XML and Microsoft SQL.

Related Information

- The *Presto Basic Administrator's Guide* and online help, which are installed with *Presto*, explain the basic operations available in the user interface, including how to perform standard configuration tasks using the **Configure** menu.

- Supplemental information is supplied in various *Configuration Guides*.

Configuration Methods

Presto is highly customizable. The user interface and functionality can be adjusted in the following ways.

Standard Configuration

Within *Presto*, use the **Configure** menu to adjust Content Types (including Screens and Views), Collections, Blogs, Forums, Reports, Value Lists, Home Page, Menus and Menu Bar, Users, Roles, Events, and Global Alerts. For instructions, see the online help or the *Presto Basic Administrator's Guide* installed with the product.

Advanced Configuration

You can use the following advanced configuration methods:

- Adjust the Application Settings (**Configure>Administrator Tools>Application Settings**)
- Modify the Presto **Web.config** file
- Modify **<application>.exe.config** files
- Configure log files
- Modify other files outside of *Presto*, such as css files, JavaScript, icons, etc.
- Customize and/or localize text that appears in the *Presto* user interface.

All of these methods are described in this document.

Web.config

The Presto **Web.config** file is an XML file that defines *Presto* Web application settings for ASP.NET and the .NET Framework. This file is configured when you install *Presto* and rarely needs to be modified. Changes can be made using any Unicode text editor that supports UTF-8 (such as Windows Notepad).

Note: Every ASP.NET application has its own Web.config file located in the application's root directory. For more information about ASP.NET Web.config files, see the MSDN documentation available online. For example: http://msdn.microsoft.com/en-us/library/ackhksh7(vs.71).aspx.

Location:	<install-dir>\WebApps\Mercury**Web.config**
Example:	C:\Program Files\Inmagic\Presto\WebApps\Mercury\Web.config
Purpose:	Intended primarily for IT staff, this is the governing file for a single *Presto* instance. It identifies the *Presto* Windows identity and defines *Presto* application settings for ASP.NET and the .NET Framework, including which Membership Provider to use for authentication (such as Active Directory), which bindings to use for Web services, e-mail server configuration, and similar settings.
Guidelines:	Rarely needs to be modified. Only *Presto* application engineers, IT administrators, or other qualified personnel should edit this file.
Editor:	XML (Unicode UTF-8)

See the following sections for information about features that can be adjusted by editing **Web.config**. Additional features that are not listed below may be covered in separate *Configuration Guides*.

Feature	Page
presto.ApplicationSettingsSection of Web.config	6
SQL Server or Windows Authentication	6
Log Files	8
Session State	24
Timeout	25
Failed Password Attempts	28
Active Directory and Single Sign On (AD SSO)	30

presto.ApplicationSettingsSection of Web.config

The Presto **Web.config** file includes a section that specifies the presto.ApplicationSettings:

- IsHosted, hostedUserName, hostedPassword: Reserved for use by Inmagic hosted systems.

- prestoIdentity: The 'PrestoUser' account which serves as the Presto application identity. This account is used to control file system access or Windows operating system services access.

- prestoEmailSettings: Specifies e-mail addresses for urgent administrative e-mail to Inmagic, Inc.

- prestoMSMQSettings: Should never be modified.

- prestoWebApplicationParameters: Set by the installation program. Identifies the *Presto* Web application's root path, directory name, protocol, server name, and port number. Should be changed only by trained personnel.

The sample below shows a typical <presto.ApplicationSettingsSection> from the *Presto* **Web.config** file. Your configuration may differ.

```
<presto.ApplicationSettingsSection>
  <prestoInstanceSettings instanceName="Presto"
   isHosted="false"
   hostedUserName="99FBA1BAFB6DB792E00S3546A0BAC243AEE7E0C5"
   hostedPassword="D315244C4360F9E581E69CEC79RDC1A43DB9E24D" />
  <prestoIdentity userName="PrestoUser" password="8lx65pOOafJgP1WmhW1ikw==" domain="Inmagic" />
  <prestoEmailSettings
   adminEmail="admin@presto.Inmagic.com"
   notificationEmail="notify-noreply@presto.Inmagic.com" />
  <prestoMSMQSettings isMSMQEnabled="true" computerName="." />
  <prestoWebApplicationParameters rootPath="C:\Program Files\Inmagic\Presto\"
   directoryName="Mercury"
   serverName="MYSERVER"
   protocol="http"
   portNumber="" />
</presto.ApplicationSettingsSection>
```

SQL Server or Windows Authentication

Presto can use SQL Server Authentication or Windows Authentication to connect to SQL server. To change the preferred authentication method, modify the ASPNETDB connection string in the Presto **Web.config** file, as described below. The connection strings for v3.5 and earlier are identical, except for the default authentication method.

For *Presto* v3.5 or later:

Presto v3.5 uses SQL Server Authentication and typically uses an encrypted connection string. **Note:** For complete information, see *Using the ConfigUtil.exe Tool Configuration Guide*. The commands are summarized below.

- To use Windows Authentication: Use the **Configutil** tool to either a) Remove the SQL username and password (**rmupdbinfo**) from the connection string; or b) Decrypt the connection string (**unprotectdbinfo**) so you can then modify it as desired. There is no need to protect (encrypt) the connection string for Windows Authentication.

- To use SQL Server Authentication (the default): Use the Configutil **setdbinfo** command to specify the user name and password and protect (encrypt) the connection string.

When you encrypt (protect) or decrypt (unprotect), you affect *all* connection strings in the <connectionStrings> section. In the examples below, the ADCONNSTRING would be affected, as well as the ASPNETDB string.

For earlier versions of *Presto*:

- Earlier versions of *Presto* use Windows Authentication (unencrypted). Simply modify the unencrypted connection string to change the authentication method.

Example #1: SQL Server Authentication connection string

Typically, the SQL Server Authentication connection string is encrypted. However, the example below shows an unencrypted string, since an encrypted example would not be useful.

```
<connectionStrings>
  <remove name="LocalSqlServer" />
  <add name="ASPNETDB" connectionString="Server=CompanySqlServer\CompanyName;
Initial Catalog=PrestoCatalog;Trusted_Connection=False;Min Pool Size=5;Max Pool Size=100;
Connect Timeout=120;MultipleActiveResultSets=true;Integrated Security=false;User
ID=PrestoSQLUser;Password=Cle@rT3xt;"
    providerName="System.Data.SqlClient" />
  <add name="ADCONNSTRING"
connectionString="LDAP://DC1.SOMEWHERE.LOCAL/CN=USERS,DC=SOMEWHERE,DC=LOCAL" />
  </connectionStrings>
```

Example #2: Windows Authentication connection string

```
<connectionStrings>
  <remove name="LocalSqlServer"/>
  <add name="ASPNETDB" connectionString="Server=CompanySqlServer\CompanyName;
Integrated Security=true;Initial Catalog=PrestoCatalog;Min Pool Size=5;Max Pool Size=100;
Connect Timeout=120;MultipleActiveResultSets=true;"
    providerName="System.Data.SqlClient"/>
  <add name="ADCONNSTRING"
connectionString="LDAP://DC1.SOMEWHERE.LOCAL/CN=USERS,DC=SOMEWHERE,DC=LOCAL" />
</connectionStrings>
```

\<application\>.exe.config Files

*The \<application\>.exe.config files should **not** be modified, except in rare troubleshooting situations, when advised by Inmagic Engineering and Support.*

Per standard NET architecture, *Presto* includes one Windows application configuration file for each internal application. For example, **ConfigUtil.exe** has an associated configuration file called **ConfigUtil.exe.config**.

- Each file includes a connection string to the *Presto* Catalog on SQL Server.
- Each file identifies membership providers.
- Each file references Web.config for its common (shared) configuration settings.

An example of when you might need to modify these files is in the unlikely event that the SQL database is renamed after installing *Presto*. In that case, you would need to change the connection string in each file. If you modify these files, be sure to use an XML editor that supports Unicode UTF-8.

The \<application\>.exe.config files reside in the following location:

> `<install-path>\WebApps\Mercury\bin\`**`<application>.exe.config`**

For example:

> `C:\Program Files\Inmagic\Presto\WebApps\Mercury\bin\`**`ConfigUtil.exe.config`**

Presto includes the following \<application\>.exe.config files:

```
BootStrapper.exe.config
ConfigUtil.exe.config
ContentExport.exe.config
ContentLoader.exe.config
Upgrade.exe.config
RebuildTermIndexes.exe.config
```

Log Files

There are two configuration files for configuring log files:

- C:\Program Files\Inmagic\Presto\WebApps\Mercury**Web.config**
- C:\Program Files\Inmagic\Presto\Presto Services**Inmagic.Mercury.Services.PrestoServices.exe.config**

Web.config specifies the logging configuration for the *Presto* Web application. The location of each log file is specified in the \<loggingConfiguration\> section of Web.config. For example:

> C:\Program Files\Inmagic\Presto\WebApps\Mercury\system\log**Errors.log**
> C:\Program Files\Inmagic\Presto\WebApps\Mercury\system\log**CriticalErrors.log**
> C:\Program Files\Inmagic\Presto\WebApps\Mercury\system\log**Upgrade.log**
> C:\Program Files\Inmagic\Presto\WebApps\Mercury\system\log**Presto.log**

Inmagic.Mercury.Services.PrestoServices.exe.config specifies the logging configuration for the Presto Services executable, which is the shell that starts up the service workers. Default locations are shown below:

> C:\Program Files\Inmagic\Presto\Presto Services\log**Errors.log**
> C:\Program Files\Inmagic\Presto\Presto Services\log**Presto.log**

To change the location or attributes of a log file, search the appropriate .config file for the \<loggingConfiguration\> section, where you can adjust the following attributes in the \<listeners\> section.

Attribute	Description
timestampFormat="yyyy-MM-dd [HH-mm-ss-fff]"	The date/time format that will be used in the file.
ageThreshold="0"	How long the log file will be retained before it is automatically deleted from disk.
ageUnit="None"	The unit for the ageThreshold.
sizeThreshold="5"	The maximum allowed size of the file, as defined by its sizeUnit. When the threshold is reached, the file is renamed to have a timestamp at the end, and a new file is created. For example, the existing Errors.log gets renamed to Errors<timestamp-info>.log and a new Errors.log is created.
sizeUnit="Megabytes"	The unit of the sizeThreshold.
maximumNumberOfLogs="5"	Maximum number of log files.
fileName="<path>\<filename>.log"	The full path and name of the log file, on the server where *Presto* is installed.

Application Settings

Application Settings are typically modified immediately after installing *Presto*, and periodically after that, if necessary. **Important!** Changing these settings can affect the stability, security, and performance of the *Presto* Web application. Only advanced users should adjust these settings.

Note: For explanations of each setting, see the *Glossary of Application Settings* on page 11.

To adjust Application Settings

1. Log in to *Presto* with Administrator rights.

2. Choose **Configure>Administrator Tools**.

3. Click **Application Settings** to access the page shown below.

4. Modify the settings, as described in this document. When making changes to long entries, you can edit the XML in-place or copy it into a UTF-8 Unicode editor, such as Windows Notepad, make your

changes, then copy it back into the field. Each setting is described in the *Glossary of Application Settings*, starting on page 11.

5. Click **Save** to save your changes, or click **Cancel** to close without saving.

To revert Application Settings to the last-saved values

If you accidentally make changes to Application Settings that render *Presto* inoperable, you can follow the instructions below to revert to a previously-saved setting.

Important! This procedure involves directly editing a SQL database table and should be done only by Inmagic personnel or administrators who have SQL database experience.

1. Start **Microsoft SQL Management Studio** on the SQL database server. Note that SQL Server Tools must be installed.

2. Open the **dbo.AppConfig table** in the *Presto* Catalog.

3. Locate the column called **PreviousValue** (you may need to scroll to the right) and copy the value from that column into the **Value** column.

   The following example shows the **dbo.AppConfig** table open for a catalog called PrestoCatalog. The name of your catalog may be different.
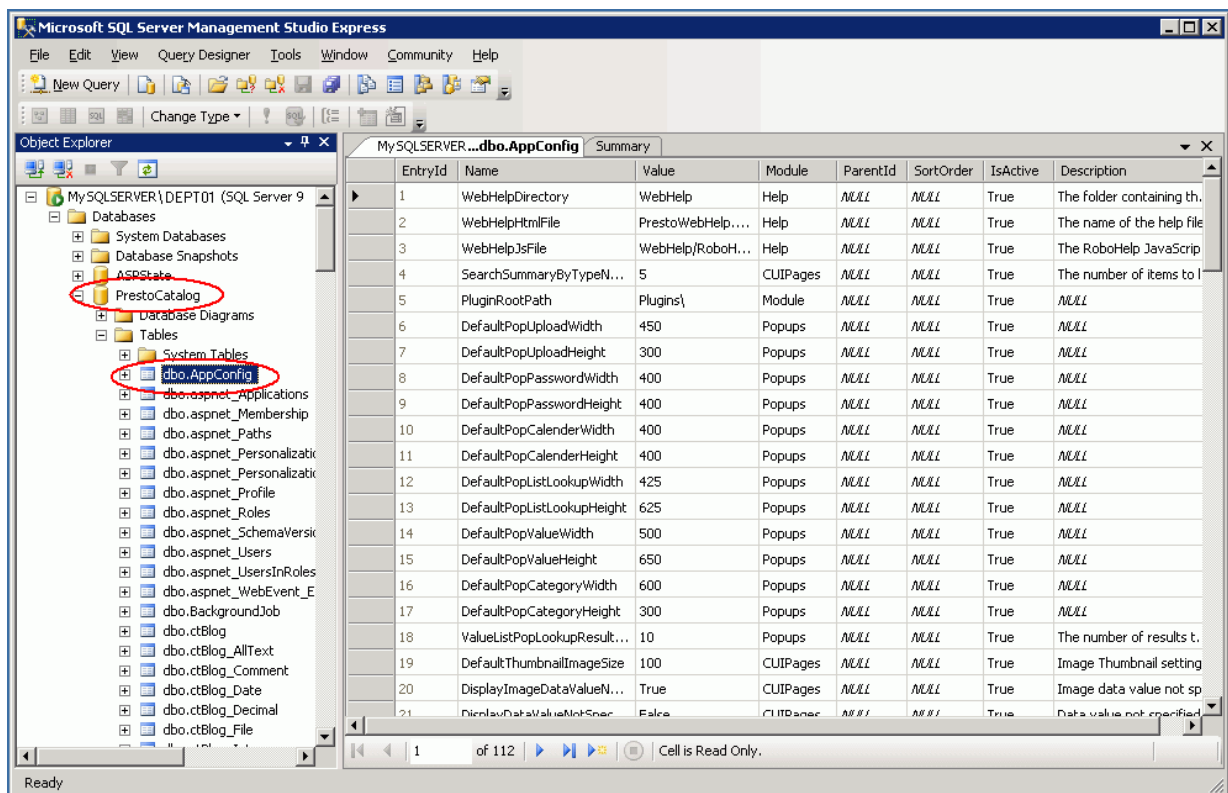
4. At a Command prompt, enter IISRESET to restart IIS.

   Note: IISRESET is recommended because it also restarts the World Wide Web Publishing Service, Simple Mail Transfer Protocol (SMTP), and HTTP SSL. If you choose to restart the IIS Admin service from Windows Services, you must manually restart the other stopped services as well.

Glossary of Application Settings

This Glossary defines all of the Application Settings available by choosing **Configure>Administrator Tools> Application Settings**.

Warning! Changing these settings can affect the stability, security, and performance of this application. Only advanced users should adjust these settings. Click 'Cancel' to close the page without saving. Clicking 'Save' writes the changes to the SQL dbo.AppConfig database table for this Presto Web application.

Tip! When editing long entries, you can either edit the XML in-place or copy it into a UTF-8 Unicode editor, make your changes, then copy it back into the field.

ActionSettings

Choose **Configure>Administrator Tools>Application Settings** to adjust the **ActionSettings**:

EnableRightClickActionMenu: Turn the right-click menu on or off, for the Search Results Screen and Detail Screen. This menu contains the same settings as the Act on Content toolbar. The default is True (on). You may set it to False if you encounter display errors.

AddToCategoryActionSettings: Plug-ins defined in this section control Category actions on the Browse page.

BrowseContentActionSettings: Plug-ins defined in this section will appear on the Browse Collection page. Each <PluginSetting> determines how a particular plug-in will operate. For example: MaxNumRecordsOnAction specifies the maximum number of records that can be acted on by the plug-in.

ContentActionSettings: Plug-ins defined in this section will appear on the Search Results, Detail, and Infocart pages. Each <PluginSetting> determines how a particular plug-in will operate. The GroupName= parameter on the <Plugin...> tag specifies which drop-down menu the option appears on, if any.

> <PluginSetting id="download_pdf_action" name="Download Action" **GroupName="DownloadAs"**
> description="Download a selected content type as PDF">

If no GroupName= appears, the option appears on the menu bar instead of being added to a drop-down menu. You can only add choices to existing drop-down menus (Add To, Download As, Status, etc.).

DetailContentActionSettings: Controls Detail Content actions. Each <PluginSetting> determines how a particular plug-in will operate.

ApplicationSetting

Choose **Configure>Administrator Tools>Application Settings** and locate the **ApplicationSetting** section to adjust these settings:

RSSMode: To lock down RSS feeds, set the RSSMode to **guestonly** (which uses Guest privileges for all RSS feeds even if the user name ID in the URL is correct) or **disabled** (which turns off RSS feeds completely, for all users). By default, RSS feeds are **enabled**. This setting also affects the Inmagic Presto SharePoint Federated Search Web Part, which relies on an RSS feed.

Note: To affect an individual user, you can edit a user account and select **Reset User ID** to cause all RSS feeds already specified in feed readers to use the Guest account privileges instead of that user's privileges until a new feed URL is specified. This may be necessary if someone has provided RSS feed URLs to users who should not have them. This has no effect on the Inmagic Presto SharePoint Federated Search Web Part.

DBCommandTimeout: Timeout in seconds for a database/Linq command.

SyncLockFileTimeout: Timeout in seconds (default is 20) for a file-based synchronization lock. This setting is for Development use only and should not be changed.

SearchSummaryType: The type of summary to display in search results from a Quick Search. The only reason to change this is if the "Summary by type" view is slow for some reason and performance cannot be adjusted in some other way (e.g., by removing huge images from the view, changing the number of records per content type back to 5, etc.). Possible values are:

- **both** (default) displays both the "Summary by type" (first) and the "Quick summary" views.
- **none** displays neither view in the list.
- **bothdefaultsummary** displays both views with "Quick summary" first.
- **summaryonly** displays just the "Quick summary"
- **summarybytypeonly** displays just the "Summary by type" view.

ValueListSize: Maximum number of characters permitted in an individual value list value. This setting controls data entry. The default is 1000 characters.

DecimalDataTypeSize: SQL-compatible size of the Decimal logical data type (how many characters are allowed before and after the decimal point).The default is 15,4. We recommend that you do not change this setting.

CriticalTextSize : The size at which Text files are considered unlimited in size, and represented by NVarChar(max). Default is 300.

MaxTermIndexSize: Maximum size of an entry in the Term index. Longer entries are truncated using this value. Default is 250 characters.

CategoryItemDelimiter: The character used to delimit categories in fully-qualified category names. Default is a pipe character (|).

TaxonomyAddCategoryToTop: Add new categories to the top or bottom of the parent category, when using "Add subcategory" when browsing a collection. Default is False (add to bottom).

TaxonomyAddContentToTop: Add new content to the top or bottom of the list when using "Add to this Category." Default is False (add to bottom).

ValidImageExtensions: Comma-separated list of file extensions to indicate which files will be treated as image files. Do not include periods or spaces. Default is bmp,gif,jpeg,jpg,png.

AdditionalInvalidSpecialCharacters: Special characters that are invalid in object names (such as Content Types, Value Lists, and special Menu items), in addition to the internally-defined characters ([]~|<> -). Define using RegEx escape characters if appropriate. Default is blank (no additional characters defined).

MaxScheduledAlertEvents: Maximum number of results (Event records) to send to user as result of executing a Scheduled Alert. Default is 50. Example: If you set up an alert to notify you weekly of all the records added in the previous week, the e-mail will only include the most recent 50.

MaxRSSEventAgeInHours: Maximum age in hours of Event instances which will be delivered to an RSS feed. Use 0 to indicate no limit. The default is 48. Example: Say a "Content added" RSS feed shows 10 items (the default). If 2 records were added yesterday, and no others have been added for months, instead of showing the 2 recent ones and 8 really old ones, the feed will show only the 2 within the 48 hour time period.

DefaultObjectNamingOption: Default object naming option for ContentType export. Values are NamesAndIds, NamesOnly, IdsOnly. Default is NamesAndIds. For more information, see the *Import and Export Configuration Guide*.

ContentTypeIconPath: The path for Content Type icons (when you select **Configure>Content Types**, create or edit a Content Type, then click the Replace link next to the Icon Image on the Overview tab). The default value is images/icons/types/, which points to <install-path>\WebApps\Mercury\images\icons\types.

MaxRSSItems: Maximum number of posts to return in an RSS feed. The default value is 10. This setting is used for all RSS feeds except Global Alerts and Blogs. The number of posts in Global Alerts is specified in the individual alert; blog feeds include all posts in the blog.

AppContext : The *Presto* instance name; corresponds to the virtual directory name in IIS. Example: Presto.

ServiceHost: The name of the IIS server. Example: localhost.

LANStartIP: The start IP address of the network IP address range. If SSO is enabled, any user whose IP address falls within the LANStartIP and LANEndIP range is considered an intranet user and will not see the *Presto* Login screen. Unless you have reason to restrict the address to a specific IP address range, leave the default setting (0.0.0.0). See *Step 5. Enable AD Single Sign On (ADSSO) [optional]*, in *Enabling Active Directory and SSO*.

LANEndIP: The end IP address of the LAN IP address range, if SSO is enabled. Unless you have reason to restrict the address to a specific IP address range, leave the default setting (255.255.255.255). See *Step 5. Enable AD Single Sign On (ADSSO) [optional]*, in *Enabling Active Directory and SSO*.

JQueryUIThemePath: Specifies the partial path under the \WebApps\Mercury\styles\jquery_themes\ folder for the theme/styles (.css file) to use for jquery objects (for example, the Application Settings and Edit Role pages).

Archive

Choose **Configure>Administrator Tools>Application Settings** to adjust the **Archive** settings:

ArchiveJobSettings: Determines how archiving will occur. See *Archiving Configuration*.

ArchiveJobTime: Archive UTC Time HH:MM:SS. Example: Set this time to 07:30:00 if your archive time is 2:30 AM EST time. See *Archiving Configuration*.

Authentication

Choose **Configure>Administrator Tools>Application Settings** to adjust the **Authentication** settings:

ADSSOEnabled: Use Single Sign On with Active Directory. **False** (default) means users must use the *Presto* Login screen. **True** means use Single Sign On with Active Directory. Intranet users who are logged into the network directly or through a VPN will not see the *Presto* Login screen. Instead, their Windows credentials will be used for authentication. If *Presto* is installed on an external public Web server that can be accessed via the Internet and is also part of the company's domain, then domain users will go directly to the Home page. Non-domain users will see the *Presto* Login screen. If a *Presto* user profile does not exist, *Presto* will create an account for the user with the same privileges as the Guest account and log in the user. See *Enabling Active Directory and SSO*.

ActiveDirectoryBlockSize: Block size for Active Directory user account imports. Default is 1000. See *Importing Active Directory Accounts*.

OPACEnabled: Bypass the *Presto* Login screen and grant rights according to the built-in Guest user account. Default is False. See *OPAC (Guest) Logins*.

ExternalSSOEnabled: True means use an external SSO security token for authentication, so users sign into an external/corporate site (for authentication) and navigate between that site and *Presto* seamlessly using Single Sign-On. False (the default) means users must log in to *Presto*. See *External SSO*.

EnableNativeLogin: Used in conjunction with **ExternalSSOEnabled** ONLY. True means native login is also allowed. False (the default) means users will *not* be allowed to log in through the *Presto* login page (they must be authenticated through the external site). If **EnableNativeLogin** is False, there is an override to get to the login page, so administrators can get into *Presto*: Append **?sso_or=true** to the end of the URL. For an example of how to use EnableNativeLogin, see See *External SSO*.

SSOFailedRedirectURL: Used in conjunction with **ExternalSSOEnabled** ONLY. The URL where users will be redirected if External SSO fails. If blank (the default), users will be redirected to the *Presto* login page. See *External SSO*.

SSOLogoutRedirectURL: Used in conjunction with **ExternalSSOEnabled** ONLY. The URL where users will be redirected if they click the Logout button. If blank (the default), users will be redirected to the *Presto* login page. See *External SSO*.

SSOTokenExpiration: Used in conjunction with **ExternalSSOEnabled** ONLY. The expiration time for the SSO security token. The default is 2 Mins. See *External SSO*.

SSOCreatedUserPassword: Used in conjunction with **ExternalSSOEnabled** ONLY. The default password that is set for all users created with External SSO. The default is blank. See *External SSO*.

EnableRememberMeLogin: Allows users to bypass the Login page when returning to the site. True means the next time the user goes to *Presto* they will not have to log in. False (the default) means users need to log in every time. See *Remember Me (Login page)*.

RememberMeExpiration: Used in conjunction with **EnableRememberMeLogin** ONLY. The expiration for the Remember Me security token. The default is 3 Days Rolling. See *Remember Me (Login page)*.

Beta

Choose **Configure>Administrator Tools>Application Settings** to adjust the **Beta** settings, if any. These settings may or may not be included in future versions.

BrowseCollection

Choose **Configure>Administrator Tools>Application Settings** to adjust the **BrowseCollection** settings:

DisplayCategoryContentCount: Display the record count for each Category on the Browse Collection page. The count is shown in parentheses after the category name. Default is True.

DisplayCumulativeCategoryContentCount: Display the cumulative record count for each Category and all subcategories on the Browse Collection page. For example, if the top-level folder is empty but it contains two subfolders, each with (3) items, then the top-level folder will display a count of (6). Default is True.

Configure

Choose **Configure>Administrator Tools>Application Settings** to adjust the **Configure** settings:

DefaultNumberOfFragments: How many snippets appear in each record in Search Results. Default is 5. Note that fewer snippet fragments may appear if Number*Size (the number of fragments times the highlight fragment size) is greater than the Maximum Visible Characters specified in the current view.

ShowCreateNewUserLink: True (default): Displays the Create Account link on the Login page, so users can create their own accounts and log in to *Presto*. False: Hides the Create Account link on the Login page.

ShowForgotPasswordLink: Show or hide the Reset Password link on the Login page. See *Passwords and User Accounts*.

DefaultHighlightFragmentSize: How many characters appear in each snippet of search results. Default is 80 characters.

HighlightTags: Determines if and how highlighting appears for search results. To turn off highlighting, retain the comma but replace the text before and after the comma with a space (so the value is simply space, comma, space). When specifying highlighting, Pre and Post tags must be separated by a comma. This example shows HTML source from a Search Results or Detail Screen. The bolded strings are controlled by what's specified for HighlightTags:

> Search results can be **** highlighted ****.

To change the formatting of highlighted text (e.g., green instead of yellow), create a CustomBrandingStyles.CSS file and define the "searchmatch" style there.

Default value (sets highlighting on and uses the "searchmatch" style defined in BrandingStyles.CSS, which is bold yellow by default):

> ,

For more information, see *Cascading Style Sheets* on page 57.

FileDataTypeFileType – The values in this field are used to populate the Available File Types list that appears when creating or editing a field when defining a Content Type. (Choose **Configure>Content Types**, click an existing content type or create a new one, then create or edit a field for the "File" Data Type.) Default values are shown below:

> doc^Word Documents (.doc)|docm^Word Documents (.docm)|docx^Word Documents (.docx)|dotm^Word Documents (.dotm)|dotx^Word Documents (.dotx)|xls^Excel Documents (.xls)|xlam^Excel Documents (.xlam)|xlsb^Excel Documents (.xlsb)|xlsm^Excel Documents (.xlsm)|xlsx^Excel Documents (.xlsx)|xltm^Excel Documents (.xltm)|xltx^Excel Documents (.xltx)|pdf^Adobe Documents (.pdf)|ppt^PowerPoint Documents (.ppt)|potm^PowerPoint Documents (.potm)|potx^PowerPoint Documents (.potx)|ppam^PowerPoint Documents (.ppam)|ppsm^PowerPoint Documents (.ppsm)|ppsx^PowerPoint Documents (.ppsx)|pptm^PowerPoint Documents (.pptm)|pptx^PowerPoint Documents (.pptx)|rtf^Rich Text Documents (.rtf)|txt^Text Documents (.txt)|bmp^BMP Images (.bmp)|jpeg^JPEG Images (.jpeg)|jpg^JPG Images (.jpg)|gif^GIF Images (.gif)|png^PNG Images (.png).

CUIPages

Choose **Configure>Administrator Tools>Application Settings** to adjust the page display and layout settings for the Customizable User Interface (CUI).

SearchSummaryByTypeNumResults: The number of items to list on the Search Results page for each type of content when using the Summary by type view. Default is 5.

DefaultThumbnailImageSize: The thumbnail size of records that contain images. Specify any reasonable value, in pixels. Default is 100 (one inch).

DisplayImageDataValueNotSpecified: Determines whether the image "not-available-image.gif" is displayed when image data is not provided. This only applies to controls associated with File fields containing images, when "Suppress Control If Associated Field Is Empty" is unchecked for the control in the screen/view definition. Default is True: Display image (`<install-dir>\images\ui\ not-available-image.gif`). False means do not display image.

DisplayDataValueNotSpecified: Determines if "Value not specified" should be rendered for properties of records that do not contain data when "Suppress Control If Associated Field Is Empty" is not checked for the control. This is similar to DisplayImageDataValueNotSpecified, but this setting is used when rendering non-image data. True: Display "Value not specified" text. Default is False: Do not display this text.

PrintFriendlySummaryMaxCharacters: Maximum characters a word can be before entire value is truncated (per item) on Print Friendly Screen Summary Section. Default is 50.

DefaultPageSize: Default number of results to show for "non-pageable" views. Default is 10.

SelectablePageSizesForCitationView: Comma-delimited list of the selectable page sizes for a citation "pageable" view. The first value is used by default when displaying results, so the items may be listed in non-numerical order. Default is 10,20,30,50,150.

SelectablePageSizesForGalleryView: Comma-delimited list of the selectable page sizes for a gallery "pageable" view. The first value is used by default when displaying results. Default is 4,8,16,32,64,128.

SelectablePageSizesForGridView: Comma-delimited list of the selectable page sizes for a grid "pageable" view. The first value is used by default when displaying results. Default is 25,50,75,100,125,150.

MaxImmediateActionItems: Maximum number of items (records) for certain actions to run in foreground mode (while client waits). If exceeded, action is executed as background action. Default is 20.

DefaultRecordSeparatorText: HTML character(s) used to separate records in all views (except Grid View). For example: `<add key="DefaultRecordSeparatorText" value="------" />`. Default is empty (there is an implied <br> between records, and this separator does not need to be specified here).

HTMLTextBoxToolbarLayout: Toolbar layout for the HTMLTextbox control (FreeTextBox). Default is: ParagraphMenu,FontFacesMenu,FontSizesMenu,FontForeColorsMenu,FontForeColorPicker,FontBackColorsMenu,FontBackColorPicker|RemoveFormat;Bold,Italic,Underline,Strikethrough|JustifyLeft,JustifyRight,JustifyCenter,JustifyFull;BulletedList,NumberedList,Indent,Outdent|CreateLink,Unlink|InsertImage,InsertImageFromGallery.

DetailBodySectionImageWidth: The image width (in pixels) of the detail body section image when **PerformIframeDetailSectionSinglePropertyLogic** is enabled.

PerformIframeDetailSectionSinglePropertyLogic: Determines if Detail screens with a single File field specified in the Body Section should display the file contents or a link to the file. True (default): Display the file contents. False: Display a link to the file.

TransformFilesDirectory: Directory for CUI transform files. Default is WebApps\Mercury\system\transforms.

MaxGridLinkItems: Maximum number of records to display in Link Grid or Link Edit controls. Default is 100. It is recommended that a reasonable default value is set (do not leave blank or enter 0).

CalendarYearRange: Range of years to display in calendar controls. Leave empty for direct text entry. Ex: 1980:2020, -10:+10 (relative to today), c-5:c+5 (relative to selected date). The default value is c-10:c+10.

DefaultViews

Choose **Configure>Administrator Tools>Application Settings** to adjust the **DefaultViews** settings:

UserLoginViews: Format login and logout events that appear in alert messages and RSS feeds.

SystemJobViews: Format system job events that appear in alert messages and RSS feeds.

FileUpload

Choose **Configure>Administrator Tools>Application Settings** to adjust the **FileUpload** settings to limit the size of files that may be uploaded from an Add/Edit screen or from Import Content and Export Content (in Administrator Tools).

MaxRequestLength: Total file size that can be uploaded, specified in bytes. Default is 4194304 (equivalent to 4MB).

MaxRequestLengthLabel: Text that indicates the maximum file size (should match the MaxRequestLength specified above). This text will be inserted into the error message that appears if users attempt to upload a file that exceeds the MaxRequestLength. The default is 4 mb, which matches the default MaxRequestLength.

Help

You should not need to adjust the **Help** settings. For related information, see *Customizing Help Topics.*

WebHelpDirectory: The folder containing the Presto online Help files (WebHelp).

WebHelpHtmlFile: The compiled *Presto* help file (PrestoWebHelp.htm).

WebHelpJsFile: A required JavaScript file (WebHelp/RoboHelp_CSH.js).

HomePage

Choose **Configure>Administrator Tools>Application Settings** to adjust the **HomePage** settings.

DisplayCriticalErrorsOnHomePage: True shows critical errors on the Home page. Default is False.

HomePage: Determines which Home page is loaded by default into the Home page designer in *Presto*. Possible values are **~/home/Default.aspx or ~/home/Default_v2.aspx**. The default page (Default.aspx) uses a gray and blue color scheme. The alternate default page (Default_v2.aspx) uses a white and green color scheme. The corresponding .CSS and .SKIN files are located in subfolders under `~/App_Themes`. For more information, see the *Presto Default Home Page Configuration Guide*.

HomePageImagePath: The folder where images are stored for use on the Home page (**~/images/homepage/**).

MimeTypeMapping

Choose **Configure>Administrator Tools>Application Settings** to adjust the MimeTypeMapping. This section lists the MIME type for each extension, to determine which application on the client machine (e.g., Adobe Reader, Microsoft Word 2007) should be used to display File fields in the Body Section of the Detail Screen. You cannot directly edit this field. To add an application that is not listed, or for more information, search the Presto Knowledgebase for MIME.

PagePluginSettings

Choose **Configure>Administrator Tools>Application Settings** to adjust the **PagePluginSettings** settings:

PagePluginSettings: Only Inmagic Professional Services should adjust this setting.

PluginRootPath: Only Inmagic Professional Services should adjust this setting. The default is Plugins\.

Popups

Choose **Configure>Administrator Tools>Application Settings** to modify the **Popups** settings:

DefaultPopUploadWidth: The width in pixels of the pop-up for the file edit control. Default is 450.

DefaultPopUploadHeight: The height in pixels of the pop-up for the file edit control. Default is 300.

DefaultPopPasswordWidth: The width in pixels of the pop-up for the password edit control. Default is 400.

DefaultPopPasswordHeight: The height in pixels of the pop-up for the password edit control. Default is 400.

DefaultPopListLookupWidth: The width in pixels of the pop-up for controls that use a lookup. Default is 425.

DefaultPopListLookupHeight: The height in pixels of the pop-up for controls that use a lookup. Default is 625.

DefaultPopValueWidth: The width in pixels of the pop-up for controls in administrative pages. Default is 500.

DefaultPopValueHeight: The height in pixels of the pop-up for controls in administrative pages. Default is 650.

DefaultPopCategoryWidth: The width in pixels of the pop-up controls for adding or editing a category on the Browse Collection page. Default is 600.

DefaultPopCategoryHeight: The height in pixels of the pop-up controls for adding or editing a category on the Browse Collection page. Default is 300.

ValueListPopLookupResultsToShowPerPage: The number of results to show per page in Value List lookup pop-up. Default is 10 (lines).

DefaultPopRelationshipListLookupHeight: The default height in pixels of the Relationship popup window. Default is 625.

DefaultPopRelationshipListLookupWidth: The default width in pixels of the Relationship popup window. Default is 625.

PrestoApplication

Choose **Configure>Administrator Tools>Application Settings** to adjust the **PrestoApplication** settings:

ApplicationName - The *Presto* application name as it appears in the Presto UI. Change as needed to meet customer requirement. Default is Inmagic® Presto.

SerialNumber - The serial number for this installed *Presto* instance. Established by the Installer, and should not be altered except by the Installer or Service Pack Installer.

BinaryFileExportPath - The default directory in which binary files are written by the Presto Export facility. If specified, this must be the full path name of a directory that is permissioned for read and write access by the Presto identity. Default is blank (no path). Typically, the binary export file path, if any, is specified in the Export Request XML file. For more information, see the *Import and Export Configuration Guide*.

PrestoServices

Choose **Configure>Administrator Tools>Application Settings** to adjust the **PrestoServices** settings:

PrestoServices: The Presto Services Worker configuration for each client worker. One worker is provided for each long-running operation that is not implemented directly in the *Presto* Web application server. For instructions on how to configure the listeners and queue processors, see *Inmagic Presto Services*.

DocumentCaptureSettings: Settings for the Presto Services Document Capture worker. For information, see the *Document Capture Configuration Guide*.

FileCaptureSettings: Settings for the Presto Services File Capture (drag & drop) worker. The default is shown below. For more information, see the *Presto Drag & Drop Files Configuration Guide*.

```
<FileCaptureSettings>
 <xmlSerializerSection type="Inmagic.Mercury.Components.FileCapture.FileCaptureSettings, Inmagic.Mercury.Components,
Version=1.1.2714.30958, Culture=neutral, PublicKeyToken=null">
  <FileCaptureSettings xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
   <FileCaptureFolders>
   </FileCaptureFolders>
  </FileCaptureSettings>
 </xmlSerializerSection>
</FileCaptureSettings>
```

DeferredIndexingSettings: Settings for the Presto Deferred Binary Content Import worker. For more information, see the *Import and Export Configuration Guide*. Also read about the DeferredIndexingQueueMessageProcessor in *Service Types (Workers)* on page 68.

PrestoSoftwareVersion

Choose **Configure>Administrator Tools>Application Settings** to view the **PrestoSoftwareVersion** settings:

MajorVersion: Presto major release number. Established by the *Presto* software, and should not be altered.

MinorVersion: Presto minor release number. Established by the *Presto* software, and should not be altered.

Revision: Presto revision or Service Pack number. Established by the *Presto* software and/or Service Pack Installer, and should not be altered.

BuildNumber: Presto build number. Established by the *Presto* software build process, and should not be altered.

Reporting

Choose **Configure>Administrator Tools>Application Settings** to adjust the **Reporting** settings.

*As of v3.5, SQL reports have been replaced by Web reports, which are explained in the help and the *Basic Administrator's Guide*. Settings marked with an asterisk (*) apply only to SQL Reports.

EnableNavigationInfoRecording: Enable or disable the recording of user navigation information (a list of *Presto* users and which pages each user has visited).The default is **False**, so information is not saved and the User Navigation Information report is empty. If set to **True**, the information is saved in the *Presto* database and reported in the User Navigation Information report. To periodically clear the information from the *Presto* database, choose **Configure>Administrator Tools>Database Cleanup>Remove All Navigation Information**.

MaxReportRecordsWithoutPrompt: The maximum number of records that can be viewed in a report before the user is prompted to Continue, Cancel, or Download. If the report contains fewer records, the report will be displayed automatically when selected. If you specify blank or 0, the prompt will never appear. This is not recommended, but may be acceptable if all of the reports include a limited number of fields, or are all filtered, and users never encounter timeouts or other issues.

ReportHost*: The name of the Microsoft SQL Server Reporting Services Report Server that *Presto* is using. Example: localhost/reportserver

ReportServerFolder*: The name of the folder being used for SQL reports, on the specified ReportHost. Example: /PrestoReports

ReportServerProtocol*: The protocol to access the SQL Report Server. Possible values are **http:** or **https:**. Default is http:

ReportNames*: SQL Reports only (not used in v3.5; retained for upgrade purposes).

SqlReportNames*: This setting appears only if you upgraded from 3.3 or 3.2. It includes only *custom* SQL reports (if any), separated by pipes (for example: New_Report|AnotherReport). It does not include out-of-the-box reports. The listed reports can be seen by choosing **Reports>View SQL Reports**.

Search

Choose **Configure>Administrator Tools>Application Settings** to adjust **Search** settings:

MeaninglessLeadingWords: Specifies Leading Articles, which are words that come at the beginning of a term and are meant to be ignored when sorting or searching. For example, *The Sun Also Rises* will be sorted under *Sun*, not *The*. Default values are a,an,the.

NonStemmedWords: Specifies words that should not be stemmed when searching, when stemming is on. The word "us" is in the list by default because "us" stems to "we" and "we" is a noise word (by default in SQL). If "us" were not in the list, it is possible that a search for "us" would find no records. Specify any word or words, separated by commas. Default value is *us*.

SearchAdapterSettings: External Search Adapters Settings (Federated Search). Defines search providers, such as Google, Bing, and Wikipedia, to appear as tabs on the Search Results page after a Quick Search. By default, no search adapters are configured. For information, see *Searching Other Products from Presto*.

RefreshFullTextNoiseWordJobTime: Determines the time at which the full-text noise word index will be refreshed, for the purposes of searching. (This is not related to repopulating the full-text catalogs, which must be done in SQL after updating noise/stop words.) Use the format HHMMSS. For example, 020000 (default) is 2:00 a.m. 233000 is 11:30 p.m.

SqlOptimizeProcessorJobTime: The time at which the SqlFullTextOptimizeProcessor in Presto Services runs, to optimize the Sql Full Text Catalog to improve relevance ranking behavior.

SearchDefaultStemming: Indicates whether stemming (^) is on or off by default. True (default) means stemming is on. False means stemming is off. Stemming behavior is determined by the Primary Stemming Language designated during *Presto* installation. For example, if you are using French and English, but French is the primary language, then stemming is performed only with French. In addition to the language stemmers supplied with SQL, new ones can be developed and added. See Microsoft's SQL Server documentation for details: http://msdn.microsoft.com/en-us/library/ms142509.aspx.

SystemMenuConfiguration

Choose **Configure>Administrator Tools>Application Settings** to view the system menu configuration settings. **Note**: The **systemMenuConfiguration** should not be altered.

SystemMetadataConfiguration

Choose **Configure>Administrator Tools>Application Settings** to view the system metadata configuration settings. **Note**: The **systemMetadataConfiguration** should not be altered.

Locale

Choose **Configure>Administrator Tools>Application Settings** to adjust the **Locale**. The locale identifies regional and language settings to determine the appearance of numbers, dates, time, language, and Home page for the logged-in user. By default, only one locale, English – United States (en-US), will appear.

Administrators for non-US clients should make any desired locales active (set **Active** to **True**) and assign each locale a **Sort Order** to determine its order in lists, using **0** to indicate the default. A locale can then be selected from lists when updating a profile or configuring user accounts, Home pages, content types, collections, and menus.

The table below shows the default values for Active and Sort Order. *Presto* only supports the listed Locales:

Locale	Value	Active	Sort Order
en-US	English – United States	True (will appear)	0 (default)
fr-CA	Français – Canada	False (will not appear)	1
en –GB	English – Great Britain	False	2
en-CA	English – Canada	False	3
en-AU	English – Australia	False	4
en-NZ	English – New Zealand	False	5
es-ES	Español – España	False	6

Localization and Customization

Setting Locales

The locale identifies regional and language settings to determine the appearance of numbers, dates, time, language, and Home page for the logged-in user. To specify which locale(s) are available in *Presto*, choose **Configure>Administrator Tools>Application Settings** and adjust the **Locale** settings, as explained in the *Glossary of Application Settings*.

Customizing and Localizing Text

All text and many messages that appear in the *Presto* UI can be customized and localized. For example, you can change the explanatory text that appears on various pages. This involves modifying the **CustomResources.resx** file with value(s) that will override the supplied values, then recompiling, as described in the instructions below.

All text that appears in the Presto UI, and many messages that can be viewed through the UI, are global resources which are:

- Compiled into the Components DLL (Inmagic.Mercury.Components.dll) for the *Presto* default locale (U.S. English, en-US).

- Compiled into satellite assemblies for each locale for which a translation or locale-specific variant exists, such as Canadian French, fr-CA.

Presto's organization of these localized resources conforms to .NET 2.0 standards. For more information, see the online MSDN topics "Globalizing and Localizing Applications" and "Hierarchical Organization of Resources for Localization."

Occasionally, *Presto* customers may want to change some of these text resources. For example, they might want to change the empty Category message "*There are no items in this category*" to "*See folders in left frame.*" Custom resource files, whose text resources override the compiled-in values, allow you to make changes such as these.

Creating and using custom resource files does not change the compiled resources or the *Presto* application configuration. They simply override the compiled resources.

To create custom resources, follow the steps below.

Step 1. Identify the required changes.

Work with the customer to identify which text strings they would like altered. Try to limit changes to the minimum possible. Then contact Inmagic Development, who will inspect an internally available Global Resource file to identify the resources that need to be changed.

Example: Say you want to change the following string in the Browse UI: "*There are no items in this category.*" Inmagic Development will search a Global Resource file to identify the resource as shown below:

```
<data name="GeneralLabel.NoItemsInCategory" xml:space="preserve">
  <value>There are no items in this category.</value>
</data>
```

Step 2. Prepare the .resx file.

In this step, you will prepare XML file(s) containing data for the modified text:

- These files are standard .NET .resx files (XML files using key-value pairs to specify text resources).

- You will create a file for the base language, and possibly one or more locale-specific files, depending on the locales supported by the *Presto* application in question.

Presto ships with a skeleton resource file to be used for customizing resources:

```
<install-dir>\WebApps\Mercury\system\customresources\CustomResources.resx
```

Using an editor that supports Unicode UTF-8, such as Windows Notepad, add the resource entry you identified in Step 1, and modify it to contain the new string ("*See folders in left frame.*"). The entry should come at the end of the file, after the last closing </resheader> tag and just before </root>. For example:

```
...
  </resheader>
  <data name="GeneralLabel.NoItemsInCategory" xml:space="preserve">
    <value>See folders in left frame.</value>
  </data>
</root>
```

Step 3. Compile the .resx file.

Now you will use the .NET SDK **resgen.exe**. tool to compile each .resx file created in the previous step. The result will be a file called **CustomResource.resources**, which contains the new value.

Note: Resgen.exe is available in the .NET Framework Developer's SDK. This can be downloaded for free, or you can acquire the program by itself from various resources within Inmagic. It must be the version for ASP.NET 2.0.

1. Open a command prompt window.

2. Change directory to `<install-dir>\WebApps\Mercury\system\customresources`, then issue the following command to compile CustomResources.resx:

    ```
    resgen CustomResources.resx
    ```

    The result will be the compiled resource file **CustomResource.resources**.

3. If you created localized versions of CustomResources.resx, compile them in the same way. For example:

    ```
    resgen CustomResources.fr-ca.resx
    ```

4. At a Command prompt, enter IISRESET to restart IIS.

    Note: IISRESET is recommended because it also restarts the World Wide Web Publishing Service, Simple Mail Transfer Protocol (SMTP), and HTTP SSL. If you choose to restart the IIS Admin service from Windows Services, you must manually restart the other stopped services as well.

Step 4. Test the new .resources file.

Run the *Presto* application in which you performed the above actions and confirm that the new resource string appears as expected.

Step 5. Deploy the custom resource file.

In this final step, you will move the files into the `.\system\customresources directory` for the customer application. The resources will be used automatically by *Presto*. If the *Presto* application in which you have been working is not the customer's installed application, deploy your customized resources:

1. At the customer site, copy the new **CustomResources.resources** file to the following location:

    ```
    <install-dir>\WebApps\Mercury\system\customresources
    ```

    Localized versions, called **CustomResources.<locale>.resources**, must also be in this folder. For example: **CustomResources.fr-ca.resx**.

    Important! In order for a localized custom resource file (e.g., **Customresources.fr-CA.resources**) to be recognized, the standard custom resource file (**CustomResources.resources**) must be present. It need not contain any customized messages; for example, you could compile the provided Customresources.resx file.

2. At a Command prompt, enter IISRESET to restart IIS.

3. Confirm that the new resource string appears as expected.

You have now completed the resource customization process.

Localization of Alert E-mail Templates

If you place localized versions of the three alert e-mail templates in a `<locale>` subfolder of the template directory, they will be used instead of the installed templates. For example, *Presto* would first look here:

```
..\WebApps\Mercury\system\template\fr-ca
```

before defaulting to the template folder:

```
..\WebApps\Mercury\system\template\
```

Session State

The Session State in the *Presto* **Web.config** is most commonly set to either `InProc` (default, stores session data on the Server in memory) or `SqlServer` (stores session data in a SQL Server database, where it can be shared by multiple Web servers). For more information about Session-State modes, please see http://msdn.microsoft.com/en-us/library/ms178586.aspx.

Changing the Session State

The installed *Presto* **Web.config** file defaults to persisting session state in-process, as shown below:

```
<sessionState mode="InProc" cookieless="false" timeout="1080" />
```

To change from InProc to SqlServer, follow the instructions below.

Step 1. Set up SQL Server for Persistent Session State

Note: During *Presto* installation, this step is performed by the installer for the ASPState database. If for some reason you need to configure it manually, follow the instructions below. Otherwise, go to step 2.

The SQL Server that you choose to persist Session State for needs to be set up correctly. This is accomplished by running the ASP.NET SQL Server Registration Tool (Aspnet_regsql.exe). The tool is described in detail at http://msdn.microsoft.com/en-us/library/ms229862(VS.80).aspx.

For Presto 3.0 or later, use the Framework 2.0 version of the Aspnet_regsql.exe tool, located at:

```
<system drive>\WINNT\Microsoft.NET\Framework\ is v2.0.50727\aspnet_regsql.exe
```

The following instructions set up databases using a *default* setup, which creates the ASPState database as a container for required stored procedures and uses the system tempdb database to store session data.

It is recommended that the tool be run from the command line, and that wizard mode not be used.

1. Open a command window on the Web application server and connect to the folder specified above.

2. Run aspnet_regsql.exe with the following parameters:

    ```
    aspnet_regsql -S ASPState -U <sql_login>  -P <password> -ssadd -sstype t
    ```

    Note that the sql_login must have database create permission. This login will be db owner of the ASPState database that is created.

Step 2. Modify the Presto Web.config File

To change the session state from InProc to SQLServer:

1. Open the *Presto* **Web.config** file (<install-path>\Webapps\Mercury\Web.config).

2. Comment out the line that sets sessionState mode to "**InProc**":

    ```
    <sessionState mode="InProc" cookieless="false" timeout="1080" />
    ```

3. Uncomment the line that sets sessionState to "**SQLServer**" and edit it to specify the SQL Server instance name on which you configured the Session State database in Step 1. Be sure to specify a SQL user (login) that has:

    - Execute rights to all stored procedures in the ASPState database
    - db_datawriter permission in the system tempdb database

Timeout Adjustments

You can configure the following timeout values.

HTTP Timeout

The default HTTP timeout value is 90 seconds. Increase this value if you get errors (in Errors.log) saying simply "Request timed out", for example when uploading a large file on an Edit screen.

1. Open the **Web.config** file (<install-path>\Webapps\Mercury\Web.config).

2. Adjust the timeout value (defined in seconds) in the following line:

   <httpRuntime **executionTimeout="90"** maxRequestLength="2048000" />

SQL Timeouts

You can adjust the following SQL timeout settings.

Database Command Timeout

If you get errors indicating there are SQL timeouts, you may need to increase the **DBCommandTimeout** value. Log into *Presto* as an administrator and choose **Configure>Administrator Tools>Application Settings**, then adjust the **DBCommandTimeout** value. This sets the timeout in seconds for a database/Linq command. The default value is 600 seconds. **Note**: This is distinct from the connection timeout in the *Presto* Web.config.

Connection Timeout

The **Connect Timeout** value for the SQL server connection string in the *Presto* **Web.config** file gets or sets the length of time (in seconds) to wait for a connection to the SQL server before terminating the attempt and generating an error.

Important! This value generally does NOT need to be adjusted. Always use caution when editing the connection string, since this controls *Presto*'s communication with SQL server.

The connection string in **Web.config** is typically encrypted, in which case you need to use the ConfigUtil tool to decrypt it, in order to see or change the string. After editing the string, use the ConfigUtil tool to encrypt it again. For an example of an unencrypted connection string, and information about how to use ConfigUtil, see *SQL Server or Windows Authentication*.

Session Timeout

The session timeout is a security measure that determines the length of inactivity that will result in being disconnected from *Presto*. For example, a timeout of 20 minutes means that if you do not submit anything to *Presto* for 20 minutes, you will be logged out. You cannot eliminate the session timeout completely, but you can increase it to avoid frequent timeouts or decrease it if you have security concerns.

To adjust the session timeout

1. Open the **Web.config** file (<install-path>\Webapps\Mercury\Web.config).

2. Increase or decrease the **timeout** attribute for the sessionState tag. A timeout of 1080 minutes is shown below. If there are system resource issues (limited RAM), decreasing this setting is recommended. Also consider decreasing this setting for security reasons (to help prevent unauthorized use of a system that has been left unattended).

```
<sessionState mode="InProc" cookieless="false" timeout="1080" />
```

   Note: The sessionState tag may include other parameters, in addition to those shown above, and it may use a mode of "SQLServer". You can leave them as-is. Just change the **timeout=** value.

3. If you are using Presto v3.5, you can skip this step. If you are using a version prior to 3.5, verify that the Application Pool settings are correct for the Web server:

   a. Right-click **My Computer** and choose **Manage**.

   b. Expand **Services and Applications**, **Internet Information Services (IIS) Manager**, **Application Pools, Presto**.

   c. Right-click the Presto instance and choose **Properties**.

   d. On the **Performance** tab, uncheck **Shutdown worker processes after being idle for…**.

   e. On the **Recycling** tab: Check *only* the option **Recycle worker processes at the following times**, then use the **Add** button to specify an off-peak time (such as 3:00 a.m.). For more information about the options for recycling worker processes, see the Microsoft TechNet article "Configuring Worker Processes for Recycling".

   f. Click **OK**, then restart IIS using **IISRESET** from a command prompt.

Online Time Window for Forums

This feature applies to *Presto* v3.5 and later.

Forums are configured entirely within the *Presto* user interface, except for the userIsOnlineTimeWindow attribute, which controls the **Online** status of users displayed on the Forum topic page. The default is 20 minutes. You might want to decrease this value if *Presto* indicates that a user is still **Online** when the user actually stopped using *Presto* several minutes ago.

1. Open the **Web.config** file (<install-path>\Webapps\Mercury\Web.config).

2. Locate the <membership> section and adjust the userIsOnlineTimeWindow setting defined for the defaultProvider. The default is 20 minutes.

   If you are using Active Directory authentication, the default setting might look like this:

   <membership defaultProvider="**PrestoADProvider**" userIsOnlineTimeWindow="20">

   If you are using native authentication, the setting might look like this:

   <membership defaultProvider=" **PrestoSQLProvider**" userIsOnlineTimeWindow="20">

Every time a user performs any activity in *Presto*, the software updates the LastActivityDate column in the aspnet_Users table. If the LastActivityDate column is within the time period specified in the userIsOnlineTimeWindow, the user is considered Online (even if the user has logged out, closed the browser, or remained in *Presto* for more than 20 minutes without performing any activities).

User Login and Authentication Settings

This section discusses configuration settings related to *Presto* user accounts, passwords, and authentication for logins.

Mixed Authentication

Presto supports four types of user validation. These validators can be enabled or disabled in Application Settings. *Presto* will attempt to validate the user using these validators in the following order:

1. Active Directory (AD) authentication (ADSSOEnabled)
2. External SSO authentication (ExternalSSOEnabled)
3. OPAC mode (OPACEnabled)
4. Native authentication (EnableNativeLogin)

All of these methods are discussed in this section.

Remember Me (Login page)

A **Remember Me** option can be added to the *Presto* Login page. This allows users to log in once and have the information stored on their machine for later use. The user will automatically be logged in to the *Presto* site upon returning.

To add the Remember Me checkbox to the *Presto* Login page, choose **Configure>Administrator Tools>Application Settings** and set **EnableRememberMeLogic** to True. The default is False, meaning users need to log in every time.

To set the expiration for the Remember Me security token, use the **RememberMeExpiration** Application Setting. The default is 3 Days Rolling. This gives the user the ability to return to *Presto* at a later time and bypass the login.

Passwords and User Accounts

To create and maintain *Presto* native user accounts, choose **Configure>Users**, as described in the help. To make additional adjustments, follow the instructions below. Note that passwords cannot be changed for Active Directory users.

Reset Password (Login page)

To allow users to reset their own passwords, include the Reset Password link on the Login page:

1. Choose **Configure>Administrator Tools>Application Settings**.
2. In the **Configure** section, set **ShowForgotPasswordLink** to True (the default) to show the Reset Password link on the Login page. Set it to False to hide the link.

Users can then reset their own passwords from the *Presto* Login page.

Create Account (Login page)

You can show or hide the <u>Create Account</u> link on the Login page. Including the link allows users to create their own user accounts. User-created accounts have the same Role as the built-in Guest user account. Therefore, *Presto* System Administrators should ensure that the Guest account is configured appropriately (with restrictive rights). Administrators can modify user-created accounts to provide additional roles or change the primary role (which determines which home page the user sees).

To show or hide the Create Account link:

1. Choose **Configure>Administrator Tools>Application Settings**.

2. In the **Configure** section, set **ShowCreateNewUserLink** to True (the default) to display the <u>Create Account</u> link on the Login page, so users can create their own accounts and log in to *Presto*. Set it to False to hide the link.

Failed Password Attempts

To determine how many failed password attempts are allowed before a user is locked out of *Presto*, administrators can adjust the following settings for the "**PrestoSQLProvider**" in the Membership providers section of the *Presto* **Web.config** file. (Note that the "**PrestoSQLProvider**" is used for native users in *Presto*.)

- **maxInvalidPasswordAttempts**: The number of failed password attempts before the user account is locked out. A typical default is 5.

- **passwordAttemptWindow**: Indicates the time window, in minutes, during which failed password attempts and failed password answer attempts are tracked.

If Native users are locked out of *Presto*

If a *Presto* user types an invalid password a certain number of times (**maxInvalidPasswordAttempts**) within the interval specified (**passwordAttemptWindow**), that user will be locked out of *Presto*. If the **Reset Password** feature is implemented on the Login page, a user can unlock his or her own account by specifying the correct Security Question and Answer. Otherwise, to unlock the user, choose **Configure>Users, Set Password** and change the user's password in *Presto*.

If Active Directory users are locked out of *Presto*

If you are using Active Directory (AD) and a user becomes locked out of *Presto*, a SQL administrator will need to reset the account in the SQL database. Do this by editing the dbo.aspnet_Membership table in the SQL database to set the IsLockedOut column to False and the FailedPasswordAttemptCount to 0. Note that the only human-readable column in this table to identify the user is e-mail address, so you might have to go into the dbo.aspnet_Users table to determine the GUID to use in the aspnet_Membership table.

Note: If you are using AD Single Sign On (SSO) and you are running into an issue where users are prompted repeatedly for a login and are then denied access when the maximum number of invalid password attempts is reached, you probably need to <u>enable NTLM Authentication in IIS</u>, as explained on page 33.

OPAC (Guest) Logins

The Online Public Access Catalog (OPAC) feature enables users to access *Presto* without providing login information. OPAC authenticates against the *Presto* SQL database using the *Presto* Guest account.

If you enable OPAC as well as Active Directory and SSO (which uses LDAP), the user will be authenticated against LDAP first, and if the user is not found, OPAC will be used.

To enable or disable OPAC:

1. Log in to *Presto* as an administrator.
2. Choose **Configure>Administrator Tools>Application Settings**.
3. Set **OPACEnabled** to True or False:
   - **True** means users will be authenticated without logging in to *Presto*. They will bypass the *Presto* Login screen and will be granted rights according to the built-in Guest user account. Users who prefer to log in using their own account can click the Log In button on the *Presto* menu bar to display the normal *Presto* Login screen.
   - **False** means users must log in to *Presto*.
4. Save the Application Settings.
5. Ensure that the Guest user in *Presto* is assigned to a role with appropriately locked-down permissions (**Configure>Users**).

Configuring Multiple OPAC (Guest) Logins

Presto accepts login credentials in the URL using the parameter "uc". The username and password must be delimited with a pipe. The `username|password` string can optionally be base64 encoded, by copying it into one of the many base64 encoders available online, such as http://ostermiller.org/calc/encode.html. Examples:

```
http://www.your-presto-url.com/Presto?uc=MyUserName|MyPassword
http://www.your-presto-url.com/Presto?uc=bXlVc2VyTmFtZXxteVBhc3N3b3Jk
```

To enable multiple OPAC logins, a *Presto* System Administrator must do the following:

1. Create a user account in *Presto* for each guest user.
2. Assign the new user accounts to Roles with appropriately locked-down permissions.
3. Configure a Home page for the Primary Role assigned to each new user.
4. Add a link on your corporate intranet (*Presto* portal page) that passes the "uc" credentials. For example, to allow members of your Marketing department to log in to *Presto*, you could add a link called "Marketing", whose URL passes the "uc" credentials, as shown in the second example above.

When users click the link, they see the appropriate Home page. For example, Guest1 sees the Home page for the Primary Role assigned to Guest1. Guest2 sees the Home page for the Primary Role assigned to Guest2.

Note: OPAC does not need to be enabled to support this functionality.

Direct Access to Images (without Login)

You can permit users to link directly to image files without having to log in, even without enabling normal OPAC logins (i.e., with OpacEnabled=false). Simply create a Role with sufficient privileges to view the images in the Content Type, and add that Role to the Guest account. This will also permit images to be displayed in *Presto* RSS feeds (e.g., profile images for blog posts) without logging in to *Presto*. If the Guest user does not have sufficient privileges to view the images, a generic image appears instead.

Active Directory and Single Sign On (AD SSO)

Presto can be configured to authenticate against Microsoft Active Directory via LDAP, in addition to using *Presto* native database authentication. If Active Directory support is enabled, *Presto* can also be configured to use Single Sign On. Active Directory support and SSO are typically configured as part of the *Presto* installation. If these features were not enabled at installation time, you can enable them after the installation.

Terminology

The following terms are used:

- **Microsoft Active Directory**: A directory service that provides the means to manage the identities and relationships that make up network environments. Active Directory users are maintained in Active Directory and are authenticated through Windows. When an Active Directory user accesses *Presto* (with or without SSO), *Presto* creates a profile for that user if one does not already exist.

- A **native user** is one whose Profile (name, password, locale) is created and maintained in *Presto*.

- **LDAP** (Lightweight Directory Access Protocol): A network protocol and architecture for organizing directory data for network resources including users, files, and devices.

- **Single Sign On (SSO)**: The ability to bypass the *Presto* Login screen and go directly to the Home page, after being authenticated by Active Directory or another mechanism external to *Presto*.

- **OPAC** (Online Public Access Catalog): A feature that enables users to access *Presto* without providing login information. Unlike Single Sign On, OPAC does not require Active Directory, because OPAC authenticates against the *Presto* SQL database using the *Presto* Guest account.

Enabling Active Directory and SSO

If Active Directory support was not configured during the *Presto* installation, you can enable it by following the instructions below. You can also optionally enable Single Sign On (SSO).

Step 1. Confirm Service Account Active Directory Permissions

The *Presto* application identity (PrestoUser), or Service Account, must be a domain account with access to query Microsoft Active Directory. This can be done by making this user a member of the "**Pre-Windows 2000 Compatible Access**" group.

Step 2. Adjust the Membership Provider

Open the *Presto* **Web.config** file:

```
C:\Program Files\Inmagic\Presto\WebApps\Mercury\Web.config
```

In the <membership> section, confirm the following or make the necessary changes:

- The defaultProvider must be "**PrestoADProvider**", and

- There must be one and only one "**PrestoSQLProvider**", not commented out, which is using "**ASPNETDB**", and

- There must be one and only one "**PrestoADProvider**", not commented out, which is using "**ADCONNSTRING**".

For example:

```
<membership defaultProvider="PrestoADProvider" userIsOnlineTimeWindow="20">
  <providers>
   <remove name="AspNetSqlMembershipProvider" />
   <add name="PrestoSQLProvider" connectionStringName="ASPNETDB"
       enablePasswordRetrieval="false"
       enablePasswordReset="true"
       requiresQuestionAndAnswer="false"
       applicationName="/"
       requiresUniqueEmail="false"
       passwordFormat="Hashed"
       maxInvalidPasswordAttempts="5"
       minRequiredPasswordLength="1"
       minRequiredNonalphanumericCharacters="0"
       passwordAttemptWindow="10"
       passwordStrengthRegularExpression=""
       type="System.Web.Security.SqlMembershipProvider, System.Web, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=b03f5f7f11d50a3a" />
     <add name="PrestoADProvider"
       type="System.Web.Security.ActiveDirectoryMembershipProvider, System.Web, Version=2.0.0.0, &#xD;&#xA;
Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
       connectionStringName="ADCONNSTRING"
       attributeMapUsername="sAMAccountName"
       applicationName="/"
       enableSearchMethods="true"/>
  </providers>
</membership>
```

Keep Web.config open and go on to the next step.

Step 3. Adjust the LDAP Connection String

Important! The specified LDAP server must be able to communicate with the *Presto* Web server.

Note: A script is available which extracts the LDAP connection string from the network. Please check the Presto Knowledgebase or contact Inmagic, Inc. to obtain the script. Rather than constructing the connectionString yourself, you can run the script to obtain the connection string, then replace the existing string in Web.config with the extracted string.

In **Web.config**, find the <connectionStrings> section and modify the "**ADCONNSTRING**" connectionString to match your Active Directory LDAP configuration. Use the following format:

```
LDAP://<FQ Domain Controller>/,CN=<AD Container>,DC=<Domain>,DC=<Domain Qualifier>"/>
```

The example below shows the Fully Qualified Domain Server (server.company.com), followed by the container name (CN=Users), and the domain components (DC=company,DC=com).

```
<connectionStrings>
   <remove name="LocalSqlServer"/>
   <add name="ASPNETDB" connectionString="SERVER=CompanySqlServer\Inmagic;Integrated Security=true;Initial
Catalog=PrestoCatalog;User ID=PrestoUser;Min Pool Size=5;Max Pool Size=100;Connect
Timeout=120;MultipleActiveResultSets=true;" providerName="System.Data.SqlClient"/>
   <add name="ADCONNSTRING"
connectionString="LDAP://server.company.com/CN=Users,DC=company,DC=com"/>
</connectionStrings>
```

Depending on your configuration, you may need to include the LDAP port (389) in the connection string, as shown in the following example:

```
connectionString="LDAP://server.company.org:389/DC=company,DC=org"
```

If you are using SSL, the LDAPS port is 636:

> connectionString="LDAPS://server.company.org:**636**/DC=company,DC=org"

Note that the "**ASPNETDB**" connectionString is also required, whether you are using Active Directory or not. It defines the *Presto* connection to the SQL database server. You should not need to modify this string.

Keep Web.config open and go on to the next step.

Step 4. Set the Preferred Name Format [optional]

This step is optional, if you want to use UPN instead of sAMAccountName for Active Directory user logins and import. You can specify one of the following formats, by modifying the attributeMapUsername option for the "**PrestoADProvider**" defined in the Presto **Web.config** file:

- **attributeMapUsername="sAMAccountName"**
  This is the default setting. The *Presto* Login screen will accept the simple form of the username (`jdoe`), and Active Directory import will use this form for the imported UserName.

- **attributeMapUsername="userPrincipalName"**
  This is the Microsoft preferred form. The UPN form (`jdoe@INMAGICINC.LOCAL`) will be required on the Login screen and will be used for Active Directory import.

An example is shown below.

```
<add name="PrestoADProvider"
     type="System.Web.Security.ActiveDirectoryMembershipProvider, System.Web, Version=2.0.0.0, &#xD;&#xA;
Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
     connectionStringName="ADCONNSTRING"
     attributeMapUsername="sAMAccountName"
     applicationName="/"
     enableSearchMethods="true"/>
  </providers>
</membership>
```

Step 5. Enable AD Single Sign On (ADSSO) [optional]

Important! The *Presto* Web server and the client machines accessing *Presto* must be part of the Active Directory domain.

To enable SSO for Active Directory, log in to *Presto* as an administrator, choose **Configure>Administrator Tools>Application Settings**, and set **ADSSOEnabled** to **True**. You can also define a LAN IP address range. Save your settings, close Application Settings, and log out of *Presto*. For descriptions of these settings, see the *Glossary of Application Settings*.

Step 6. Confirm Login and Import AD Accounts

To confirm that you can log in, point your Web browser to the *Presto* URL:

- If you enabled Active Directory and SSO, you should be directed to the *Presto* Home page without seeing a login page. If you run into any issues, see *Troubleshooting SSO*.

- If you did not enable SSO, you should be prompted for your login. Enter your Windows username, using the name format you specified in *Step 4. Set the Preferred Name Format [optional]*. You should then be directed to the *Presto* Home page.

At this point, you may choose to import Active Directory profiles, as described in *Importing Active Directory Accounts* on page 35.

Troubleshooting SSO

To confirm that SSO has been enabled, point your Web browser to the *Presto* URL. You should be directed to the *Presto* Home page without seeing a login page. If you run into any issues, you can try making the adjustments described below.

Note: Single Sign On for Active Directory (ADSSOEnabled) requires that Active Directory is enabled.

Client Browser Configuration for SSO

If you enabled SSO but you see a Connect To pop-up requesting a Windows login, adjust the client browser settings to trust the *Presto* site, for each client that will access *Presto*:

- Internet Explorer: Select **Tools>Internet Options>Security**, select **Local intranet**, click the **Sites** button, then click the **Advanced** button. You can then add the *Presto* server/url to this security zone.

- Firefox: In the Firefox address bar, enter "`about:config`" (without the quotes). Double-click the Preference Name "`network.negotiate-auth.trusted.uris`" and add the *Presto* domain name as the value. For example, add **.Inmagicinc.com** (include the leading period, as shown).

LDAP Connection String

Symptom: The **PrestoADProvider** is enabled in the Web.config file and you can log in to *Presto* with admin/admin but you cannot log in with an Active Directory user.

Solution: Some configurations require the connection string to include the LDAP port number. For instructions, see the explanation earlier in this document (*Step 3. Adjust the LDAP Connection String*).

NTLM Authentication in IIS

Symptom: You are using Single Sign On (SSO), and you are unable to log into *Presto*. Symptoms include being prompted repeatedly for the login name, then being denied access. You may see the error "HTTP Error 401.1 - Unauthorized: Access is denied due to invalid credentials" or "The User Name is invalid", and the `errors.log` file may include the message "Error establishing a user session with user=<username>".

Explanation: This is a Microsoft security feature when accessing an IIS Web site that is configured to use Integrated Windows authentication. For more information, see
http://support.microsoft.com/default.aspx?scid=kb;en-us;871179.

Solution: Run the following command on the IIS server to enable NTLM Authentication in IIS:

1. Start a command prompt.

2. Change to the directory containing **Adsutil.vbs** (**C:\Inetpub\Adminscripts**, by default).

3. Run the following command:

```
cscript adsutil.vbs set w3svc/NTAuthenticationProviders "NTLM"
```

To verify that the NtAuthenticationProviders metabase property is set to NTLM, enter this command:

```
cscript adsutil.vbs get w3svc/NTAuthenticationProviders
```

The following text should be returned:

```
NTAuthenticationProviders        : (STRING) "NTLM"
```

IIS Settings for WinLogin.aspx

If you enabled SSO, IIS should automatically redirect the *Presto* URL from Default.aspx to WinLogin.aspx. To support this redirect, WinLogin.aspx must use Integrated Windows Authentication.

1. Open Microsoft Internet Information Services (IIS) Manager.

2. Under Default Web Site, go to the *Presto* virtual directory where *Presto* is deployed.

3. In the **Winlogin** folder, right-click the **WinLogin.aspx** file, click **Properties**, then go to the **File Security** tab to edit the authentication and access control for this single file.

4. On the Authentication Methods dialog, clear **Enable anonymous access** and check **Integrated Windows authentication**. Click **OK** to save your changes.
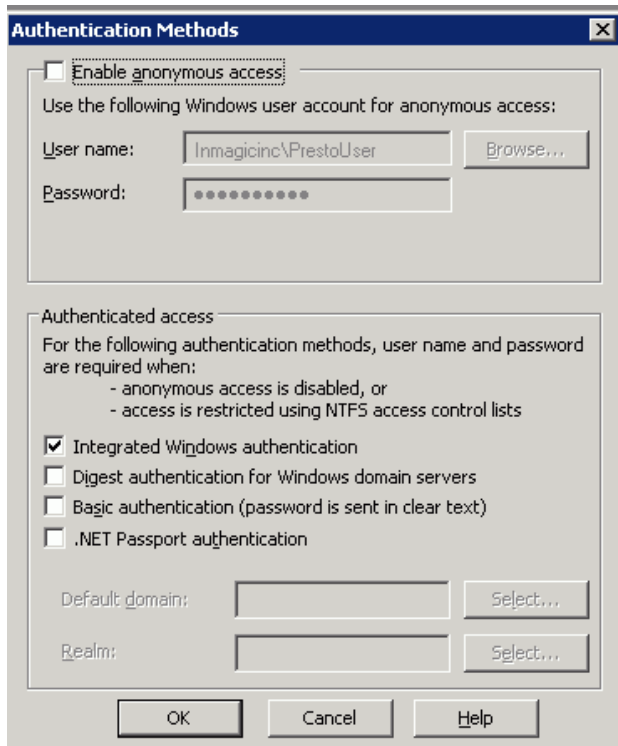
5. Go to the **Custom Errors** tab of the WinLogin.aspx Properties and **Edit** the various 401 errors to assign a custom redirection to C:*<presto-install-path>*\WebApps\Mercury\Redirect401.htm. See the screen capture below.
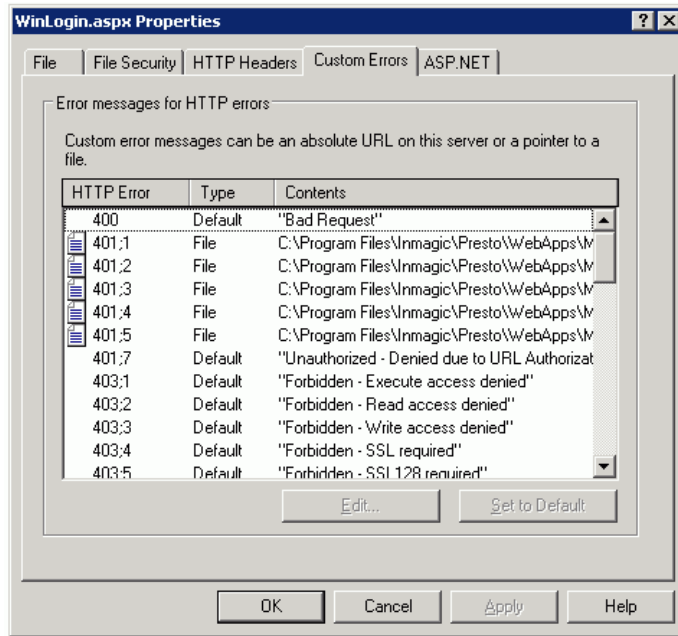
6. Click **Apply** and close the dialog box. This issue should now be resolved.

Creation of Active Directory User Profiles in *Presto*

If Active Directory support is enabled (with or without SSO) and a user logs in who does not have a *Presto* account, the *Presto* user account will be created automatically after the user is authenticated. This new *Presto* user account will be based on the built-in *Presto* **Guest** user, which by default is assigned the Everyone role.

For each of these automatically-created *Presto* user accounts, the *Presto* administrator can choose **Configure>Users** to adjust the User and Role settings. Note that the **Password** fields are not available when editing an Active Directory user, since authentication is handled by Windows.

As an alternative to allowing *Presto* to create user accounts at authentication time, it may be more efficient to import Active Directory accounts, as described below.

Importing Active Directory Accounts

Administrators may prefer to import Active Directory user accounts, rather than waiting for each account to be created by *Presto* at authentication time. Importing users is more efficient because administrators can assign the appropriate role permissions in batches instead of having to edit multiple individual user accounts.

To import existing Active Directory accounts, choose **Configure>Administrator Tools>Import Active Directory Profiles**. This tool is available only if Active Directory support has been enabled. The groups that appear in the *Presto* UI are those for the LDAP connection specified in Web.config.

The following configuration settings can affect the Active Directory profile import process:

- **Application Settings**: Active Directory users are imported a block at a time until all users in an Import configuration are processed. The block size is controlled by the **ActiveDirectoryBlockSize** (default value 1000) specified in **Configure>Administrator Tools>Application Settings**.

- **Web.config**: The **attributeMapUsername** setting indicates the preferred form for both login and import of AD user accounts. The form **attributeMapUsername="sAMAccountName"** results in usernames in the simple form `jdoe`. The form **attributeMapUsername="userPrincipalName"** results in the UPN form (`jdoe@INMAGICINC.LOCAL`).

  Note: If you are planning to import both forms of user names, you can switch Web.config between the two forms (sAMAccountName and userPrincipalName). However, only the currently configured form can be used for login.

  For more information, see *Step 4. Set the Preferred Name Format [optional]* on page 32.

Important! For a list of items that you should be aware of before running an import, please read about importing Active Directory user profiles in the *Presto* online help or the *Presto Basic Administrator's Guide*.

Disabling Active Directory Support

To disable Active Directory support so that only native *Presto* authentication is in effect, open the *Presto* Web.config file, find the <membership> section, and set defaultProvider=" **PrestoSQLProvider**", as shown below (in an actual file, the <providers> section would be filled in, but is omitted below):

```
<membership defaultProvider=" PrestoSQLProvider" userIsOnlineTimeWindow="20">
  <providers>
   …
  </providers>
 </membership>
```

Then log in to *Presto* as an administrator, choose **Configure>Administrator Tools>Application Settings**, and set **ADSSOEnabled** to **False**.

Note: The userIsOnlineTimeWindow attribute shown above is not related to Active Directory support. For more information, see *Online Time Window for Forums*.

External SSO

External Single Sign On (SSO) allows users to sign into an external/corporate site and then navigate seamlessly between that site, where they were authenticated, and *Presto*.

To Configure External SSO

1. Download the **Inmagic Presto API toolkit**, then install and configure it, as explained in the instructions provided with the toolkit. Within the API is a method for creating an encrypted user security token. The documentation provided with the API toolkit explains how to do this.

   > **Important!** The **Inmagic Presto API toolkit** includes sample code that you can copy and customize for your purposes. The toolkit also includes a readme file with important information about configuring security, and a help file (CHM) that describes all of the API methods and calls.

2. Choose **Configure>Administrator Tools>Application Settings**, locate the **Authentication** section, and adjust the settings to support External SSO, starting with setting **ExternalSSOEnabled** to **True**.

3. Perform further customization, such as configuring a custom URL error page for a failed redirect, as explained in *How External SSO Works*.

The following scenarios provide a high-level look at how external SSO works. They are followed by a more detailed diagram and explanation.

Scenario 1

User logs into Third Party Site and then wishes to go to Inmagic Presto.
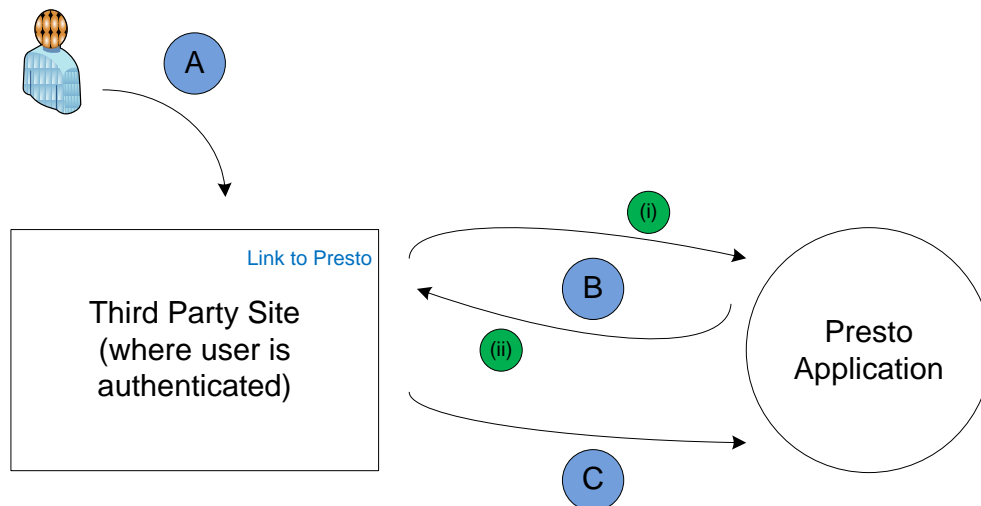
A) The user arrives at Third Party Site and is authenticated.

B) The user clicks a link to go to *Presto*.

    i.   The Third Party Site issues one or more API calls to *Presto*. These calls validate, create, and/or update the *Presto* user account for this user.

    ii.   The call(s) return an encrypted SSO security token.

C) The Third Party Site redirects the user to SSO.ASPX (in the *Presto* application), passing the encrypted SSO token in the query string. *Presto* validates the token and the user. The user then enters the *Presto* application.

Scenario 2

User attempts to enter Inmagic Presto.
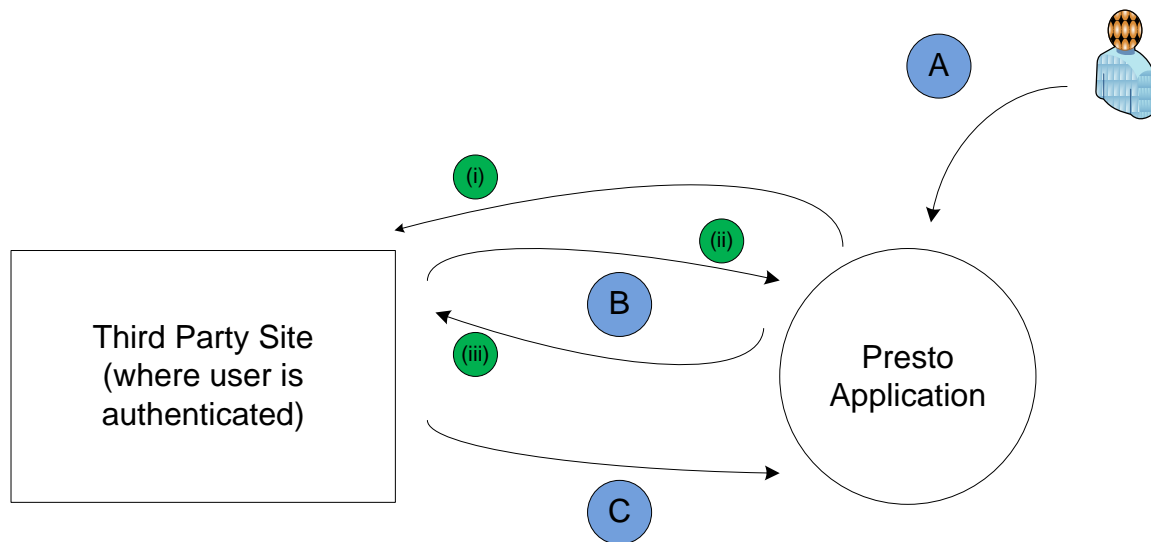
A)   The user attempts to enter *Presto*.

B)   The user cannot be authenticated by *Presto*.

   i.   The user is redirected to Third Party Site (with information that the user is arriving via a failed *Presto* login). The user is authenticated by Third Party Site.

   ii.   The Third Party Site issues one or more API calls to *Presto*. These calls validate, create, and/or update the *Presto* user account for this user.

   iii.   The call(s) return an encrypted SSO security token.

C)   The Third Party Site redirects user back to *Presto*, passing the encrypted SSO token in the query string. *Presto* validates the token and the user. The user then enters the *Presto* application.

Note: If the user attempted to enter *Presto* via an RSS link (link to a specific page), then this URL can be passed back and forth between *Presto* and the Third Party Site so that the user is redirected to this *Presto* page after authenticating at the Third Party Site.

Scenario 3

Presto can be configured for mixed authentication (Active Directory SSO, External SSO, OPAC, native). In this example, users are authenticated by both Active Directory and an external site.
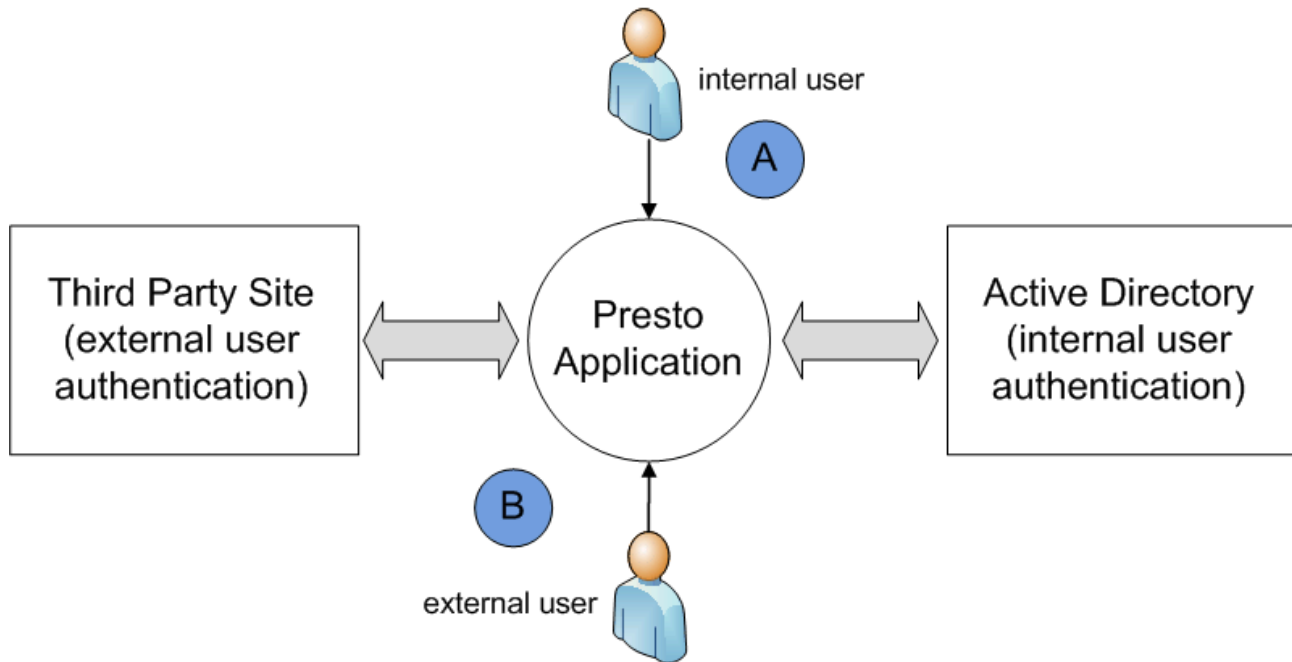
A) An internal user attempts to enter *Presto*. The user is authenticated via AD.

B) An external user attempts to enter *Presto*. The user is authenticated via the Third Party Site.

External SSO Flowchart
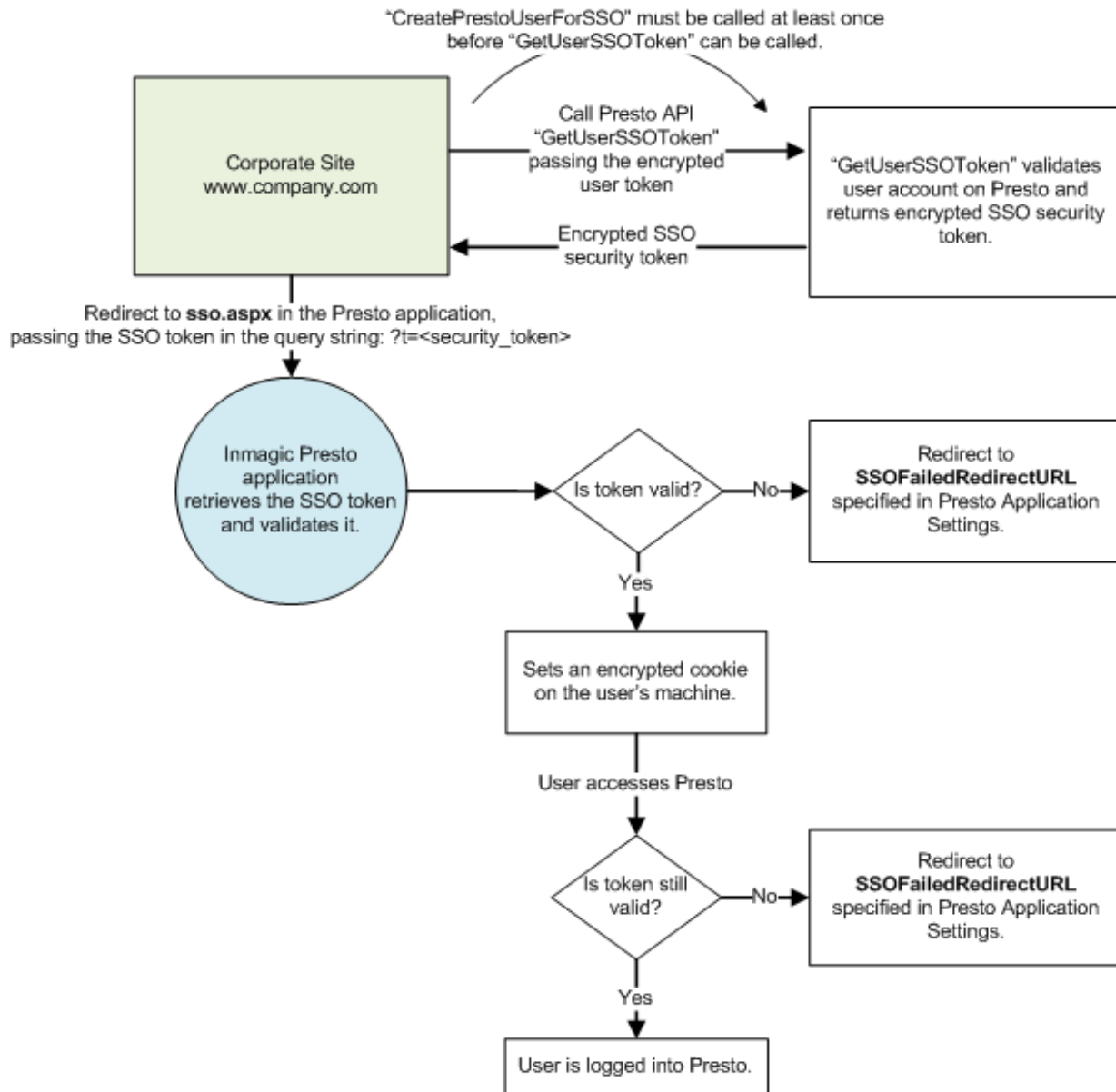
How External SSO Works

As explained in the **Inmagic Presto API toolkit**, the method to create the user token requires the following fields:

- Username

- First Name

- Last Name

- Email Address

- Optional fields can be added to the token for user account creation (a list of *Presto* Roles to add the user to, upon creation)

The "create token" method will return an encrypted string containing the above information.

The token is encrypted using a set of keys that are added to the customer's application Web.config / app.config file, if calling the Presto API from a Windows application they built, as well as the *Presto* application Web.config file. The customer's security keys are used for encryption and the *Presto* security keys are used for decryption. These keys can be changed at any time but need to match on both applications. This is explained in the *Presto* API toolkit documentation.

Upon receiving this encrypted token, the following calls can be made to perform user operations:

- **CreatePrestoUserForSSO**. Using the encrypted token, a new user account will be created with the common SSO password. This must be called once for each user before SSO with *Presto* will work.

- **GetUserSSOToken**. Used for login. Using the encrypted token, this method will check the encrypted token to make sure that it is a valid token (user data and token expiration). If the token is valid and the operations are successful, a new SSO encrypted security token will be returned to the calling site. This method must be called before each SSO attempt, depending on the expiration time of the token.

- **ResetUserSSOPassword**. Using the encrypted token, this method will reset the user's membership password with the current version in the Application Settings area. This method is for security purposes. If for some reason the "common password" must be changed, this method must be called for each existing user before SSO will be valid again.

Upon receiving the SSO security token from the *Presto* API toolkit method, the external site will need to redirect to the Presto SSO.aspx page, passing the security token as a querystring value.

Example: https://www.inmagic.com/presto/sso.aspx?t=<SSO-security-token>

The SSO.aspx page will read and decrypt the user security token. After decryption, the token will be used to validate the user information against the *Presto* user accounts. If the token has not "expired" (explained below) and the user account is valid, a cookie will be set on the user's machine, which can be used for logging in at a later time (if the token has not expired). After the cookie has been set on the user's machine, the user will be logged into *Presto* and redirected to the requested page.

A "returnURL" querystring parameter can also be added to the "SSO.aspx" redirect, if the user has requested to go to a specific record.

Example: https://www.inmagic.com/presto/sso.aspx?t=<SSO-security-token>&returnURL=<record-URL>

For more security on the system, the security token can be set with an expiration time, which can be configured in the *Presto* Application Settings: **SSOTokenExpiration**. For example, if you specify 30 seconds, the user will be de-authorized 30 seconds after clicking a link to go to *Presto*.

If a security token has expired or the user is not valid, the user will then be redirected from the SSO.aspx page to a custom URL (typically an error page or a customer login page). This custom URL can be configured in the *Presto* Application Settings: **SSOFailedRedirectURL**.

If the user attempts SSO without a valid *Presto* account, the user will be denied. **CreatePrestoUserForSSO** needs to be called at least once before SSO will work for each user. **CreatePrestoUserForSSO** can be called multiple times or every time if you want to be sure that the user exists in *Presto* before SSO is attempted. If the user already exists in *Presto*, the method will return a status of "RecordAlreadyExists".

A shared password, which is unknown to the user, will be used for the account password. The shared password can be set in the *Presto* Application Settings: **SSOCreatedUserPassword**.

Note about changing the password: If you change the password in Application Settings (for security reasons), and any automatically created accounts have already been created, SSO will no longer work for those user accounts, unless a call to the "ResetUserSSOPassword" method is made for each user. This call can be made at any time or before each SSO call, if desired for security reasons.

If a user (*who has used External SSO at least once*) attempts to navigate to a page in the *Presto* application without logging in to the corporate site first (for example, if they previously bookmarked a page in *Presto*) and/or their security token has expired, the user will be redirected to the custom URL configured in the *Presto* Application Settings: **SSOFailedRedirectURL**. If a custom URL has not been configured, then the user will be redirected to either a default error page, or to the Presto Login page, depending on the EnableNativeLogin setting (see bullets below).

If a user (*who has used never used External SSO*) navigates to *Presto* or follows a link sent to them by another user, the following occurs:

- If **EnableNativeLogin** is **False** (the default): They are redirected to the configured custom URL. If no custom URL is configured, they get a message that states that they do not have permission to view the page.

- If **EnableNativeLogin** is **True**: They are redirected to the *Presto* native login page. Since the user does not know their password, it is advised that the customer place instructions on the login page (using **LoginMessage.htm**) telling the user that they need to log in through the corporate site.

An example is shown below.

Link to Presto Document   User is logged into external Site
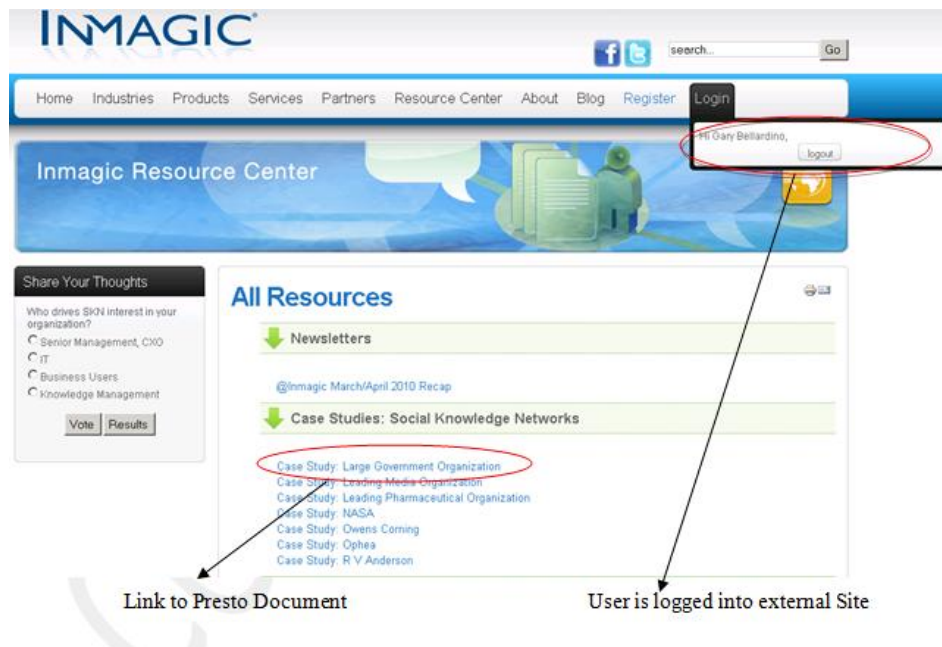
1. User clicks the link to a Presto record.
2. Security Token is set on the user's machine.
3. User is redirected to the SSO.aspx page in Presto.
4. Token is read and validated.
5. If valid, the user is redirected to the requested record detail page.
6. If NOT valid, or if expired, the user is redirected to the custom URL error page (defined by the **SSOFailedRedirectURL** Application Setting).

Note: As indicated in the example above, the security token is created when the user clicks the link to go to *Presto*. In this way, the token can have its expiration time set (**SSOTokenExpiration**) to expire after a certain period of time, such as 30 seconds. Because the user is authenticated just before going to *Presto*, this makes it possible to quickly de-authorize a user.

Updating User Roles due to Membership Changes

A user's roles or profile can be updated on the fly before SSO by calling the **Update Profile** API method. This addresses the need to update user roles at a later time if the user's membership (on the external site) changes. The role needs to exist in *Presto* before making this call. Roles cannot currently be created through API calls.

Application Settings for External SSO

Choose **Configure>Administrator Tools>Application Settings** and locate the **Authentication** section to adjust the settings related to External SSO:

- **ExternalSSOEnabled**: True means use an external SSO security token for authentication, so users sign into an external/corporate site (for authentication) and navigate between that site and *Presto* seamlessly using Single Sign-On. False (the default) means users must log in to *Presto*.

- **EnableNativeLogin**: Used in conjunction with **ExternalSSOEnabled** ONLY. True means native login is also allowed. False (the default) means users will *not* be allowed to log in through the *Presto* login page (they must be authenticated through the external site). If native login has been disabled, there is an "override" to get to the login page, so administrators can get into *Presto*: Append **?sso_or=true** to the end of the URL.

- **SSOFailedRedirectURL**: Used in conjunction with **ExternalSSOEnabled** ONLY. The URL where users will be redirected if External SSO fails. If blank (the default), users will be redirected to the *Presto* login page.

- **SSOLogoutRedirectURL**: Used in conjunction with **ExternalSSOEnabled** ONLY. The URL where users will be redirected if they click the Logout button. If blank (the default), users will be redirected to the *Presto* login page.

- **SSOTokenExpiration**: Used in conjunction with **ExternalSSOEnabled** ONLY. The expiration time for the SSO security token. The default is 2 Mins.

- **SSOCreatedUserPassword**: Used in conjunction with **ExternalSSOEnabled** ONLY. The default password that is set for all users created with External SSO. The default is blank.

- **EnableRememberMeLogin**: Allows users to bypass the Login page when returning to the site. True means the next time the user goes to *Presto* they will not have to log in. False (the default) means users need to log in every time. See *Remember Me (Login page)*.

- **RememberMeExpiration**: Used in conjunction with **EnableRememberMeLogin** ONLY. The expiration for the Remember Me security token. The default is 3 Days Rolling. See *Remember Me (Login page)*.

Search Configuration

Administrators can choose **Configure>Administrator Tools>Application Settings** to modify the following search-related settings. **Note**: For descriptions of each setting, see the *Glossary of Application Settings*.

In the **Search** section:

- MeaninglessLeadingWords
- NonStemmedWords
- SearchAdapterSettings
- RefreshFullTextNoiseWordJobTime
- SqlOptimizeProcessorJobTime
- SearchDefaultStemming

In the **ApplicationSetting** section:

- SearchSummaryType

In the **Configure** section:

- DefaultNumberOfFragments
- DefaultHighlightFragmentSize
- HighlightTags

In the **CUIPages** section:

- SearchSummaryByTypeNumResults

Configuring Noise Words (Stop Words)

To prevent a full-text index from becoming bloated, SQL Server has a mechanism that discards commonly occurring strings that do not help the search. These discarded strings are called noise words (SQL 2005) or stopwords (SQL 2008). These words are omitted from the full-text index; therefore, *Presto* will ignore them in phrase searches, within the search criteria as well as the record text.

Typical noise words in English are *a*, *and*, *is*, and *the*. Noise-word files can be edited, so organizations can add specific words that should be ignored. **Important:** If you edit a noise-word file, you must repopulate the SQL full-text catalogs before the changes will take effect. Use SQL Management Studio, find Full Text Catalogs under Storage, right click on each catalog, and click Build.

For information about configuring noise words in SQL, see the following documentation:

- **SQL Server 2005**: Noise-word files are located in the `$SQL_Server_Install_Path\Microsoft SQL Server\MSSQL.1\MSSQL\FTDATA\` directory. This directory is created and the noise-word files are installed when you set up SQL Server with Full-Text Search support. For information, see http://msdn.microsoft.com/en-us/library/ms142551(SQL.90).aspx.

- **SQL Server 2008**: Noise words have been replaced by stopwords, which are managed in databases using objects called stoplists. See http://msdn.microsoft.com/en-us/library/ms142551.aspx.

An example of a possible list of words to exclude is shown below:

```
about|1|after|2|all|also|3|an|4|and|5|another|6|any|7|are|8|as|9|at|0|be|$|because|bee
n|before|being|between|both|but|by|came|can|come|could|did|do|does|each|else|for|from|
get|got|has|had|he|have|her|here|him|himself|his|how|if|in|into|is|it|its|just|like|ma
ke|many|me|might|more|most|much|must|my|never|no|now|of|on|only|or|other|our|out|over|
re|said|same|see|should|since|so|some|still|such|take|than|that|the|their|them|then|th
ere|these|they|this|those|through|to|too|under|up|use|very|want|was|way|we|well|were|w
hat|when|where|which|while|who|will|with|would|you|your|A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|
P|Q|R|S|T|U|V|W|X|Y|Z|
```

Enabling Synonym Searching

Synonym searching is the ability to precede a word or phrase with a tilde (~) to expand the search to include synonyms (so a search for *~run* will also find *jog*).

This capability is not enabled by default. To enable synonym searching, you must uncomment the **tsENU.xml** file that is supplied with Microsoft SQL Server and populate it with synonyms. Brief instructions are provided below. For complete instructions, see http://msdn.microsoft.com/en-us/library/ms345186(SQL.90).aspx.

Important: When you are editing thesaurus files using text editor tools, the files must be saved in Unicode format and Byte Order Marks must be specified.

1. Using a Unicode editor, such as Microsoft Notepad, open the thesaurus file (**tsENU.xml**) supplied with MS SQL Server. The default path on the SQL Server machine is:

   ```
   <SQL-instance-path>\MSSQL.1\MSSQL\FTData\tsENU.xml
   ```

2. Uncomment the file, then add an expansion set for each set of synonyms you want, as shown below:

   ```xml
   <XML ID="Microsoft Search Thesaurus">
   <!--  Commented out
       <thesaurus xmlns="x-schema:tsSchema.xml">
         <diacritics_sensitive>0</diacritics_sensitive>
         <expansion>
             <sub>cat</sub>
             <sub>feline</sub>
         </expansion>
         <expansion>
             <sub>run</sub>
             <sub>jog</sub>
         </expansion>
   ```

   If your *Presto* installation is using multiple locales, be sure to include synonyms in both languages, such as English and French. For example:

   ```xml
           <expansion>
               <sub>cat</sub>
               <sub>feline</sub>
               <sub>chat</sub>
           </expansion>
   ```

3. Save the file, then restart the SQL Server full-text search service.

Presto should now support synonym searching.

Searching Other Products from *Presto* (Federated Search)

The Federated Search capability allows users to search other products (SharePoint MOSS, Google, Metacarta, etc.) from *Presto*. Each provider's search results are displayed on a dedicated tab within *Presto*. **Note**: It is the customer's responsibility to obtain the proper licensing for each configured product.

Overview

Presto can be configured to perform searches across multiple products ("providers") when using the *Presto* Quick Search box. This functionality is enabled by creating specific search adapters for each product.

Users with appropriate rights can then do a search in *Presto* and the search results will be displayed on individual tabs for each configured provider, such as Google™ or Microsoft SharePoint®.

The following illustration shows the *Presto* Search Results user interface when Federated Search has been configured. Each configured provider has its own Search Results tab. As explained in *Permissions*, the user executing the search must be granted the proper Permissions for the tabs to appear. Note that the Act on Content actions are not available when a search provider tab is selected.
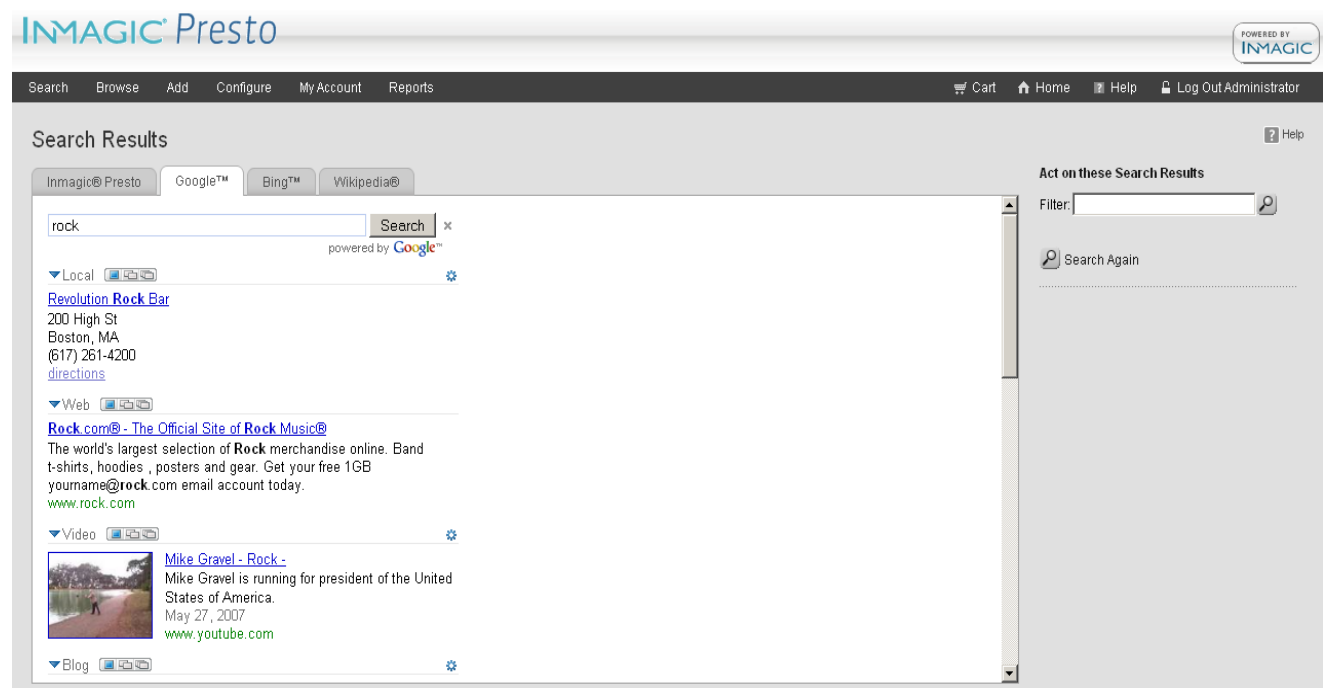
Supported Search Providers

Essentially, any search provider is supported. The level of integration is ultimately determined by the provider's capabilities. Examples of possible providers include the following:

- Google™
- Bing™
- Wikipedia®
- Microsoft SharePoint® (MOSS)
- Scitopia
- Metacarta

Configuring Search Adapters for Federated Search

By default, no search adapters are configured. To configure search adapters, choose
Configure>Administrator Tools>Application Settings, and modify the **SearchAdapterSettings** in the
Search section.

Under <SearchAdapterSettings>, add a <PluginSetting> section for each product that should appear as a
Search Results tab (see example below).

Note: Further configuration is required for Google, SharePoint, and any search adapter that has the
"UrlIsInternal" value set to **True**. For more information, see the *Google Example* and *SharePoint Example* later
in this document.

The following example shows search adapters configured for Google, SharePoint, Bing, Wikipedia, Scitopia,
and Metacarta. Each provider is highlighted, to indicate where the section begins.

```
<!-- Always edit this section with a Unicode editor that supports UTF-8 -->
<SearchAdapterSettings>
 <xmlSerializerSection type="Inmagic.Mercury.Components.Search.Configuration.SearchAdapterSettings,
Inmagic.Mercury.Components, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null">
  <SearchAdapterSettings>
   <PluginSetting id="google_search" DisplayName="Google™">
    <ConstructorParams/>
    <Params>
     <Param key="Url" type="string">
      <Value><![CDATA[searchadapters/GoogleSearchAdapter.aspx]]></Value>
     </Param>
     <Param key="UrlIsInternal" type="bool">
      <Value>True</Value>
     </Param>
    </Params>
   </PluginSetting>
   <PluginSetting id="SharePoint_search" DisplayName="SharePoint">
    <ConstructorParams/>
    <Params>
     <Param key="Url" type="string">
      <Value><![CDATA[searchadapters/SharePoint.aspx]]></Value>
     </Param>
     <Param key="UrlIsInternal" type="bool">
      <Value>True</Value>
     </Param>
    </Params>
   </PluginSetting>
   <PluginSetting id="bing_search" DisplayName="Bing™">
    <ConstructorParams/>
    <Params>
     <Param key="Url" type="string">
      <Value><![CDATA[http://www.bing.com/search?q={0}]]></Value>
     </Param>
     <Param key="UrlIsInternal" type="bool">
      <Value>False</Value>
     </Param>
    </Params>
   </PluginSetting>
   <PluginSetting id="wiki_search" DisplayName="Wikipedia®">
    <ConstructorParams/>
    <Params>
     <Param key="Url" type="string">
```

```
                    <Value><![CDATA[http://en.wikipedia.org/wiki/Special:Search?search={0}]]></Value>
        </Param>
        <Param key="UrlIsInternal" type="bool">
         <Value>False</Value>
        </Param>
       </Params>
      </PluginSetting>
      <PluginSetting id="scitopia_search" DisplayName="Scitopia">
       <ConstructorParams/>
       <Params>
        <Param key="Url" type="string">
                    <Value><![CDATA[http://www.scitopia.org/scitopia/result-list/fullRecord:{0}]]></Value>
        </Param>
        <Param key="UrlIsInternal" type="bool">
         <Value>False</Value>
        </Param>
       </Params>
      </PluginSetting>
      <PluginSetting id="metacarta_search" DisplayName="Metacarta">
       <ConstructorParams/>
       <Params>
        <Param key="Url" type="string">
                    <Value><![CDATA[http://geosearch.metacarta.com/?keyword={0}]]></Value>
        </Param>
        <Param key="UrlIsInternal" type="bool">
         <Value>False</Value>
        </Param>
       </Params>
      </PluginSetting>
     </SearchAdapterSettings>
    </xmlSerializerSection>
</SearchAdapterSettings>
```

The following tables describe the attributes and elements of a search adapter *PluginSetting*.

PluginSetting Attributes

Attribute Name	Description	Parameter and value
Id	The ID of the PluginSetting. This value must be unique and may not contain spaces.	Unique String. Required.
DisplayName	The name to appear on the *Presto* Search Results tab.	String. Required.

PluginSetting Elements

Parameter Name	Description	Parameter and value

UrlIsInternal	Specifies whether the URL is an **internal** or **external** resource. To indicate an internal resource, use True:
	
	`<Param key="UrlIsInternal" type="bool"> <Value>True</Value>`
	
	To indicate an external resource, use False:
	
	`<Param key="UrlIsInternal" type="bool"> <Value>False </Value>`
	
	Google and SharePoint are considered **internal** resources, and must be set to True. For more information, see the *Google Example* (page 50) and the *SharePoint Example* (page 51).
		Boolean. Required.

| Url | The path to the resource of the search adapter provider, specified inside a <![CDATA[]]. This is the URL that is used when creating the iframe on the *Search Results* page.
• If the resource is **external**, the path must be absolute (http://...). Note that **external** resources have a {0} in the string. This value will be replaced with the user-specified query. The query expression is passed directly to the search adapter provider. There is no mapping of a *Presto* query expression to the search adapter query expression.
• If the resource is **internal**, the path must be relative to the root of the *Presto* Web application (the *Presto* virtual directory root). For example: searchadapters/GoogleSearchAdapter.aspx. Google and SharePoint are both considered **internal** resources. Note that internal resources do not contain the value {0} in the string, because the adapter itself controls whether mapping occurs. For more information, see the *Google Example* (page 50) and the *SharePoint Example* (page 51). | String. Required. |

Permissions for Configured Search Adapters

To use the configured search adapters (to see the tabs when search results are displayed), the user must be a member of a role which has been granted the System Permission PERFORM FEDERATED SEARCH. Use **Configure>Roles** to change System Permissions (see below). Use **Configure>Users** to assign users to roles.

Note: The PERFORM FEDERATED SEARCH permission allows the user to access *all* search adapter tabs. It is not possible to specify individual permissions for each specific search adapter.
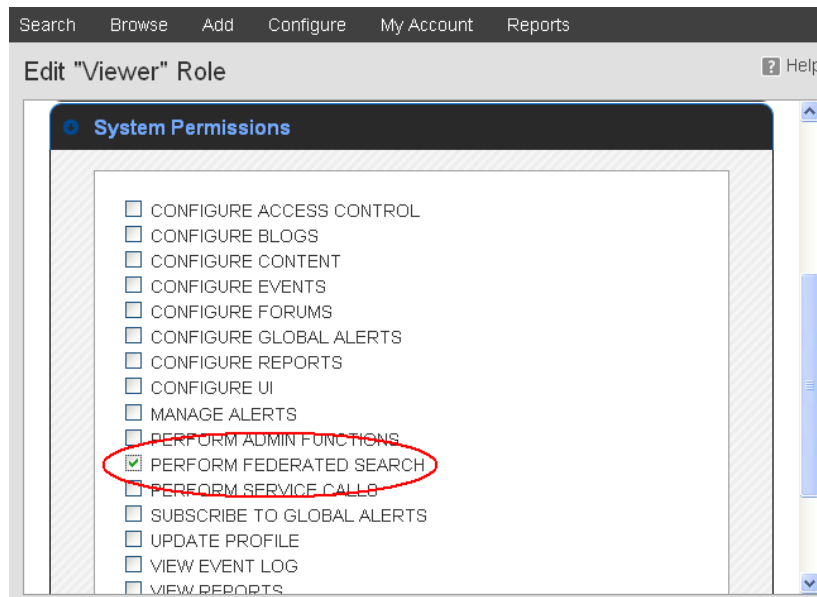
Google Example

The Google search adapter uses their Ajax Search API (http://code.google.com/apis/ajaxsearch/). Google provides numerous examples of using this API and it is well documented. To use this API, a new Web page must be created; therefore, in the context of *Presto* search adapters, it is considered an internal resource ("UrlIsInternal").

The **GoogleSearchAdapter.aspx** page referenced earlier is shown below. This ASPX page is a modified version of an example provided by Google. It can be further modified as desired. Note that the ASPX page contains references to *Presto* objects, which are needed to extract the search query entered by the user. The ASPX page can be placed anywhere in the *Presto* virtual directory. The location of the page must be noted in the Google search adapter settings (see *PluginSetting Elements*).

```
<!--
 Copyright (c) 2008 Google Inc.

 You are free to copy and use this sample.
 License can be found here: http://code.google.com/apis/ajaxsearch/faq/#license
-->
<%@ Page Language="c#" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
 <head>
   <script language="CS" runat="server">
    System.Text.StringBuilder query;
    void Page_Load(object sender, System.EventArgs e)
    {
      query = new
System.Text.StringBuilder(Inmagic.Mercury.Components.Web.SessionManager.SessionCUISearchContext.UserQueryString);
      string filterQuery = Inmagic.Mercury.Components.Web.SessionManager.SessionCUISearchContext.FilterQueryString;
      if (!string.IsNullOrEmpty(filterQuery)) {query.Append(" " + filterQuery);}
    }
   </script>
  <meta http-equiv="content-type" content="text/html; charset=utf-8"/>
  <title>Google Search</title>
  <style type="text/css">

  body {
   background-color: white;
   color: black;
   font-family: Arial, sans-serif;
   font-size : 13px;
   margin: 15px;
  }

  #searchcontrol .gsc-control { width : 400px; }

  </style>
  <script src="http://www.google.com/jsapi" type="text/javascript"></script>

  <script type="text/javascript">
  //<![CDATA[

  google.load('search', '1.0');

  function OnLoad() {
   // Create a search control
   var searchControl = new google.search.SearchControl();
```

```
    // Add in a full set of searchers
    var localSearch = new google.search.LocalSearch();
    searchControl.addSearcher(localSearch);
    searchControl.addSearcher(new google.search.WebSearch());
    searchControl.addSearcher(new google.search.VideoSearch());
    searchControl.addSearcher(new google.search.BlogSearch());
    searchControl.addSearcher(new google.search.NewsSearch());
    searchControl.addSearcher(new google.search.ImageSearch());
    searchControl.addSearcher(new google.search.BookSearch());
    searchControl.addSearcher(new google.search.PatentSearch());

    // Set the Local Search center point
    localSearch.setCenterPoint("Boston, MA");

    // tell the searcher to draw itself and tell it where to attach
    searchControl.draw(document.getElementById("searchcontrol"));
    // execute an initial search
    searchControl.execute("<%=query%>");
   }
  google.setOnLoadCallback(OnLoad, true);
  //]]>
  </script>
 </head>
 <body>
  <div id="searchcontrol">Loading</div>
 </body>
</html>
```

SharePoint Example

Follow the steps below to enable users to search SharePoint from within *Presto*. Search results will appear on a **SharePoint** tab on the *Presto* Search Results screen.

Note: This feature requires the Microsoft Office SharePoint Server 2007 (MOSS) Federated Search connector, and therefore is supported only on MOSS, not Windows SharePoint Services (WSS).

1. Obtain the URL for the SharePoint site that will be searched. This can be the main SharePoint URL or some subset or site within the main SharePoint site.

2. Run **WSDL.exe** against the SharePoint site URL to create a C# file in the *Presto* bin folder (required location). **Note**: WSDL.exe is an installed file that typically resides in the SDK folder (C:\Program Files\Microsoft.NET\SDK\v1.1\Bin) , but may be in another location on your system.

The following example creates a C# file called **SharePointSearch.cs**:

```
WSDL.exe /out: c:/Program Files/Inmagic/Presto/WebApps/Mercury/bin/SharePointSearch.cs
http://SharePointServer/_vti_bin/search.asmx?wsdl
```

3. Use the **csc** command to compile the .cs file and create a DLL in the Presto bin folder (required location).

```
csc /out: c:/Program Files/Inmagic/Presto/WebApps/Mercury/bin/Search.DLL  SharePointSearch.cs
```

4. Create an ASPX page that will consume the Search.DLL. Note the following requirements:
   – The ASPX page must be in the *Presto* WebApps folder (default location C:\Program Files\Inmagic\Presto\WebApps) or one of its subfolders. The URL parameter in the configuration settings must point to the page.

- The page can have any name (for example, SharePointSearch.aspx).

- The ASPX page must include this line in the Page_Load (highlighted in the following example):

  query = new
  System.Text.StringBuilder(Inmagic.Mercury.Components.Web.SessionManager.SessionCUISearchContext.UserQueryStrin
  g).ToString();

- The ASPX page must specify **queryService.Credentials** that will be passed to the SharePoint service. The example line highlighted below shows how to use the **DefaultCredentials**.

- A corresponding user account must be defined on SharePoint.

The following example shows an ASPX page that meets these criteria. The most important lines are highlighted.

```
<%@ Page Language="c#" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
 <head>
   <script language="CS" runat="server">
    System.Text.StringBuilder query;

     void Page_Load(object sender, System.EventArgs e)
    {
       string query = string.Empty;

       query = new
System.Text.StringBuilder(Inmagic.Mercury.Components.Web.SessionManager.SessionCUISearchContext.UserQueryString).ToStrin
g();

       QueryService queryService = new QueryService();

       //queryService.Credentials = System.Net.CredentialCache.DefaultCredentials.GetCredential(new Uri(queryService.Url),
"NTLM");
       //queryService.Credentials = System.Net.CredentialCache.DefaultCredentials;

       queryService.Credentials = System.Net.CredentialCache.DefaultCredentials;

       //queryService.UseDefaultCredentials = true;
       queryService.PreAuthenticate = true;

       System.Text.StringBuilder queryXml = new System.Text.StringBuilder();

       string cQuery = "<QueryPacket xmlns=\"urn:Microsoft.search.Query\">";

       queryXml.Append(cQuery);
       queryXml.Append("<Query>");
       queryXml.Append("<Context>");
       queryXml.Append("<QueryText language=\"en-US\" type=\"STRING\">" + query + " </QueryText>");
       queryXml.Append("</Context>");
       queryXml.Append("<Range>");
       queryXml.Append("<Count>30</Count>");
       queryXml.Append("</Range>");
       queryXml.Append("</Query></QueryPacket>");

       try
       {
         System.Xml.XmlDocument doc = new System.Xml.XmlDocument();
         string status = queryService.Status();
```

```
        string SearchInfo = queryService.GetPortalSearchInfo();
        string result = queryService.Query(queryXml.ToString());
        System.Data.DataSet queryResults = queryService.QueryEx(queryXml.ToString());
        rptSharePointSearch.DataSource = queryResults;
        rptSharePointSearch.DataBind();
    }
    catch (Exception ex)
    {
        string msg = ex.Message;
    }
  }
 </script>
<meta http-equiv="content-type" content="text/html; charset=utf-8"/>
<title>SharePoint Search</title>
<style type="text/css">

body {
  background-color: white;
  color: black;
  font-family: Arial, sans-serif;
  font-size : 13px;
  margin: 15px;
}

#searchcontrol .gsc-control { width : 400px; }

</style>
 //<![CDATA[

 function OnLoad() {
   // Create a search control
 }

 //]]>
 </script>
 </head>
<body>
  <form id="form1" runat="server">
  <div>
<asp:Repeater ID="rptSharePointSearch" runat="server">
<HeaderTemplate>
<table style="border: 1px solid Navy; background-color:Silver">
<tr style="width: 1200px">
<td style="width:100px">
Title
</td>
<td style="width:480px">
Description
</td>
<td style="width:500px">
URL
</td>
</tr>
</table>
</HeaderTemplate>
<ItemTemplate>
<table style="border: 1px solid Navy; ">
<tr style="width: 1200px">
```

```
<td style="width:100px">
<%#Eval("Title")%>
</td>
<td style="width:500px">
<%#Eval("Description")%>
</td>
<td style="height: 24px; width: 500px;" valign="top">
        <asp:HyperLink ID="httpURL" NavigateUrl='<%# DataBinder.Eval(Container.DataItem, "Path")%>' runat="server"
Text='<%# DataBinder.Eval(Container.DataItem, "Path") %>' >
        </asp:HyperLink>
          </td>
</tr>
</table>
</ItemTemplate>
</asp:Repeater>
   </div>
   </form>
    </body>
</html>
```

Continue with the steps below.

5. If you have not already done so, log in to *Presto* as an administrator and choose
    Configure>Administrator Tools>Application Settings, and update the **SearchAdapterSettings** to
    include an entry for SharePoint.

Be sure to specify the ASPX page name for the CDATA value, as shown in the example below.

```
    <PluginSetting id="SharePoint_search"
DisplayName="SharePoint">
     <ConstructorParams/>
     <Params>
      <Param key="Url" type="string">
       <Value><![CDATA[SharePoint.aspx]]></Value>
      </Param>
      <Param key="UrlIsInternal" type="bool">
       <Value>True</Value>
      </Param>
     </Params>
    </PluginSetting>
```

6. In order to see the SharePoint tab, the user who is searching must be a member of a role which has
    been granted the System Permission PERFORM FEDERATED SEARCH. For more information, see
    *Permissions for Configured Search Adapters*Permissions.

Categories (Browse Collections) Settings

Choose **Configure>Administrator Tools>Application Settings** and locate the **ApplicationSetting** section
to adjust the following settings:

-   CategoryItemDelimiter

-   TaxonomyAddCategoryToTop

Locate the **BrowseCollection** section to adjust these settings:

-   DisplayCategoryContentCount

■ DisplayCumulativeCategoryContentCount

Note: For descriptions of each setting, see the *Glossary of Application Settings*.

Look-and-Feel Adjustments

You can adjust the appearance of the *Presto* user interface in the following ways.

Home Page Modifications

Administrators can choose **Configure>Administrator Tools>Application Settings** to adjust settings related to the Home page. For more information, see:

■ *Glossary of Application Settings* (page 17).

■ *Presto Default Home Page Configuration Guide*

Login Page Modifications

You can make the following changes to the Login page:

■ Replace the Inmagic logo: See *Image Files and Branding*.

■ Show or hide the Create Account and Change Password links: See *Reset Password (Login page)*.

■ Add an image and/or text to the left side of the page, as described below.

You can add information to the left side of the Login page. The example below shows the addition of an image and the text **Presto Knowledgebase**.

To add text or an image, modify the **LoginMessage.HTM file** in the `\WebApps\Mercury` folder. Replace the information in the <body> section shown below with the information that should appear on the Login page.

```
<body>
  <!--Information about Inmagic&#0174; <i>Presto</i> goes here.-->
</body>
```

For example:

```
<body bgcolor="#ffffff" background="/Presto/images/custom/einstein.jpg"
 style="background-position: bottom; background-repeat: no-repeat;
 background-attachment: fixed; font-family: Arial; font-weight: bold">

  Presto Knowledgebase

</body>
```

Image Files and Branding

You can add or replace images that appear throughout the UI. For example, you can replace the supplied "Inmagic Presto" images with your corporate logo. *Presto* image files are located in the following folder or its subfolders: `<install-dir>\WebApps\Mercury\images`. The default location is:

```
C:\Program Files\Inmagic\Presto\WebApps\Mercury\images
```

Replacing an Image

The recommended method for overriding an image depends on how it is linked to in the product code:

- If the image is linked using a "background-image" attribute in a CSS class, then it is recommended that this attribute be overridden using the respective "Custom" style CSS file.

- If the image is linked using HTML, it is recommended that the image be replaced with a custom image with an identical name.

To avoid formatting issues, ensure that the new image has similar dimensions to the original image. The images listed below are commonly overridden.

Image	Description	File Name	Style Property
INMAGIC Presto	The logo displayed in the upper left corner of the application, including the Login screen.	..\images\branding\ presto-logo.gif	**File:** BrandingStyles.css **Class:** .branding-image
POWERED BY INMAGIC	Displayed in the upper right corner throughout the application.	..\images\branding\ poweredby.gif	**File:** BrandingStyles.css **Class:**.poweredby-image
?	Displayed when an image is not available for controls associated with image files (bmp, gif, jpg, etc.). To show or hide this image, go to **Application Settings** and set **DisplayImageDataValueNotSpecified** to True or False.	..\images\ui\ not-available-image.gif	Not linked using CSS

Removing an Image

To remove an image in *Presto*, use the same concept as replacing an image, but substitute the following supplied image, which is transparent:

```
C:\Program Files\Inmagic\Presto\WebApps\Mercury\images\null_image.gif
```

By replacing the predefined image with this image using the identical name, the image will not be visible in a Web browser.

Adding or Changing Content Type Icons

Every Content Type in *Presto* can have an icon associated with it, as shown in the example below.
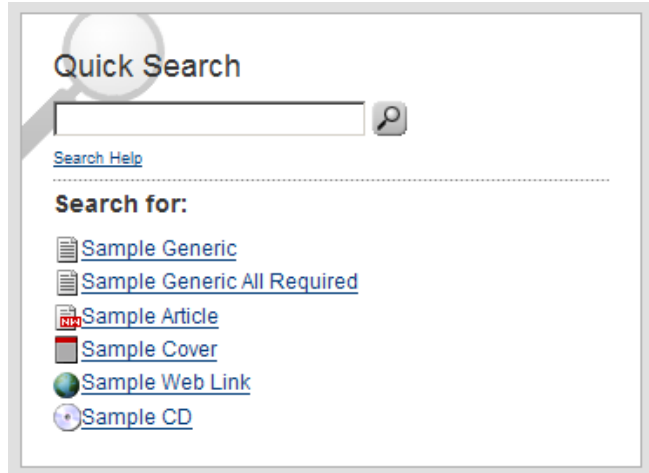
Content Type icons are located in the following directory by default:

```
C:\Program Files\Inmagic\Presto\WebApps\Mercury\images\icons\types
```

You can add icon image files to this directory as needed, then use **Configure>Content Types** in *Presto* to associate an image file with a Content Type.

Cascading Style Sheets

Presto includes a number of cascading style sheets (CSS files) that control the appearance of the user interface (colors, fonts, etc.). Each installed CSS file has a **custom CSS file** associated with it that can be used to override any CSS classes defined in the parent file.

To override the styles defined in the installed CSS files, you must create custom CSS files with the names shown below. Custom files are not supplied with *Presto*, and must be created manually if they are to be used. To determine which CSS files are being used on a page, use the browser's "View Source" functionality.

The installed CSS files reside in the following folder by default:

```
C:\Program Files\Inmagic\Presto\WebApps\Mercury\styles
```

Note: Unlike the CSS files discussed in this section, CSS files used for the Home Pages are located in folders under `~/App_Themes` and may be modified directly.

Important! Do not modify the installed CSS files.

Installed Files (do not modify)	Custom Files (must be created manually)	Defines styles used in the following areas
BaseStyles.css	CustomBaseStyles.css	Standard (not pop-up) pages and PDF download
BlogStyles.css	CustomBlogStyles.css	Blog pages
BrandingStyles.css	CustomBrandingStyles.css	About.aspx, PopActionHtmlHead.ascx, PopupHtmlHead.ascx, StyleSheets.ascx, and PDF download
ButtonStyles.css	CustomButtonStyles.css	About.aspx, PopActionHtmlHead.ascx, StyleSheets.ascx
EmailAndDownloadHTMLStyles.css	CustomEmailAndDownloadHTMLStyles.css	E-mail and Download content pages (HTML format only)
FreeTextBoxStyles.css	CustomFreeTextBoxStyles.css	HTML Text controls (Add/Edit screens, Configure Home Page, and so on)
gridStyle.css	CustomgridStyle.css	StyleSheets.ascx
HomeStyles.css	CustomHomeStyles.css	About.aspx, Default.aspx (Login screen), home\Default.aspx (home page), PopJobInProgress.aspx, RSS Feed Web Part.

ListStyles.css	CustomListStyles.css	About.aspx, StyleSheets.ascx
MenuStyles.css	CustomMenuStyles.css	StyleSheets.ascx
PopActionStyles.css	CustomPopActionStyles.css	Action pop-up pages and PDF download
PopupStyles.css	CustomPopupStyles.css	Non-action pop-up pages
tabStyle.css	CustomtabStyle.css	Pages with tabs
TreeStyle.css	CustomTreeStyle.css	ContentCollectionCategoriesTree.aspx, PopContentCopy.aspx, PopContentMove.aspx, home\Default.aspx (home/portal page), PopCategorySelect.aspx

Note: For CSS modifications to take effect for PDF download, an `iisreset` is necessary. PDF download uses the following CSS files (in this order): BaseStyles.css, BrandingStyles.css, PopActionStyles.css.

Example of a CSS Override

This example shows how to change the color used for highlighting search results. The **HighlightTags** setting turns highlighting on or off, and the "searchmatch" class controls the appearance of the highlighting. Here is a portion of the HTML source from a Search Results or Detail Screen:

Search results can be **** highlighted ****.

The "searchmatch" class is defined in the supplied file called `..\styles\BrandingStyles.CSS`. The supplied .CSS files should never be modified, so in order to change the formatting of highlighted text (e.g., make it green instead of yellow), you can create a file called `..\styles\CustomBrandingStyles.CSS` and define the `"searchmatch"` style in that file. For example:

```
.searchmatch {
       font-weight:bold;
       background-color: #00FF00
}
```

Any styles defined in CustomBrandingStyles.CSS will override those defined in BrandingStyles.CSS.

Alert E-mail Templates

E-mail alerts use the following templates, which can be modified if necessary, using a text editor:

- AlertEmailTemplate.htm
- CollectionEventItemSummaryTemplate.htm
- EventItemSummaryTemplate.htm

The e-mail alert templates are in the following location:

```
<install-dir>\WebApps\Mercury\system\template\
```

For example:

```
C:\Program Files\Inmagic\Presto\WebApps\Mercury\system\template\
```

Note: Any changes will need to be redone after an upgrade. Modifications to these templates are *not* retained when upgrading to a new version of *Presto*.

Note that the e-mail alert templates contain variables for data that will be substituted at run-time. These variables are case-sensitive and are also noted in the comments of the template files.

- Alert_Date: The date associated with the alert
- Message_Content: The main message content - the notifications
- Base_Url: The base URL for the *Presto* Web application

- Manage_My_Alerts_Link: The "Manage My Alerts" page link
- User_First_Name: The first name from the user's profile page link
- User_Last_Name: The last name from the user's profile page link

For additional information, see *Localization of Alert E-mail Templates*.

Default Views for Alert Messages and RSS Feeds

Administrators can choose **Configure>Administrator Tools>Application Settings** and locate the **DefaultViews** section to make the following adjustments.

Setting	Description
UserLoginViews	Format login and logout events that appear in alert messages and RSS feeds.
SystemJobViews	Format system job events that appear in alert messages and RSS feeds.

Page Display and Layout (CUI Pages)

Administrators can choose **Configure>Administrator Tools>Application Settings** to adjust the settings for **CUIPages** (Customizable User Interface). For descriptions of each setting, please see the *Glossary of Application Settings*.

Pop-up Windows

Choose **Configure>Administrator Tools>Application Settings** to configure pop-up windows in *Presto*. For descriptions of each setting, please see the Popups section in the *Glossary of Application Settings*.

Custom JavaScript

You can use JavaScript to add functionality to *Presto*. A custom script file called `CustomCommonScripts.js` is included on every page. This is a global include; therefore, any scripts added to this file will be available on every page in *Presto*.

To add custom scripts, create a file called `CustomCommonScripts.js`, in the following location:

```
<install-path>\WebApps\Mercury\jscript\CustomCommonScripts.js
```

All custom scripts should then be then entered in this file. It is recommended that only functions be added to this file, which may then be called from a Custom Value control or from a modified ASPX page.

ASPX Page Modifications

Modifying ASPX pages should only be done if absolutely necessary, as these pages are heavily associated with compiled code. (One exception is editing the Home page files (Default.aspx or Default_v2.aspx), which is commonly done.) Rigorous testing should be performed on any changes made to ASPX pages. To modify ASPX pages, you can use any text editor. Please consult Inmagic Engineering for instructions.

Important! Modifications to ASPX pages are not carried over during an upgrade.

RSS Feed Configuration

Presto supports several types of RSS feeds:

- **Search RSS feeds**. Click the RSS button in the upper right corner of any Search Results page.

- **User Alert RSS feeds**. Click the Subscribe button next to any Alert on the My Account>Manage My Alerts page.

- **Global Alert RSS feeds**. Click the Subscribe button next to any Alert on the My Account>Subscribe to Global Alerts page. (To create global alerts, choose **Configure>Global Alerts.**)

- **RSS feeds for Forums or Forum topics**. Click the RSS button at the top of the Forum or Forum Topic page.

- **Blog RSS feeds**. Click the RSS icon on the blog toolbar.

To add an RSS feed Web Part to a *Presto* home page

For instructions on how to configure an RSS feed Web Part on a *Presto* home page, see the online help or the *Basic Administrator's Guide*.

Note: The Short Description display type for the RSS Feed Web Part uses classes for the Title, Date, Description and Main Box styles (e.g., class="rssTitle"), so those styles can be customized. The styles are under "RSS Short Description Feed Styles" in HomeStyles.css. See *Cascading Style Sheets*.

To adjust RSS settings

1. Choose **Configure>Administrator Tools>Application Settings** and locate the **ApplicationSetting** section, where you can make adjustments to the following settings: **MaxRSSItems**, **MaxRSSEventAgeInHours**, **RSSMode**. For example, you can use the RSSMode setting to lock down RSS feeds. For information, see the *Glossary of Application Settings*.

2. If feeds in the RSS Web Part on the Home page are not working because the *Presto* application server has no direct external access to the Internet, you can specify a proxy URL to use for RSS feeds, as follows.

   Important! The following procedure involves directly editing a SQL database table and should be done only by administrators who have SQL database experience.

   a. Start **Microsoft SQL Management Studio** on the SQL database server.

   b. Open the **dbo.AppConfig** table in the *Presto* Catalog, and add the following rows:
      - RssProxyUrl: The value is a server name or IP address with port (e.g., 192.168.200.6:8080).
      - RssProxyUsername: Optional, not all proxy servers require username/password.
      - RssProxyPassword: Optional, not all proxy servers require username/password.

   c. At a Command prompt, enter IISRESET to restart IIS. **Note**: IISRESET is recommended because it also restarts the World Wide Web Publishing Service, Simple Mail Transfer Protocol (SMTP), and HTTP SSL. If you choose to restart the IIS Admin service from Windows Services, you must manually restart the other stopped services as well.

To determine how RSS feeds appear

To determine how RSS feeds (and alert e-mails) appear, configure the Alert View and/or RSS Feed View for each content type (**Configure>Content Types>Configure Views**).

SMTP E-mail Server Configuration

For more information: The SMTP server is accessed internally through the .Net Framework 2.0 SmtpClient class. Configuration through the <system.net> element is a feature of the .Net Framework, not *Presto*. For MSDN documentation, search for "<mailSettings> Element (Network Settings)", or follow this link: http://msdn2.microsoft.com/en-us/library/w355a94k.aspx.

Adjusting the e-mail server configuration in Web.config

Configuring *Presto* to use an e-mail server is typically done during installation. However, you can change the configuration by editing the <system.net> element of the *Presto* **Web.config** file. The Web.config file contains three typical configurations, as shown below. Only one mailSettings section should be enabled at a time. The others should be commented out.

```
<system.net>
   <!-- Email settings for the Framework 2.0 SmtpClient
      Attributes are:
      <network
         host="relayServerName"
         port="portNumber"
         userName="username"
         password="password" />
   -->
   <!-- Settings to use SMTP server when Presto Services is running as a domain user (userName/password not
required)-->
   <mailSettings>
    <smtp>
     <network host="serverName"/>
    </smtp>
   </mailSettings>

<!-- Settings to use the IIS Default SMTP Virtual Server installed on this machine -->
   <mailSettings>
    <smtp>
     <network host="localhost"/>
    </smtp>
   </mailSettings>

   <!-- Settings to use SMTP server when Presto Services is running as a local user (userName/password required)-->
   <mailSettings>
    <smtp>
     <network host="serverName" userName="username" password="password"/>
    </smtp>
   </mailSettings>
  </system.net>
```

Important! You must restart Inmagic Presto Services after modifying e-mail settings in Web.config. You may also need to restart the "IIS Admin Service" in the system's **Administrative Tools>Services** dialog.

Adjusting e-mail notification settings and SMTP Client Encryption (SSL) in Web.config

You can edit the prestoEmailSettings in Web.config to make the following adjustments:

- Specify e-mail addresses that *Presto* will use for administration and notification e-mails.
- Specify whether the SmtpClient uses Secure Sockets Layer (SSL) to encrypt the connection. (Introduced in v3.5 Service Pack 1 – Do not use this setting unless you have SP1 or later.)

The following line shows an example:

< prestoEmailSettings adminEmail="presto-admin@company.com " notificationEmail=notify-noreply@company.com smtpClientEnableSsl="false"/>

These attributes are explained below.

- **AdminEmail** (required): The administrator's e-mail address. This should be a real or aliased e-mail address set up for administrator use. *Presto* users can send e-mail to this address to request administrative assistance – for example, to recover a forgotten password. The adminEmail address should match the e-mail of the Presto System Admin account. Using a correctly constructed pseudo e-mail address is recommended.

- **NotificationEmail** (required): This address is used as the "From" address for notification messages from *Presto*; for example, when *Presto* needs to send e-mail to all subscribers to notify of a server shutdown. This e-mail address is also the "From" address for all e-mail Alert notifications. The notificationEmail address should be specific to the company. Needs to be set up by IT for sending out e-mails that do not require replies (like Alerts). Using a correctly constructed pseudo e-mail address is recommended.

- **smtpClientEnableSsl** (optional): Set this to "true" if you want to encrypt the connection using SSL or "false" if you do not want to use SSL. The default is false. (Introduced in v3.5 Service Pack 1 – Do not use this setting unless you have SP1 or later.)

Important! You must restart Inmagic Presto Services after modifying e-mail settings in Web.config. You may also need to restart the "IIS Admin Service" in the system's **Administrative Tools>Services** dialog.

Troubleshooting E-mail Server Issues

If you are having e-mail server issues, the first thing you should do is restart Inmagic Presto Services. Any changes you have made to e-mail settings in Web.config will not take effect until you do so. Other commonly reported issues are addressed below. Additional information may be available in the Inmagic Presto Knowledgebase.

The following message appears when clicking Send e-mail:

```
The following error occurred when trying to send as e-mail the selected content(s).
Please contact your system administrator for further assistance.

No valid E-mail Server specified.
```

This typically indicates a firewall or security issue. To troubleshoot this problem:

- If you have virus-scanning software installed on the *Presto* Web Server, it is recommended that you add the Presto Services executable (**Inmagic.Mercury.Services.PrestoServices.exe**) to the excluded processes list for blocked processes on your e-mail port.

- Check with the IT department to make sure that the server is up and is able to send e-mail.

Emails are not being sent

To troubleshoot this problem:

- You must always restart Inmagic Presto Services after modifying e-mail settings in Web.config.

- Verify that Presto Services are running properly (choose **Configure>Administrator Tool**s, click **Monitor Services**, and verify that the table appears and that the timestamps are recent).

- Confirm that the prestoServicesClient webApplicationPath in the Inmagic.Mercury.Services.PrestoServices.exe.config file is correct. For example:

```
<prestoServicesClient webApplicationPath="C:\Customers\App\Inmagic\Presto\WebApps\Mercury\" />
```

- Confirm that the specified SMTP server is working.

- Confirm that your SMTP server allows relay from the *Presto* application server. By default, most SMTP servers prevent SMTP relay. The way you do this depends on your particular implementation (for example, whether the SMTP server is on the local machine or remote or IIS or another third-party tool). On **IIS**, right-click the SMTP default server, select **Properties**, select the **Access** tab, click the **Relay** button to see a list of machines, then **Add** the IP address of the application or database server.

- Confirm that the AdminEmail and NotificationEmail addresses specified in Web.config are valid e-mail addresses.

Archiving Configuration

Presto can be configured to deactivate and/or delete records that exceed a days-old limit. Configuration is on a per-Content Type basis. Records may need to be archived for several reasons:

- They are time-based and expire after a specific period, such as material from a wire feed service that must be removed after 30 days because of contractual obligations.

- They are "hot topics" that become less relevant as time passes.

- They are produced continually (such as Event records) and the oldest ones must be removed to prevent unchecked growth of the database.

Archiving is implemented by the ArchivingProcessor (worker), which wakes up every 300 seconds (or other interval defined by its ServiceWait), and checks the ArchiveJobSettings and ArchiveJobTime (Application Settings). Note that archiving only occurs once a day. For more information, see *Inmagic Presto Services*.

Whenever a Content Type is added, deleted, or renamed, you should review and update the archiving settings. Any changes you make will take effect after Inmagic Presto Services is restarted and the specified **ArchiveJobTime** occurs.

Error Logging

If there are any errors in the archiving configuration, such as an incorrectly identified Content Type, the errors will be recorded in the Errors.log file the next time archiving occurs. To ensure that archiving information is written to the Event Log, choose **Configure>Events** and enable the "Data archived" System Event. To see the event log, choose **Reports>View Event Log**.

To configure archiving

1. Choose **Configure>Administrator Tools>Application Settings**.

2. Locate the **PrestoServices** section and define the **Service Wait** for the ArchivingProcessor. This determines how often the worker will check the ArchiveSettings.

3. Locate the **Archive** section and define the **ArchiveJobTime** to specify what time archiving will occur daily.

4. For **ArchiveJobSettings**, add entries between the `<ArchivingSettings>` `</ArchivingSettings>` tags to determine how archiving will occur. Typical settings are shown below:

```
<ArchivingSettings>
    <Setting Name="Sample Web Link" Type="UserContentType" Action="Deactivate" AgeInDays="30"/>
    <Setting Name="Sample Web Link" Type="UserContentType" Action="Delete" AgeInDays="60"/>
    <Setting Name="Local News Feed" Type="UserContentType" Action="Delete" AgeInDays="14"/>
    <Setting Name="Job" Type="SystemContentType" Action="Delete" AgeInDays="180"/>
    <Setting Name="User Login/Logout Events" Type="UserLoginEvents" Action="Delete" AgeInDays="180"/>
    <Setting Name="System Events" Type="JobEvents" Action="Delete" AgeInDays="180"/>
    <Setting Name="Content Events" Type="ContentEvents" Action="Delete" AgeInDays="180"/>
    <Setting Name="Collection Events" Type="ContentCollectionEvents" Action="Delete" AgeInDays="180"/>
    <Setting Name="User Profile Events" Type="UserProfileEvents" Action="Delete" AgeInDays="180"/>
    <Setting Name="Content Type Events" Type="ContentTypeEvents" Action="Delete" AgeInDays="180"/>
    <Setting Name="Blog Events" Type="BlogEvents" Action="Delete" AgeInDays="180"/>
</ArchivingSettings>
```

The Setting attributes are explained below. Save the Application Settings when you are done.

Type

The **Type** is required to determine what should be archived (UserContentType, BlogEvents, etc.).

Name

The **Name** is required if you specify Type="UserContentType", and is optional in all other cases. The **Name** can be either a user-defined Content Type or one of the Event or Job names shown below.

- Name="<any-user-defined-content-type>" Type="UserContentType"
- Name="Job" Type="SystemContentType"
- Name="User Login/Logout Events" Type="UserLoginEvents"
- Name="System Events" Type="JobEvents"
- Name="Content Events" Type="ContentEvents"
- Name="Collection Events" Type="ContentCollectionEvents"
- Name="User Profile Events" Type="UserProfileEvents"
- Name="Content Type Events" Type="ContentTypeEvents"
- Name="Blog Events" Type="BlogEvents"

Action

The Action can be one of the following:

- **Delete**: Deleting records permanently removes them from the database. You can delete any user Content Type or System Content Type record. This keeps the database from getting too big. The record, job, or event record will no longer be available in *Presto*.

- **Deactivate**: Deactivating records lets you retain them for a period of time and is typically done as an intermediate step prior to deletion. You can deactivate any user Content Type record. Deactivated records will be retained for the specified amount of time. *Presto* users who are allowed to view Inactive content will see the records grayed out and identified by the word "INACTIVE" in red.

To specify both actions for a single content type, include separate lines for each action. For example:

```
<Setting Name="Sample Web Link" Type="UserContentType" Action="Deactivate" AgeInDays="30"/>
<Setting Name="Sample Web Link" Type="UserContentType" Action="Delete" AgeInDays="60"/>
```

AgeInDays

Specify the number of days after which a record will be deleted or deactivated. You can specify any integer greater than 0. The criterion is *days old* based on *DateModified*.

Inmagic Presto Services

Inmagic Presto Services is a Windows service that provides background processing for the following *Presto* operations:

- Job execution for import and export

- Raising *Presto* alerts

- Email queuing and archiving

Inmagic Presto Services should always be running in order to support the Inmagic *Presto* Web application. By default, Inmagic Presto Services starts automatically when the server boots up. As with any Windows service, it is controlled through the Microsoft Windows Services Control Panel.
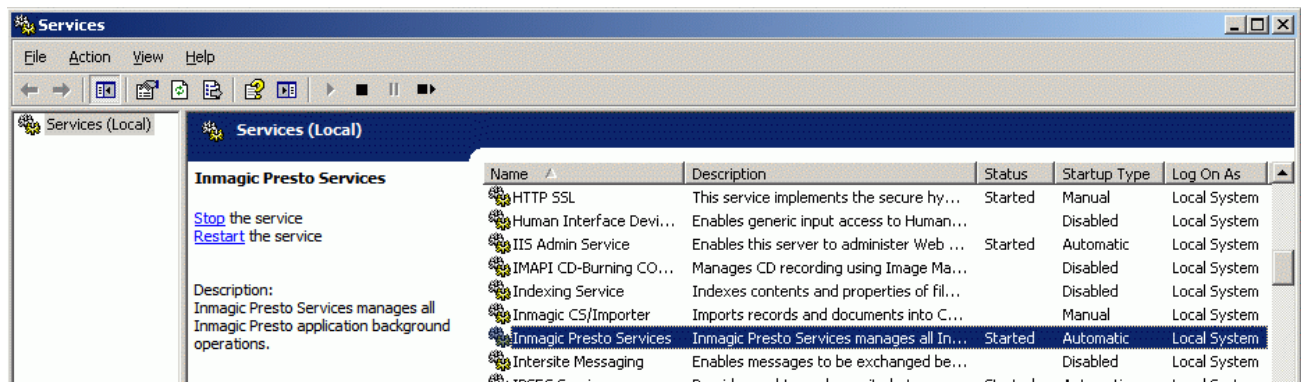
Configuring Inmagic Presto Services

Follow the steps below to configure Inmagic Presto Services.

Step 1. Configure the *Presto* Web Application to use Inmagic Presto Services.

A "Shared Services" file called `Inmagic.Mercury.Services.PrestoServices.exe.config` is installed with *Presto*. Under normal circumstances, you should not have to edit this file. However, in actual practice it is important to ensure that this file has been configured properly.

1. Open the **PrestoServices.exe.config** file in a UTF-8 Unicode editor (such as Windows Notepad):

   C:\Program Files\Inmagic\Presto\Presto Services\Inmagic.Mercury.Services.PrestoServices.exe.config

2. Ensure that the path to the installed *Presto* Web application is correct.

   The <presto.ClientApplicationSettingsSection> of this file includes one prestoServicesClient element for each *Presto* Web application that is using the Inmagic Presto Services. For example, if there are two installed instances (**Presto** and **PrestoTest**), the section might look like this:

```
<presto.ClientApplicationSettingsSection>
   <prestoClients>
    <prestoServicesClient webApplicationPath="C:\Program Files\Inmagic\Presto\WebApps\Mercury\" />
    <prestoServicesClient webApplicationPath="C:\Program Files\Inmagic\PrestoTest\WebApps\Mercury\" />
   </prestoClients>
```

```
</presto.ClientApplicationSettingsSection>
```

3. Save and close the file.

4. **Important!** Configuration changes do not take effect until the Inmagic Presto Services Windows service is restarted. *However, before you restart the service, please read this entire note.* Stopping or restarting services can be disruptive to *Presto* applications using those services. If possible, avoid restarting Inmagic Presto Services during peak hours and ensure that no jobs are being processed (look at **Configure >Administrator Tools>Monitor Jobs** to see which jobs are in process). Ideally, you should also stop the *Presto* Web site in IIS before restarting Inmagic Presto Services to ensure that no additional jobs are submitted.

5. Restart Inmagic Presto Services, using the Microsoft Windows Services Control Panel.

Step 2. Adjust the PrestoServices Worker Configuration

Use the *Presto* Application Settings to adjust the worker configuration:

1. Log in to *Presto* with Administrator rights.

2. Choose **Configure>Administrator Tools>Application Settings**.

3. Locate the **PrestoServices** section.
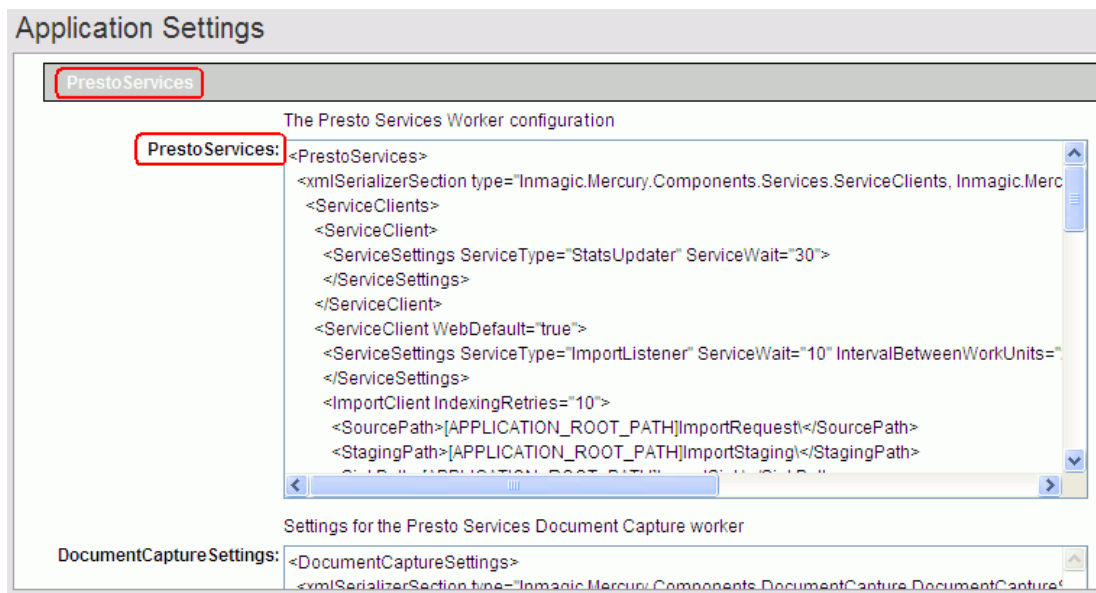
This section lists the configuration for each client worker. One worker is provided for each long-running operation that is not implemented directly in the *Presto* Web application. Note that the Shared Services configuration file (Inmagic.Mercury.Services.PrestoServices.exe.config) identifies the *Presto* instance that is using Inmagic Presto Services.

4. Follow the instructions below to edit the content in the field labeled **PrestoServices**. You can either edit the XML in-place or copy it into a UTF-8 Unicode editor (such as Windows Notepad), make your changes, then copy it back into the field. Remember that XML is case-sensitive.

5. Click **Save** to save your changes (or click **Cancel** to close without saving).

PrestoServices Worker Configuration

The default PrestoServices worker configuration defined in Application Settings is shown below. Each item is described later in this section. Note that the placeholder [APPLICATION_ROOT_PATH] for the ImportListener and ExportListener is automatically replaced with the correct physical path when Inmagic Presto Services starts; the appearance of this placeholder is normal and correct and should *not* be modified.

```
<PrestoServices>
  <xmlSerializerSection type="Inmagic.Mercury.Components.Services.ServiceClients, Inmagic.Mercury.Components,
Version=1.0.0.0, Culture=neutral, PublicKeyToken=null">
    <ServiceClients>
     <ServiceClient>
      <ServiceSettings ServiceType="StatsUpdater" ServiceWait="30">
      </ServiceSettings>
     </ServiceClient>
     <ServiceClient WebDefault="true">
      <ServiceSettings ServiceType="ImportListener" ServiceWait="10" IntervalBetweenWorkUnits="250"
ItemsPerWorkUnit="10">
      </ServiceSettings>
      <ImportClient>
       <SourcePath>[APPLICATION_ROOT_PATH]ImportRequest\</SourcePath>
       <StagingPath>[APPLICATION_ROOT_PATH]ImportStaging\</StagingPath>
       <SinkPath>[APPLICATION_ROOT_PATH]ImportSink\</SinkPath>
       <ExceptionFilePath>[APPLICATION_ROOT_PATH]ImportException</ExceptionFilePath>
      </ImportClient>
     </ServiceClient>
     <ServiceClient>
      <ServiceSettings ServiceType="ExportListener" ServiceWait="30" IntervalBetweenWorkUnits="250"
ItemsPerWorkUnit="10">
      </ServiceSettings>
      <ExportClient>
       <SourcePath>[APPLICATION_ROOT_PATH]ExportRequest\</SourcePath>
       <SinkPath>[APPLICATION_ROOT_PATH]ExportSink\</SinkPath>
       <ResultPath>[APPLICATION_ROOT_PATH]Export\</ResultPath>
      </ExportClient>
     </ServiceClient>
     <ServiceClient>
      <ServiceSettings ServiceType="AlertListener" ServiceWait="30" IntervalBetweenWorkUnits="250"
ItemsPerWorkUnit="10">
      </ServiceSettings>
     </ServiceClient>
     <ServiceClient>
      <ServiceSettings ServiceType="JobListener" ServiceWait="10" IntervalBetweenWorkUnits="250"
ItemsPerWorkUnit="10">
      </ServiceSettings>
     </ServiceClient>
     <ServiceClient>
      <ServiceSettings ServiceType="EmailQueueMessageProcessor" ServiceWait="5" IntervalBetweenWorkUnits="125"
ItemsPerWorkUnit="10">
      </ServiceSettings>
     </ServiceClient>
       <ServiceClient>
      <ServiceSettings ServiceType="ArchivingProcessor" ServiceWait="30">
      </ServiceSettings>
     </ServiceClient>
     <ServiceClient>
      <ServiceSettings ServiceType="SqlFullTextOptimizeProcessor" ServiceWait="30">
      </ServiceSettings>
     </ServiceClient>
    </ServiceClients>
  </xmlSerializerSection>
</PrestoServices>
```

Service Types (Workers)

The following Service Types (also called *workers*) are defined in the **PrestoServices** section of Application Settings (see *PrestoServices Worker Configuration*).

- **StatsUpdater**: A utility worker that updates the ServiceWorkerStats table for all other workers. This avoids contention for pooled database connections. [Only 1 per configuration, required.]

- **ImportListener**: Listens for import files in an ImportClient SourcePath folder and creates import jobs for each file that appears. The ImportListener always processes the oldest file first, which ensures that the most recent changes in other products (Genie, Blogs, etc.) are added last. [1 or more per configuration, required.] See the *Import and Export Configuration Guide* for more information.

- **ExportListener**: Listens for export files in an ExportClient SourcePath folder; creates export jobs for each export request file that appears. [1 or more per configuration, required.] See the *Import and Export Configuration Guide* for more information.

- **AlertListener**: Listens for any Query Alerts which are due to be processed. Places each ready Query Alert in the Alert queue to be processed. [Only 1 per configuration, required.]

- **JobListener**: Listens for newly created jobs. Places each new job in the Job Processing queue to be processed. [Only 1 per configuration, required.]

- **EmailQueueMessageProcessor**: Retrieves queued e-mail messages and sends them using the SMTP server configured for this *Presto* instance. [Only 1 per configuration, required.]

- **ArchivingProcessor**. Deletes or activates outdated content items and/or deletes outdated events/jobs. By default, it wakes up every 300 seconds (ServiceWait) and checks the Archive JobTime and ArchiveJobSettings (in the **Archive** section of Application Settings), to see when and how it should run. It only runs once a day. [Only 1 per configuration, required.]

- **SqlFullTextOptimizeProcessor:** A processor that runs once a day and executes a stored procedure which optimizes the Sql Full Text Catalog to improve relevance ranking behavior. By default, it wakes up every 30 seconds (ServiceWait) and checks the SqlOptimizeProcessorJobTime (in the Search section of Application Settings) to see when it should run. [Only 1 per configuration, optional but present by default.]

- **DeferredIndexingQueueMessageProcessor**: A queue message processor that handles queued requests to index the content of large binary files. If this worker is added to the configuration, then as many instances as are specified in the DeferredIndexingSettings entry in Application Settings are created. Multiple instances allow simultaneous indexing of multiple files. Configuring this worker can improve the speed of large Import operations on servers that have relatively large hardware resources available - that is, multiple processors and fast disk drives such as SAN. [Optional.]

- **DocumentCaptureListener**: Can optionally be added as the last entry in the <PrestoServices> section, but is not included in the default configuration. [Only 1 per configuration, optional.] **Note**: After defining a DocumentCaptureListener in the **PrestoServices** section, you must configure the DocumentCaptureSettings to determine how the binary files will be loaded. This is explained in the *Document Capture Configuration Guide*.

- **FileCaptureListener**: Can optionally be added as the last entry in the <PrestoServices> section, but is not included in the default configuration. A FileCaptureListener is more powerful than a DocumentCaptureListener. [Only 1 per configuration, optional.] **Note**: After defining a FileCaptureListener in the **PrestoServices** section, you must configure the FileCaptureSettings to determine how the binary files will be loaded. This is explained in the *Drag & Drop Files Configuration Guide*.

General Service Settings

Each Service Type (worker) has the following attributes that can be adjusted:

- **ServiceType**: The type of work that is performed by this worker (ImportListener, ExportListener, JobListener, etc.).

- **ServiceWait**: The number of **seconds** the worker sleeps before checking for work to do.

- **IntervalBetweenWorkUnits**: The number of **milliseconds** to sleep between processing work units.

- **ItemsPerWorkUnit**: The number of items to process per work unit.

Help Configuration

The **Help** section of Application Settings (**Configure>Administrator Tools>Application Settings**) includes several settings that identify the components required for *Presto* online help. In general, you should not adjust these settings. For more information, see the *Glossary of Application Settings*.

Customizing Help Topics

To customize individual help topics, follow the instructions below.

Administrators can add a custom link to any help topic, to provide access to additional resources. For example, add a link that points to an HTML page, a PDF file, or an external URL. You can specify whether the custom content should open in the Help window or a new window. To make these changes, modify the supplied JavaScript text file **custom_linklist.JS**, as explained below.

To modify a *Presto* help topic by adding a custom link:

1. (If you are planning to direct users to an external link that contains help files, such as a corporate Web site or intranet, you can skip this step, and go to step 2.)
   If you are going to direct users to customized content, create the content and place it in the `CustomHelp` subfolder of the *Presto* installation folder. Be sure to include any referenced images or other supporting files. File names must not include spaces. For example:

   ```
   C:\Program Files\Inmagic\Presto\WebApps\Mercury\WebHelp\CustomHelp\mynewpage.htm
   ```

   ```
   C:\Program Files\Inmagic\Presto\WebApps\Mercury\WebHelp\CustomHelp\mylogo.gif
   ```

2. Make a copy of the file **custom_linklist.JS**, so you have a backup in case you need to revert to it. This file is located in the `WebHelp` subfolder of the *Presto* installation folder. For example:

   ```
   C:\Program Files\Inmagic\Presto\WebApps\Mercury\WebHelp\custom_linklist.JS
   ```

3. Open the **custom_linklist.JS** file using a text editor and follow the instructions. Basically, you will edit this file to include a pointer to the new content in the `CustomHelp` folder (such as mynewpage.htm) or other target (such as http://www.mywebsite.com).

4. Back up the modified **custom_linklist.JS** file and move it to another location, in case the custom_linklist.JS file gets overwritten in a future release of the application. You will want to be able to copy content from your edited version of the JavaScript file to the original custom_linklist.JS file that is installed with a software upgrade, if necessary. Also, back up all of the files in the CustomHelp subfolder and keep them in a safe place.

Be sure to test the modified help before making it available to users.