

# Configuration File Changes

Sitecore Experience Platform 8.1 Update 1



**sitecore**<sup>®</sup>  
Own the experience<sup>™</sup>

## 1.1 Config file changes in Sitecore 8.1 rev. 151207 (Update-1) compared to Sitecore 8.1 rev. 151003 (Initial Release).

This update requires you to make changes to the following files:

- `Web.config`
- `App_Config\FieldTypes.config`
- `App_Config\Sitecore.config`
- `App_Config\Include\Sitecore.ContentSearch.Analytics.config`
- `App_Config\Include\Sitecore.ContentSearch.config`
- `App_Config\Include\Sitecore.ExperienceEditor.config`
- `App_Config\Include\Sitecore.ExperienceEditor.Speak.Requests.config`
- `App_Config\Include\Sitecore.PathAnalyzer.config`
- `App_Config\Include\ContentTesting\Sitecore.ContentTesting.config`
- `App_Config\Include\FXM\Sitecore.FXM.config`
- `App_Config\Include\ListManagement\Sitecore.ListManagement.config`
- `App_Config\Include\Social\Sitecore.Social.ExperienceProfile.config`
- `App_Config\Include\XdbCloud\Sitecore.Cloud.Xdb.config.disabled`
- `App_Config\Include\XdbCloud\Sitecore.ContentSearch.Cloud.Index.Analytics.config.disabled`
- `App_Config\Include\Z.SwitchMasterToWeb\SwitchMasterToWeb.config.example`

Depending on the search provider that you are using, you should also update the following files:

### Lucene

- `App_Config\Include\Sitecore.ContentSearch.Lucene.DefaultIndexConfiguration.config`
- `App_Config\Include\ListManagement\Sitecore.ListManagement.Lucene.IndexConfiguration.config`

### Solr

- `App_Config\Include\Sitecore.ContentSearch.Solr.DefaultIndexConfiguration.config.example`
- `App_Config\Include\ListManagement\Sitecore.ListManagement.Solr.IndexConfiguration.config`

In this update, the following configuration file has been added:

- `App_Config\Include\Sitecore.DefaultLanguage.config`

You can download the clean configuration files for Sitecore 8.1 rev. 151207 (Update-1):

- `app_config_files_81rev151207.zip`
- `Web.config_81rev151207.zip`

To download these files, on the [Sitecore Experience Platform 8.1 Update-1](#) page, in the **Upgrade options** section, click *Config files for upgrade*.

# Configuration File Changes

## Tip

Most customers have only applied a small number of changes to the `Web.config` file. In this case, we recommend that you download the clean config file and re-apply any changes that you've made to save time and to reduce the risk of making mistakes when adding the new settings, pipelines, and so on to the file.

### 1.1.1 Changes to Web.config

To upgrade your existing config file to Sitecore 8.1 rev. 151207 (Update-1), you must make the following changes to the `Web.config` file:

1. In the `<modules>` section, after the `<add type="Sitecore.Web.HttpModule, Sitecore.Kernel" name="SitecoreHttpModuleExtensions" />` node, add the following module:

```
<add type="Sitecore.Web.HttpModuleDisabler, Sitecore.Kernel"
name="SitecoreHttpModuleDisabler" />
```

2. Before of the `<pages>` section, add the following note:

```
<!-- If requestValidationMode attribute of httpRuntime node is set to 2.0,
Sitecore requires this setting to be set to "false" for Sitecore client to
work and it shouldn't be changed.
You can however set ValidateRequest attribute in the @Page directive to
"true" for your layout .aspx files.
-->
```

### 1.1.2 Changes to FieldTypes.config

1. After the `<fieldType name="Name Value List" type="Sitecore.Data.Fields.NameValueListField, Sitecore.Kernel" />` node, add the following node:

```
<fieldType name="Name Lookup Value List"
type="Sitecore.Data.Fields.NameValueListField, Sitecore.Kernel" />
```

### 1.1.3 Changes to Sitecore.config

1. In the `<renderField>` section, after the `"Sitecore.Pipelines.RenderField.GetDocxFieldValue"` processor, add the following nodes:

```
<processor type="Sitecore.Pipelines.RenderField.GetIntegerFieldValue,
Sitecore.Kernel" />
<processor type="Sitecore.Pipelines.RenderField.GetNumberFieldValue,
Sitecore.Kernel" />
```

2. In the `<publishItem>` section, move the `"Sitecore.Publishing.Pipelines.PublishItem.MoveItems"` processor and insert it after the `"Sitecore.Publishing.Pipelines.PublishItem.DetermineAction"` processor.
3. Before of the `<counters>` section, add the following note:

```
<!--
List of the system counters whose values should be written to the Sitecore
log file by the Sitecore health monitor

Supported attributes:
category: The name of the performance counter category that this
performance counter is associated with.
name: The name of the performance counter that has an entry in the
performance monitor.
instanceType: The name of the performance counter category.
valueType: [raw|calculated]. This attribute is optional. If set to
'raw', the counter outputs the raw value from the performance counter. If set to 'calculated',
the counter outputs the calculated value. If no attribute is set, the counter outputs the raw
value from the performance counter.
-->
```

# Sitecore Experience Platform 8.1 Update 1

4. In the beginning of the `<getMediaStream>` section, add the following processor:

```
<processor type="Sitecore.Resources.Media.SaveColorProfileProcessor, Sitecore.Kernel" />
```

5. At the end of the `<getMediaStream>` section, add the following processor:

```
<processor type="Sitecore.Resources.Media.RestoreColorProfileProcessor, Sitecore.Kernel" />
```

6. After the `<setting name="ItemCloning.NonInheritedFields" value="" />` setting, add the following nodes:

```
<!-- ITEM CLONING - INHERIT WORKFLOW DATA
Specify whether you want the workflow and workflow state fields to be
inherited by clones from their source item.
Note: When you change the setting value, clones move in or out the
workflow, which may affect their behavior, for example, on publishing.
Default value: false
-->
<setting name="ItemCloning.InheritWorkflowData" value="false" />
<!-- ITEM CLONING - FORCE UPDATE
Specify whether clones should be updated automatically when:
- a new version is added to the original item.
- a new language is added to the original item.
- a new sub-item is added to the original item.
Note: If true, all the clones of the original item are updated
automatically.
Default value: false
-->
<setting name="ItemCloning.ForceUpdate" value="false" />
```

7. After the `<setting name="LicenseFile" value="$(dataFolder)/license.xml" />` setting, add the following node:

```
<!-- LINK DATABASE - MAXIMUM BATCH SIZE
This setting specifies the maximum number of records that Compact
method can process in a single iteration.
If you set this value too low, it can cause excessive amounts of
network traffic and may have a significant impact
on system throughput.
If you set this value too high, it may affect tasks that require
communication with SQL Server. For example, it may affect the size of transactions
sent to SQL Server and this might require you to set higher 'lock
timeout' values as it takes longer to process large batches.
Default: 1000
-->
<setting name="LinkDatabase.MaximumBatchSize" value="1000" />
```

8. After the `<setting name="Media.FileFolder" value="/App_Data/MediaFiles" />` setting, add the following setting:

```
<!-- MEDIA - FILE SYSTEM WATCHER - BUFFER SIZE
The size (in bytes) of the internal buffer used by the file system
watchers that monitor the file system change
notifications and raise events when a directory, or file in a directory,
changes.
If you get log entries about internal buffer overflow, increasing the
value of this setting may solve this problem.
The minimum size is 4096 (4 KB).
The maximum size is 65536 (64 KB).
Be aware that increasing this value could hurt performance. See
http://msdn.microsoft.com/en-us/library/system.io.filesystemwatcher.internalbuffersize%28v=vs.110%29.aspx.
Default value: 8192
-->
<setting name="Media.FileSystemWatcher.BufferSize" value="8192" />
```

9. After the `<setting name="Media.WhitespaceReplacement" value=" " />` setting, add the following setting:

```
<!-- REMOVE UNUSED BLOBS ON SAVE
Specifies whether unused blobs should be deleted from an item when it is
saved. This setting only checks items that contain blob fields.
To improve performance during a bulk upload of media files, set this
setting to "false".
-->
```

# Configuration File Changes

periodically. When this setting is set to "false", you must run the CleanupDatabase task

```
Default value: true.
-->
<setting name="RemoveUnusedBlobsOnSave" value="true" />
```

10. After the `<setting name="MediaResponse.SlidingExpiration" value="" />` setting, add the following setting:

```
<!-- MEDIA RESPONSE - VARY HEADER
Specifies the value for the Vary HTTP header to use in media response
headers.
Set it to "" to omit this header.
Default value: ""
-->
<setting name="MediaResponse.VaryHeader" value="" />
```

11. After the `<setting name="ViewStateStore" value="Sitecore.Web.UI.Sheer.FileViewStateStore, Sitecore.Kernel" />` setting, add the following setting:

```
<!-- WEB EDIT JQuery UI STYLESHEET
The stylesheet to include in the page editor in WebEdit mode.
Default value:
/sitecore/shell/Controls/Lib/jQuery/jQueryUI/1.10.3/smoothness/jquery-ui-1.10.3.min.css
-->
<setting name="WebEdit.JQueryUIStylesheet"
value="/sitecore/shell/Controls/Lib/jQuery/jQueryUI/1.10.3/smoothness/jquery-ui-
1.10.3.min.css" />
```

12. At the end of the `<settings>` section, add the following node:

```
<!-- HOST NAME VALIDATION
Regular expression for validating Host Name
-->
<setting name="HostNameValidationPattern" value="^[a-zA-Z0-9]+([a-zA-Z0-9-
.]*)([a-zA-Z0-9]+)$" />
```

## 1.1.4 Changes to App\_Config\Include\Sitecore.ContentSearch.Analytics.config

1. In the beginning of the `<settings>` section, add:

```
<!-- CONTENT SEARCH - ANALYTICS - DEFAULT LANGUAGE
Defines the default language that is used for indexing and searching
analytics data.
-->
<setting name="ContentSearch.Analytics.DefaultLanguage" value="en"/>
```

## 1.1.5 Changes to App\_Config\Include\Sitecore.ContentSearch.config

1. Change this comment:

```
<!-- PARALLEL INDEXING MAX THREAD LIMIT
This setting allows you to limit the number of threads used for indexing
operations when parallel indexing is enabled.
If the value is set to 0, there is no limit to the number of threads.

Default value: 3
```

To:

```
<!-- PARALLEL INDEXING MAX THREAD LIMIT
This setting allows you to limit the number of threads used for indexing
operations when parallel indexing is enabled.
If the value is set to 0, there is no limit to the number of threads.

Default value: 3

Note: Consider to change the ContentSearch.IndexUpdate.BatchSize value
while changing this setting value. The larger MaxThreadLimit value, the larger
BatchSize value should be to prevent a performance penalty. BatchSize
value should not be less than ContentSearch.ParallelIndexing.MaxThreadLimit * 100.
```

```
-->
```

2. After the "ContentSearch.Indexing.DisableDatabaseCaches" setting, add the following node:

```
<!-- INDEX UPDATE BATCH SIZE
    Determines the amount of items that should be processed during the index
    update within a batch.
    Default value: 300
-->
<setting name="ContentSearch.IndexUpdate.BatchSize" value="300" />
```

## 1.1.6 Changes to App\_Config\Include\Sitecore.ExperienceEditor.config

1. In the <pipelines> section, remove the following node:

```
<renderField>
  <processor
type="Sitecore.ExperienceEditor.Pipelines.RenderField.EditAllVersions,
Sitecore.ExperienceEditor"
patch:before="processor[@type='Sitecore.Pipelines.RenderField.RenderWebEditing,
Sitecore.Kernel']" />
</renderField>
```

2. At the end of the <httpRequestBegin> pipeline, add:

```
<processor
type="Sitecore.ExperienceEditor.Pipelines.HttpRequest.ResolveContentLanguage,
Sitecore.ExperienceEditor"
patch:after="processor[@type='Sitecore.Pipelines.HttpRequest.ItemResolver, Sitecore.Kernel']"
/>
```

3. At the end of the <getPageEditorNotifications> pipeline, add:

```
<processor
type="Sitecore.Pipelines.GetPageEditorNotifications.GetLockingNotification,
Sitecore.ExperienceEditor" />
<processor
type="Sitecore.Pipelines.GetPageEditorNotifications.GetWriteAccessDeniedNotification,
Sitecore.ExperienceEditor" />
```

4. After the "WebEdit.ContentEditorStylesheet" setting, add the following node:

```
<!-- WEB EDIT JQuery UI STYLESHEET
    The stylesheet to include in the page editor in WebEdit mode.
    Default value:
/sitecore/shell/Controls/Lib/jquery/jqueryUI/1.10.3/smoothness/jquery-ui-1.10.3.min.css
-->
<setting name="WebEdit.JQueryUIStylesheet"
value="/sitecore/shell/Controls/Lib/jquery/jqueryUI/1.10.3/smoothness/jquery-ui-
1.10.3.min.css"/>
```

## 1.1.7 Changes to App\_Config\Include\Sitecore.ExperienceEditor.Speak.Requests.config

1. Before this node:

```
<request name="ExperienceEditor.Save.HasWritePermission"
type="Sitecore.ExperienceEditor.Speak.Ribbon.Requests.SaveItem.HasWritePermission,
Sitecore.ExperienceEditor.Speak.Ribbon"/>
```

Add the following comment:

```
<!-- NOTE: This request is obsolete and will be removed in the next product
version. -->
```

## 1.1.8 Changes to App\_Config\Include\Sitecore.PathAnalyzer.config

1. Change this setting:

```
<!-- PATH ANALYZER - CACHING - TREE CACHE EXPIRATION
    Sets the absolute expiration on cached tree data.
    A value of 00:00:00 disables automatic expiration.
```

# Configuration File Changes

```
        Default value: 00:00:30 (30 seconds)
    -->
    <setting name="PathAnalyzer.Caching.TreeCacheExpiration" value="00:00:30" />
```

To:

```
<!-- PATH ANALYZER - CACHING - TREE CACHE EXPIRATION
    Sets the absolute expiration on cached tree data.
    A value of 00:00:00 disables automatic expiration.

    Default value: 00:15:00 (15 minutes)
    -->
    <setting name="PathAnalyzer.Caching.TreeCacheExpiration" value="00:15:00" />
```

## 1.1.9 Changes to App\_Config\Include\Sitecore.ContentSearch.Lucene.DefaultIndexConfiguration.config

1. In the beginning of the <contentSearch> section, add the following section:

```
<!-- REINDEX ON PUBLISH END SINGLE INSTANCE
    This strategy is triggered on publish:end and uses the EventQueue to
    incrementally rebuild the index. The strategy contains all indexes which belong to it.
    -->
    <onPublishEndAsyncSingleInstance
type="Sitecore.ContentSearch.Maintenance.Strategies.OnPublishEndAsynchronousSingleInstanceStrategy, Sitecore.ContentSearch" singleInstance="true">
    <param desc="database">web</param>
    <!-- Whether or not a full index rebuild should be triggered when the number
of items in the EventQueue exceeds the number specified
    in ContentSearch.FullRebuildItemCountThreshold. -->
    <CheckForThreshold>true</CheckForThreshold>
    </onPublishEndAsyncSingleInstance>
```

2. In the beginning of the <fieldReaders> section, add the following node:

```
<param desc="id">defaultFieldReaderMap</param>
```

## 1.1.10 Changes to App\_Config\Include\Sitecore.ContentSearch.Solr.DefaultIndexConfiguration.config.example

1. In the beginning of the <fieldReaders> section, add the following node:

```
<param desc="id">defaultFieldReaderMap</param>
```

## 1.1.11 Changes to App\_Config\Include\Z.SwitchMasterToWeb\SwitchMasterToWeb.config.example

1. Remove the following node:

```
<index id="sitecore_list_index">
    <patch:delete/>
</index>
```

## 1.1.12 Changes to App\_Config\Include\ContentTesting\Sitecore.ContentTesting.config

1. After the "ContentTesting.CommandRoutePrefix" setting, add the following node:

```
<!-- CONTENT TESTING - COMMAND ROUTE PREFIX MVC
    The prefix used for controller routes for mvc controllers.
    Default value: sitecore/shell/api/ct/mvc
    -->
```

# Sitecore Experience Platform 8.1 Update 1

```
</setting name="ContentTesting.CommandRoutePrefix.Mvc" value="sitecore/shell/ct/" />
```

2. After the "ContentTesting.PhantomJS.Timeout" setting, add the following nodes:

```
<!-- CONTENT TESTING - SCREENSHOT - VIEWPORT - DEFAULT HEIGHT
The height of the view port used to generate screenshots of pages if not
specified in the screenshot task.
Default value: 800
-->
<setting name="ContentTesting.Screenshot.ViewPort.DefaultHeight" value="800"/>

<!-- CONTENT TESTING - SCREENSHOT - VIEWPORT - DEFAULT WIDTH
The width of the view port used to generate screenshots of pages if not
specified in the screenshot task.
Default value: 1200
-->
<setting name="ContentTesting.Screenshot.ViewPort.DefaultWidth" value="1200"/>

<!-- CONTENT TESTING - SUGGESTED TEST INDEX NAME
The default suggested test index name.
Default value: sitecore_suggested_test_index
-->
<setting name="ContentTesting.SuggestedTestIndexName"
value="sitecore_suggested_test_index"/>

<!-- CONTENT TESTING - SUGGESTED TESTS - MAXIMUM
The maximum number of suggested tests to show.
Default value: 50
-->
<setting name="ContentTesting.SuggestedTests.Maximum" value="50"/>
```

3. After the "ContentTesting.TestingIndexName" setting, add the following nodes:

```
<!-- CONTENT TESTING - VISITOR - COMBINATION STORAGE
Specifies the storage mechanism to use for the visitors test combinations.
Must be one of ( xdbfacet | cookie )
Default value: xdbfacet
-->
<setting name="ContentTesting.Visitor.TestCombinationStorage" value="xdbfacet"/>
```

4. At the end of the <allowedControllers> section, add the following node:

```
<allowedController
desc="CT_TestSummaryController">Sitecore.ContentTesting.Requests.Controllers.TestSummary.Tests
ummaryController, Sitecore.ContentTesting</allowedController>
```

5. Before the <pipelines> section, add the following section:

```
<model>
  <elements>
    <element
interface="Sitecore.ContentTesting.Model.Entities.ITestCombinationsData,
Sitecore.ContentTesting.Model"
implementation="Sitecore.ContentTesting.Model.Entities.TestCombinationsData,
Sitecore.ContentTesting.Model"/>
  </elements>
  <entities>
    <contact>
      <facets>
        <facet name="TestCombinations"
contract="Sitecore.ContentTesting.Model.Entities.ITestCombinationsData,
Sitecore.ContentTesting.Model" />
      </facets>
    </contact>
  </entities>
</model>
```

6. In the <cancelTest> section, move the

"Sitecore.ContentTesting.Pipelines.CancelTest.ProcessTestWorkflow" processor and insert it after the

"Sitecore.ContentTesting.Pipelines.CancelTest.CheckCanCancelTest" processor.

7. Replace the <determineTestOwner> section, with:

```
<determineTestOwner>
```



# Configuration File Changes

```
        <processor
type="Sitecore.ContentTesting.Pipelines.DetermineTestOwner.GetFromContext,
Sitecore.ContentTesting"/>
        <!-- Enable this processor if you want to determine test owner by the workflow
states transition user.
        You just need to specify the transitional workflow states that are used
as from workflow states and
        the order of the transition history. -->
        <!--<processor
type="Sitecore.ContentTesting.Pipelines.DetermineTestOwner.GetFromWorkflowHistory,
Sitecore.ContentTesting">
        <TransitionalWorkflowStates hint="list:AddTransitionalWorkflowState">
        <WorkflowState>/sitecore/system/Workflows/Sample
Workflow/Draft</WorkflowState>
        </TransitionalWorkflowStates>
        <FirstTransition>true</FirstTransition>
        </processor>-->
        <!-- Enable this processor if you want to determine test owner by the [Updated
By] field of the host item.-->
        <!--<processor
type="Sitecore.ContentTesting.Pipelines.DetermineTestOwner.GetFromStatistics,
Sitecore.ContentTesting" />-->
        </determineTestOwner>
```

8. After the `<getRenderingRules>` node, add the following node:

```
        <!-- GET SCREENSHOT
        Generate and scale screenshots
        -->
        <getScreenshot>
        <processor
type="Sitecore.ContentTesting.Pipelines.GetScreenshot.GenerateFilename,
Sitecore.ContentTesting" />
        <processor
type="Sitecore.ContentTesting.Pipelines.GetScreenshot.CheckCachedImage,
Sitecore.ContentTesting" />
        <processor
type="Sitecore.ContentTesting.Pipelines.GetScreenshot.CheckDisabler, Sitecore.ContentTesting"
/>
        <processor
type="Sitecore.ContentTesting.Pipelines.GetScreenshot.GenerateScreenshotFile,
Sitecore.ContentTesting" />
        <processor type="Sitecore.ContentTesting.Pipelines.GetScreenshot.ScaleImage,
Sitecore.ContentTesting" />
        </getScreenshot>
```

9. In the `<getScreenShotForURL>` section, after the

```
        <processor
type="Sitecore.ContentTesting.Pipelines.GetScreenShotForURL.CheckDisabler,
Sitecore.ContentTesting" />
```

processor, add the following processor:

```
        <processor
type="Sitecore.ContentTesting.Pipelines.GetScreenShotForURL.PopulateUrlParameters,
Sitecore.ContentTesting" />
```

10. In the `<getTestToRun>` section, after the

```
        <processor
type="Sitecore.ContentTesting.Pipelines.GetTestToRun.GetFromVisualization,
Sitecore.ContentTesting"/>
```

processor, add the following processor:

```
        <processor
type="Sitecore.ContentTesting.Pipelines.GetTestToRun.EnsurePageTest,
Sitecore.ContentTesting"/>
```

11. In the `<getTestToRun>` section, replace the

```
        <processor
type="Sitecore.ContentTesting.Pipelines.GetTestToRun.GetFromVisualization,
Sitecore.ContentTesting"/>
```

# Sitecore Experience Platform 8.1 Update 1

processor with the following processor:

```
<processor
type="Sitecore.ContentTesting.Pipelines.GetTestToRun.EnsureCorrectLanguage,
Sitecore.ContentTesting">
  <excludeFields hint="list:AddExcludeVariableTemplateId">
    <pageLevel>{AB86861A-6030-46C5-B394-E8F99E8B87DB}</pageLevel>
  </excludeFields>
</processor>
```

12. In the beginning of <initialize> section, add the following processor:

```
<processor
type="Sitecore.ContentTesting.Pipelines.Initialize.RegisterContentTestingCommandRoute,
Sitecore.ContentTesting"
patch:before="processor[@type='Sitecore.Mvc.Pipelines.Loader.InitializeRoutes, Sitecore.Mvc']"
/>
```

13. In the <startPageTest> section, after the

```
<processor
type="Sitecore.ContentTesting.Pipelines.StartPageTest.DeployComponentTestDependencies,
Sitecore.ContentTesting" />
```

processor, add the following processor:

```
<processor type="Sitecore.ContentTesting.Pipelines.StartPageTest.ProtectTest,
Sitecore.ContentTesting" />
```

14. In the <startTest> section, after the

```
<processor
type="Sitecore.ContentTesting.Pipelines.StartTest.CreateTestDefinition,
Sitecore.ContentTesting" />
```

processor, add the following processor:

```
<processor type="Sitecore.ContentTesting.Pipelines.StartTest.ProtectTest,
Sitecore.ContentTesting" />
```

15. In the <sitecore.experienceeditor.speak.requests> section, remove this node:

```
<request name="Optimization.SuggestedTests.Count"
type="Sitecore.ContentTesting.Requests.ExperienceEditor.SuggestedTestsCountRequest,
Sitecore.ContentTesting" />
```

## 1.1.13 Changes to App\_Config\Include\FXM\Sitecore.FXM.config

1. At the end of the <saveUI> section, add the following processor:

```
<processor
type="Sitecore.FXM.Pipelines.SaveUI.ExcludeUpdatedElementReplacersFromSaveList, Sitecore.FXM"
patch:after="processor[@type='Sitecore.FXM.Pipelines.SaveUI.AddElementReplacersProcessor,
Sitecore.FXM']" />
```

## 1.1.14 Changes to App\_Config\Include\ListManagement\Sitecore.ListManagement.config

1. Before the <pipelines> section, add the following node:

```
<!-- HOOKS -->
<hooks>
  <hook
type="Sitecore.ListManagement.Analytics.Hooks.UnlockContactListsAgentLoader,
Sitecore.ListManagement.Analytics">
    <param name="agent"
type="Sitecore.ListManagement.Analytics.UnlockContactListsAgent,
Sitecore.ListManagement.Analytics">
      <param name="listManager" ref="contactListManager" />
      <param name="mapper"
type="Sitecore.ListManagement.ContentSearch.BatchIdMapper,
Sitecore.ListManagement.ContentSearch" />
    <param name="operationManager"
type="Sitecore.Analytics.Data.Bulk.Contact.ContactBulkUpdateManager, Sitecore.Analytics" />
```

# Configuration File Changes

```
</param>
  <param name="interval">00:00:10</param>
</hook>
</hooks>
```

2. In the beginning of "Sitecore.ListManagement.ContentSearch.Pipelines.AssociateContact.AssociateContacts" processor, add the following node:

```
<ContactRepository ref="contactRepository" />
```

3. In the beginning of "Sitecore.ListManagement.ContentSearch.Pipelines.RemoveContactAssociations.RemoveAssociationsFromIndex" processor, add the following node:

```
<ContactRepository ref="contactRepository" />
```

4. Remove the <scheduling> section.

## 1.1.15 Changes to App\_Config\Include\ListManagement\Sitecore.ListManagement.Lucene.IndexConfiguration.config

1. After the <field fieldName="name"> node, add the following node:

```
<field fieldName="description" storageType="YES" indexType="TOKENIZED"
vectorType="NO" boost="1f" type="System.String"
settingType="Sitecore.ContentSearch.LuceneProvider.LuceneSearchFieldConfiguration,
Sitecore.ContentSearch.LuceneProvider">
  <analyzer
type="Sitecore.ContentSearch.LuceneProvider.Analyzers.LowerCaseKeywordAnalyzer,
Sitecore.ContentSearch.LuceneProvider" />
</field>
```

2. After the <field fieldName="recipients"> node, add the following nodes:

```
<field fieldName="query" storageType="YES" indexType="TOKENIZED"
vectorType="NO" boost="1f" type="System.String"
settingType="Sitecore.ContentSearch.LuceneProvider.LuceneSearchFieldConfiguration,
Sitecore.ContentSearch.LuceneProvider">
  <analyzer
type="Sitecore.ContentSearch.LuceneProvider.Analyzers.LowerCaseKeywordAnalyzer,
Sitecore.ContentSearch.LuceneProvider" />
</field>
<field fieldName="source.includedlists" storageType="YES"
indexType="NOT_ANALYZED" vectorType="NO" boost="1f" type="System.String"
settingType="Sitecore.ContentSearch.LuceneProvider.LuceneSearchFieldConfiguration,
Sitecore.ContentSearch.LuceneProvider" />
<field fieldName="source.excludedlists" storageType="YES"
indexType="NOT_ANALYZED" vectorType="NO" boost="1f" type="System.String"
settingType="Sitecore.ContentSearch.LuceneProvider.LuceneSearchFieldConfiguration,
Sitecore.ContentSearch.LuceneProvider" />
<field fieldName="source.predefinedsourcetype" storageType="YES"
indexType="NOT_ANALYZED" vectorType="NO" boost="1f" type="System.String"
settingType="Sitecore.ContentSearch.LuceneProvider.LuceneSearchFieldConfiguration,
Sitecore.ContentSearch.LuceneProvider" />
<field fieldName="source.predefinedparameters" storageType="YES"
indexType="NOT_ANALYZED" vectorType="NO" boost="1f" type="System.String"
settingType="Sitecore.ContentSearch.LuceneProvider.LuceneSearchFieldConfiguration,
Sitecore.ContentSearch.LuceneProvider" />
<field fieldName="source.usealldatabase" storageType="YES"
indexType="NOT_ANALYZED" vectorType="NO" boost="1f" type="System.Boolean"
settingType="Sitecore.ContentSearch.LuceneProvider.LuceneSearchFieldConfiguration,
Sitecore.ContentSearch.LuceneProvider" />
```

3. In the beginning of the <fields hint="raw:AddComputedIndexField"> section, add the following nodes:

```
<field fieldName="source.includedlists"
itemFieldName="IncludedSources">Sitecore.ListManagement.ContentSearch.Fields.MappedIndexField,
Sitecore.ListManagement.ContentSearch</field>
```

```

        <field fieldName="source.excludedlists"
itemFieldName="ExcludedSources">Sitecore.ListManagement.ContentSearch.Fields.MappedIndexField,
Sitecore.ListManagement.ContentSearch</field>
        <field fieldName="source.predefinedsourcetype"
itemFieldName="PredefinedSourceType">Sitecore.ListManagement.ContentSearch.Fields.MappedIndexF
ield,Sitecore.ListManagement.ContentSearch</field>
        <field fieldName="source.predefinedparameters"
itemFieldName="PredefinedParameters">Sitecore.ListManagement.ContentSearch.Fields.MappedIndexF
ield,Sitecore.ListManagement.ContentSearch</field>
        <field fieldName="source.usealldatabase"
itemFieldName="UseAllDatabase">Sitecore.ListManagement.ContentSearch.Fields.MappedIndexField,S
itecore.ListManagement.ContentSearch</field>

```

#### 4. After the <fields> section, add the following section:

```

<!-- INDEX DOCUMENT TO PROPERTY MAPPER
Maintains a collection of all the possible Convertors for the provider.
-->
<indexDocumentPropertyMapper
type="Sitecore.ContentSearch.LuceneProvider.DefaultLuceneDocumentTypeMapper,
Sitecore.ContentSearch.LuceneProvider">
    <!-- OBJECT FACTORY
Constructs search result objects based on the type that is passed in
.GetQueryable<T>() and the rules defined in this section.
-->
    <objectFactory
type="Sitecore.ContentSearch.DefaultDocumentMapperObjectFactory, Sitecore.ContentSearch">
        <rules hint="list:AddRule">
            <rule fieldName="_template" comparison="Equal"
value="c0caf6988a424b669eaf7d442b46f722" valueType="System.String, mscorlib"

creationType="Sitecore.ListManagement.ContentSearch.Model.SegmentedList,
Sitecore.ListManagement.ContentSearch"

baseType="Sitecore.ListManagement.ContentSearch.Model.ContactList,
Sitecore.ListManagement.ContentSearch"

type="Sitecore.ContentSearch.DefaultDocumentMapperFactorySimpleRule, Sitecore.ContentSearch">
            <param desc="fieldName">$(fieldName)</param>
            <param desc="comparison">$(comparison)</param>
            <param desc="value">$(value)</param>
            <param desc="type">$(valueType)</param>
            <param desc="creationType">$(creationType)</param>
            <param desc="baseType">$(baseType)</param>
        </rule>
    </rules>
</objectFactory>
</indexDocumentPropertyMapper>

```

## 1.1.16 Changes to App\_Config\Include\ListManagement\Sitecore.ListManagement.Solr.IndexConfiguration.config.disabled

### 1. Replace the <fieldNames hint="raw:AddFieldByFieldName"> section, with:

```

<fieldNames hint="raw:AddFieldByFieldName">
    <field fieldName="name" returnType="string" storageType="YES"
indexType="TOKENIZED" vectorType="NO" boost="1f" type="System.String" />
    <field fieldName="description" returnType="text" stored="true"
indexed="true" termVectors="false" boost="1f" />
    <field fieldName="owner" returnType="string" storageType="YES"
indexType="TOKENIZED" vectorType="NO" boost="1f" type="System.String" />
    <field fieldName="recipients" returnType="int" storageType="YES"
indexType="TOKENIZED" vectorType="NO" boost="1f" type="System.Int32"/>
    <field fieldName="query" returnType="string" stored="true"
indexed="false" termVectors="false" boost="1f" />
    <field fieldName="source.includedlistsources"
returnType="stringCollection" stored="true" indexed="false" termVectors="false" boost="1f" />
    <field fieldName="source.excludedlistsources"
returnType="stringCollection" stored="true" indexed="false" termVectors="false" boost="1f" />
    <field fieldName="source.predefinedsourcetype" returnType="string"
stored="true" indexed="false" termVectors="false" boost="1f" />
    <field fieldName="source.predefinedsourceparameters"
returnType="stringCollection" stored="true" indexed="false" termVectors="false" boost="1f" />

```

# Configuration File Changes

```
        <field fieldName="source.usealldatabase" returnType="bool" stored="true"
indexed="false" termVectors="false" boost="1f" />
        <field fieldName="__fullcreateddate" returnType="string"
storageType="YES" indexType="TOKENIZED" vectorType="NO" boost="1f" type="System.DateTime"
format="yyyyMMddTHmmss" />
        <field fieldName="__fullupdateddate" returnType="string"
storageType="YES" indexType="TOKENIZED" vectorType="NO" boost="1f" type="System.DateTime"
format="yyyyMMddTHmmss" />
    </fieldNames>
```

2. In the beginning of the `<fields hint="raw:AddComputedIndexField">` section, add the following nodes:

```
        <field fieldName="source.includedlistsources"
itemFieldName="IncludedSources">Sitecore.ListManagement.ContentSearch.Fields.MappedListIndexField,Sitecore.ListManagement.ContentSearch</field>
        <field fieldName="source.excludedlistsources"
itemFieldName="ExcludedSources">Sitecore.ListManagement.ContentSearch.Fields.MappedListIndexField,Sitecore.ListManagement.ContentSearch</field>
        <field fieldName="source.predefinedsourcetype"
itemFieldName="PredefinedSourceType">Sitecore.ListManagement.ContentSearch.Fields.MappedIndexField,Sitecore.ListManagement.ContentSearch</field>
        <field fieldName="source.predefinedsourceparameters"
itemFieldName="PredefinedParameters">Sitecore.ListManagement.ContentSearch.Fields.MappedListIndexField,Sitecore.ListManagement.ContentSearch</field>
        <field fieldName="source.usealldatabase"
itemFieldName="UseAllDatabase">Sitecore.ListManagement.ContentSearch.Fields.MappedIndexField,Sitecore.ListManagement.ContentSearch</field>
```

3. After the `<fields>` section, add the following section:

```
    <!-- INDEX DOCUMENT TO PROPERTY MAPPER
        Maintains a collection of all the possible Convertors for the provider.
    -->
    <indexDocumentPropertyMapper
type="Sitecore.ContentSearch.SolrProvider.Mapping.SolrDocumentPropertyMapper,
Sitecore.ContentSearch.SolrProvider">
        <!-- OBJECT FACTORY
            Constructs search result objects based on the type that is passed in
            .GetQueryable<T>() and the rules defined in this section.
        -->
        <objectFactory
type="Sitecore.ContentSearch.DefaultDocumentMapperObjectFactory, Sitecore.ContentSearch">
            <rules hint="list:AddRule">
                <rule fieldName="_template" comparison="Equal"
value="c0caf6988a424b669eaf7d442b46f722" valueType="System.String, mscorlib"

creationType="Sitecore.ListManagement.ContentSearch.Model.SegmentedList,
Sitecore.ListManagement.ContentSearch"

baseType="Sitecore.ListManagement.ContentSearch.Model.ContactList,
Sitecore.ListManagement.ContentSearch"

type="Sitecore.ContentSearch.DefaultDocumentMapperFactorySimpleRule, Sitecore.ContentSearch">
                    <param desc="fieldName">$(fieldName)</param>
                    <param desc="comparison">$(comparison)</param>
                    <param desc="value">$(value)</param>
                    <param desc="type">$(valueType)</param>
                    <param desc="creationType">$(creationType)</param>
                    <param desc="baseType">$(baseType)</param>
                </rule>
            </rules>
        </objectFactory>
    </indexDocumentPropertyMapper>
```

## 1.1.17 Changes to App\_Config\Include\Social\Sitecore.Social.ExperienceProfile.config

1. After the `<settings>` section, add the following note:

```
<api>
  <services>
    <configuration>
      <allowedControllers hint="list:AddController">
```

```
<allowedController
desc="FacebookDataController">Sitecore.Social.ExperienceProfile.Client.Controllers.FacebookDataController, Sitecore.Social.ExperienceProfile.Client</allowedController>
  <allowedController
desc="GooglePlusDataController">Sitecore.Social.ExperienceProfile.Client.Controllers.GooglePlusDataController, Sitecore.Social.ExperienceProfile.Client</allowedController>
  <allowedController
desc="GooglePlusRecentActivityController">Sitecore.Social.ExperienceProfile.Client.Controllers.GooglePlusRecentActivityController,
Sitecore.Social.ExperienceProfile.Client</allowedController>
  <allowedController
desc="LinkedInDataController">Sitecore.Social.ExperienceProfile.Client.Controllers.LinkedinDataController, Sitecore.Social.ExperienceProfile.Client</allowedController>
  <allowedController
desc="TwitterDataController">Sitecore.Social.ExperienceProfile.Client.Controllers.TwitterDataController, Sitecore.Social.ExperienceProfile.Client</allowedController>
  <allowedController
desc="TwitterRecentActivityController">Sitecore.Social.ExperienceProfile.Client.Controllers.TwitterRecentActivityController, Sitecore.Social.ExperienceProfile.Client</allowedController>
</allowedControllers>
</configuration>
</services>
</api>
```

## 1.1.18 Changes to App\_Config\Include\XdbCloud\Sitecore.Cloud.Xdb.config.disabled

1. After "Sitecore.Cloud.Xdb.DeploymentType" setting, add the following note:

```
<!-- DEPLOYMENT REGION
Identifies the region for xDB Cloud that this Sitecore installation should
attempt to provision in.
Important: Before you run Sitecore for the first time, you must specify a
value for the DeploymentRegion setting.
You can do this after you configure Sitecore to use xDB Cloud. Thereafter
Sitecore will ignore the DeploymentRegion setting and automatically connect to the xDB Set
that you specified.
For a list of supported regions, go to:
https://doc.sitecore.net/cloud/81/xdm_cloud/configure_a_sitecore_xdb_cloud_connection.
Default value: "" (must be specified before connecting to xDB Cloud for the
first time)
-->
<setting name="Sitecore.Cloud.Xdb.DeploymentRegion" value="" />
```

2. Replace the <hooks> section, with the following:

```
<hooks>
  <hook type="Sitecore.Cloud.Xdb.DisableAggregation, Sitecore.Cloud.Xdb"
patch:after="hook[@type='Sitecore.Analytics.Aggregation.AggregationLoader,
Sitecore.Analytics.Aggregation']"/>
  <hook type="Sitecore.Cloud.Xdb.DisableProcessing, Sitecore.Cloud.Xdb"
patch:after="hook[@type='Sitecore.Analytics.Processing.ProcessingLoader,
Sitecore.Analytics.Processing']"/>
  <hook type="Sitecore.Cloud.Xdb.DisableAutomation, Sitecore.Cloud.Xdb"
patch:after="hook[@type='Sitecore.Analytics.Automation.Aggregation.SubsystemLoader,
Sitecore.Analytics.Automation.Aggregation']"/>
  <hook type="Sitecore.Cloud.Xdb.DisablePathAnalyzerProcessing,
Sitecore.Cloud.Xdb" patch:after="hook[@type='Sitecore.PathAnalyzer.Processing.SubsystemLoader,
Sitecore.PathAnalyzer']"/>
  <hook type="Sitecore.Cloud.Xdb.DisableExperienceAnalyticsReduce,
Sitecore.Cloud.Xdb"
patch:after="hook[@type='Sitecore.ExperienceAnalytics.Reduce.SubSystemLoader,
Sitecore.ExperienceAnalytics']"/>
</hooks>
```

## 1.1.19 Changes to App\_Config\Include\XdbCloud\Sitecore.ContentSearch.Cloud.Index.Analytics.config.disabled

1. Replace the <fieldReaders> node with:

# Configuration File Changes

```
<fieldReaders  
ref="contentSearch/indexConfigurations/defaultCloudIndexConfiguration/fieldReaders">
```