

Data Illusion's FeedbackServer

Table of Contents

About	1
History	3
Requirements	5
Buy now!	7
Help on help	9
Installation	11
Sharepoint 2007	11
WSS / Sharepoint 2007 Extension	11
MOSS active directory setup	12
MOSS .NET 3.5SP1	14
Web setup	16
Step 1 - Web Server Installation	16
Step 2 - SQL Server Installation	17
Upgrades	18
Web Installer Installation Upgrade	18
Registered License Upgrade	18
Active directory setup	18
Configure IIS for Integrated Security	19
Active directory and forms	20
Active directory and anonymous access	21
Configure IIS for Integrated Security	22
ASP.NET membership provider	22
SQL Server user provider	23
System settings	24
Settings Encryption	28
Form Management	29
Completion Actions	29
Completion action	29
Conditional thanks message	30
Completion Alerts	30
Completion Alerts	30
Score	33
Scoring	33
Form creation	34
General settings	34
Form home	37
Form export	38
Form Content Editor	39
Question wizard	39
Insert question	39
Expert mode	39
Question types	39
Static question	39
Single question	40
Matrix question	40
Side by side matrix	41
Pool linked question	41
Templated question	42

Hidden question	43
Question library	44
Question library	44
Library directory	44
Library templates	44
Repeatable sections	44
Repeatable sections	44
Sections management	46
Answer types	46
Selection based types	46
Field based types	48
Xml bound types	54
Sql bound types	56
List items bound types	57
Misc. types	58
Answer types	60
Create answer types	60
Answer skip logic conditions	63
Entry validators	63
Pages	64
Page Behaviors	64
Pages	68
Flow page	68
Free layout page	68
Reports page	69
Respondent answers	69
Content edition	69
Form builder	69
Preview / Test	70
Page Options	70
Question Options	70
Free form builder	71
Form Builder Functions	71
Form Elements	72
Form Behaviors	72
Current limitations / Known Issues	73
Question editor	73
Question Options	73
Matrix question editor	75
Matrix Question Options	75
Answer editor	76
Answer Overview	76
Answer Settings	76
Page options	77
Regular Expression Editor	77
Edit Regular Expression	78
Test Regular Expression	78
Pipe source selector	78
Answer Pipe Value	78
Question Answers Pipe	79

Dynamic Content	79
Querystring Variable Value	79
Session Variable Value	79
Cookie Variable Value	79
Web Server Variable Value	80
Panel Attribute Value	80
Form printer	80
Entry mask	80
Question pool	81
Linked panel question editor	82
Folder Management	83
Folders	83
Folder Settings	83
Trashcan	83
Trashcan Settings	83
Deployment / Distribution	85
Deployment	85
Link deployment	85
Friendly link	85
Popup windows	86
Popup Settings	86
Take survey	86
Web control deployment	87
WSS / Sharepoint deployment	87
Branding / Style	89
Style	89
Style editor	89
Style Settings	89
Media library	90
Media Folder Settings	90
Edit media item	90
Edit Media	90
Campaigns	91
Email Campaign	91
Email distribution	91
Email list	91
CSV file	91
Users and groups	92
Invitation message	92
Mailing status	93
Panel Campaign	93
Panel campaign	93
Invitation message	93
Mailing status	94
Campaigns	94
Mailing batch	95
Message template	95
Conditional Logic	97
Conditions	97
Condition Rules	97

Skip logic.....	98
Branching.....	99
URL redirection	99
Dynamic Content	101
Dynamic content	101
How does it work ?	101
Dynamic content conditions	101
Panel Management	103
Panelist Mapping	103
Panelist mapping.....	103
User provider panelist mapping.....	104
Security addins panelist mapping.....	104
Panel views.....	104
Panel views	105
Panel view settings.....	105
Panel view members	105
Panel Connectors	105
FeedbackServer connector	105
SQL Connector.....	106
Active Directory Connector.....	108
SalesForce.com Contact Connector.....	109
SalesForce.com Lead Connector	111
Panels.....	113
Panelists Entries.....	113
The Rosetta Stone.....	115
Panel information	115
Panel settings	116
Panel attributes	117
Attributes Overview	118
Attributes Types Settings.....	118
Panelist manager	118
Entry approval.....	119
Panelist editor	119
Edit Panelist	119
Attributes	120
Survey Answers	120
Pending Survey Invitation.....	120
Panel participation	120
Link panel to form	121
Reporting.....	123
Report Items	123
Report items	123
Graphic charts	124
Free text report.....	125
Respondent entries	126
Matrix grid report	126
Cross tabulation	127
Individual reach for multiple answers.....	127
Radar rating chart.....	128
Web Html graphics	129

Ranking report.....	129
Constant sum	129
Static text.....	129
Data export	129
Data export.....	129
CSV individual answers.....	130
CSV separate selection / fields.....	132
Xml export	134
SPSS export.....	135
Reports	136
Report settings.....	136
Report Options	136
Report Information.....	136
Answer Filters.....	136
Report builder	137
Page Options.....	137
Report Item Options	137
Insert report item.....	138
Extended filters	138
Report analysis	139
Filters.....	139
File manager	139
Uploaded Files	139
Export All Files To Server's Directory	140
Report filters.....	140
Rating groups.....	140
Rating Groups Options	140
Report printer	141
Filters.....	141
Data import	141
Data Import Step 1	141
Data Import Step 2 - The mapping	142
Tips & Tricks To Import / Export Using FeedbackServer.....	142
Data generator	142
Respondent Analysis	143
Archiving respondents.....	143
Security.....	145
Security add ins	145
IP protection	145
Cookie protection.....	145
Password protection	145
EMail code protection	145
Token protection.....	146
Asp.net security context	146
FeedbackServer security context	146
Panel security context	147
Active directory security.....	148
Entry number limitation.....	148
Image password	148
Email code confirmation	148

Token Security	149
Security tokens	149
Token generator	149
Token status	150
Security	150
Form security	151
Security Options	151
Security Add In Options	152
Insert security addin	152
Key provider	153
Multi-Language	155
Multi language forms	155
Multi language settings	155
Multi language export / import	156
Translation XLIFF Export	156
Translation XLIFF Export	156
System messages	156
Messages Manager	157
Messages Translation	157
User Management	159
User management	159
User manager	160
User creation	161
Group manager	162
Roles manager	162
User import	162
Metadata	165
Metadata	165
Metadata Editor	165
Glossary	Erreur ! Signet non défini.
Index	166

About

FeedbackServer™ is a web based software application that will let you create web-based surveys and forms to gather feedback from your customers, employees, friends or web site visitors and export or analyze the results through integrated reporting tools.

FeedbackServer does not require any specific technical knowledge to be used and is based on the industry standard Microsoft .NET framework.

Here are some key features you will find in it :

- A powerful web based and easy to use AJAX enabled form builder
- Free page layout base form editor
- Full free forms and template based design editor
- Complete panel management
- Panel connectors to external sources (SQL Server, Active Directory, Salesforce CRM ...)
- Web & Email distribution
- WYSIWYG editors
- Multiple answer types from selection to complex field types
- Multiple questions types including matrix type questions
- An easy to use report and results analysis builder
- Unique results filtering capabilities
- Multi-languages surveys
- Rating / scaling
- Branching features
- Answer piping
- Token based security
- Active directory integration
- ASP.net 2.0 membership provider support
- Word reporting
- Sharepoint portal server integration
- Native SPSS support
- Freely extensible architecture through FeedbackServer's SDK
- And much, much more!

For the latest complete features list you can also check
<http://www.feedbackserver.com>

We hope that you will enjoy this application as much as we did to create it,
The FeedbackServer Team

History

1st September 2010 - FeedbackServer 5.0.0

New features

- Specific print CSS has been setup when browser is used in print mode
- Respondent details can be show as pages within a form
- Reports can be shown as pages within a form
- Respondent can see his answers highlighted in the reports on the thanks page
- Report will automatically be shown on the thanks page if only one report has been set as public
- Ability to skip answers on the same page as the conditions that trigger them
- 3D Matrix (Side by side) questions support
- Matrix questions creation and edition interface has been completely reworked
- Matrix selection based answers can be displayed in drop down lists
- Replaced charting engine, charts can be rendered in 3D
- Customize width / height for each chart
- Display multiple chart series using filters or dates on one single graphic report item
- Added response date range rule for report filter conditions
- Added respondent language rule for report filter conditions
- Respondent can be set in archive mode to re-allow submissions from security add ins
- Pipe current respondent (voter) unique id using the {{vuid}} piping tag
- Set a default value for piping tags which can be displayed on reporting screens when no effective value can be piped
- Develop piper add ins to create custom tags and to pipe values from code
- Security add ins settings are cloned as well when cloning a survey
- Ranking setup can be set to use drop down lists instead of text boxes
- Media gallery with folder / sub folder organization support
- Upload and use any files (image, flash, video etc..) directly from the media gallery in your forms
- Develop custom condition add ins to return true / false to FS's condition (skip, branching etc..) using custom .net code
- Setup custom images in styles to replace the HTML radio / check box inputs form elements
- Added a small mail server test interface to test the current mail server settings
- Ability to force system buttons (next, previous, restore, save progress, submit) to be shown or hidden
- Automatic clearing of existing answers on skipped or branched pages can be turned off
- Page navigation using either navigation boxes or drop down lists to jump directly to a given page
- Page navigation boxes can be customized using CSS to give a "tabbed" navigation style
- Allow respondent to navigate only to already "visited" pages
- Set validation on/off for the current page when navigating to another page using page navigation
- Ignore all pages with invalid / missing answers when submitting a form with page navigation enabled
- Files are also available for download to respondent after he uploaded them
- Set title for the deployment page
- Set a permanent redirection url for deleted or closed forms
- Setup custom metadata attributes for forms, questions, answers and respondents
- Copy existing questions or free form based question into free form based pages
- Link panel questions in free form based pages

Fix

- Minor caching issues under webkit based browsers
- Ad groups didn't get access to question / answer details
- Unable to export forms using templates on multiple pages
- Reach was not calculated properly on web based graphical reports with multiple answers
- Random control tree errors on the form preview screen
- Card ranking did always show the first answer as the last choice
- Respondent details print used wrong language in multi-languages surveys
- Filter rules didnt retrieve the answers without values on not conditions

Requirements

FeedbackServer requires following configuration to run :

- Any Windows 2003 / 2008 Edition (32bit or 64bit), Windows Vista, Windows 7 or Windows XP Professional
- SQL Server 2005 / 2008 or the free SQL Server Express 2005 / 2008
- Internet Information Services 6.0 or above with ASP.NET 3.5SP1 or above
- Microsoft .NET Version 3.5 Service Pack 1
- Internet Explorer V7.x+ or FireFox V2.x+ for the web administration part
- Any browser to display and fill the forms

Note : Databases configured using Turkish localization are not compatible with FeedbackServer.

Buy now!

In order to offer the biggest flexibility to our customers, FeedbackServer is offered with several licensing models to fit small businesses needs as well as enterprise needs. You can buy the license that fits to your business online and get your FeedbackServer license file right after as an electronic download.

In order to purchase Feedback Server or check for the latest versions, please visit : <http://www.feedbackserver.com>

Thanks for supporting FeedbackServer!

Help on help

This online help system was designed to make it easy to find what you need. There are three ways to search:

- Use the Table of Contents to view the Help system like a book.
- A complete index - alphabetical by subject.
- The Search feature lets you search for specific text.

For example, if you are looking for information about generating reports features, you might look in the Table of Contents under Reporting or by using the index or search for the word 'Reports'.

At any time you can use the navigation bar across the top of each page to move up in the help system hierarchy. The file that is currently open is right-most in the chain, and you can return to any upper-level file using the links.

For the latest version of this help file, visit the FeedbackServer web site at :
<http://www.feedbackserver.com>

Installation

Sharepoint 2007

WSS / Sharepoint 2007 Extension

FeedbackServer is a very extensible survey and form application offer extensions for popular portals. One of the supported portal is Microsoft Sharepoint and Microsoft Windows Sharepoint Services (WSS).

To configure FeedbackServer with its MOSS extension and Active Directory integration we must follow the steps below :

How to Configure FeedbackServer Sharepoint Extensions.

1. Install the .net 3.5 SP1 framework on your MOSS installation. The .net 3.5 SP1 framework is required to run FS. To install .net 3.5 on your MOSS server. You may follow this guide
2. Make sure that you run under full trust by adding following item before the `</system.web>` element in your MOSS web.config. If you already have the `<trust ..>` element in your web.config simply change its level value to Full.

The MOSS web.config can usually be found in
C:\inetpub\wwwroot\wss\VirtualDirectories\80

```
<trust level="Full" originUrl="" />
```

- 3.
4. Copy following directory and file to your MOSS website root directory. This directory is usually located at : C:\inetpub\wwwroot\wss\VirtualDirectories\80
Files and directories to copy :
Bin
FeedbackServer.lic
fs.aspx
nreport.aspx
5. Create a new DataIllusion folder at the root of your MOOS website directory and copy following directories and their content in this new directory. This directory is usually located at :
C:\inetpub\wwwroot\wss\VirtualDirectories\80\DataIllusion
Directories and their content to copy to this new DataIllusion folder :
FeedbackServer
FS_Install
FSXMLData
Images
6. Open your SQL Server management interface and create a new empty FeedbackServer database and grant the DBO rights to the current Application Pool identity user. Usually this user is "Network Service".

Open your browser and enter the url of your MOSS site along the FS_Install directory.

eg: a `http://yourwsssite/dataillusion/fs_install`

7. Enter your SQL Server information, credentials and the name of the database you have created in step 6 and setup the Database. DOT NOT use and check the Write DB Connection to web.config option
8. Setup all FS configuration elements in your MOSS web.config. Please use this documentation to configure your MOSS config file accordingly.
9. Deploy the FeedbackServer MOSS package using the provided DeployWSP.bat file located in the MOSS_Install directory
10. Open your sharepoint site and from the Site Settings / Site Administration / Site Features enable the Feedback Server feature.

Once enabled a link to the administration section will be added to your quick launch tool bar and all FS related web parts will be available in your site.

MOSS active directory setup

This files contains the different configuration parts to enable FS on your MOSS installation

Make sure that you have installed .NET 3.5 and configured ASP.NET 3.5 on your MOSS Installation before installing FS.

The MOSS web.config is usually located in : `c:\inetpub\wwwroot\wss\virtualdirectories\80`

Copy the element below right after the `<configSections>` in your MOSS web.config

```
<section name="FeedbackServerConfig"
type="DataIllusion.FeedbackServer.Config.FeedbackServerSection,
DataIllusion.FeedbackServer.Core" />
```

Copy the element below right before the `<system.web>` in your MOSS web.config and set following attributes with your organization configuration details :

`connectionString="yourdatabaseconnectionstring"`

`smtpServer="yourmailserver"`

`adController="yourcontrollerpdc"`

`adUserName="youradname"`

`adPassword="youradpassword"`

`adDomain="youraddomain"`

Set following attribute to the account that will always have administrative privileges within FS

`feedbackServerADUserName="youraccount"`

Account name must be entered without its domain ("domain\)") prefix.

```
<FeedbackServerConfig connectionString="server=(local);database=FeedbackServer;Integrated
security=SSPI" enableMOSS="true" adminRoot=~/.DataIllusion/FeedbackServer"
imagesPath=~/.DataIllusion/images/" xmlDataPath=~/.DataIllusion/FSXmlData/"
languagesPath=~/.DataIllusion/FSXmlData/Languages/"
savTempPath=~/.DataIllusion/SAVTempExport/"
surveyTemplatePath=~/.DataIllusion/FSXmlData/Templates/Surveys/"
panelSyncThreadsNumber="2" smtpServer="localhost" smtpPort="25" smtpAuthUserName=""
smtpAuthPassword="" emailingProviderAssembly="DataIllusion.FeedbackServer.Core"
emailingProviderClass="DataIllusion.FeedbackServer.Emailing.SystemWebEmailing"
```

```
mailingThreadsNumber="2" mailingBatchPoolingDelay="1" sqlBasedAnswerTypesAllowed="true"
uploadedFileDeleteTimeOut="24" sessionUploadedFileDeleteTimeOut="336" serverCaching="true"
cacheTimeOut="-1" userProviderAssembly="DataIllusion.FeedbackServer.Core"
userProviderClass="DataIllusion.FeedbackServer.UserProvider.ADUserProvider"
formUserProviderSingleMode="false" adController="yourcontrol" adUserName="youradname"
adPassword="youradpassword" adDomain="youraddomain" bypassImpersonation="true"
adToDBSyncTimeOut="1440" feedbackServerADUserName=""
adSearcherFilter="( & (objectCategory=person)(objectClass=user))"
feedbackServerMembershipUserName="" feedbackServerMembershipApplicationName=""
enableTrashCan="true" compressSurveyViewState="true"
compressAjaxWebServices="true" compressAdminPages="true"
compressSurveyDeploymentPage="true" />
```

Copy the elements right before the </httpHandlers> in your MOSS web.config

```
<add verb="*" path="fs-*.aspx"
type="DataIllusion.FeedbackServer.Web.Handlers.FSFriendlyHandler,DataIllusion.FeedbackServer.C
ore"/>
<add verb="*" path="fsexport.aspx"
type="DataIllusion.FeedbackServer.Web.Handlers.DataExportHandler,DataIllusion.FeedbackServer.C
ore"/>
<add verb="*" path="fsstyle.aspx"
type="DataIllusion.FeedbackServer.Web.Handlers.StyleHandler,DataIllusion.FeedbackServer.Core"/>
<add verb="*" path="fslang.aspx"
type="DataIllusion.FeedbackServer.Web.Handlers.JSResourceHandler,DataIllusion.FeedbackServer.
Core"/>
```

Copy the element below right before the </httpModules> in your MOSS web.config

```
<add name="PanelScheduler"
type="DataIllusion.FeedbackServer.Web.HttpModules.FSContextModule,DataIllusion.FeedbackServer
.Core"/>
```

Copy the element below right before the </modules> in your MOSS web.config

```
<remove name="FSContext"/>
<add name="FSContext"
type="DataIllusion.FeedbackServer.Web.HttpModules.FSContextModule,DataIllusion.FeedbackServer
.Core"/>
```

Copy the elements below right before the </handlers> in your MOSS web.config

```
<add name="FSStyles" verb="*" path="fsstyle.aspx"
type="DataIllusion.FeedbackServer.Web.Handlers.StyleHandler,DataIllusion.FeedbackServer.Core"/>
<add name="FSLang" verb="*" path="fslang.aspx"
type="DataIllusion.FeedbackServer.Web.Handlers.JSResourceHandler,DataIllusion.FeedbackServer.
Core"/>
<add name="FSFriendly-Integrated" verb="*" path="fs-*.aspx"
type="DataIllusion.FeedbackServer.Web.Handlers.FSFriendlyHandler,DataIllusion.FeedbackServer.C
ore"/>
<add name="FSDataExport-Integrated" verb="*" path="fsexport.aspx"
type="DataIllusion.FeedbackServer.Web.Handlers.DataExportHandler,DataIllusion.FeedbackServer.C
ore"/>
```

Replace your exiting <pages ...> element by the page one below

```
<pages enableSessionState="false" enableViewState="true" enableEventValidation="false"
enableViewStateMac="true" validateRequest="false"
pageParserFilterType="Microsoft.SharePoint.ApplicationRuntime.SPPageParserFilter,
Microsoft.SharePoint, Version=12.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c"
```

```
asyncTimeout="7">
```

MOSS .NET 3.5SP1

This document contains the different configuration elements to enable .NET AJAX 3.5 on your MOSS web.config once you have downloaded and installed .net 3.5 SP1. The MOSS web.config is usually located in : c:\inetpub\wwwroot\wss\virtualdirectories\80

DOWNLOAD AND INSTALL .NET 3.5 SP1 ON YOUR MOSS SERVER

you may download it from following location :

<http://www.microsoft.com/downloads/details.aspx?FamilyID=ab99342f-5d1a-413d-8319-81da479ab0d7>

Copy the element below right after the <configSections> in your MOSS web.config

```
<sectionGroup name="system.web.extensions"
type="System.Web.Configuration.SystemWebExtensionsSectionGroup,System.Web.Extensions,
Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35">
<sectionGroup name="scripting"
type="System.Web.Configuration.ScriptingSectionGroup,System.Web.Extensions, Version=3.5.0.0,
Culture=neutral, PublicKeyToken=31BF3856AD364E35"> <section name="scriptResourceHandler"
type="System.Web.Configuration.ScriptingScriptResourceHandlerSection, System.Web.Extensions,
Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35" requirePermission="false"
allowDefinition="MachineToApplication"/>
<sectionGroup name="webServices"
type="System.Web.Configuration.ScriptingWebServicesSectionGroup, System.Web.Extensions,
Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35">
<section name="jsonSerialization"
type="System.Web.Configuration.ScriptingJsonSerializationSection, System.Web.Extensions,
Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35" requirePermission="false"
allowDefinition="Everywhere" />
<section name="profileService" type="System.Web.Configuration.ScriptingProfileServiceSection,
System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"
requirePermission="false" allowDefinition="MachineToApplication" />
<section name="authenticationService"
type="System.Web.Configuration.ScriptingAuthenticationServiceSection, System.Web.Extensions,
Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35" requirePermission="false"
allowDefinition="MachineToApplication" />
<section name="roleService" type="System.Web.Configuration.ScriptingRoleServiceSection,
System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"
requirePermission="false" allowDefinition="MachineToApplication" />
</sectionGroup>
</sectionGroup>
</sectionGroup>
```

Copy the element below right after the <SafeControls> in your MOSS web.config

```
<SafeControl Assembly="System.Web.Extensions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35" Namespace="System.Web.UI" TypeName="*" Safe="True"
/>
```

Copy the element below right before the </httpHandlers> in your MOSS web.config

```
<add verb="*" path="*.asmx" validate="false"
type="System.Web.Script.Services.ScriptHandlerFactory, System.Web.Extensions, Version=3.5.0.0,
Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>
```

```
<add verb="*" path="*_AppService.axd" validate="false"
type="System.Web.Script.Services.ScriptHandlerFactory, System.Web.Extensions, Version=3.5.0.0,
Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>
<add verb="GET,HEAD" path="ScriptResource.axd"
type="System.Web.Handlers.ScriptResourceHandler, System.Web.Extensions, Version=3.5.0.0,
Culture=neutral, PublicKeyToken=31BF3856AD364E35" validate="false"/>
```

Copy the element below right before the </httpModules> in your MOSS web.config

```
<add name="ScriptModule" type="System.Web.Handlers.ScriptModule, System.Web.Extensions,
Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>
```

Copy the element below right after the <assemblies> in your MOSS web.config

```
<add assembly="System.Core, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=B77A5C561934E089"/>
<add assembly="System.Web.Extensions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35"/>
<add assembly="System.Data.DataSetExtensions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=B77A5C561934E089"/>
<add assembly="System.Xml.Linq, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=B77A5C561934E089"/>
```

Copy the element below right after the <pages> in your MOSS web.config

```
<controls>
<add tagPrefix="asp" namespace="System.Web.UI" assembly="System.Web.Extensions,
Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>
<add tagPrefix="asp" namespace="System.Web.UI.WebControls"
assembly="System.Web.Extensions, Version=3.5.0.0,
Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>
</controls>
```

Copy the element below right before the </configuration> in your MOSS web.config

```
<system.web.extensions>
<scripting>
<webServices>
</webServices>
</scripting>
</system.web.extensions>
<system.webServer>
<validation validateIntegratedModeConfiguration="false"/>
<modules>
<remove name="ScriptModule" />
<add name="ScriptModule" preCondition="managedHandler"
type="System.Web.Handlers.ScriptModule, System.Web.Extensions, Version=3.5.0.0,
Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>
</modules>
<handlers>
<remove name="WebServiceHandlerFactory-Integrated"/>
<remove name="ScriptHandlerFactory" />
<remove name="ScriptHandlerFactoryAppServices" />
<remove name="ScriptResource" />
<add name="ScriptHandlerFactory" verb="*" path="*.asmx" preCondition="integratedMode"
type="System.Web.Script.Services.ScriptHandlerFactory, System.Web.Extensions, Version=3.5.0.0,
```

```
Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>
<add name="ScriptHandlerFactoryAppServices" verb="*" path="*_AppService.axd"
preCondition="integratedMode" type="System.Web.Script.Services.ScriptHandlerFactory,
System.Web.Extensions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35"/>
<add name="ScriptResource" preCondition="integratedMode" verb="GET,HEAD"
path="ScriptResource.axd"
type="System.Web.Handlers.ScriptResourceHandler, System.Web.Extensions, Version=3.5.0.0,
Culture=neutral, PublicKeyToken=31BF3856AD364E35" />
</handlers>
</system.webServer>
```

Web setup

The web installer has been designed for users who wish to get the maximum of flexibility during their installation process. The installation files are delivered inside a single compressed ZIP file and require you to configure Internet Information Services manually.

Before starting the installation make sure that you have following component installed on your server :

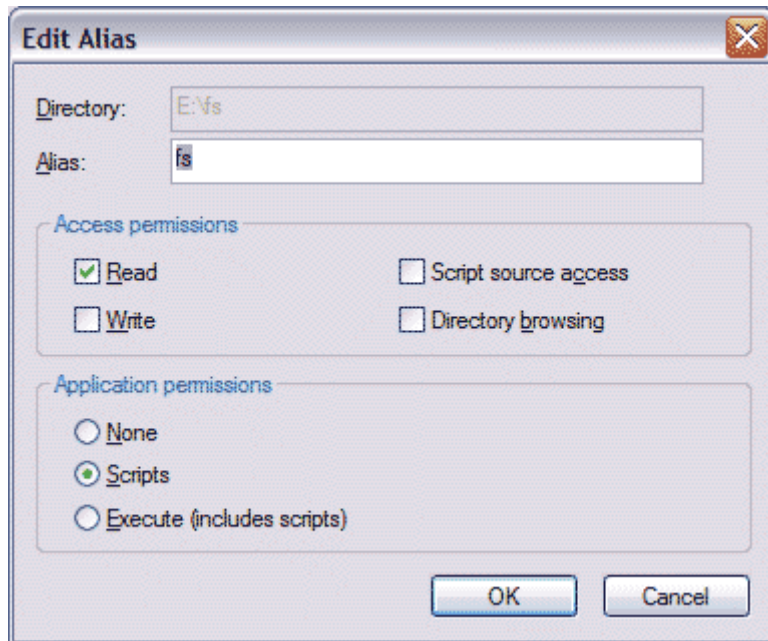
- **NET 3.5 Service Pack 1 or .NET 4.0.** You may download the .net 3.5 or the 4.0 package from Microsoft's web site at :
<http://www.microsoft.com/downloads/details.aspx?FamilyID=9cfb2d51-5ff4-4491-b0e5-b386f32c0992&displaylang=en>
- **Microsoft Charting Components** which can be downloaded from Microsoft's website at :
<http://www.microsoft.com/downloads/details.aspx?FamilyId=130F7986-BF49-4FE5-9CA8-910AE6EA442C&displaylang=en>

You may also follow this animated tutorial to install FeedbackServer on your server.

Step 1 - Web Server Installation

Unzip the compressed ZIP file inside a directory of your choice, for example Feedback Server. Once you have unzipped the directory you will need to configure it as a virtual directory for Internet Information Services.

To do this part right click on the directory and select its properties and under Web sharing choose share this folder, keep the default settings, click ok and then apply the changes.



Once you have finished these steps Internet Information Services will be configured with this new virtual directory and should be able to access it to configure the SQL database using your browser and URL : http://yourserver/fs_install/

Step 2 - SQL Server Installation

Using your browser navigate to follow url http://yourservername/fs_install.

This page will allow you install Feedback Server® on one of your existing SQL Server databases.

Its highly recommended to delete the fs_install directory once you're done with the SQL configuration.

Upgrades

Upgrading an existing installation of FeedbackServer is very straightforward. Even though upgrading will NEVER delete any of your data or database **we do strongly recommend making a backup of your database before any upgrades.**

Before starting the installation make sure that you have following component installed on your server :

- **NET 3.5 Service Pack 1 or .NET 4.0.** You may download the .net 3.5 or the 4.0 package from Microsoft's web site at :
<http://www.microsoft.com/downloads/details.aspx?FamilyID=9cfb2d51-5ff4-4491-b0e5-b386f32c0992&displaylang=en>
- **Microsoft Charting Components** which can be downloaded from Microsoft's website at :
<http://www.microsoft.com/downloads/details.aspx?FamilyId=130F7986-BF49-4FE5-9CA8-910AE6EA442C&displaylang=en>

Web Installer Installation Upgrade

Make sure to delete all existing files from your FeedbackServer web installation. Once all files are deleted, copy all new files supplied with the web installer zip file to your directory and go to the following URL : http://yourserver/fs_install/

From there you can specify your existing SQL Server and existing FeedbackServer database and the SQL Installer will automatically detect and upgrade your existing FeedbackServer database to the latest version.

Registered License Upgrade

If you purchased FeedbackServer you have to make sure that you have also downloaded the latest registered license for it and copy the registered feedbackserver.lic file over the provided trial license in your FeedbackServer directory. Each version of FeedbackServer requires to download the latest license from our customer section at <http://www.dataillusion.com>.

Active directory setup

FeedbackServer comes out of the box with a complete Active Directory user provider letting you connect to your enterprise Active Directory server and use its user information instead of FeedbackServer's built-in user provider.

Once configured the provider will redirect all built in user features of FeedbackServer to Active Directory. For security reasons some features related to the users eg:

password or profile management are not available.

To configure the Membership provider you will need to remove the original web.config and rename the existing web.AD.config.

Once you have renamed your web.AD.config to web.config you will need to setup following elements using a text editor like notepad to edit your web.config file.

1. connectionString is the connection string that FeedbackServer will use to connect to your SQL Server.
2. Setup following keys with your configuration :
 - ADController is the ip or server name of your Active directory primary controller.
 - ADUserName is the username required to connect to your Active Directory server.
 - ADPassword is the password of the username that has rights on your Active Directory server.
 - ADDomain is the domain of your Active Directory infrastructure.
 - FeedbackServerAdminADUsername is the Active Directory Username that will always have administrator privileges in Feedback Server. Generally it must be the username of the person who first will access and configure Feedback Server. The name must be entered without the domain\ prefix.
 - ADSearcherFilter is the Active Directory filter that is used to retrieve the user list in the user management interface of Feedback Server. You may change the LDAP query based on your personal needs. By default this query will return all users and groups.
 - ADToDBSyncTimeOut is the time in minutes after which the local FeedbackServer user database is re synchronized with your remote Active Directory server. It is recommended to keep the value high to avoid network traffic, especially on high volumes Active Directory servers.

Configure IIS for Integrated Security

1. Open the Internet Information Services Management from the "Administrative Tools menu" on the "Start Menu" or in the "Control Panel".
2. Expand your server in the tree on the left until you see the Web site or Virtual Directory you want to configure. Right click the site or directory and choose Properties.
3. Switch to the "Directory Security" tab and click the "edit" button in the Anonymous access and authentication control area.

4. Make sure you that "Anonymous access" is not checked and make sure that the "Integrated Windows authentication" is checked.

Once you have configured the IIS and the new user provider, you may access FeedbackServer using any administration page of FeedbackServer. You do not need to access FeedbackServer administration pages through the default FeedbackServer login page as Active Directory authentication is handled by the browser.

Active directory and forms

Sometimes its not possible to configure the web server to enable integrated windows authentication for user login. FeedbackServer provides a way to still have the benefits of Active Directory authentication by showing a web authentication interface without having to enable integrated authentication on the server.

In order to configure you will need to remove the original web.config and rename the web.AD.Forms.config.

Once you have renamed your web.AD.Forms.config to web.config you will need to setup following elements using a text editor like notepad to edit your web.config file.

1. connectionString is the connection string that FeedbackServer will use to connect to your SQL server.
2. Setup following keys with your configuration :
 - ADController is the ip or server name of your Active directory primary controller.
 - ADUserName is the username required to connect to your Active Directory server.
 - ADPassword is the password of the username that has rights on your Active Directory server.
 - ADDomain is the domain of your Active Directory infrastructure.
 - Feedback ServerAdminADUsername is the Active Directory Username that will always have administrator privileges in Feedback Server. Generally it must be the username of the person who first will access and configure Feedback Server. The name must be entered without the domain\ prefix.
 - ADSearcherFilter is the Active Directory filter that is used to retrieve the user list in the user management interface of Feedback Server. You can change the LDAP query based on your personal needs.
 - ADToDBSyncTimeOut is the time in minutes after which the local FeedbackServer user database is re synchronized with your remote Active Directory server. It is recommended to keep the value high to avoid network traffic, especially on high volumes Active Directory servers.

Once you have configured the new web.config, you may access FeedbackServer using any the login page of Feedback Server which will show you a login box that will authenticate the user against the Active Directory domain controller you have configured.

Active directory and anonymous access

FeedbackServer can be configured to run at the same time under Active Directory for user authentication in the administration section while still allowing anonymous user who don't have an Active Directory account to take surveys on the public site.

To configure FeedbackServer in order to support both mode at the same time we have to follow the steps below :

1. Remove the original web.config and rename the web.AD.Public.config to web.config. Once you have renamed your files you will need to setup following elements using a text editor like notepad to edit your web.config file.
2. connectionString is the connection that Feedback Server® will use to connect to your SQL 2000 / 2005 / 2005 Express or MSDE server where you have installed your Feedback Server database using the web installer.
3. Setup following keys with the feedback server configuration to setup Active Directory:
 - ADController is the ip or server name of your Active directory primary controller.
 - ADUserName is the username required to connect to your Active Directory server.
 - ADPassword is the password of the username that has rights on your Active Directory server.
 - ADDomain is the domain of your Active Directory infrastructure.
 - Feedback ServerAdminADUsername is the Active Directory Username that will always have administrator privileges in Feedback Server. Generally it must be the username of the person who first will access and configure Feedback Server. The name must be entered without the domain\ prefix.
 - ADSearcherFilter is the Active Directory filter that is used to retrieve the user list in the user management interface of Feedback Server. You can change the LDAP query based on your personal needs.
 - ADToDBSyncTimeOut is the time in minutes after which the local Feedback Server® user database is re synchronized with your remote Active Directory server. It is recommended to keep the value high to avoid network traffic, especially on high volumes Active Directory servers.

Configure IIS for Integrated Security

1. Open the Internet Information Services Management from the "Administrative Tools menu" on the "Start Menu" or in the "Control Panel".
2. Expand your server in the tree on the left until you see the Web site or Virtual Directory you want to configure. Select the FeedbackServer Directory and click on its properties to display the property windows.
3. Switch to the "Directory Security" tab and click the "edit" button in the Anonymous access and authentication control area.
4. Make sure that the "Anonymous access" is not checked and make sure that the "Integrated Windows authentication" is checked.

Once you have configured the directories you will be able to login to the administration section using the standard windows authentication of your browser while keeping the survey and report deployment pages available for public users. If you want public users to use active directory to enter a survey you have to setup the Active Directory Security Addin on your survey.

ASP.NET membership provider

ASP.NET features a built in user membership provider that can be easily extended and which also offers by default an Active directory provider. FeedbackServer can replace its default provider with an asp.net compatible user provider enabling FeedbackServer to be fully integrated in your user management systems based on the asp.net provider.

Once configured the provider will redirect all built in user features of FeedbackServer to the asp.net membership provider.

To setup the membership provider you will need to remove the original web.config and rename the web.ASP.config.

Once you have renamed your web.ASP.config to web.config you will need to setup following elements using a text editor like notepad to edit your web.config file.

1. connectionString is the connection that FeedbackServer will use to connect to your SQL server.
Note that its not the same as the ASP.net membership connection string in step 2.

2. Setup the ASP.net membership connection string

```
<connectionStrings>
<remove name="LocalSqlServer"/>
<add name="LocalSqlServer" connectionString="server=yoursqlserver;integrated security=true;Initial
Catalog=yourmembershipdatabase" />
</connectionStrings>
<roleManager enabled="true" />
```

```
<membership>
  <providers>
    <add connectionStringName="LocalSqlServer" name="MySQLMembershipProvider"
type="System.Web.Security.SqlMembershipProvider, System.Web, Version=2.0.0.0,Culture=neutral,
PublicKeyToken=b03f5f7f11d50a3a" />
  </providers>
</membership>
```

3. Note that this settings are related to asp.net membership configuration and you will need to setup them accordingly to your current .net configuration. If you dont know how to setup the membership provider we suggest following tutorial to start with : Scott Guthrie's membership provider
4. <add key="FeedbackServerMembershipUserName" value="" /> < BR> <add key="FeedbackServerMembershipApplicationName" value="" />
5. FeedbackServerMemberhipName will be the user defined in the asp.net membership provider that will hold by default admin rights inside FeedbackServer. This is required as we need at least one administrator when we start FeedbackServer for the first time.
6. FeedbackServerMemberhipApplicationName (optional) is the application from which the asp.net membership provider will retrieve the users and roles. Leave blank or commented out if you don't know your application name.

Once you have configured the new user provider, you may access Feedback Server using any asp.net user membership provider available on the market.

SQL Server user provider

The SQL Server user provider replaces the existing FeedbackServer user management system by our own SQL Server based user and groups tables. Thanks to this provider we can directly re-use existing SQL server tables which contains information about users and groups to which we will grant access to the administration section of FeedbackServer.

Once configured the provider will use the users and groups from our custom SQL Server tables.

In order to configure the SQL Server provider you will need to remove the original web.config and rename the web.SQL.config.

Once you have renamed your web.SQL.config to web.config you will need to setup following elements using a texteditor like notepad to edit your web.config file.

1. sqlUserProviderConnectionString is the connection that FeedbackServer will use to connect to your SQL Server database containing the user and group tables.

2. `sqlUserProviderFeedbackServerAdminUserName` is the user within our custom user table that will always have administrative privileges within FeedbackServer.
3. `sqlUserProviderPasswordEncryptionType` if our user table has its password encrypted we can choose the method to validate the passwords given by the users to authenticate within FeedbackServer against our user table password column. Following password encryption validation method are supported :
 1. None
 2. Sha5
 3. MD5

SQL queries to our SQL user and groups tables can be specified using following attributes. Note that `@username`, `@Password`, `@UserId` text is replaced dynamically by FeedbackServer at runtime with the corresponding runtime value.

1. `sqlUserProviderAuthenticationQuery` is the query used by FeedbackServer to authenticate a user from the administration section login screen.
2. `sqlUserProviderUsersQuery` is the query to get all users from our custom SQL user table. Feedback Server will always take the first column in the select as the ID and the second one as the username.
3. `sqlUserProviderGroupsQuery` is the query to get all groups from our custom SQL group table. Feedback Server will always take the first column in the select as the group ID and the second one as the group name. Leave blank if no custom group table is available.
4. `sqlUserProviderUserGroupsQuery` is the query to get all groups for the given user id.
5. `sqlUserProviderUserNameQuery` is the query to get the username based on the user id.

Once you have configured the new user provider, you may access FeedbackServer using any user and password available within the custom SQL Server user table.

System settings

FeedbackServer provides two ways to set its internal settings, either through the web settings interface or directly by changing the `web.config` file using a text editor like notepad.



Some security related settings can only be configured through the `web.config`.

- `connectionString` is the connection that FeedbackServer will use to connect to your SQL server.
- `smtpServer` is the mail server that will be used by FeedbackServer mail features.
- `smtpAuthUserName` is the username required to connect to the mail server to relay mails.
- `smtpAuthPassword` is the password required to connect to the mail server to relay mails.
- `sslEnabledSmtp` enables secure communication between FeedbackServer and mail servers supporting the ssl protocol.
- `smtpPort` is the port of the mail server.
- `adminRoot` is the directory name of your FeedbackServer administration directory.
- `imagesPath` is the directory where Feedback Server's image are located.
- `xmlDataPath` is the directory holding all Xml based files used by the Xml Answer Types.
- `languagesPath` is the directory holding the system languages files.
- `savTempPath` is the directory used by the SPSS export to save its temporary file. This directory must have its write right set to true.
- `panelSyncThreadsNumber` is the number of threads that synchronises panel information.
- `emailingProviderAssembly` is the assembly used by the mailing features.
- `emailingProviderClass` is the class used by the mailing features.
- `mailingThreadsNumber` is the number of threads that will be used to send out emails.
- `mailingBatchPoolingDelay` is the number of minutes FeedbackServer will wait before checking if some emails need to be sent out.
- `disableDataExportLink` removes the access to the data export pages.
- `sqlBasedAnswerTypesAllowed` allows us to restrict if we dont want to allow the creation of SQL based answer types.
- `uploadedFileDeleteTimeOut` is how many hours can unvalidated uploaded files stay in the database.
- `sessionUploadedFileDeleteTimeOut` is how many hours can uploaded files for sessions that has no yet been resumed stay in the database. Leave some time

because user wont get notified if its session have been deleted when he resumes.

- serverCaching is the internal caching feature of FeedbackServer to improve performances.
 - ⚠ • Set caching to false if you are running FeedbackServer in a **cluster environment** where several servers running the same instances of FeedbackServer against the same FeedbackServer DB.
- cacheTimeout is the time in minutes after which the object stored in the cache will be refreshed. This option is very useful in a cluster environment where you need to synchronize "manually" the caching sources based on the cache timeout.
 - 0 = no cache
 - -1 = no timeout.
- userProviderAssembly is the assembly used to located the object that will handle user management.
- userProviderClass is the class that used to handle user management.
- formUserProviderSingleMode allow us to turn off user management and run like a single user without any credential or login need.
- adController is the server name or ip to which FeedbackServer will connect to retrieve the active directory users and groups.
- adUserName is the username required to connect to your Active Directory server.
- adPassword is the password of the username that has rights on your Active Directory server.
- adDomain is the domain of your Active Directory infrastructure.
- adAuth1Controller is the additional server name or ip to which FeedbackServer will connect to retrieve the active directory users and groups if we want to allow the Feedback Server active directory security addin to authenticate respondents against another domain.
- adAuth1UserName is the additional username required to connect to your Active Directory server if we want to allow the Feedback Server active directory security addin to authenticate respondents against another domain.
- adAuth1Password is the additional password of the username that has rights on your Active Directory server if we want to allow the Feedback Server active directory security addin to authenticate respondents against another domain.
- adAuth2Domain is the additional domain of your Active Directory infrastructure if we want to allow the Feedback Server active directory security addin to authenticate respondents against another domain.

- `adAuth2Controller` is the additional server name or ip to which FeedbackServer will connect to retrieve the active directory users and groups if we want to allow the Feedback Server active directory security addin to authenticate respondents against another domain.
- `adAuth2UserName` is the additional username required to connect to your Active Directory server if we want to allow the Feedback Server active directory security addin to authenticate respondents against another domain.
- `adAuth2Password` is the additional password of the username that has rights on your Active Directory server if we want to allow the Feedback Server active directory security addin to authenticate respondents against another domain.
- `adAuth2Domain` is the additional domain of your Active Directory infrastructure if we want to allow the Feedback Server active directory security addin to authenticate respondents against another domain.
- `adToDBSyncTimeOut` is the number of minutes we want to wait until we resync the local DB with AD. It is strongly advised to keep this to a high value to avoid any network and AD server overload.
- `feedbackServerADUserName` is the username required to connect to your Active Directory server.
- `adSearcherFilter` is the Active Directory filter that is used to retrieve the user list in the user management interface of Feedback Server. You can change the LDAP query based on your personal needs. It returns by default all users.
- `feedbackServerMembershipUserName` is the asp.net membership user who will be the first FeedbackServer admin. You will need to login with this user the first time you use FeedbackServer with the membership API to get admin rights.
- `feedbackServerMembershipApplicationName` Optional : Application name to assign to the asp.net membership provider to define from which application users will be used.
- `enableTrashcan` let us disable the trashcan system from the system allowing the removal of surveys without any way to recover them.
- `compressSurveyViewState` enables the main survey control to compress and decompress the asp.net generated viewstate to reduce download size on the page and improve performances.
- `compressAjaxWebServices` let us compress all requests being made to web services within the FS application. Its important to note that if you run FeedbackServer along with other applications within the same IIS app space this might have an impact on other applications too and you may want to disable it.
- `compressAdminPages` let us compress all responses of administration pages called within FeedbackServer.

FeedbackServer

- `compressAdminPages` let us compress all responses of administration pages called within FeedbackServer.
- `compressSurveyDeploymentPage` let us compress the output of the survey deployment page (`fs.aspx`). Its important to note that this will not compress a third page where we would have incorporated only the survey web control.
- `dateFormat` let us format the dates that are show in Feedback Server. By default universal date format is used ("u") but you can use any of the formats available here.

Settings Encryption

Using the online settings interface you can choose wether or not to encrypt the `web.config` part containing FeedbackServer's settings.



Each machine has different encryption settings and once the section is encrypted you will not be able to copy & paste the FeedbackServer's settings to another `web.config` on another machine.

Form Management

Completion Actions

Completion action

Once our respondents have finished to answer our survey and submitted their answers we can setup FeedbackServer to trigger a completion action to run at the end of our form.

Completion actions that can be executed are described below :

"Thanks you" Message Action

This action allows us to show any text / html message to the respondent once he finished the survey. Its generally used to thank the user or maybe give him information about the survey he took. Using the conditional thanks message feature we can also show a specific message based on the user answers.

- **Edition Language** is the language in which the "thank you" message text is edited and to which language it will be shown. We can select a language, change the text and when we will click on the update button the text will be assigned to the selected language. This feature is only available if we have turned on FeedbackServer's Multi-Language features.
- **Thanks Message** is the message that will be shown to the end user once he finished the survey. Its also possible to embed specific text tags that will be replaced dynamically at runtime by FeedbackServer and show extra information like :

::score:: tag will display the total score of the user. Note that this features requires scoring to be activated.

::keycode:: tag will display the unique code provided by the Key Provider, this tag is generally used if the user can change their answers and if the default display of the key code has been disabled.

URL Redirection Action

This action allows us to redirect the user to specific page or URL.

- **Edition Language** for which the redirection will occur it is also the language in which the "thank you" message text is edited and to which language it will be

shown. This feature is only available if we have turned on FeedbackServer's Multi-Language features.

- **Thanks Message** is the message that will be shown to the respondent once he finished the survey. This message will only show up if you didn't specify any redirection URL. Its mostly used in case you have multiple languages and want for one language a redirection but not for the others.
- **Redirection URL** is the URL to which the user will be redirected. It must be in following format : <http://www.youdomain.com/yourpage.aspx>.
- **Redirection Delay** is the number of seconds the thank you message will be shown to the respondent until he gets redirected to the redirection URL.

Reports Display Action

This action allows us to show a "thank you" message at the end of the survey which has attached at his bottom a list of reports that the user can display.

- **Edition Language** in which the "thank you" message text is edited and to which language it will be shown. This feature is only available if we have turned on FeedbackServer's Multi-Language features.
- **Thanks Message** is the message that will be shown to the respondent once he finished the survey. This message will only show up if you didn't specify any redirection URL. Its mostly used in case you have multiple languages and want for one language a redirection but not for the others.
- **Reports To Show** is the list of reports that will be available to the respondent once he finished the survey. Note that in order to be able to use reports publicly we must be set at least one public reports from the General Report Settings.

Conditional thanks message

Thanks message conditions allow us to setup logical rules based on respondent's answers to show a specific message at the end of our survey. We could for example show a specific message only to people who answered choice B at question 5.

Each condition is based on a set of rules that you can define. The first condition that will be met will show its "thanks" message to the user. You can have as many conditions as you wish and order or re-order them at any time.

Completion Alerts

Completion Alerts

Completion alerts are a great way to get or send email based notifications once a user has finished a survey. Usage can range from simple scenarios like receiving a list of respondent's answers to more complex scenario where we can create templates with file attachments and trigger alerts only if the respondents have answered specific answers.

FeedbackServer offers several completion alerts that can be triggered :

Short Message Alert

This alert will notify us by a small text stating that a new person has filled out our form. No details about the answers will be send.

- **Email From Name** is the name from which the email the notification will originate. Respondent's answers can be piped in it using the `[[youranswerpipealias]]`.
- **Email From** is the email from which the notification will be sent from. Respondent's answers can be piped in it using the `[[youranswerpipealias]]`.
- **Email Notification To** is the email that will receive the notification. If we want to send alerts to several emails we can separate emails through a ; char. Respondent's answers can be piped in it using the `[[youranswerpipealias]]`.
- **Email Notification Subject** is the subject of the notification email. Respondent's answers can be piped in it using the `[[youranswerpipealias]]`.
- **Attachments** is the attachment files that will be send along with this alert. There are no limits on the number of attachment that can be send along the alert. In order to add a new attachment click on the on the browse button, select your file and click on the update button of the alert. We must keep in mind to use small size attachments as some mailboxes are limited in size.

Full Entry Report Alert

This alert will sent us a complete report where all respondent's answers will be shown including those he didn't answer.

- **Email From Name** is the name from which the email the notification will originate. Respondent's answers can be piped in it using the `[[youranswerpipealias]]`.
- **Email From** is the email from which the notification will be sent from. Respondent's answers can be piped in it using the `[[youranswerpipealias]]`.

- **Email Notification To** is the email that will receive the notification. If we want to send alerts to several emails we can separate emails through a ; char. Respondent's answers can be piped in it using the `[[youranswerpipealias]]`.
- **Email Notification Subject** is the subject of the notification email. Respondent's answers can be piped in it using the `[[youranswerpipealias]]`.
- **Attachments** is the attachment files that will be send along with this alert. There are no limits on the number of attachment that can be send along the alert. In order to add a new attachment click on the on the browse button, select your file and click on the update button of the alert. We must keep in mind to use small size attainments as some mailboxes are limited in size.

Answers Only Alert

This action will send a partial report where only the respondent answers will appear. This report will not show unanswered entries.

- **Email From Name** is the name from which the email the notification will originate. Respondent's answers can be piped in it using the `[[youranswerpipealias]]`.
- **Email From** is the email from which the notification will be sent from. Respondent's answers can be piped in it using the `[[youranswerpipealias]]`.
- **Email Notification To** is the email that will receive the notification. If we want to send alerts to several emails we can separate emails through a ; char. Respondent's answers can be piped in it using the `[[youranswerpipealias]]`.
- **Email Notification Subject** is the subject of the notification email. Respondent's answers can be piped in it using the `[[youranswerpipealias]]`.
- **Attachment** is the attachment files that will be send along with this alert. There are no limits on the number of attachment that can be send along the alert. In order to add a new attachment click on the on the browse button, select your file and click on the update button of the alert. We must keep in mind to use small size attachments as some mailboxes are limited in size.

Template Notification

This action will send the template that we've created. A template can be either composed of plain text or rich HTML. Each template can also contain any number of respondent answers through the use of pipe alias with the `[[youranswerpipealias]]` tag.

- **Email From Name** is the name from which the email the notification will originate. Respondent's answers can be piped in it using the `[[youranswerpipealias]]`.
- **Email From** is the email from which the notification will be sent from. Respondent's answers can be piped in it using the `[[youranswerpipealias]]`.
- **Email Notification To** is the email that will receive the notification. If we want to send alerts to several emails we can separate emails through a ; char. Respondent's answers can be piped in it using the `[[youranswerpipealias]]`.
- **Email Notification Subject** is the subject of the notification email. Respondent's answers can be piped in it using the `[[youranswerpipealias]]`.
- **Attachments** is the attachment files that will be send along with this alert. There are no limits on the number of attachment that can be send along the alert. In order to add a new attachment click on the on the browse button, select your file and click on the update button of the alert. We must keep in mind to use small size attachments as some mailboxes are limited in size.
- **Template** is body of the alert email that will be sent. Can be composed of respondent answers through pipe aliases.

Alert Triggers

Each alert can be triggered based on respondent answers. If no triggers are specified the alert will always be triggered.

Score

Scoring

Enabling scoring in FeedbackServer will allow us to set a given number of points for each selectable answer available in our questions.

The score will be calculated dynamically as the respondent is taking the survey. Generally scoring system are used in training institutions to create multiple choices tests. Score can be activated on our survey by checking the score checkbox from the general settings tab.

Lets see a small example in order to understand scoring. Suppose have a question with following answers and that each answer has specific score value.

What is the capital of Switzerland ?

- 1) Paris (scores 1)
- 2) Washington (scores 2)
- 3) Bern (scores 3)
- 4) Geneva (scores 4)

We can, based on the selected answer, know if the respondent has correctly answered this question. If his score on the question is 3 we know that he did answer the question correctly.

If we have multiple questions we can at the end of the survey use a thanks message condition to show a specific text based on the respondent total score eg: you've failed, you've passed.

Form creation

The first step after having installed FeedbackServer is to create a new form.

We can either create a new survey from scratch using the quick creation screen or we can import an existing form Xml file that has been exported using FeedbackServer's export function.

To create our first form we can follow these steps :


1. Create a form using the form creation wizard
2. Customize survey settings
3. Add a new question from the form builder using the question wizard
4. Customize your question settings
5. Repeat step 3 to 4 as much as you need it
6. Setup security using security add ins
7. Deploy the survey through a web link or by email
8. Gather results
9. Create a report
10. Analyze the results

General settings

The general settings allows us to setup and customize our form features.

- **Title** is used mainly as logical information to make the difference between one or the other of our surveys. Its not used or shown on the survey control seen by the survey respondents.
- **Deployment page title** is the title that will be displayed in the browser window title.
- **Page Header Image** can be used to display an image on top of the page that displays the survey.
- **Permanent redirection URL** can be used once a form has been closed or deleted to redirect all users trying to access the form to a specific URL.

- **Survey Language** forces a survey to run under a specific language culture which is not depended from the current server configured culture.
- **Style Template** lets us choose between different graphical styles we have created using the WYSIWYG Style Editor . The selected style will be applied to all elements in the survey from the security add ins interfaces, the survey control and completion message or public reports.
- **Open Date** is the date at which our survey will accept new entries. No entries will be accepted before that date, leave blank if our survey has no time limit to take it.
- **Close Date** is the date at which our survey will close and not accept entries anymore. No entries will be accepted after that date, leave blank if our survey has no deadline.
- **Buttons Row Display** let us choose the position of the survey buttons (e.g.: submit, next page etc..). We can either show them on the footer, the header or both on the survey.
- **Rich client / Ajax support** enables rich web 2.0 survey features like no page refresh, watermarks, entry masks. This feature should be turned off for surveys which require full accessibility level compliance.
- **Footer Progress Display** allows us to set what will be shown in the survey control footer. We can either choose to show nothing or to show specific information on the current progress by selecting page numbers, text or graphical percentage progress.
- **Disable Question Numbers** if we don't want to show the current question number on the left of each question in the survey control.
- **Score** can be enabled in order to let us specify a score for each answers and then see the scores for each respondent. If we want to know more about scoring make sure to read our Score Introduction.
- **Allow Respondents To Download Their Answers** using this feature we can allow the user to download all answers he has entered during the survey as word document when he finishes and submits the survey.
- **Show Summary Of Respondent's Answers** allow us to set the survey to show to the respondent a complete summary of his answers before he does the final submit. If navigation is activated he will be able to navigate back to change his answers, if its not activated a button allowing him to come back to page 1 and change his is made available.
- **Force https secure connection** forces a survey to run under the secure HTTPs protocol. Responsents will be automatically redirected to the equivalent secure deployment survey link.
- **Anonymous Respondents** let us disable the logging of respondents IP addresses.

 The IP Protection Addin will stop working correctly after activating this function as the same "dummy" IP will be assigned to respondents.

- **Clear Branched / Skipped Pages Answers** will clear all respondent answers on pages that have been already displayed once but skipped by the respondent at a later stage by navigating back and forth.
- **Always Display Submit Button** forces the display of the submit button.
- **Previous / Next Page Navigation** we can allow the user to navigate back and forth in the survey's pages while he is taking it. By default the user can only move forward in the survey and cannot come back to change his answers.
- **Display Next Page Button** forces the display of the next page button.
- **Display Previous Page Button** forces the display of the previous page button.
- **Page Navigation Type** displays all the pages and respondent choose which pages he wants to display.
- **Resume Of Progress** we can allow the user to save and resume his progress while taking our survey.

We can choose to save the unique code provided by Key Provider in the destination storage of the key provider. By default the FeedbackServer key provider's destination storage is a cookie, so when the respondent comes back he will not need to enter any code manually. The drawback of this approach is that he will need to resume the survey on the same computer when he saved the progress on.

We can also set the resume to manually, in this case a the unique code will be provided as a text message on screen and he will be able to resume at any time and from anywhere his progress by clicking the resume progress button on our survey.

- **Allow User To Change His Answers** Once a user has finished the survey and has reached the completion message, the user cannot change his answers.

However we can set FeedbackServer to allow the user to change his answers, in this case a unique code (UID) generated by the Key Provider will be shown with the completion message when the user has finished the survey. He can then log back at any time, enter his unique code and change his previous answers.

We can also choose not to show the unique code, this will allow us to customize and embed the code in the Completion Message using the `::keycode::` tag.

- **Display Save Progress Button** forces the display of the save respondent progress button.

- **Display Resume Progress Button** forces the display of the resume respondent progress button.
- **Display Restore Button** forces the display of the restore answers button.
- **Survey Description** let us write some descriptive text of what our survey is about. This text will be show at the survey home page.
- **Metatdata** let us fill the form metadata attributes that we have setup using the metadata editor

Form home

The home page provides general information about our survey and also provides an introduction to people who will take the survey if you have setup the introduction message from the general settings tab. The page will also show us the different reports we've created.

- **Web Access Link** is the link that we can use to access the survey from a browser. We may also go to the deployment section to get further ways to deploy our survey.
- **Survey Status** is the current status of the survey. Status can be either :
 - Open - will allow respondents to take the survey.
 - Closed - no respondents will be allowed to take the survey.
 - Deleted - no respondent can take the survey and it will only be available within the trashcan folder.
- **Creation Date** is the date on which the survey was created.
- **Last Entry On** is the date on which the last response to the form was recorded.
- **Average Response Time** average time taken by the respondents to fill the form.
- **Display Times** is the number of times the survey web control has been shown. The number is not per person / session, it will increase each time the survey control has been rendered.
- **Number Of Respondents** who filled our form.
- **Unvalidated Progress Entries** is the number of participants who have saved their progress to resume it later on but have not validated their answers yet. We can delete all unvalidated entries but this will also delete all their answers and respondents who saved their progress wont be able to resume it.

- **Delete Responses** Deletes all respondent answers that were posted for our survey. It is not possible to restore the answers once they have been deleted.
-
- **Delete** will delete our survey and all his questions, answers and respondent answers. It is not possible to recover a survey once it has been deleted. If the trashcan system is enabled surveys will be first moved to the trashcan.
- **Restore** when a survey has been moved to the trashcan we can restore it and enable it again as an active survey.
- **Clone** will create a full copy of our form settings and content. The respondent answers are not copied to the cloned form.
- **Open, Close** will open or close the survey to respondents.

Form export

The export page lets us export our form as an Xml format which we can later on re-use to re-import our form into another FeedbackServer installation or as a form template.

- **Include Respondent Answers** do we need to export respondent answers with the form structure.
- **Include Media** do we need to export images, animations with the form structure.
- **Include Metadata** do we need to export the form metadata with the survey structure.

Form Content Editor

Question wizard

Insert question

The insert question page is composed of several wizard which will let us add quickly new questions to our survey. One of the wizard, called the expert mode, allows us to create a "raw" question without any preset settings so we can fully customize it.

Copy Existing Question

FeedbackServer allows us to copy into the current survey an existing question either from another survey we own, from the Question Library or we can also link a panel to the survey.

Import XML Question

It is also possible to import an existing question from an Xml file that was exported using the export feature of the question editor.

Expert mode

The expert mode of the question wizard creation tool allows us to insert a new question inside our form and let us handle all addition and settings of the question manually.

We can choose between three types of questions :

1. Single Question
2. Matrix Question
3. Hidden Question
4. Pool Linked Question
5. Templated Question
6. Static Free Text

Question types

Static question

The static question is the most basic question type available in FeedbackServer.

It allows us to add any free text / pictures to our form. This question type cannot handle answers, it can only render static text or HTML to the respondent and integrate images.

Single question



A single question can be composed of any number of answer types. FeedbackServer does not have the concept of radio, checkbox or field question. FeedbackServer is much more modular and allow you to compose your own question based on the Answer Type you need inside your question.

Single Question Examples

A single question composed of field answer types and xml bound types (Country)

Please provide some information about yourself :

we will keep all your information confidential

Name :	<input type="text"/>
Email :	<input type="text"/>
Age :	<input type="text"/>
Country :	<input type="text" value="[Select Country]"/> 
Last time you eat an hamburger :	<input type="text"/> 


A single question composed of selection answer types with a radio button selection.

What's your current position in your company ?*

☐ C.T.O ☐ Developer ☐ Software architect ☐ Other :

A single question composed of selection answer types with a drop down selection.

What version of ASP.net are you using ?*

As you can see, by composing the answer types together you can almost create any kind of questions you want.

Matrix question

A matrix question is a question that can be composed of rows and column using the matrix question editor.

Each row is basically a single question and each column can be composed of any of the available answer types. Each row will share the same answer type of the column, it is not possible to have one row that has a different answer type inside the same column.

Matrix questions are handy when we run out of space in our page or if we have a set of questions that related together.

Matrix Question Example

A matrix question with a radio buttons layout and large field, it is also possible to have matrix questions with checkboxes to allow multiple choices.

How would you rate our ASP.net product ?*

	very bad	bad	ok	good	amazing!	n/a	suggestions
Features	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="text"/>
Usability	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="text"/>
Quality	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="text"/>
Performances	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="text"/>

Side by side matrix

Side by side matrix questions also known as 3D matrix question can be composed of multiple sub-matrix groups. Each matrix group can have its own set of answers and answer items and selections can be displayed either as radio buttons, checkbox or within drop down lists.

6. Let us know more about your condition

	Have you ever been diagnosed with these conditions		Are you taking medications		How strong were the symptoms
	yes	no	yes	no	
Hearth attack	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="text"/>
High blood pressure	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="text"/>
Pneumonia	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="text"/>

Pool linked question

The pool linked question allows us to virtually show at runtime a question that has been created within the question pool. The question shown will be picked in a random fashion from the linked pool questions that have been selected from the interface.

Once a respondent takes the survey one of the selected questions will be shown to him allowing us to create a "virtual" form with randomized questions shown to the respondent.

Templated question

The templated questions in FeedbackServer is one of the most powerful question types along with the free form question type. Question templates let us create a specific template based on any kind of html, plain text we want, once this templated has been created it will automatically be rendered at runtime using the current answer items available within our templated question.

In order to create a question we first need to setup a template in the Question Templates library.

The template management provides following options to setup :

- **Edition Language** templates can be different from each language and by selecting the language we can choose for which language we are currently editing the template.
- **Template Title** we can set the title of template shown when we select the template from the templated question screen.
- **Template Header** allows us to setup a header which will appear on top of our answers list.
- **Template Body** allows us to setup the body, this body will be repeated as many times as there are answers available for the question. Each body can have following tags which will be replaced at runtime.
 1. [fs:answeritemlabel] : Will be replaced at runtime by the label text of the answer being setup in the answer settings screen.
 2. [fs:answeritem] : Will be replaced at runtime by the answer of the question.
 3. [fs:answerstyle] : Will be replaced by the current answer style of the survey answers styles. Note that the style generated at runtime is a standard html style based text.
 4. [fs:questionstyle] : Will be replaced by the current answer style of the survey questions styles. Note that the style generated at runtime is a standard html style based text.

- **Template Alternating Body** is repeated each 2nd answer generated by the question.
- **Template Footer** allows us to set up a footer after all answers have been generated

Template Question Examples

Using following template FeedbackServer will generate following output for each templated question based on this template :

Template Header

```
<table style='[answerstyle]'
```

Template Body

```
<tr style= 'background:red'><td>[fs:answeritemlabel]</td><td  
valign="top">[fs:answeritem]</td></tr>
```

Template Alternating Body

```
<tr style='background:yellow'><td  
valign="top">[fs:answeritemlabel]</td><td>[fs:answeritem]</td></tr>
```

Template Footer

```
</table>
```

This simple HTML table based template will produce the output we can see below. We can of course use any kind of tags, text we want to create our own personalized template, there is really no limitation on how the output can look like.

4. Do you like templated questions ?

Yes these are great!	<input type="radio"/>
Not really	<input checked="" type="radio"/>
Please comment :	<div></div>

Hidden question

Hidden questions are questions being part of the form at runtime but which are not displayed to the respondent. These kind of questions can be used to collect data or make calculations that are no shown to the respondent.

Question library

Question library

FeedbackServer allows us to create a question library where we can store frequently used questions to avoid creating them again and again and re-use them from the question editor.

As the question library can be shared across all users in a multi-user configuration, it is also a good way to share questions with other users of FeedbackServer.

Library directory

The library directory allows us to create new directories where we could put our questions sorted by specific topics.

- **Add New Library** allows us to create a new library directory. At this time only one level directories are supported, it is not possible to create sub-directories.
- **Edit** will let us change the library name.
- **Library Name** is the name of the name of library as shown to the users.
- **Delete Library** will delete the directory including all its question templates.

Library templates

The library templates allows us to add new questions to a library that will be available as templates to be re-used from the Question Editor.

- **Insert New / Existing Question** allows us to add a new question to the template either by creating a new one or by copying an existing one from a survey.

Repeatable sections

Repeatable sections

FeedbackServer's repeatable sections allows us to enable the respondent to copy a question as many times as he needs it.

For example let's suppose we have a question to ask the respondent to enter information about books he has read during the past year. We would have two choices here, either we choose to add a huge number of answer fields to our question to make sure the respondent will have enough fields or we only add the required answer fields for a book one time and enable the repeatable sections. This will let the respondent have the opportunity to add as many new books answers as he needs it.



When exporting the results, data corresponding to each section are grouped by a same section number.

A repeatable section in "full repeat mode".

1. Please tell us what book you've read during the past year :

[Add Book](#)

Book Name :

Book Author :

Book ISBN :

[Add Book](#) | [Delete Book](#)

Book Name :

Book Author :

Book ISBN :

[Add Book](#) | [Delete Book](#)

Book Name :

Book Author :

Book ISBN :

A repeatable section in "grid repeat mode".

1. Please tell us what book you've read during the past year :Book Name : Book Author : Book ISBN :

Book Name :	Book Author :	Book ISBN :			
My Book	My Author	121312323	Add Book	edit	Delete Book
Nice Book	Nice Author	12132356	Add Book	edit	Delete Book

Sections management

Answer types

Selection based types

Selection types

Selection types will base their rendering on the selection mode that was chosen in the question editor. Depending on the selection mode the type will be rendered as a radio button, check box or grouped along with other selection - text types inside a drop down list.

Following selection types are available :

- Selection - Text
- Selection - Other
- Selection - Image Rating

Selection types share following properties :

- **Answer Text** is the text that will be shown next to the selection item (radio, check box) or inside the drop down list.
- **Image URL** we can give an image URL (<http://www.mydomain.com/myimage.gif>) in order to show it instead of showing the answer text.

- **Type** allows us to change the current type of the answer we want in our survey. Respondent answers already collected will not be deleted if we change the type.
- **Reporting Alias** is the text that can be shown instead of the answer text inside our reports.
- **Selected Answers** we can choose to have the selection item selected / checked by default.
- **Score Point** lets us specify a score value if the answer is selected. This score will be used to calculate the question's score and survey overall score of the respondent. This feature is only available if we have turned on FeedbackServer's Scoring features from the form settings page.
- **Rating Weight** let us define a numeric weight value that will be used in the rating / scale calculation inside reports. This feature is only available if we have turned on rating on the question of the answer type.
- **Fixed Display Order** allows us to stick the answer's display order and to avoid taking part in the randomization process of the answers. This feature is only available if we have turned on answer randomization on the question of the answer type.

Selection text

A selection type will base its rendering on the selection mode that was chosen in the question editor. Depending on the selection mode the type will be rendered as a radio button, check box or grouped along with other selection - text types inside a drop down list.

Selection - other

A selection other type will allow the user to enter custom text for his selection and selection rendering is based on the selection mode that was chosen in the question editor. Depending on the selection mode the selection will be rendered as a radio button, check box or grouped along with other selection - text types inside a drop down list.

We can set following properties to customize our selection other types :

- **Default Text Value** allows us to set a default value inside the text box field. It is also possible to fill the text box with a default value coming from an external source using following tags :

`##yourquerystringvariablename##` will set the default text with a query string variable's value.

`@@yoursessionvariablename@@` will set the default text with a session

variable's value.

&&yourcookievariablename&& will set the default text with a cookie variable's value.

%%servervariablename%% will set the default text with a server side variable's value.

- **Pipe Alias** allows us to specify an alias that we can use in other questions to pipe the text box field value.

Selection - image rating

Selection image rating types can be used to display an image instead of the standard radio button when used with rating enabled question types. All image rating types within the question will be used to calculate the rating chosen by the respondent.

We can set following properties to customize our selection other types :

- **Rating Image Style** allows us to set the type of image that we wish to display instead of the standard browser form item.

Field based types

Field types

Field types will allow us to ask the respondent for any free text he might want to enter or we want to receive. Field types range from very basic text boxes to complex field types like ranking fields or constant sum fields. It is also very easy to create new field types using the online type editor.

Following field types are available :

- Field - Basic
- Field - Large
- Field - Required
- Field - Email
- Field - Calendar
- Field - Ranking
- Field - Constant Sum
- Field - Hidden
- Field - Password

Field based types share following properties :

- **Answer Text** is the text that will be shown next to the field.
- **Image URL** we can give an image URL (<http://www.mydomain.com/myimage.gif>) in order to show it instead of showing the answer text.
- **Type** allows us to change the current type of the answer we want in our survey. Respondent answers already collected will not be deleted if we change the type.
- **Respondent Display Behavior** allows us to define how the type will be rendered to a respondent. We can either choose to show / hide the panelists answer, to allow a new entry by the respondent or to make it read only for existing panelist answers. This feature is only available when the type is linked to a panel attribute.
- **Default Text Value** allows us to set a default value inside the text box field. It is also possible to fill the text box with a default value coming from an external source using following tags :


##yourquerystringvariablename## will set the default text with a query string variable's value.

@@yoursessionvariablename@@ will set the default text with a session variable's value.

&&yourcookievariablename&& will set the default text with a cookie variable's value.

%%servervariablename%% will set the default text with a server side variable's value.

- **Watermark Text** allows us to define a watermark information text which will be shown to the respondent within the field before he enters his information. Note that the watermark is not saved as a response if respondent doesn't enter anything within the field and keeps it "blank". This feature is only available when the survey rich client mode has been enabled from the survey settings page.
- **Pipe Alias** allows us to specify an alias that we can use in other questions to pipe the text box field value.
- **Mailing Pipe Alias** is the alias that we can use inside our survey invitation body. The invitation engine will replace the given alias by its corresponding panelist value while sending the invitation. This feature is only available when the type is linked to a panel attribute.
- **Reporting Alias** is the text that can be shown instead of the answer text inside our reports.
- **Extended Report Filter** we will be able to use directly the text answers of the respondent as a filter for our reports.

- **Fixed Display Order** allows us to stick the answer's display order and to avoid taking part in the randomization process of the answers. This feature is only available if we have turned on answer randomization on the question of the answer type.
- **Regex Server Side Validation** allows us to choose one regular expression from the regular expression library to validate the respondent entry.
- **Field Entry Max Length** allows us to limit the number of characters that the respondent can enter in the field.
- **Mandatory** us to stick the answer's display order and to avoid taking part in the randomization process of the answers.
- **Forbid Duplicates** will check if the respondent entry has already been entered and saved for this answer on another form entry.
- **Field Width** overrides the width of the field answer type that has been setup in the answer type editor.
- **Field Height** overrides the height of the field answer type that has been setup in the answer type editor.
-  Let us choose from piping, dynamic content or external values we want to include in the text at runtime.

Field basic

This type will render a large text box that support multiple lines comments from the respondent. You can use this type if you expect a large amount of comments from your respondents. There is no maximum length limit by default. If we need a to change the size of the field or its properties it is possible to create our own field type using the answer type creator.

Field large

Field required

This type will render a standard text box. There is no maximum length limit by default. If we need a to change the size of the field or its properties it is possible to create our own field type using the answer type creator.

This field has a client side validation script to make sure it has been filled. It is recommended to also activate the server-side checking using the Mandatory option of the type settings.

field email

This type will render a standard text box. There is no maximum length limit by default. If we need a to change the size of the field or its properties it is possible to create our own field type using the answer type creator.

This field has a client side validation script to make sure that a correct email has been provided. It is also recommended to activate the Regular expression's email validation from the answer settings.

Field calendar

This type will render a standard text box with a calendar where respondent can pick a date from.

This field has a no client side validation script but a server-side validation is done to see if the text entered is a valid date. The field is not mandatory by default, if you want to make it mandatory you make check the mandatory option in the answer settings.

Field ranking

Ranking is a method that will allow us to setup and group multiple ranking fields together to ask the respondent a specific ranking based on the number of ranking field types we have linked in our question. Using the dynamic ranking type we can also generate interactive drag / drop based ranking questions.

Please rank these companies :

Swiss :	<input type="text"/>
Lufthansa	<input type="text"/>
British airways :	<input type="text"/>
Air France	<input type="text"/>

In the example above we have a question with four ranking fields linked using the connections of the type settings. The respondent will need to answer each field uniquely with a number from 1-4 to rank the flight companies.

In order to make ranking work we always need to have a "master" ranking type that will manage the values of the group of ranking fields. In this case the "master" type is "Swiss". "Swiss" has subscribed to the other ranking types value using the connections of the type settings.

The field is not mandatory by default, if you want to make it mandatory you make check the mandatory option in the type settings.

We can set following properties to customize our ranking types :

- **Display type** allows us to set the way a respondent can choose its ranking.
- **Max Ranking Count** allows us to set limit the required ranking number that the respondent needs to enter. By default the ranking count is the total of all ranking fields setup in the question.

Field constant sum

Constant sum is a method that will allow us to setup and group multiple constant sum fields together to ask the respondent to enter a values that will match the total value we've specified.

Please enter a constant sum up to 100

Quality :	<input type="text"/>
Welcome :	<input type="text"/>
Help :	<input type="text"/>

In the example above we have a question with three constant sum fields linked using the connections of the type settings. The respondent will need to answer each field with a number and the total of all fields must reach 100.

In order to make constant sum work we always need to have a "master" ranking type that will manage the values of the group of constant sum fields.

In this case the "master" type is "Quality". "Quality" has subscribed to the other constant sum types value using the connections of the type settings. We have also set the total of the sum to reach from the extended settings of the "Quality" type.

The field is not mandatory by default, if you want to make it mandatory you make check the mandatory option in the type settings.

We can set following properties to customize our ranking types :

- **Constant Sum To Reach** is the total value that the respondent must share across all the constant sum based answers in the question.

Field hidden

This type will not be shown to the respondent. The main purpose of this type is to gather extra information without having the respondent knowing it.

For example if we want to capture the browser used by the respondent we could add a Field - Hidden type to our question and by setting the default text to :
%%HTTP_USER_AGENT%% we would automatically capture the browser of the respondent for our results.

Field password

This type will render a password text box but from which text cannot be read by the respondent

Field compare

A compare field is a great way to compare two or more fields between themselves. One of the application is for example if you want to confirm an entry from a previous field with another one (e.g.: Password and password confirmation fields).

In order to setup compare answer we need to achieve following steps :

1. Add two answer item of type Field - Compare to our question
2. Edit one of these answer items and go to the connections tab
3. Once we are in the connection tab, choose from the publishers our second compare answers
4. From the extended properties we can choose which kind of comparison we want to validate between both fields including their data types.
5. Thats it! Our compare field is now ready to use .

We can set following properties to customize our ranking types :

- **Compare Condition** is type of comparison that will be done between the publisher answers chosen in the connection tab and the current answer for which the comparaison has been setup.
- **Compare Data Type** is the kind of type the system will use to do the comparison. Note that will not validate the field content, we may use the features offered by the entry validation tab to setup a specific entry validation check.

Field image code

This type will render an image that will show a random generated code that the user will have to type in to confirm that its a human respondent who is filling the form and not a bot.

Xml bound types

Xml types

Xml bound types are bound to an Xml file available on the server.

Xml bound type will read the Xml file to which they are bound to and show the choices inside a drop down list. It is very easy to create and bind new Xml bound types to your own Xml files using the online type creator.

Following xml bound types are available :

- Xml - Gender
- Xml - Country List
- Xml - US States

Xml files are based on the current language settings. FeedbackServer will try to read Xml file based on the language FeedbackServer is running in, before reading the default one. e.g.: If we bind our Xml type to a file called "country.xml" this will be our default file, if we switch to French for example we could have another Xml file called "country.fr.xml" that will be automatically read in when the survey is in French. This way it is very easy to provide translated versions of an single Xml bound type to different respondents based on their language.

Selection types share following properties :

- **Answer Text** is the text that will be shown next to the drop down list.
- **Image URL** we can give an image URL (<http://www.mydomain.com/myimage.gif>) in order to show it instead of showing the answer text.
- **Type** allows us to change the current type of the answer we want in our survey. Respondent answers already collected will not be deleted if we change the type.
- **Respondent Display Behavior** allows us to define how the type will be rendered to a respondent. We can either choose to show / hide the panelists answer, to allow a new entry by the respondent or to make it read only for existing panelist answers. This feature is only available when the type is linked to a panel attribute.
- **Default Text Value** allows us to set a default list item inside the drop down list. It is also possible to set a default list item with a default value coming from an external source using following tags :


`##yourquerystringvariablename##` will set the default text with a query string

variable's value.

@ @yoursessionvariablename@ @ will set the default text with a session variable's value.

&&yourcookievariablename&& will set the default text with a cookie variable's value.

%%servervariablename%% will set the default text with a server side variable's value.

- **Pipe Alias** allows us to specify an alias that we can use in other questions to pipe the text box field value.
- **Mailing Pipe Alias** is the alias that we can use inside our survey invitation body. The invitation engine will replace the given alias by its corresponding panelist value while sending the invitation. This feature is only available when the type is linked to a panel attribute.
- **Reporting Alias** is the text that can be shown instead of the answer text inside our reports.
- **Extended Report Filter** we will be able to use directly the text answers of the respondent as a filter in the General Report Settings. To learn more about piping and extended report filters we suggest reading the Extended Filter Introduction.
- **Fixed Display Order** allows us to stick the answer's display order and to avoid taking part in the randomization process of the answers. This feature is only available if we have turned on answer randomization on the question of the answer type.
-  Let us choose from piping, dynamic content or external values we want to include in the text at runtime.

Xml country

This type is bound to and will read the country xml file located in the XmlData directory to show a list of countries, it is also possible to create our own Xml bound type using the answer type creator.

If we want to select a default element in the drop down list we can do it by setting the "Default Text Value" property to a value that is available inside the bound drop down list.

Xml us states

This type is bound to and will read the United States xml file located in the XmlData directory to show a list of the United States, it is also possible to create our own Xml bound type using the answer type creator.

If we want to select a default element in the dropdown list we can do it by setting the "Default Text Value" property to a value that is available inside the bound drop down list.

Xml gender

This type is bound to and will read the gender xml file located in the XmlData directory to show a list of gender, it is also possible to create our own Xml bound type using the answer type creator.

If we want to select a default element in the drop down list we can do it by setting the "Default Text Value" property to a value that is available inside the bound drop down list.

Subscriber xml list

This type will load and bind dynamically an Xml file to a dropdownlist based on the value posted by the answer publisher its subscribing to.

Out of the box a subscriber type can be used to be connected to an Xml - Country List type publisher to get a list of regions based on the country that has been selected. e.g.: if United States are selected in the Xml - Country List, the Subscriber - Xml List type will receive US and try to load the us.xml file.

This type can subscribe to any answer publisher but its better to subscribe to Xml bound types.

Sql bound types

Sql types

Sql based query answer types are another powerful feature of FeedbackServer.

Using Sql based type you can expose and use existing data directly in your survey.

At this time the data will be exposed as a drop down list, you can choose if you want to have it mandatory or not. By default there aren't any out of the box Sql based answer types as these are related to your own database.

If you wish to create your own Sql type you will need to :

1. Go to the form builder / answer type editor section.
2. Click on create new type
3. Give it a new name eg : Customer list

4. Select Sql query as a datasource
5. Enter your Sql query eg: select customerid, customername from tbcustomers.
Only Sql "select" based queries are allowed.
6. If it requires a selection check the "selection required" box.
7. Create type.

You can now use this new answer type in any new or existing question.

Answer Piping

You can pipe answers from previous pages using the standard `[[pipealias]]` tags directly in your Sql query, so in our previous example we could have for example a form that ask the user on the first page for his country and we could then retrieve through piping on the other pages the list of customers of the selected country.

e.g. : select customerid, customername from tbcustomer where country = `[[countryalias]]`

Its also possible to use other piping tags like :

`##yourquerystringvariablename##`

`@@yoursessionvariablename@@`

`&&yourcookievariablename&&`

`%%servervariablename%%`

Security considerations

In some scenarios allowing select queries against a database can be a security threat. By default all FeedbackServer administrator can create / change sql based type.

If you create a new user you will need to give him explicitly the rights to create sql based answer type, its strongly recommended that if you don't give sql based answer type right to also remove Xml import rights for the user as a user could change the Xml file and inject his own Sql code.

You can also disable the feature for all admin from the web.config by setting the `SqlBasedAnswerTypesAllowed` to false.

List items bound types

List item collection

Item collection types let us create a custom collection of text / value items using an easy to use administration interface.

Unlike Xml bound type where we need to edit manually Xml files and then upload them to server to make them available, item collection types let us add and edit collection members directly on the server.

If you wish to create your own collection based type you will need to :

1. Go to the form builder / answer type editor section.
2. Click on create new type
3. Give it a new name eg : Country list
4. Select list items as a datasource
5. Choose the layout that will be shown to the user
6. Create type.
7. Once you created the type you can select it again to add items to its collection

You can now use this new answer type in any new or existing question.

Item Collection Editor

The item collection editor let us add, edit or remove items from the collection of the type

- **Item Text** is the text that will be shown to the user.
- **Item Value** is the value that will be stored inside the database once the respondent has submitted his results.
- **Edition Language** allows us to edit the item texts or values in a specific language. Feedback Server® will automatically show at runtime the right version to the user based on the language he has chosen. In order to edit an item in a specific language, select the edition language in which you want to edit the item, click on the edit button to edit the item, enter its translation and click update to save the translation to the database.

Misc. types

Filed upload

This type will allow the respondent to upload any number of files. There are no limitation on the number of file upload types per question or per page in a form. You can manage the uploaded file using the file manager .

This field has no extra client side validation check but if we require the respondent to upload at least one file we can check the mandatory check box inside the type settings.

Type Settings

- **Answer Text** is the text that will be shown next to the upload field.
- **Image URL** we can give an image URL (<http://www.mydomain.com/myimage.gif>) in order to show it instead of showing the answer text.
- **Type** allows us to change the current type of the answer we want in our survey. Respondent answers already collected will not be deleted if we change the type.
- **Reporting Alias** is the text that can be shown instead of the answer text inside our reports.
- **Fixed Display Order** allows us to stick the answer's display order and to avoid taking part in the randomization process of the answers. This feature is only available if we have turned on answer randomization on the question of the answer type.

Extended Type Settings

Extended type settings are extra specific properties of a type that are only available once a type has been created.

To edit the extended settings of a type we will have to click on the edit button of the answer overview once we've added our type to our question or changed an existing type to a new one by clicking the update button.

- **Max. File Upload Number** is the number of files a respondent can upload for this type instance.
- **Max. File Size** is the maximum size in bytes allowed for upload. Note that by default ASP.net maximum size is 4mb, if we want to allow bigger upload. You may check following link on how to allow bigger uploads on your website.
- **Accepted Content Type** let us specify if we want to accept only specific files eg: pdf, gif etc ..

Boolean

This type will render a check box. If the check box is checked the value "true" will be saved as an answer, if it's not checked the value "false" will be saved as an answer.

Birthday

This type will render a set of drop down lists allow the user to select month, day and year that will be saved as a date.

Answer types

Answer types will let us create our questions as we want in order to get the feedback we need. FeedbackServer provides out of the box a large number of answer types that can be used to compose a question. Each item has its specific set of properties and all items can be composed together in the same question.

Here are the items provided out of the box :

- Selection - Text
- Selection - Other
- Field - Basic
- Field - Large
- Field - Required
- Field - Email
- Field - Calendar
- Field - Ranking
- Field - Constant Sum
- Field - Hidden
- Field - Password
- Field - Image code
- File - Upload
- Boolean
- Xml - Country List
- Xml - US States
- Subscribers - Xml List

It is also possible to create new items either using the answer type creator or through the FeedbackServer SDK. In order to create reduce choices lists we may use the answer skip logic system to hide answers based on survey conditions.

Create answer types

FeedbackServer provides us a way to create our own answer types without any programming knowledge. Using the type creator we can create four types of answer types that we can then re-use inside our questions :

- Selection Types
- Field Types

- Xml Bound Types
- Sql Bound Types
- List Item Collection Types

Following properties can be set to customize each of our new or existing answer type.

Selection Types

- **Field Name** is the name of the type that will be shown in the type selection inside the answer editor.
- **Allow Selection** if we are creating a selection type we need to check this option.
- **Field Entry** is our selection type offering an alternate entry like the Selection - Other type.

Field Types

- **Field Name** is the name of the type that will be shown in the type selection inside the answer editor.
- **Allow Selection** if we are creating a field type we don't need to check this option.
- **Field Entry** we need to check this option to display the extra settings related to field types.
- **Field Width** of our field.
- **Field Height** of our field. If the height is more than one our field will be automatically rendered as a multi-line text box.
- **Javascript Function Name** is the name of the javascript function that will be called to validate the field content. The javascript function should be defined inside the javascript code option.
- **Error Message** is the error message that will show up if the javascript function return false.
- **Javascript Code** is the javascript code that will validate the field content. The function must return true if the condition is matched or false if the method could not validate the content of the field.

Code Example to validate a mandatory field :

```
function isFilled(sender)
{
    if (sender.value.length == 0)
    {
        sender.focus();
        return false;
    }
    else
    {
        return true;
    }
}
```

Xml Bound Types

- **Field Name** is the name of the type that will be shown in the type selection inside the answer editor.
- **Collection Layout Rendering** is the user interface mode in which the data items will be displayed to the user inside the survey.
- **Xml File Name** is the file name of the Xml file that the type will be bound to.

Note that the Xml file has to be in the directory specified by FeedbackServerXmlDataPath element of the web.config. At this time it is not possible to edit or create Xml files through the administration interface.

The format of the Xml file must be as following :

```
<?xml version="1.0" standalone="yes"?>
<FSDataSource xmlns= "http://www.feedbackserver.com/FSDataSource.xsd">
<XmlDataSource>
    <RunTimeAnswerLabel>Label To Show : </RunTimeAnswerLabel>
    <XmlAnswers>
        <XmlAnswer>
            <AnswerValue></AnswerValue><AnswerDescription>[Select an
answer]</AnswerDescription>
        </XmlAnswer>
        <XmlAnswer>
            <AnswerValue>yourvalue1</AnswerValue><AnswerDescription>your
answer</AnswerDescription></XmlAnswer>
        </XmlAnswers>
    </XmlDataSource>
```

```

    <AnswerValue>yourvalue2</AnswerValue><AnswerDescription>your
second answer</AnswerDescription></XmlAnswer>
  </XmlAnswers>
</XmlDataSource>
</FSDDataSource>

```

Sql Bound Types

- **Field Name** is the name of the type that will be shown in the type selection inside the answer editor.
- **Collection Layout Rendering** is the user interface mode in which the data items will be displayed to the respondent.
- **Sql Query** is the Sql query that will populate the drop down list.

List Items Types

Field Name is the name of the type that will be shown in the type selection inside the Answer Editor.

Collection Layout Rendering is the user interface mode in which the data items will be displayed to the user inside the survey.

Answer skip logic conditions

Skip logic conditions allow us to setup logical rules based on respondent's answers, session, querystring, user, group or language to hide an answer to the respondent.

Each condition is based on a set of rules that you can define. The first condition that will be met will hide the answer to which the skip logic applies. You can have as many conditions as you wish and order or re-order them at any time.

Entry validators

Beside providing us with a rich answer types system FeedbackServer also provides us with extensive ways of validating respondents entries in order to have gather the highest possible data quality.

Entry validators are server side add ins that does a certain validation depending on what they have be developed for. The main advantage of entry validators is that any developers can develop new entry validator using FeedbackServer's SDK and these can then be used right away on any available answer type.

Here are entry validators provided out of the box :

- **MSSQL - Integer**
Entry value from whole numbers data from -2^{31} (-2,147,483,648) through $2^{31} - 1$ (2,147,483,647)

- **MSSQL - Big integer**
Entry value from whole number data from -2^{63} (-9223372036854775808) through $2^{63}-1$ (9223372036854775807)
- **MSSQL - Small integer**
Entry value from 2^{15} (-32,768) through $2^{15} - 1$ (32,767)
- **MSSQL - Tiny integer**
Entry value from 0 through 255
- **MSSQL - Decimal**
Entry value from fixed precision and scale numeric data from $-10^{38} + 1$ through $10^{38} - 1$
- **MSSQL - Float**
Entry value from floating precision number data from $-1.79E + 308$ through $1.79E + 308$
- **MSSQL - Real**
Entry value from floating precision number data from $-3.40E + 38$ through $3.40E + 38$
- **MSSQL - Bit**
Entry value with either a 1 or 0 value
- **MSSQL - Date time**
Entry value from January 1, 1753, through December 31, 9999
- **MSSQL - Small date time**
Entry value from January 1, 1900, through June 6, 2079
- **MSSQL - Money**
Entry value from data values from -2^{63} (-922,337,203,685,477.5808) through $2^{63} - 1$ (+922,337,203,685,477.5807)
- **MSSQL - Small money**
Entry value from data values from -214,748.3648 through +214,748.3647

Pages

Page Behaviors

Update answers

The update behavior generates a button that will submit respondent answers for update. Note that this button is only available once a respondent has his form set in change mode to update an existing answer set.

Behavior Options

- **Button Text** overrides the default text of the button.
- **Image URL** let us chose an image that will replace the standard button input
- **Class Name** overrides the default css that is applied to the button.
- **Default Postback Trigger** let us chose if this is the button that will be triggered by default when the respondent presses the return key.

Countdown timer

The count down timer will display the time left to fill the form.

Dropdown page navigator

The drop down page navigator will display a drop down list with all the pages to let the respondent navigate across the different pages.

Next page

The next page generates a button that will take the respondent to the next page of our form.

Behavior Options

- **Button Text** overrides the default text of the button.
- **Image URL** let us chose an image that will replace the standard button input
- **Class Name** overrides the default css that is applied to the button.
- **Default Postback Trigger** let us chose if this is the button that will be triggered by default when the respondent presses the return key.

Page Navigator

The page navigator will display a table with all the pages to let the respondent navigate across the different pages.

Behavior Options

- **Display Mode** display the box either vertically or horizontally.

Previous page

The previous page generates a button that will take the respondent to the previous page of our form.

Behavior Options

- **Button Text** overrides the default text of the button.
- **Image URL** let us chose an image that will replace the standard button input
- **Class Name** overrides the default css that is applied to the button.
- **Default Postback Trigger** let us chose if this is the button that will be triggered by default when the respondent presses the return key.

Progress indicator

The progress indicator will display the current progress of the respondent in the form.

Behavior Options

- **Progress Display** display the progress either using the current pages, as a percentage or as a percentage graphic bar.

Restore answers

The restore answers generates a button that will allow the respondent to restore a previously saved answer set.

Behavior Options

- **Button Text** overrides the default text of the button.
- **Image URL** let us chose an image that will replace the standard button input
- **Class Name** overrides the default css that is applied to the button.
- **Default Postback Trigger** let us chose if this is the button that will be triggered by default when the respondent presses the return key.

Resume progress

The resume progress generates a button that will allow the respondent to resume a previous form state.

Behavior Options

- **Button Text** overrides the default text of the button.

- **Image URL** let us chose an image that will replace the standard button input
- **Class Name** overrides the default css that is applied to the button.
- **Default Postback Trigger** let us chose if this is the button that will be triggered by default when the respondent presses the return key.

Review answers

The review answers generates a button that will allow the respondent to display a page with all his current answers for review.

Behavior Options

- **Button Text** overrides the default text of the button.
- **Image URL** let us chose an image that will replace the standard button input
- **Class Name** overrides the default css that is applied to the button.
- **Default Postback Trigger** let us chose if this is the button that will be triggered by default when the respondent presses the return key.

Save progress

The save progress generates a button that will allow the respondent to save his current progress and answers in the form.

Behavior Options

- **Button Text** overrides the default text of the button.
- **Image URL** let us chose an image that will replace the standard button input
- **Class Name** overrides the default css that is applied to the button.
- **Default Postback Trigger** let us chose if this is the button that will be triggered by default when the respondent presses the return key.

Submit answers

The submit answers generates a button that will allow the respondent to save his answers and submit the form.

Behavior Options

- **Button Text** overrides the default text of the button.

- **Image URL** let us chose an image that will replace the standard button input
- **Class Name** overrides the default css that is applied to the button.
- **Default Postback Trigger** let us chose if this is the button that will be triggered by default when the respondent presses the return key.

System messages

System messages will display any error or confirmation messages related to the form.

Pages

FeedbackServer provides a very flexible paging system which allows us to insert various page types into our form flow.

Currently you can use any of the following pages any times you wish in your forms :

- Flow page
- Free layout page
- Reports page
- Respondent answers

Flow page

A flow page is a page that has a fixed question display layout going from the top to the bottom of the page with a header and/or footer showing the different actions (submit, next page) that the respondent can do. The layout can't be changed.

Free layout page

Free layout based pages allows us to layout and position our questions anywhere within the canvas as such we have full control over the layout and how we want to display our questions.

Beside questions we can also add specific actions in our page using behaviors. Behaviors are component that we can use in our free layout based form to trigger actions or to display specific information to the respondents.

Following behaviors are available out of the box :

Update answers
Countdown timer
Drop down page navigator

Next page
Page navigator
Previous page
Progress indicator
Restore answers
Resume progress
Review answers
Save progress
Submit answers
System messages

Free layout based pages support fully extra page options

Question Options

- **Display Header / Footer** let us chose if we want to display the standard form buttons displayed on the top or at the bottom of our forms.
- **Use default page navigator** let us chose if we want to display the default page navigator (box or drop down) in case we have activated page navigation in our form.

Reports page

Respondent answers

Content edition

FeedbackServer's content editor allows us to create and manage our survey forms in an easy way.

Each form is composed of types of questions that are themselves composed of types of answers. Unlike other survey packages there is no limitation as you can mix different answer types inside a question.

You could for example mix field answer types and selection answer types or SQL bound types in the same question. This mixed system will give you a maximum of freedom to create your forms as you need it.

Form builder

The form builder is the heart of FeedbackServer as it allows to create and edit the survey form we are going to deploy and that our user will take.

By default FeedbackServer offers already many types of questions and answer items to build our forms and using the SDK its also possible to develop our own custom answer items.

Preview / Test

This screen allows us to test our form and display it the same way as the final output In preview / test mode no answers are saved.

Page Options

- ▲ will move the page up until it reaches the next page break above.
- ▼ will move the page down until it reaches the next page break below.
- **Delete** will delete the page.
- **Clone** will delete the page.
- **Disabled** will delete the page.
- **Insert Question** inserts a new question at the end of the page. If we want to insert a question before another one you may use the insert question link at the question options level.
- **Insert Line Breaks** insert a new line break at the end of the page. Behind the scene a line break is just a static question that shows an HTML line break tag.
- **Edit Branching** allows us to setup branching rules. If we have multiple pages we can redirect the respondent to a defined page based on his answers. This option is only available if you have multiple pages in your form.

Question Options





- ▲ will move a question's position up.
- ▼ will move a question's position down.
- **Edit Question** depending on the type of question we will able to use either the question editor or to the matrix question editor.
- **Edit Answers** on non-matrix based type questions.
- **Edit Rows** of a matrix based question.
- **Edit Columns** of a matrix based question.





- **Edit Matrix Group** allows us to edit the groups that compose a side by side based matrix
- **Delete the question**, its answers and all related respondent answers. Note that it's not possible to recover the question or its answers afterward.
- **Clone** makes an exact copy of the question.
- **Disable** the question will hide it from the respondent and make it inactive
- **Insert Question** inserts a new question before the current question. If we want to insert a question at the end of the page you may use the insert question link at the page options level.
- **Insert Page Break** inserts a page break before the current question.
- **Insert Line Break** inserts a new line break before the current question. Behind the scene a line break is just a Static Question that shows an HTML line break tag.
- **Skip Logic** let us creates Skip Logic Conditions to hide the question based respondent's answers on the previous pages. This option is only available if you have multiple pages in your form.

Free form builder

Sometimes its just easier to have a complete control over how our forms will look like. Thanks to the free form builder of FeedbackServer we have all the tools to build freely and without any limitation the forms we need.

Form Builder Functions

-  Reload will reload the latest saved version on the server of our form.
-  Save will save the form on the server. Note that redo/undo features are reset after saving to the server and previous forms cannot be recovered once a new version is saved.
-  Background color allows us to change the background color of the form elements which support it. Current elements support background color are table, cells and html zone elements.
-  Hide table editing borders allows us to preview our form without the editing borders of the tables.




-  **Enable / Disable Undo** us step disable the undo / redo features of the editor as undo can currently be a resource intensive operation and can slow down the browser. Note that Internet Explorer is currently not supported by the undo / redo functions.
-  **Undo** let us step back in our editing process. Currently the form builder supports up to 5 history undo.
-  **Redo** let us go forward in the editing process once we used the undo feature.
-  **Reset form** will reset the form to its Feedback Server initial state which is a single empty table.

Form Elements

FeedbackServer forms can be created using any answer type provided by FeedbackServer or any type created either using the answer type editor or the SDK.

Each type can be drag/dropped on the interface and maybe re sized. Using the right mouse button we may also change the properties of the form item which have been dropped on the designer zone.

Custom form elements are also available, out of the box current elements are available :

-  **Text / Html** let us drag/drop and edit a text zone within the form. We can use and show any HTML we want.
-  **Table** let us drag/drop and edit a table. Tables can be placed anywhere within the designer screen including within other tables. Nested tables are fully supported by Feedback Server.
-  **Image** allows us to drag/drop and edit an image from the Feedback Server image library.

Note that its very easy to create and add new elements using the client side form builder framework of Feedback Server.

Form Behaviors

Form behaviors are components within FeedbackServer that control the flow of the form or allow complex interactions between the respondent and the forms. We can for example add progress based behaviors to display the current respondent progress or we can also add custom buttons to let the respondents save their progress, submit their answers.

Current limitations / Known Issues

- Undo / Redo features are only available under Firefox.
- Changing properties on existing answer elements can not be undone using the undo/redo features.
- Form builder is not fully working when running administration in right to left languages eg: Arabic, Hebrew.

Question editor

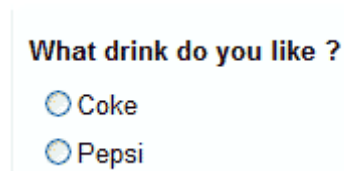
The question editor let us setup the configuration and constraints of our question.

If we want to add answers to the question we will need to use the Answer Editor after having created our question.

Question Options

- **Selection Mode** is the mode in which Selection Answer Types will be rendered by the FeedbackServer engine.

The radio button mode will show Selection Answer Types inside the question as radio buttons. The respondent can select only one answer in this mode.

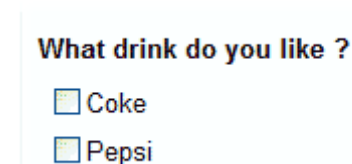


What drink do you like ?

☐ Coke

☐ Pepsi

The check box mode will show Selection Answer Types inside the question as check boxes. The respondent can select multiple answers in this mode.



What drink do you like ?

☐ Coke

☐ Pepsi

The drop down list mode will group and show Selection Answer Types inside a

drop down list. The respondent can select only one answer in this mode.

What drink do you like ?

[Select an answer] ▼

- **Edition language** lets us select in which mode we are editing the question texts. This feature is only available if we have turned on FeedbackServer's Multi-Language features.
- **Exclude Answer** let us define which answer within the question will act as "none of the other" answer. The answer defined will, when running the survey, uncheck all other checked answers of the question when its checked by the respondent and wont allow any other selection on the question answers. This feature is only available for multiple choice type questions.
- **Display Mode** is way we want to layout the answer items inside the question. We can either choose to order the layout in a vertical manner or to order the layout in an horizontal manner.
- **Show Question Header** let us hide the question header at runtime. Note that question having their header hidden will be skipped by the question numbering process.
- **Number Of Columns** is the number columns in which the question layout will be split into.
- **Fields Layout** we can show Field Answer Types of the question in two different ways to the respondent.
In the left to right mode the answer item label will be shown on the same line as the text box.
In the top to bottom mode the answer item label will be shown on top of the text box.
- **Randomize Answers Order** we can choose to show the answer's in a random order to each respondent to avoid the "order bias effect".
- **Rating Weight** we can activate the answers rating / scaling options in our question's Selection Answer Types. To know more about rating and scaling we suggest reading the Rating / Scaling Introduction.
- **Min. Selection Required** is the minimum answers that are mandatory to select in the question. Only Selection Answer Types are calculated in the selection number count, if we want to make Field Answer Types mandatory we can do it by checking the mandatory check box in the Field Type answers options.
- **Max. Selection Allowed** is the maximum of answers that can be selected in the question. Only Selection Answer Types are calculated in the selection number count, if we want to make Field Answer Types mandatory we can do it by checking the required field in the Field Type answers options.

- **Pipe Alias** we can specify an alias that we can use to pipe respondent question's answers in subsequent questions labels or answers labels.
- **Hide Validation Asterix** will hide the default * char shown after each question that has mandatory answers.
- **Question Disabled** will not use the question in the form at run time.
- **Reporting Alias** we can specify an alias that we can afterward show up in the reports instead of showing the full question text.
- **Metadata** let us fill the metadata attributes that were setup for questions using the metadata editor.

Matrix question editor

The matrix question editor let us setup the configuration and constraints of the parent question of our matrix based questions.

Matrix Question Options

- **Column Header** we can display a custom text on top of the matrix rows.
- **Question Row Width** will set the same width for all question text rows. Setting up to 0 will hide the question text row.
- **Calculation Mode** FeedbackServer will generate extra columns and rows to show the total of respondent entries with all text box columns available in the matrix.
- **Rotate** matrix will rotate the layout between the rows and columns.
- **Randomize Row Order** lets us randomize the row order when these are displayed to the respondent.
- **Hide Validation Asterix** will hide the default * char shown after each question that has mandatory answers.
- **Multiple Choices Matrix** is the mode in which the matrix Selection Answer Types columns will be rendered by the FeedbackServer engine. By default these types are rendered as radio buttons but we can also render check boxes to allow multiple choices.
- **Rating** we can activate the answers rating / scaling options in our question's selection answer types.
- **Repeatable Matrix Section** allows us to specify if the respondent will be able to duplicate the matrix in order to provide more answers.

- **Min. Selection Required** per row is the minimum answers that are mandatory to select in each row. Only Selection Answer Types are calculated in the selection number count, if we want to make field based answer types mandatory we can do it by checking the mandatory check box in the field type answers options.
- **Max. Selection Allowed** per row is the maximum of answers that can be selected in each row. Only Selection Answer Types are calculated in the selection number count, if we want to make field based answer types mandatory we can do it by checking the Required field in the field type answers options.
- **Reporting Alias** we can specify an alias that we can afterward show up in the reports instead of showing the full question text.

Answer editor

The answer editor allows us to add new answers to our question. By default FeedbackServer offers already many types of answer items but as you can see in the Form Architecture it is very easy to develop and add our own answer items inside FeedbackServer.

Answer Overview

- **Edition Language** lets us select in which mode we are editing the answers label and default texts. This feature is only available if we have turned on FeedbackServer's Multi-Language features.
- **Quick Translate** lets us select translate the answer text into multiple language from one single screen. This feature is only available if we have turned on FeedbackServer's Multi-Language features.
- **Delete** will delete the answer and all respondent answers related to this answers. Its not possible to recover it once it has been deleted.
- **Add New Answer** allows us to add a new answer type to our question.

Answer Settings

As each answer item has its specific set of properties and behavior we suggest to read the documentation of each type individually as you need it to build your survey.

- Selection - Text
- Selection - Other
- Field - Basic

- Field - Large
- Field - Required
- Field - Email
- Field - Calendar
- Field - Ranking
- Field - Constant Sum
- Field - Hidden
- Field - Password
- Field - Image code
- File - Upload
- Boolean
- Xml - Country List
- Xml - US States
- Subscribers - Xml List

Page options

FeedbackServer allows us to setup specific features for each page in our form.

- **Page Title** is the title that will be show on the page navigator if paging has been enabled on the form
- **Disable Response Timer** will stop the timer from counting if a timer has been setup for the form
- **Page Disabled** will not use the page at runtime
- **Show Submit Button** allows us to show the submit button on the page instead of showing a next page button. This option is only available on multiple pages forms. Its generally used to finish the survey on a given page where a respondent was redirected using branching.
- **Random Questions** order we can enable the page to show the questions inside it in a random order.
- **Display Header / Footer** allows us to hide the default header / footer (buttons, progress ..) on free form based pages
- **Use Default Page Navigator** allows us to hide the page navigator boxes or drop down lists on free form based pages if navigation is enabled on the form.

Regular Expression Editor

A regular expression is a string based language that allows to check if a given text matches the expression or not. We can check a string against any regular expression combination like emails, numbers, zip codes etc ... Almost any field based type in FeedbackServer can be validated against a regular expression created using the regular expression editor.

To learn more about regular expressions you may visit following site :
<http://www.regexlib.com/>

Edit Regular Expression

- **Name** is the name that the regular expression will have in Feedback Server® regular expression library.
- **Regular Expression** is the regular expression that will validate the text entry.
- **Error Message** is the error message that will be shown to the respondent if his entry didn't match the regular expression.

Test Regular Expression

- **Regular Expression** is the regular expression we want to test.
- **Value To Test** is the value we want to test against our regular expression.

Pipe source selector

The pipe source selector is an easy to use interface that will let us select from different tag sources that will be replaced at runtime with the current value based on the respondent state. We can choose between several piping sources :

Answer Pipe Value

Let us choose from a list of available answer pipe values in order to pipe a respondent text.

- **From Previous Pages** show all answers that have a pipe alias and that are available on the previous pages that the current question's page.
- **All Pages** show all answers that have a pipe alias and that are available on any pages of the form.

Question Answers Pipe

Let us choose from a list of available question pipe values in order to pipe a respondent answer(s) to a question.

- **From Previous Pages** show all questions that have a pipe alias and that are available on the previous pages that the current question's page.
- **All Pages** show all questions that have a pipe alias and that are available on any pages of the form.

Dynamic Content

Let us choose from a list of available dynamic content tags in order to pipe a dynamic text that we've setup based on the respondent's answer.

- **From Previous Pages** show all dynamic content tags that are available on the previous pages that the current question's page.
- **All Pages** show all dynamic content tags that are available for any pages of the form.

Querystring Variable Value

- **Variable Name** is the name of querystring variable from which the value will be piped into the text.

Session Variable Value

- **Variable Name** is the name of session variable from which the value will be piped into the text.

Cookie Variable Value

- **Variable Name** is the name of cookie variable from which the value will be piped into the text.

Web Server Variable Value

- **Variable Name** is the name of web server variable from which the value will be piped into the text.

Panel Attribute Value

- **Panel** is the panel from which we will choose the attributes which value will be piped into the text.

Form printer

The form printer feature let us generate a Microsoft Word compatible document of our forms.

- **Show Question Numbers** will generates the current question number in front of each question.
- **Use Survey Page Breaks** will insert the same page breaks that were setup in the survey.
- **Show Data Source Items** let us choose if we want to use a "_____" text for data source item or if we want to show the full list. Note that in case of country list this list can generate very long documents.
- **Export Language** is the language in which the survey will be rendered.
- **Print As Web** will generate a web friendly print output .

If you want to generate PDF's from your word document we suggest using the free CutePDF printer for Windows.

Entry mask

Entry masks is a way to pre fill a field with a specified format and force the respondent to enter only answers corresponding to the mask that was setup.

- **Mask** is the formatting mask used to pre-fill the fields. A mask can be composed of following template chars.
 - 9 - Only a numeric character
 - L - Only a letter
 - \$ - Only a letter or a space
 - C - Only a custom character (case sensitive)

A - Only a letter or a custom character
 N - Only a numeric or custom character
 ? - Any character
 / - Date separator
 : - Time separator
 . - Decimal separator
 , - Thousand separator
 \ - Escape character
 { - Initial delimiter for repetition of masks
 } - Final delimiter for repetition of masks

If we want to respondent to enter only numeric based chars within a field we can use for example a mask of : 9999 which will ask the respondent to enter 4 chars.

- **Mask Type** allows us to specify a specific validation on the mask currently number, date, date / time and time validations are supported.
- **Accept Negative Values** allows us to specify if we need the negative sign to be shown for numeric values. None wont show any negative sign, Left will show the sign of the left of the mask and right on the right of the mask.
- **Display Currency** allows us to show a currency symbol next to our mask.
- **Valid Charset** is the set of chars allowed for the C (custom), A, N chars used within a template.
- **Clear Text On Invalid Entry** setup the mask to clear its container field if the respondent enters an invalid entry.
- **Clear Mask On Lost Focus** allows us to clear the mask as soon as the focus on the field is lost.

Question pool

Each form can be built using a set of static questions but in more complex forms we might need to generate their structure dynamically.

The question pool feature allows us to add any number of preset questions to our pool. Once we have filled our pool with a number of preset questions we will be able to use them in our form by adding any number of pool linked questions in our form. Each pool linked question will then render in a random fashion one of the question we have made available in the pool.

In case we add more than one pool linked question, questions will be still shown in a random fashion for each pool linked questions but only once through all the pool linked questions and only if there are still questions left within the pool.

Linked panel question editor

The linked panel question editor let us setup the configuration and constraints of a question that comes from a linked panel structure. As these questions are linked directly to a panel's attributes we cannot add or change the answers of it.

Linked Panel Question Options

- **Edition language** lets us select in which mode we are editing the question texts.
This feature is only available if we have turned on FeedbackServer Multi-Language features.
- **Allow Panelist Creation** let us choose whether the respondents answers will be used to create and add a new panelist to the linked panel.

Folder Management

Folders

FeedbackServer has integrated support for folders. Folders will allow us to sort and organize our surveys by creating backup, archive folders for example. We can create or nest any number of folder we need to.

Forms or folders can be easily moved from one folder through another using drag & drop. We can even move several surveys at one time by selecting them and keeping the control key pressed before moving them to their new folder.

For large organizations we recommend giving folder root creation rights only to people with higher privileges and give only access rights to each department folders to the users responsible for the form creation in that department.

Folder Settings

- **Folder Name** is the name of the folder as it will appear in the left folder / survey pane.
- **Sort Type** is the order in which the surveys will be sorted in the display.
- **New Childs Inherits Users and Group Access** once checked all newly created surveys and folder within that folder will automatically get the same user and groups access rights of this folder. e.g. : If you have set this folder to give access to the group "makerting" all subsequent surveys or folder that will be created in this folder will be assigned to the "marking" group as well. Note that existing surveys or folder prior enabling this feature will not automatically inherits the access rights, we would need to do this manually from the user and group management interface .

Trashcan

FeedbackServer offers a trash can system to let us keep track of deleted forms or folders and restore them in case these were deleted by error. We can choose to either enable or completely disable the trash can from the system settings. If enabled, each time a user deletes a form or folder these will be first moved to the trash can where we will have following options :

Trashcan Settings

- **Empty trashcan When Survey Count Reaches** is a feature that let us setup the maximum number of surveys we do allow to be stored within the trash can.

FeedbackServer

Once this count is reached FeedbackServer will automatically deletes all surveys available and empty completely the trash can.

Deployment / Distribution

Deployment

FeedbackServer provides several ways to deploy our form to our customers.

- Web link Deployment provides us with a HTML link to our survey. We can then use this link anywhere from emails, messenger or to our website html pages.
- Friendly Link Deployment provides us with a HTML link that we can customize to create short urls. We can then use this link anywhere from emails, messenger or to our web site html pages.
- Popup Window Deployment provides us an invitation popup window that we can show on a specific frequency to our web site visitors.
- Fill Form is a special page inside FeedbackServer's administration that is meant to allow a registered FeedbackServer User to take a survey once he logged inside FeedbackServer. This page requires the user to have the "take survey" right set for a given Role
- Web Control Deployment is for more experienced users who like to customize the survey ASP.net control and integrate it inside their ASP.net pages.

Link deployment

FeedbackServer provides us with a HTML link to our survey. We can then use this link anywhere from emails, messenger or in our website HTML pages.

You can copy & past the link provided and use it in almost any application that supports HTML links.



The form id is unique and it is not possible for the respondent to alter the link in order to find out what other surveys we've created.

Friendly link

Sometimes its not possible to use the default form link. For example if we need our respondent to easily remember the url or if we just want to send a really short link within our emails.

Friendly URL will let you specify an alpha-numeric name that we can then use within a very short URL. If we set "foodsurvey" as a friendly name the generated URL to access the survey would be :

<http://www.ourdomain.com/fs-foodsurvey.aspx>

We can then copy & past the link provided and use it in almost any application that supports HTML links.

Its important to remember that friendly URL will redirect to the standard fs.aspx form page, as such you need to have the default fs.aspx available on your web site.

Popup windows

FeedbackServer allows us to deploy our form through invitation popup windows. We can set a specific frequency to the window and based on it show the invitation only after a given number of people have visited the page on which the popup code is hosted.

The popup window is simulated using a DHTML generated window and is fully compatible with browsers having popup blocker activated.

Popup Settings

- **Show Popup Every** is the frequency of visitors the popup should be shown to. A cookie is set on each visitors machine to prevent having a visitor to be counted twice in the frequency count. It is recommended to put a 0 value during testing as this will always show the window.
- **Open In New Window** allows us to open the form inside a new window.
- **Edition Language** is the language in which we are currently editing the title, link text and popup message. This feature is only available if we have turned on FeedbackServer's Multi-Language features.
- **Get Language From** allows us to choose from which source FeedbackServer will try to get the language code in order to get the correct translation for the popup window. This feature is only available if we have turned on FeedbackServer's Multi-Language features.
- **Popup Title** is the title shown at the top of the popup window.
- **Popup Survey Link Text** is the text of the link that will take the visitor to our survey.
- **Popup Message** is the message that will be shown inside the popup window.

Take survey

This is a special page that is meant to allow a registered FeedbackServer user to take a survey once he logged inside Feedback Server. This page requires the user to have the "take survey" right set.

Once the user has logged inside FeedbackServer and reach this page he will be able to select the form he has the right to fill We can also restrict the respondent to take a survey only once by activating the FeedbackServer Security Context add in from the security settings.

Web control deployment

This page is intended for more experienced users who like to customize the form asp.net control and integrate it inside their asp.net pages. Note that we can integrate as many forms controls on the same page as we want.

In order to integrate the ASP.net web control code that is provided on the deployment page inside our own asp.net pages we must follow following steps :

1. Open the ASP.net page that should integrate the survey web control in a text editor like notepad.
2. Add following code on the next line after the <%@Page ... %> page directive :

```
<%@ Register TagPrefix="vts" Namespace="DataIllusion.Feedback
Server.WebControls" Assembly="DataIllusion.Feedback Server.Core" %>
```
3. Once we have added this piece of code all we need to do is to cut & past the code provided in the Survey asp.net code section and place it anywhere in our page as long as it is after a <form>.
4. If the page is in a separate web application than the original Feedback Server® installation we must:
 1. Copy the DLL's from the FeedbackServer web application Bin directory to the Bin directory where the page that contains the ASP.net page is hosted.
 2. Copy the <FeedbackServerSettings> declaration and configuration element to the target application's web.config.
 3. Copy the images and FSXmlData directory to the target server.

WSS / Sharepoint deployment

This page let us deploy our survey as a WSS / Sharepoint list on a WSS / SPS site. Once deployed the survey will be shown in the standard WSS / Sharepoint survey list along the other Sharepoint surveys.

The MOSS list dropdown list contains all WSS sites where we have list edition rights. If we want to show the list on the quick launch toolbar of our site we can check the display on the quick launch tool bar option.

Branding / Style

Style

FeedbackServer provides several ways of defining a look & feel for our forms or reports.

1. Using the WYSIWYG Style Editor anyone can create a new style template for a form.
2. Using a custom CSS style sheet create using the CSS SDK available for download on the form style creation page.

Style editor

The style builder is a WYSIWYG style editor that will allow us create a style templates (colors, header, footers) for our forms and public reports. Once we have created a new style we can assign it to our survey using the style properties of the form settings page.

Style Settings

- **Style Name** is the name of the style.
- **Public Style** is the style shared across users in a multi-user configuration of FeedbackServer. The creator of a style is the only one along with the system administrator who can change it once it has been created.
- **Clone** creates a copy of the style template.
- **Built In** styles can only be managed by administrative user accounts.
- **Export CSS** will export the CSS equivalent of the current style created using the style builder.
- **Export Xml** will export an Xml file representing the style. This Xml file can be imported afterward on any FeedbackServer installation when creating a new style.
- **Button Images** allows us to replace all form action buttons by our own custom images
- **Selection Images** allows us to replace the default browser radio and check boxes by our own custom images.

- Survey Deployment Page allows us to brand the header and footers of the form deployment page that will be displayed to the respondent.

Media library

The media gallery is the central place in FeedbackServer where we can store and organize all our different media files (images, animation, videos, documentation). Media files are stored directly in the FeedbackServer database and can be selected from the gallery from any text based parts of FeedbackServer e.g. questions, answers, thanks messages.

Media Folder Settings

- **Folder Name** is the name of the media gallery folder.
- **Built in** is a folder (including all its content) that will be shared across all the users and that can only be modified or filled by administrators based accounts.

Edit media item

Edit Media

- **Title** is the title that will be used by the media item.
- **Width** is the width of the stored element.
- **Height** is the height of the stored element.

Campaigns

Email Campaign

Email distribution

FeedbackServer provides a full email distribution interface that will let us send email invitations to fill our forms.

Once we have send out the invitations we are then able to track who did answer to it or not, including respondent's answers. Using the email code protection add in protect our forms against multiple submissions.

Mailing is done in a multi-threaded way in order to speed up processing several emails are sent at the same time. If you want to increase the number of simultaneous connections to your mail server you can change the FSMAILingThreadNumber inside the web.config. Make sure that your destination server supports the number of connection you specify.

Email list

Using the email list mailing we can send out invitations to a list of emails that we can enter manually into the invitation text box.

- **Email Invitation List** is a comma separated list of emails to which we want to send the invitation to. eg : email1@domain.com, email2@domain.com etc ..
- **Anonymous Entries** FeedbackServer will not save a link between the email and the respondent answers if this option is checked.

CSV file

Using the csv file email mailing we can send out invitations using an existing CSV (comma separated value) based file that we can upload to the server.

- **CSV Email File** is the file that contains our list of emails that will be used to send out invitations. The format of the file must be a comma separated. e.g : email1@domain.com,email2@domain.com etc ..
- **Anonymous Entries** FeedbackServer will not save a link between the email and the respondent answers if this option is checked.

Mailing is done in a multi-threaded way in order to speed up processing several emails are sent at the same time. If you want to increase the number of simultaneous

connections to your mail server you can change the FSMailingThreadNumber inside the web.config. Make sure that your destination server supports the number of connection you specify.

Users and groups

Using the users / groups mailing features we can choose a set of users or groups being available inside FeedbackServer to send out invitations to their emails.

In order to give access to this feature to our users please note that you must grant the user directory and group directory access from the user / role management interface.

- **Users** are the users currently available inside FeedbackServer. Their email will be used as invitation recipient.
- **Groups** are the groups currently available inside FeedbackServer. The groups members emails will be used as invitation recipient.
- **Anonymous Entries** FeedbackServer will not save a link between the email and the respondent answers if this option is checked.

Invitation message

All email based campaigns can use a customizable invitation message either the first time we send out invitations or during the time we send out reminders.

- **Choose Message** Template we can either use an existing Message Template or create a new one.
- **From** is the email from which the invitation will originate.
- **From Name** is the name of the person sending the invitation.
- **Content Type** is the format in which the invitation message is written. You can either send a message in plain text or send it as HTML content.
- **Invitation Message** is the invitation message, two special optional tags that will be replaced while sending the invitation can be included in the invitation.

[--invitationid-] this tag will be replaced by a unique identifier that will be used by EMail code protection security add in to uniquely identify a user to avoid multiple submission. Note that the EMail code protection must be activated in order to avoid multiple submissions.

[--invitationtoken-] this tag will be replaced by one of the token available in your token list. If the destination email is found in token identity's email the

given token will be taken. If no token can be attached to this email a random token will be taken from our generated token list and finally if no tokens are available anymore, a new one will be created.

- **Anonymous Entries** FeedbackServer will no track a link between the email and the respondent answers if this option is checked.

Mailing is done in a multi-threaded way in order to speed up processing several emails are sent at the same time. If you want to increase the number of simultaneous connections to your mail server you can change the `FSMailingThreadNumber` inside the `web.config`. Make sure that your destination server supports the number of connection you specify.

Mailing status

We can check at any time the current status of our campaign and invitations

- **Pending** shows us a list of emails that have been invited but didn't fill our form.
- **Answered** shows us the list of panelists that have filled the form. Its important to note that deleting answered entries will delete all respondent answers as well.
- **Send Reminder** to all let us send out a reminder to all emails that did not fill out our form.

Panel Campaign

Panel campaign

Panel campaigns gives us the power to select and send out invitation directly to all or part of our existing panels and track for each of our panelist the responses that they have posted.

Its important to note that to avoid multiple panelist submissions we have to enable the panelist security add in on our form from the form security settings screen.

If we want to "re-open" the filling of the forms to panelists who took already the form we have to archive the current respondent set.

Invitation message

All email based campaigns can use a customized invitation message either the first time we send out invitations or during the time we send out reminders.

- **Choose Message Template** we can either use an existing message template or create a new one.
- **From** is the email from which the invitation will originate.
- **From Name** is the name of the person sending the invitation.
- **Content Type** is the format in which the invitation message is written. You can either send a message in plain text or send it as HTML content.
- **Invitation Message** is the invitation message, two special optional tags that will be replaced while sending the invitation can be included in the invitation.

[--panelistid-] this tag will be replaced by the panelist unique identifier. Note that the Panelist Security Addin must be activated in order to avoid multiple submissions.

[[yourpanelistattribute]] by replacing the "yourpanlistattribute" name with a valid panelist attribute we can replace at runtime the attribute by the panelist effective attribute value.

Mailing is done in a multi-threaded way in order to speed up processing several emails are sent at the same time. If you want to increase the number of simultaneous connections to your mail server you can change the FSMailingThreadNumber inside the web.config. Make sure that your destination server supports the number of connection you specify.

Mailing status

We can check at any time the current status of our campaign and invitations

- **Pending** shows us a list of emails that have been invited but didn't fill our form.
- **Answered** shows us the list of panelists that have filled the form. Its important to note that deleting answered entries will delete all respondent answers as well.
- **Send Reminder** to all let us send out a reminder to all emails that did not fill out our form.

Campaigns

FeedbackServer provides offers a unified and extensible campaign system that can be used to distribute invitations to take our forms across several medias.

Following campaign add ins are provided out of the box to distribute our invitations :

- Email list
- CSV file
- Panels
- User / Groups

Mailing batch

FeedbackServer manages each campaigns that are send as batches. A batch is a task that will run in the background in order to process each batch entry.

By default batches are checked each minutes to see if any new batch has been created to process it. You can change the pooling interval by changing the <MailingBatchPoolingDelay> inside the web.config file.

The status page gives us an overview of the current batches and which recipient have been queued, send or had errors while processing.

Message template

It's very easy to create new message templates that we can reuse across all our campaigns.

- **Add New Template** this will change the send mode in Add mode, once we've create our invitation as we want it to be we can click on the Add button to add the new template.
- **Edit Template** to edit / delete a template we need to select the template we want to edit from the dropdown list and click on the Edit Template link.

Conditional Logic

Conditions

FeedbackServer comes out of the box with a unified condition engine. Using this condition engine FeedbackServer's elements can trigger specific actions based on the conditions rules and respondent answers.

Following FeedbackServer features are using the condition engine :

- Branching Conditions
- Skip Logic Conditions
- URL Redirection Conditions
- Report Filter
- Skip Logic Conditions
- Answer Skip Logic Conditions
- Panel Views
- Dynamic Content Conditions
- Completion Alerts Trigger Conditions

A condition can be composed of one or multiple condition rules that are tied together using conditional operators like AND, OR. Conditions are evaluated either while taking the survey or once the survey is finished a we want to apply certain conditions to filter a report.

Condition Rules

A condition rule is the heart of a condition. Rules can build based on the questions and answers or panelists available in our survey form or survey answers and are evaluated a runtime while the respondent takes the survey with his answers or when a report is run.

- **Question Answered** let us setup a rule based on the question answer answers available in the survey form. Once we have selected a question we can choose whether we want to setup a rule for a specific answers or for all answers of a question.

If we choose to create a rule based on a specific answer we will have several more features available.

- Selection answers type have no extra features.
- Text entry answers allow us to setup an extra rule to see if the text entered match that rule. If we dont specify any text then Feedback Server® will consider the rule valid if any text has been entered by the respondent.

- **Score** rules let us setup rules based on the score of the respondent, it can be either the total score at a given moment or a score for specific question. Score features are only available if we have turned on FeedbackServer's Scoring features.
- **Respondent Quota** let us setup rules based on the current number of the respondents to a question or an answer of a question. Note that the quota is calculated based on the actual number of respondent who finished the survey, it doesn't include the people who are currently taking the survey but who didn't finish it.
- **QueryString Variable** let us setup rules based on on a querystring variable value.
- **Session Variable** let us setup rules based on on a server side session variable value.
- **Context User** let us setup rules based on current user logged into FeedbackServer.
- **Context User Group** let us setup rules based on the group of the current user logged into FeedbackServer.
- **Member Of Panel** let us setup rules based on the panel membership of a respondent. This feature is currently only available with report filters
- **Member Of Panel View** let us setup rules based on the members of a panel view. This feature is currently only available with report filters.
- **Language** let us setup rules based on the selected language in which the survey is currently running. This feature is only available if we have turned on FeedbackServer's Multi-Language features.
- **Response Date Range** let us setup a range of date with a start date and end date on which forms were filled. This feature is only with report filters.
- **Condition Add in** let us setup rules based on an external conditional add in.

Once we have created a new rule for the condition we can add any number of new rules to the condition as we need. If we have more than one rule we can also choose which logical operator will be used to evaluate the rules together in the condition.

Skip logic

Skip logic conditions allow us to setup logical rules based on respondent's answers, session, querystring or language to hide a question to the respondent.

To use skip logic features we must have at least two pages in your survey to be able to use the skip logic features as a question can only be hidden based on answers of

previous pages not answers on the current page of the question. Skip logic can not be used on the first page of a survey.

Each condition is based on a set of rules that you can define. The first condition that will be met will hide the question to which the skip logic applies. You can have as many conditions as you wish and order or re-order them at any time.

Branching

Branching conditions allow us to setup logical rules based on respondent's answers, session, querystring or language to branch the respondent to a specific page in the form or even terminate the form.

As branching conditions is a feature related to pages we must have at least 2 pages in our survey to be able to use the branching features. Its not possible to use branching on the last page of the form.

Each condition is based on a set of rules that you can define. The first condition that will be met will branch to the page that we defined in the branching editor. You can have as many conditions as you wish and order or re-order them at any time.

URL redirection

URL redirection conditions allow us to setup logical rules based on respondent's answers, session, querystring or language to redirect a respondent to a give URL once he finishes to fill a form or once the condition is met after a page change. Each URL must be prefixed by "http://".

Each condition is based on a set of rules that you can define. The first condition that will be met will branch to the page that we defined in the branching editor. You can have as many conditions as you wish and order or re-order them at any time.

Dynamic Content

Dynamic content

FeedbackServer offers us a nice way to pipe user-defined content into question, answers at runtime.

Lets take for example a large company that needs to make a survey about several similar products of its offer. Without dynamic content we would need to create a separate survey for each of this products in order to display the correct question and answers texts based on each product. Even worse we would need to aggregate the whole different form results once we've finished collecting the results.


We can for example add a querystring parameter to our form url that would specify the product code and define a specific condition that will show text A or B depending if the querystring parameter is A or B. We could also pipe this querystring value inside a hidden - field default value in order to keep it for our reporting, this way we will be able to know for which product the user took the survey.

Dynamic content conditions can be applied not only to querystring but also to language, session variables and current user answers in the survey.

Using dynamic content we can really offer a high level of personalization to our form.

How does it work ?

Dynamic content is a system that is based on tags. We can define any number of tags we want, each tag can have a number of conditions we define and each conditions can show a specific text/html message we define.

Once we have created a new tag we can use it afterward inside our form's question, answer answer default text or sql based types either by adding it manually directly in their text fields using following format `/*ourdynamictagname*/` or we can simply open the piping option window using the  icon and choose the tag we would like to use from the dynamic content tags list.

Once we have added the tag to our text it will be automatically replaced at runtime based on the rules defined for it and with the content we defined.

Dynamic content conditions

Dynamic conditions allow us to setup logical rules based on respondent's answers, session, querystring or language to show the respondent a specific text in an answer or question's text.

Each condition is based on a set of rules that you can define. The first condition that will be met will show the text we have entered for it. You can have as many conditions as you wish and order or re-order them at any time.

Panel Management

Panelist Mapping

Panelist mapping

FeedbackServer let us connect a panel to a survey in order to show the panel structure and values to the survey respondents.

In order to show the correct information to the correct respondent we need a way to map the information we have about a panelist and the person who is actually taking the survey that has the panel linked to it.

As a concrete example lets say that we've created a panel and filled the panel with our customer information (name, email, age, country etc..). FeedbackServer allows us to easily show the information we have about a panelists as a question inside a survey. We can then directly send the survey to the respondent to be filled out and have the updated information directly from the "source" which is actually the customer we have in our panel. No more long stressing phone calls, emails exchanges to know if the information we have in our systems is correct or not.

Its also interesting to know that we can map any number of panelists from different panels to the same FeedbackServer User. This is very useful to present a consolidated view to the respondent for a panelist for which data comes from different data sources.

Mapping Choices

FeedbackServer provides us with several ways of doing the mapping from the panelists to the respondent.

1. Manual Mapping
Using the panelist editor we can map a panelist to any existing FeedbackServer user. This method can be cumbersome as you need to assign for each panelist its FeedbackServer user and also the user must be logged inside FeedbackServer to allow the mapping.
2. Automatic Mapping
Using the user provider panelist mapping tool we can ask FeedbackServer to try to map the current logged in user to one of our panelist. The user provider panelist mapping tool provide us with several mapping option ranging from mapping the current logged in username to a panelist attributes or id. If you know that one of your panel attributes matches a FeedbackServer user id or user name this is the way to go. Note that the user must be logged inside FeedbackServer to allow the mapping.
3. Security Add in Mapping
We can use any security add in that supports the panelists mapping

architecture to automatically map the current respondent to a panelist. As mapping depends on how the security addin features set are actually managing the mapping. Main advantage is that some security add in will do mapping without having the user to be logged in, e.g.: using an external querystring variable that will hold the panelist id.

User provider panelist mapping

User provider panelist mapping allows us to set the way FeedbackServer will link a logged in user to existing panelist information in order to show the correct information to the user once he access a survey that has the panel linked to it as question.

- **Active user provider type** is the current provider name that is used to authenticate and feed the FeedbackServer's users and groups.
- **Map username to data source id** will map the panelist information against the currently logged in user's username to the data source id of the panel.
- **Map username to panelist attribute** will map the panelist information against the currently logged in user's username to the value of the panel attribute.
- **Map user id to data source id** will map the panelist information against the currently logged in user's id to the data source id of the panel.
- **Map user id to panelist attribute** will map the panelist information against the currently logged in user's username to the value of the panel attribute.

Security addins panelist mapping

In order to map a panelist information to a connected respondent that is currently taking a survey where a panel as been linked to we can use the attached security add in of the survey to handle the mapping between the connected respondent and the panelist information for us.

As each security add in has its own way and features to handle the mapping we suggest reading directly the specifications of each security add in that implements panelist mapping features.

At this time following security add ins support the panelist mapping architecture :

- Panel Security Context

Panel views

Panel views

FeedbackServer let us create panelists segment based on our panel members.

The main purpose of panel views is that once we've created a segment based on our panelists we can reuse the views to make targeted mailing invitations for surveys or reuse the segment as a filter on the survey that have been taken by our panelists.

E.g. : Send only the survey to people of our panel which are over 18 years old.

Panel view settings

Edit Panel View

- **Panel view name** is the name of our panel view.

Panel View Rules

We can add as many rules as we want against the panel attributes to narrow down our panel to create a segment of panelists.

Panel view members

The panel view member page let us see all panelist that belongs to the panel view segment. Note that it will not show panelist that have entries waiting for approval.

We can choose to show two different attributes of the panel in order to get more information in the grid. We can also sort each column by clicking on the column's name.

In order to get to the panelists details all we have to do is double click on the panelist's row we want to see.

- **Add New Panelist** will let us add a new panelist to the panel. Note that only panel connector supporting the addition of panelists will show this option.
- **Delete Selected Panelists** will delete all selected panelists directly from the panel. Note that only panel connector which support "delete" on their data store will show this option and all his related survey answers will be deleted as well from the surveys he participated to.

Panel Connectors

FeedbackServer connector

The FeedbackServer is the default built in panel connector. Its does not require any external data source to connect to and let you create a brand new panel either from scratch or using the automatic panel filling features of FeedbackServer.

Supported Panel Connector Modes

- **Read** Allows to read existing panelist in our panel
- **Update** Allows to update existing panelist in our panel
- **Delete** Allows to delete panelists from our panel
- **Add / Auto Fill** Allows to add new panelists to our panel or to fill our panel automatically when its linked to a survey

SQL Connector

The Microsoft SQL panel connector will let you connect any SQL Server table or view as a panel inside FeedbackServer. Once connected it will synchronize the information available in the source table with the local FeedbackServer database.

Its important to note that if you synchronize a set of users starting from 50k with a many columns (5+) you must make sure that your local FeedbackServer database has enough space available for the synchronized data and for the transaction log its SQL Server before synchronizing.

Following SQL Server column types are currently supported for synchronization :

- bigint
- int
- smallint
- tinyint
- bit
- decimal
- numeric
- money
- smallmoney
- float
- real
- datetime
- smalldatetime
- char
- varchar
- text
- nchar
- nvarchar
- ntext

User defined types are also supported if they are based on any of the supported types above.

SQL Connector Settings

Make sure to ask your local system administrator for the correct settings if you have any doubts.

- **Authentication Mode** is how the connector will connect to our SQL Server
 - SQL Authentication : Uses SQL Server's user ids and passwords.
 - Integrated Security : Users the current user which runs the ASP.net worker process.
 - Connection String : Supports any valid SQL Server connection string
- **SQL Server Name** is the name of our SQL Server that will hold the table we want to connect to.
- **User Id** is the SQL User id to connect to our SQL Server
- **User Password** is the password of the SQL User
- **Table To Connect To The panel** is the list of tables available from which we can choose the table to connect to.
- The account connecting to the database must have a least read access to the table in order to see it in this list.
- **Map panelists id to primary key** is the name of the primary key of the table to which we will connect.
- **Table columns to synchronize** is the name of the columns that are available for synchronizing. Its important to remember that the more columns we choose the bigger the synchronization will get.

Supported Panel Connector Modes

- **Read** Allows to synchronize structure and data from a SQL table
- **Update** Allows to update existing data in the connected SQL table

Known Limitations / Issues

The SQL Connector has some limitations listed below :

- One primary key only table with a single primary key are supported at this time
- No Structure changes its not possible to add new columns for synchronizing once the panel has been created

Active Directory Connector

The Active Directory connector will let you connect any Active Directory primary controller user base and use it as a panel inside FeedbackServer. Once connected it will synchronize the information available in the PDC with the local FeedbackServer database.

Its important to note that if you need to synchronize a large set of AD users (50k+) you must make sure that your local FeedbackServer database has enough space available for the synchronized data and for the transaction log of its SQL before synchronizing.

Following Active Directory user schema attributes are currently available and supported for synchronization :

- c (country code)
- company
- departmenet
- description
- displayName
- givenName
- initials
- l (city)
- mail (email)
- name
- physicalDeliveryOfficeName
- postalCode
- postOfficeBox
- sn (last name)
- streetAddress
- title

Active Directory Connector Settings

Make sure to ask your local system administrator for the correct settings if you have any doubts.

- **PDC Server Name** is the name of the primary domain controller that manages your active directory user structure
- **Active Directory Domain** is the domain from which to synchronize the users

- **Admin User Name** is the user name of the AD user who has read / write access to the active directory PDC
- **Admin User Password** is the password of the AD user who has read / write access to the active directory PDC
- **Search Filter** let you specify a filter for your AD Query. Only user schema is supported at this time.
- **User Attributes To Synchronize** is the name of the user Active Directory attributes that we can synchronize. Its important to remember that the more columns we choose the bigger the synchronization will get.

Supported Panel Connector Modes

Read Allows to synchronize structure and data from an active directory store

Update Allows to update existing data in the connected active directory store

Known Limitations / Issues

The AD Connector has some limitations listed below :

- Defined attributes list only the predefined user attributes list is currently supported. You may contact us on the procedure to add further attributes for synchronization.
- No Structure changes its not possible to add new attributes for synchronizing once the panel has been created

SalesForce.com Contact Connector

The SalesForce.com contact connector will let you connect your existing SalesForce.com contact base and use it as a panel inside FeedbackServer. Once connected it will synchronize the information available in your SalesForce account with the local FeedbackServer database.

Its important to note that if you need to synchronize a large set of contacts (50k+) you must make sure that your local FeedbackServer database has enough space available for the synchronized data and for the transaction log of its SQL before synchronizing.

Following SalesForces.com contact attributes are currently available and supported for synchronization :

- salutation

- firstname
- lastname
- email
- title
- department
- birth date
- phone
- home phone
- mobile phone
- other phone
- fax
- mailing street
- city
- country
- state
- postal / zip code
- other mailing street
- other city
- other country
- other state
- other postal / zip code

SalesForce.com Contact Connector Settings

Make sure to ask your SalesForces.com for the correct settings if you have any doubts.

Not all Salesforce subscription allow remote access through web services. Please contact your SalesForces account manager to know if you can connect to your account through the Salesforce web service .

- **SF User Name** is the user that will be connecting to the salesforce web service.
- **SF Password** is the password of the user connecting to the web service.
- **Entity Attributes** let you select which lead attributes to synchronize. Its important to remember that the more attributes we choose the bigger the synchronization will get.

Supported Panel Connector Modes

- **Read** Allows to synchronize structure and data from contacts inside our Dynamics CRM server
- **Update** Allows to update existing contacts on the connected Dynamics CRM server

Known Limitations / Issues

The Salesforce Contact Connector has some limitations listed below :

- Defined attributes list only the predefined contact attributes list is currently supported. You may contact us on the procedure to add further attributes for synchronization.
- No Structure changes its not possible to add new contact attributes for synchronizing once the panel has been created

SalesForce.com Lead Connector

The Salesforce.com lead connector will let you connect your existing Salesforce.com lead base and use it as a panel inside FeedbackServer. Once connected it will synchronize the information available in your Salesforce account with the local FeedbackServer database and we will be able to update leads or create new leads directly from Feedback Server's forms.

Its important to note that if you need to synchronize a large set of leads (50k+) you must make sure that your local FeedbackServer database has enough space available for the synchronized data and for the transaction log of its SQL before synchronizing.

Following SalesForces.com contact attributes are currently available and supported for synchronization :

- salutation
- firstname
- lastname
- email
- title
- department
- birth date
- phone
- home phone
- mobile phone
- other phone
- fax
- mailing street
- city
- country
- state
- postal / zip code

- other mailing street
- other city
- other country
- other state
- other postal / zip code

SalesForce.com Contact Connector Settings

Make sure to ask your SalesForces.com for the correct settings if you have any doubts. Not all Salesforce subscription allow remote access through web services. Please contact your SalesForces account manager to know if you can connect to your account through the Salesforce web service.

- **SF User Name** is the user that will be connecting to the salesforce web service.
- **SF Password** is the password of the user connecting to the web service.
- **Entity Attributes** let you select which lead attributes to synchronize. Its important to remember that the more attributes we choose the bigger the synchronization will get.

Supported Panel Connector Modes

- **Read** Allows to synchronize structure and data from contacts inside our Dynamics CRM server
- **Update** Allows to update existing contacts on the connected Dynamics CRM server
- **Add / Auto Fill Allows** to add new leads and to our panel or to fill our panel / CRM automatically with new leads when its linked to a survey

Known Limitations / Issues

The Salesforce Lead Connector has some limitations listed below :

- Defined attributes list only the predefined contact attributes list is currently supported. You may contact us on the procedure to add further attributes for synchronization.
- No Structure changes its not possible to add new contact attributes for synchronizing once the panel has been created

Panels

FeedbackServer provides us with an extensive set of features to manage a panel of external entities eg: contacts, business partners, students etc ... The advantages of a panel is that you can have a centralized set of data that you can then reuse for reporting, survey distribution or filtering without duplicating your information.

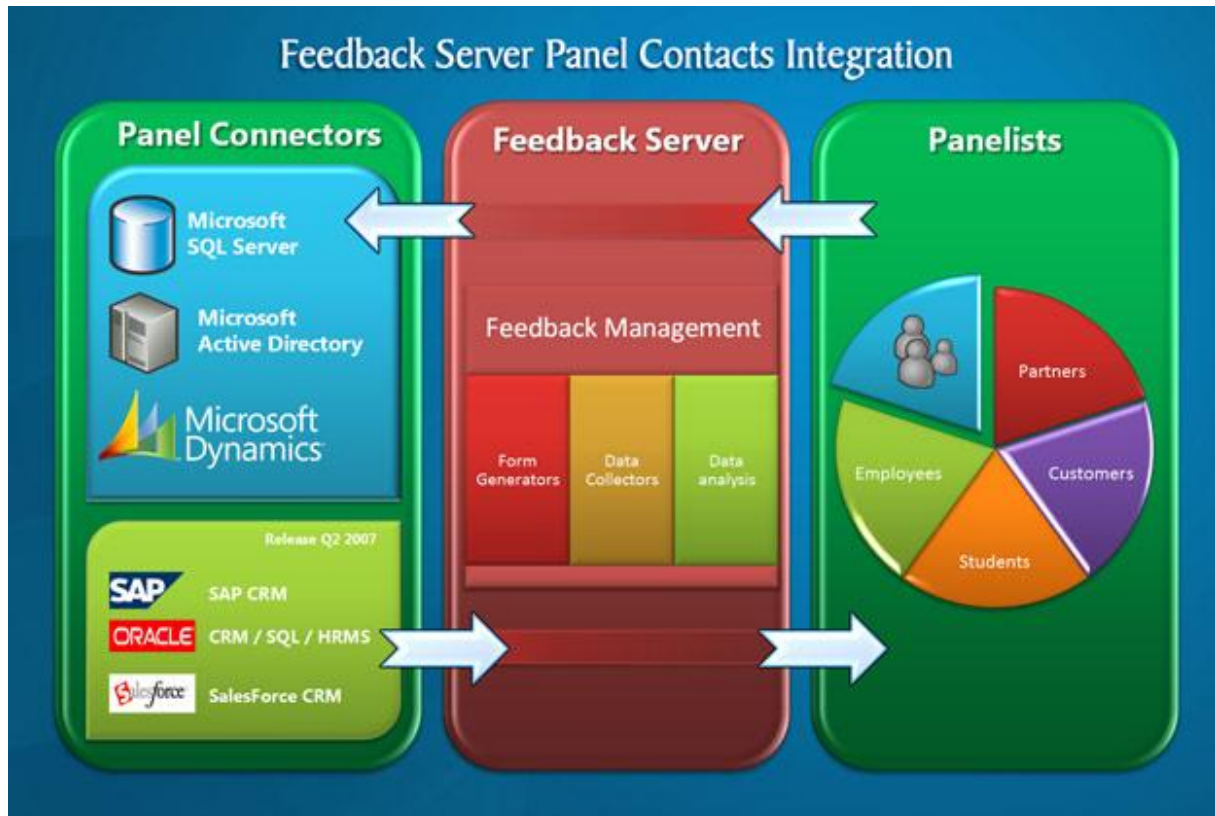
Instead of having to ask each time your customer for the same information (eg: name, country, age etc..) you can centralize all this information once for all in a panel, create the corresponding panelists and then just attach surveys to the panel. Its even possible afterward to analyze and aggregate survey results with the panel information of the panelists who took the survey.

Each panel is composed of a set of attributes that you can define in order to create your panelists structure. As attributes are based on the standard FeedbackServer answer types we can reuse any existing answer type (fields, dates, countries ...) we have registered in FeedbackServer and thus we have a great flexibility on how our panelist data structure will look like.

Panelists Entries

Not only does FeedbackServer offer you a way to manually enter your panelists but FeedbackServer goes a step further by allowing you to "import" existing sets of data from external sources through the concept of Panel Connectors. A panel connector is a plugin that will handle the synchronization between an external data store and a Feedback Server® panel.

As you can see below panels can be created from many different existing contacts data stores.



Out of the box FeedbackServer comes with following panel connectors.

1. FeedbackServer Connector.
This is the default built in connector. Its allows you to create, update, delete panelists on a panel without having to own or to connect to an external data store other that the Feedback Server® data store.
2. Microsoft SQL Connector
Using this connector we can connect our panel to any table available in any of our SQL Server databases. This connector supports read and updates from its data store.
3. Active Directory Connector
Using this connector we can connect our panel to our centralized domain controller and synchronies our panel with the users available in our active directory network. This connector supports read and updates from its data store.
4. SalesForce.com Connector
Using this connector we can connect our panel to our SalesForce.com contacts and leads database. This connector supports read, updates and creation from its data store.

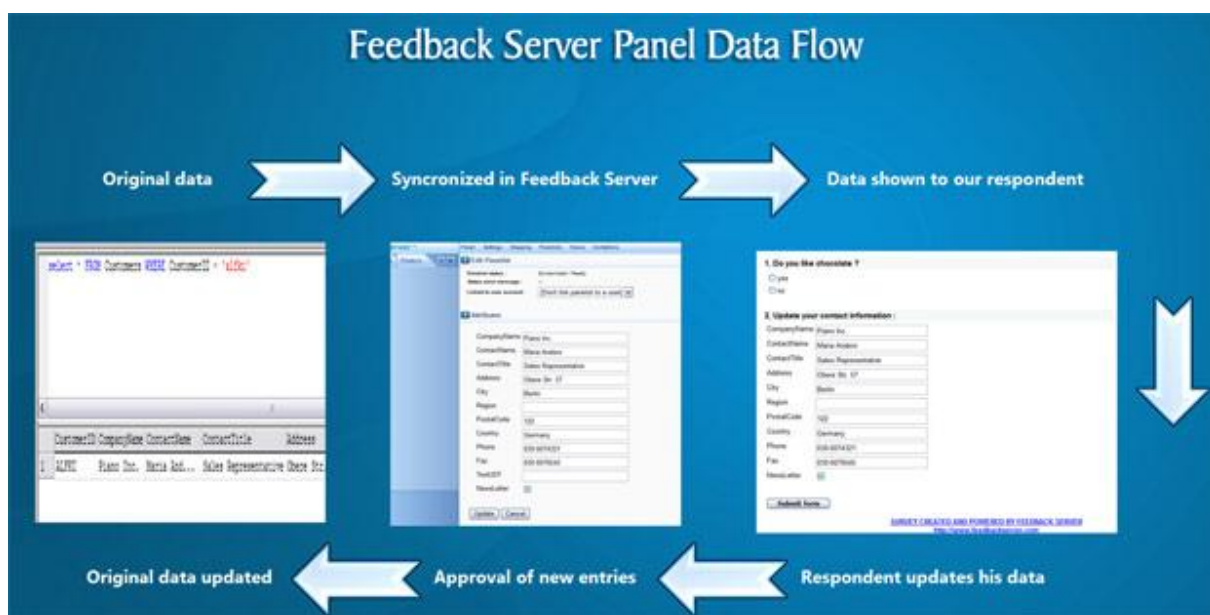
Using Feedback Server's API its also very easy to implement a new connector if you have a third party data store that is currently not supported out of the box.

The Rosetta Stone

How many times did we have incomplete information in our data store about customers ? How many times did we try to find out back this information through lengthy phone calls, emails that get never answers ? Why didn't our customer purchase anything since the last six month ?

Having information in our data store is very good but having it up to date and knowing over time how our customers act or react about our products is even better.

FeedbackServer offers an easy way to handle these issues by offering the concept of Linked Panel Questions. Using a linked panel question we can present information we have about a customer directly to the customer by linking a panel to a survey.



As you can see above the customer just see a very easy to use form with his current information, if our customer sees an error in his information he can just correct and it and we will get directly the change and best of all we can even ask him as few extra questions along the way.

Through the concept of panel views we can create small segments of panelists based on rules and send only surveys to these segments letting us effectively target the right persons for which we require information.

Panel information

The panel information page will give us the current state of our panel. If the connected panel depends from a third party data source it will also provide us information on the synchronization process and options to start or stop the process manually.

- **Panel Connector** name of the panel connector to which the panel is linked to.

- **Panelists** count is the number of panelists available in the panel.
- **Entry approval requests** is the number of panelists entries waiting to be approved.
- **Synchronies Status** is monitoring the sync process between the connector and a third party data source. It will also show a detailed error message if an error occurred during the process.
- **Synchronies Progress** is percentage of data that have been synchronised against a third party data source.
- **Start / Stop Synchronization** allows us to manually start or stop the background synchronization task of the panel connector.
- **Manage Attributes** let us edit the attributes of the panel.
- **Delete Panel** let us delete the panel. Its important to note that if we delete a panel all panelists will be deleted including all their answers to the surveys they've taken. Panel that are connected to a third party data source are not delete the third party data source for which panelists information is linked to.
- **Survey Participation** us shows us an overview of all the survey that panelists members have participated to. We can click on the panelists count to go to complete list of panelists who participated in the survey.

Panel settings

The panel settings page will let us change all settings related to the current panel and panel connector.

- **Panel name** is the name of panel.
- **Sync mode** choose if the synchronisation is triggered manually from the panel main screen or if the system synchronies the data automatically based on the sync interval.
- **Data sync type** how the data will be synced.

Add will add any new records from the source into FS.
Update will update any changes from the source into FS.
Delete will delete any deleted record in the source from FS.

- **Connector Source Update Mode** let us choose how panel information will be added/updated when it has been linked to a survey and a respondent has taken the survey and updated the panelist information.

Live link will immediately synchronise the panel connector data store with the panelist information posted by the respondent.

After approval will let us review each entry before we manually agree to synchro it with the panel connector data store.

Not allowed will show all information linked to a survey in read only mode and no updates are possible.

- **Source Synchronization Interval** is the interval in minutes when the local FeedbackServer database will be synchro with the panel connector data store of the connector.
- **ReSync On Each Application Restart** let us choose whether we want to re-sync the local FeedbackServer database and the third party data store on each application restart of the web server. We should use this option with care as a web server with high application recycles / reboots will likely surcharge itself and the third party data store.

Panel attributes

Panel attributes are the basics of a panel as it will define the structure of the panelist inside our panel and thus the data we can manage and exploit. The panel attributes editor allows us to add / delete attributes to our panel.

FeedbackServer is able to handle two kind of attributes :

- **Internal attributes** are independent from any connector and can be added manually to a panel. Creation / deletion, update of data inside these kind of attributes will have no effect on any third party data source.

Its very important to note that internal attributes are not supported in the approval entry process which means that once an attribute value for a panelist has been updated the change will be immediately reflected in the database even if a panel connector requires an approval.

- **Linked attributes** are linked to a panel connector data source and cannot be added manually. These attributes are generated automatically by the panel connector to reflect the structure to which the panel is connected to. If the connector is supporting live update of its data source changes will be updated in the local FeedbackServer database and in the remote connected data store.

It is also possible to change the answer type that is currently assigned to a linked attribute, but its not possible for validation reasons to change its entry validation settings as this would maybe break the update / add process to the connected third party data store.

Both attributes can be mixed together. For example if we use a connector that connects us to a third party data source but we are missing some key information that is not available in this data source we can add internal attributes to our linked attributes.

Attributes Overview

- **Edition Language** lets us select in which mode we are editing the attributes label and default texts.
- ▲ will move the attribute's position up.
- ▼ will move the attribute's position down.
- **Delete** will delete the attribute and all panelist data related to this attribute i. Its not possible to recover it once it has been deleted. Note that delete attributes that were linked to a third party data source by the connector will not delete any data on the third party data source.
- **Add New Attribute** allows us to add a new attribute to our panel.

Attributes Types Settings

Each attribute can be based on any answer type available inside FeedbackServer. By default FeedbackServer offers already many types of answer items it is very easy to develop and add our own answer items inside FeedbackServer using the SDK.

As each answer item has its specific set of properties and behavior we suggest to read the documentation of each type individually as you need it to build your panelist attribute structure.

Panelist manager

The panelist management page let us see all panelist that are in our panel. Note that it will not show panelist that have entries waiting for approval.

We can choose to show two different attributes of the panel in order to get more information in the grid. We can also sort each column by clicking on the column's name.

In order to get to the panelists details all we have to do is double click on the panelist's row we want to see.

- **Add New Panelist** will let us add a new panelist to the panel. Note that only panel connector supporting the addition of panelists will show this option.
- **Delete Selected Panelists** will delete all selected panelists. Note that only panel connector which support "delete" on their data store will show this option and all his related survey answers will be deleted as well from the surveys he participated to.

Entry approval

The entry approval page let us see all panelist that are in our panel that have posted changes to their attributes values and that are waiting for an approval.

We can choose to show two different attributes of the panel in order to get more information in the grid. We can also sort each column by clicking on the column's name.

In order to get to the panelists details and approve it individually all we have to do is double click on the panelist's row we want to see.

- **Approve Selected Panelists Entries** will approve all selected panelists and store the approved data in the connected third party data store of the connector.
- **Deny Selected Panelists Entries** will deny all selected panelists entries and restore the current state that is available in the connected third party data store of the connector to the local Feedback Server® database.

Panelist editor

The panelist editor let us see all details regarding a single panelist. If the panel connector allows them we are also able to update, delete the panelist information from the panel.

Edit Panelist

- **Panelist Status** is the current synchronization status of the panelists data with the panel connector's data source or the state if the panelist is waiting for an approval.
- **Status Error Message** shows us any error message that happened while synchronizing to the connector's data source .
- **Linked To User Account** will link / map the given panelist to the Feedback Server user provider logged in user. This feature should be used when you

connect the panel to a survey and you want the current logged user to be associated with panelist's information to show the correct data inside the linked panel question.

Attributes

shows the current data for the panelist that we are editing. If the panelists data requires approval we can see to compare both data the changed one and the one that is currently in the panel connector data store.

- **Add New Panelist** will let us add a new panelist to the panel. Note that only panel connector supporting the addition of panelists will show this option. Only panel connectors supporting the "add" mode will support new panelist creation.
- **Approve Entries** will approve the entries and update the panel connector data store as well.
- **Deny Entries** will deny the entries and restore the original data from the panel connector data store.
- **Update** will update all entries and update the panel connector data store. Only panel connectors supporting the "add" mode will support panelist updates.
- **Delete Selected Panelist** will delete all selected panelists. Note that only panel connector which support "delete" on their data store will show this option and all his related survey answers will be deleted as well from the surveys he participated to.

Survey Answers

Displays a list of all survey in which the panelist has participated to.

Pending Survey Invitation

Displays a list of all survey that he has been invited to but didnt yet give an answer to.

Panel participation

The panel participation page let us see all panelist that have participated to a given survey including the date at which they participated.

We can choose to show two different attributes of the panel in order to get more information in the grid. We can also sort each column by clicking on the column's name.

In order to get to the panelists details all we have to do is double click on the panelist's row we want to see.

- **Delete Selected Survey Participation** will delete all selected panelists participation. All his related survey answers will be deleted as well from the surveys he participated to.

Link panel to form

FeedbackServer let us link any of our panel to any of our forms. Once linked to a survey the panel attributes will be made available as a question that you can move anywhere inside the form.

To link a panel to a survey all we have to do is choose the insert question on our survey and from the copy question pane choose link to panel and choose the panel we want to link to the survey.

Linked panels can be used in several scenarios :

1. Panel Auto Filling

Using panel auto filling we can present the respondent of the survey with a blank fields representing the panel attributes. Once the respondent has finished the survey all connected panels will automatically create a new panelist and assign the respondent's value to it.

This is very practical if you want to create a new panel but don't want to enter all the data yourself so you can just send out a simple survey in order to fill your panel automatically with panelists.

In the case of existing data like in the Dynamics CRM Lead Connector we can easily provide a simple form for our web users to fill and they will instantly be added to our internal CRM systems as leads.

Note that only panel connector which support the insertion of new panelists are able to support this feature.

2. Panel Updating

Using panel updating along with the panelist mapping features we can show the current logged in panelist with the data that we have about him in our panel. Once the respondent has finished the survey all data related to the panel will automatically updated inside the panel.

Note that only panel connector which support the update of panelists are able to support this feature.

3. **Panel Information**

As we can setup our panel attributes to have different states depending on the context we could show any or all attributes as read only to the user for information purposes.

We also can configure each attribute behavior from the panel attributes to make a field read only, visible, updatable based on the context panel filling or panel updating.

Reporting

Report Items

Report items

Report items are the base of any report. Each report can be composed of any number of report items.

Each report item has its own analysis tools and rendering process. It is also very easy to insert our own analysis report item and extend the FeedbackServer reporting engine with it using the FeedbackServer SDK.

Following report items are provided out of the box to build our reports :

- Graphics charts
- Web graphics
- Free text report
- Cross tabulation report
- Respondent entries list
- Individual Reach For Multiple Answers (IRMA)
- Matrix grid report
- Radar rating charts
- Constant sum
- Ranking
- Static text

Report items share following properties that we can customize :

- **Report Item Title** is an optional text we can specify that will show up as information on top on the report item in the analysis mode.
- **Use Aliases** do we want to show question and answers reporting aliases instead of showing the question / answer label text.
- **Multiple Items Layout** if our report item supports multiple items selections like multiple answers or questions we can display the results either vertically or horizontally.
- **Columns Number** is the number of columns in which we want to display the items inside our report item.

Filters / Answers Filters

- **Override Report Filters** allows us to override the filters that have been set at the report level.
- **Filter Start Date** is the start date interval on which the results of the Report Items will be calculated.
- **Filter End Date** is the end date interval on which the results of the Report Items will be calculated.
- **Assign A Filter** using the report filter Editor we can create specific filters. For example if we only want to display the results of respondents who have chosen answer B to the question Z.
- **Language Filter** we can filter the results of the Report Items by the language chosen by the respondent. This feature is only available if we have turned on Feedback Server's Multi-Language features.

Extended Filters

The extended filters allow us to filter based on respondents answers. To learn more about extended filters please read extended filters introduction.

Report Questions, answers

We can select from which answers or questions we want to use in our report item.

Graphic charts

This report item will show us graphical charts and statistics for questions and free text answers.

Graphics Properties

- **Bar Chart** will render a group of bar charts to display the results.
- **Pie Chart** will render a pie chart to display the results.
- **Line Chart** will render lines to display the results.
- **Rating Chart** will render a gauge based on the mean rating of the rating weights set on the answers. This feature is only available for rating based questions.
- **Chart Width** is the width of the chart that is rendered.
- **Chart Height** is the height of the chart that is rendered.
- **3d Rendering** will render the chart output using 3D effects.

Charts Properties

The charts provide us following information during runtime analysis of :

- **Answers Total** is the total count of answers for the question, if we have multiple choices answers the total is a sum of all answers.
- **Individuals** is the number of individual respondents who did answer to this question, if we have multiple choices answers and a respondent has selected more than one answers he will be still counted as one respondent.
- **Participants** is the total number of respondents who fill the form.
- **Reach** is the percentage of respondents the question has reached. In other words its the percentage of respondent who answered the question out of the total of survey participants.
- **Rating** is the average rating value for the question based on the answer selection types that are marked as rating part. This feature is only available if we have turned on rating from the question editor.

Charts Series

The chart series allow us to define a sub-set of chart with their own filtering rules that will be included in the main chart. For example we can use chart series if we want to compare a set of answers from different time periods on the same graphic.

- **Name** is the name of the series that will be displayed inside the main report.
- **Start Date** is the start date that will be used to filter the subset of data.
- **End Date** is the end date that will be used to filter the subset of data.
- **Filter** is the filter that will be applied to the subset of data of the series.

Free text report

This report item will show us all free text entries that have been entered by the respondents. We can choose what questions answers we want to show the free text for and also choose if we want to group them by respondents or by question's answers.

Free Text Report Properties

- **Display** we can either group the display by answers or by respondents. Grouping by respondent will show us all selected answers by respondent.

- **Paging Size** we can turn on paging if we have a long list of free text answers entries.
- **Text Sort Order** we can sort the text entries by alphabetical order.
- **Show Voter Details** we can show more information about respondents like IP's, context username.

Respondent entries

This report item will list the respondents who filled the form.

Respondent Entries List Properties

- **Paging Size** we can turn on paging if we have a long list of respondents.
- **Text Sort Order** we can sort the respondent list by date on which they filled the form.

Matrix grid report

This report item will show us a global matrix view of the results of matrix question types.

Matrix Properties

The matrix provide us following information during runtime analysis of :

- **Answers Total** is the total count of answers for the question, if we have multiple choices answers the total is a sum of all answers.
- **Individuals** is the number of individual respondents who did answer to this question, if we have multiple choices answers and a respondent a selected more than one answers he will be still counted as one respondent.
- **Participants** is the total number of respondents who participated in the survey.
- **Reach** is the percentage of respondents the question has reached. In other words its the percentage of respondent who answered the question out of the total of survey participants.
- **Rating** is the average rating value for the question based on the answer selection types that are marked as rating part. This feature is only available if

we have turned on rating from the question editor from and if there is a least one selection type answer that is marked as a rate part.

- **Grid Rating** is the average rating of all rating of the child questions that are displayed in the grid. This feature is only available if we have turned on rating from the question editor from and if there is a least one selection type answer that is marked as a rate part .

Cross tabulation

This report item will allow us to make a cross tabulation results between two questions.

Suppose we have two questions :

What is your age ?

How do you find our web site ?

	How do you find our website ?		
what is your age ?	very good	bad	excellent
No answered	0% (0)	0% (0)	0% (0)
10-20 year	0% (0)	0% (0)	100% (1)
20-40 year	100% (2)	0% (0)	0% (0)
40+ year	0% (0)	100% (1)	0% (0)
Total	2	1	1

Using cross tabulation of the two question we can see the relation between their answers.

Cross Tabulation Properties

- **Base Question** is the question that has its answers used as rows in the cross tabulation.
- **Compare Question** is the question that has its answers used as columns in the cross tabulation.

Individual reach for multiple answers

This report item will allow us to see for multiple choices question the combination of choices answered by a single respondent.

Suppose we have following multiple choices question :

What do you drink ?

☐ Pepsi

☐ Water

- ☐ Milk
- ☐ Fanta

Using the other report items we would only be able to know either the total of answers for all choices or the total number of individual respondent of the question.

Using IRMA we can know exactly for the reach of individual respondent for the different answers combinations of the question.

IRMA offer us two different analysis method.

1. We can choose specific answers and see the reach for the combination of these answers.

Count of individual voters who answered following combination :

Pepsi
Water

1 single voters out of a total of 6 voters (16.67% reached)

2. We can choose a number of combinations of answers and IRMA will calculate the different combinations along with their individual respondent reach

Personal Information :

Combination	Voter total	Reach
Pepsi, Milk	2	33.33%
Fanta, Water	2	33.33%
Fanta, Milk	2	33.33%
Pepsi, Fanta	1	16.67%
Pepsi, Water	1	16.67%
Water, Milk	1	16.67%
Total voters of the survey : 6		

Individual Calculation Type

- **Answer Reached** is the selected answer combination we want to analyze against.
- **Answer Number Reached** is the number of answers we want to be part in our combination calculation.

Radar rating chart

This report item will show us a radar chart with the average of all rating groups selected to appear in the chart. A minimal of 3 groups is required to give a meaning to the radar chart processed data.

Report Groups

We can select which rating group will be included in the radar chart to calculate the mean of all groups and show the average for each group.

Web Html graphics

This report item will show us web based HTML text graphical charts and statistics for questions and free text answers.

Charts Properties

The charts provide us following information during runtime analysis of :

- **Answers Total** is the total count of answers for the question, if we have multiple choices answers the total is a sum of all answers.
- **Unique Respondents** is the number of individual respondents who did answer to this question, if we have multiple choices answers and a respondent has selected more than one answers he will be still counted as one respondent .
- **Rating** is the average rating value for the question based on the answer selection types that are marked as rating part. This feature is only available if we have turned on rating from the Question Editor .

Ranking report

This report item will allows us to display the average ranking of our ranked based answers.

Constant sum

This report item will allows us to display the total of all sums that were entered for our constant sum answers.

Static text

This report item let us set a descriptive text that we can use to set description within our report as information for our users.

Data export

Data export

FeedbackServer provides already some powerful reporting features but sometimes when we need even more analysis on our data or integrate our FeedbackServer data to an existing database we need to be able to export our data.

Using FeedbackServer's data export we are able to generate text based files of respondents answer that we can import in any third party reporting tool like Excel or SPSS, SAS, Access and any other tool supporting the import of CSV or Xml based files.

Data Export

- **Export in** we can choose if we want to export our data as an Xml, SPSS or CSV (comma separated value) file.
- **CSV output type** if we have chosen to export as CSV we have now the choice between two different output.
 1. Individual Answers
 2. Separate Selection Types And Fields

CSV individual answers

Using the individual answers export each answer inside a question will have its own column. Selection type answers which have been answered will be marked in the respondent's row using the Answer selection mark, all selection answers that have not been selected will be marked by the Answer unselected mark. Field based types will output directly the respondent text entry of the field.

Data Export

- **Export Encoding** is the format in which the file will be saved. If we are using a multi-language we strongly suggest to use UTF-8 as this will allow us to see entries written in any language.
- **Respondent Export Filter** is the filter that will be applied to respondents. We can create a filter using the report filter creator.
- **Export Answers** we can either export all answers from the survey or a subset of answers we select.
- **Export From / To Date** is the date interval from which we want to export respondents.
- **Include Security Addin Data** do we include activated security add in data. E.g: The token security add in will generate extra data in the export to export the token information.

Export Formatting

- **Columns Header Template** allows us to format as we need the header column . {0} will be replaced by the answer text or id or display order and {1} will be replaced by the question text or id or display order. Note that both are optional and that you can format them in any custom string you like.
- **Column Header Output** will define what values to use to replace the {0} / {1} inside the header template column .
- **Answer Selection Mark** is the value that will be shown in each row where a respondent has answered to an answer. Note that field type answers will show what the respondent entered in the field.
- **Answer Unselected Mark** is the value that will be shown in each row where a respondent has not answered to an answer. Note that field type answers will show what the respondent entered in the field.
- **Field Delimiter** is the char that will act as a delimiter between the exported answers columns inside a CSV.
- **Text Delimiter** is the char that will act as a delimiter between the text entries inside a CSV.
- **Replace CR** is the char that will replace "new line" character. Some tools like Excel or Access might encounter problems with "new line" characters inside a row. We suggest replacing it with a custom char before replacing it again after having imported the data inside the target tool.
- **Use Reporting Aliases** we can export the column names using the reporting aliases instead of using the full answers / questions label text.

Data Output Example

In order to better understand we have compiled a few example of questions and how their output is organized

.

Standard single selection question with three selection answer types

What is your favorite color ? <input type="radio"/> red <input checked="" type="radio"/> green <input type="radio"/> blue	VoterId	red (What is your favorite color)	green (What is your favorite color)	blue (What is your favorite color)
	1234	0	1	0

Standard single selection question mixed with three selection answer types and one field type answer

How did you find this survey ? <input checked="" type="radio"/> Great <input type="radio"/> Amazing <input type="radio"/> Lost my time Comment : I am waiting for the next survey ...	VoterId	Great (How did you find this survey ?)	Amazing(How did you find this survey ?)	Lost my time (How did you find this survey ?)	Comment :(How did you find this survey ?)
	1234	0	1	1	I am waiting for the next survey ...

Standard single selection question with multiple selection and three selection answer types

What is your favorite drink ? <input checked="" type="checkbox"/> coke <input type="checkbox"/> water <input checked="" type="checkbox"/> fanta	VoterId	coke (What is your favorite drink)	water (What is your favorite drink)	fanta (What is your favorite drink)
	1234	1	0	1

Standard single question with three field based types

4. Personal information : Name : <input type="text" value="John"/> Last name : <input type="text" value="Doe"/> Country : <input type="text" value="United States"/>	VoterId	Name :(Personal Information :)	Last name :(Personal Information :)	Country :(Personal Information :)
	1234	John	Doe	US

CSV separate selection / fields

Using the separate selections / fields export question will have its own column. Respondent's rows will contain the answer to the question being in the column. Field based type will be separated into the same number of columns as they are field based types in your questions.

Data Export

- **Export encoding** is the format in which the file will be saved. If we are using a multi-language we strongly suggest to use UTF-8 as this will allow us to see entries written in any language.

- **Respondent Export Filter** is the filter that will be applied to respondents. We can create a filter using the report filter creator.
- **Export Answers** we can either export all answers from the survey or a subset of answers we select.
- **Export From / To Date** is the date interval from which we want to export respondents.
- **Include Security Add in Data** do we include activated security add in data. E.g: The token security add in will generate extra data in the export to export the token information.

Export Formatting

- **Columns Header Template** allows us to format as we need the header column. {0} will be replaced by the answer text or id or display order and {1} will be replaced by the question text or id or display order. Note that both are optional and that you can format them in any custom string you like.
- **Column Header Output** will define what values to use to replace the {0} / {1} inside the header template column.
- **Respondent Answers Output** what value will be used in each respondent row's answer to show his selection type answers.
- **Respondent Text Answers Output** what value will be used in each respondent row's answer to show his field type answers.
- **Export Following Answers** we can either export all types of answers or select to export only selection or field based answers.
- **Multiple Answers** as this method shows the actual respondent answers in each question column we need a way to export also multiple choice questions. We can either choose to merge each respondent's answers from a multiple answer questions into a single column and separate each answer using a separator or we can choose to create a new column for each answer available in the multiple choice question.

If we create multiple columns each column will be formatted using the following format : question's text 1, question's text 2, question's text 3 etc .. until all question's answers have their own column.

- **Field Delimiter** is the char that will act as a delimiter between the exported answers columns inside a CSV.
- **Text Delimiter** is the char that will act as a delimiter between the text entries inside a CSV.

- **Replace CR** is the char that will replace "new line" character. Some tools like Excel or Access might encounter problems with "new line" characters inside a row. We suggest replacing it with a custom char before replacing it again after having imported the data inside the target tool.
- **Use Reporting Aliases** we can export the column names using the reporting aliases instead of using the full answers / questions label text.

Data Output Example

In order to better understand we have compiled a few example of questions and how their output is organized.

Standard single selection question with three selection answer types

What is your favorite color ? <input type="radio"/> red <input checked="" type="radio"/> green <input type="radio"/> blue	<table> <tr> <th>VoterId</th><th>What is your favorite color</th></tr> <tr> <td>1234</td><td>green</td></tr> </table>	VoterId	What is your favorite color	1234	green
VoterId	What is your favorite color				
1234	green				

Standard single selection question mixed with three selection answer types and one field type answer

How did you find this survey ?

☒ Great

☐ Amazing

☐ Lost my time

Comment : I am waiting for the next survey ...

VoterId	How did you find this survey ?)	Comment :(How did you find this survey ?)
1234	Great	I am waiting for the next survey ...

Standard single selection question with multiple selection and three selection answer types

What is your favorite drink ? <input checked="" type="checkbox"/> coke <input type="checkbox"/> water <input checked="" type="checkbox"/> fanta	<table> <tr> <th>VoterId</th><th>What is your favorite drink</th></tr> <tr> <td>1234</td><td>coke#fanta</td></tr> </table>	VoterId	What is your favorite drink	1234	coke#fanta
VoterId	What is your favorite drink				
1234	coke#fanta				

Standard single question with three field based types

Standard single question with three field-based types

4. Personal information :

Name :

John

Last name :

Doe

Country :

United States

VoterId	Name :(Personal Information :)	Last name :(Personal Information :)	Country :(Personal Information :)
1234	John	Doe	US

Xml export

The Xml export generates an easy to use Xml based file. Each question element has as child all its answers which in turn contains all the respondent answers. Each answer element contains already an aggregation sum of the total of its answers.

Data Export

- **Export encoding** is the format in which the file will be saved. If we are using a multi-language we strongly suggest to use UTF-8 as this will allow us to see entries written in any language.
- **Respondent Export Filter** is the filter that will be applied to respondents. We can create a filter using the report filter creator
- **Export Answers** we can either export all answers from the survey or a subset of answers we select.
- **Export From / To Date** is the date interval from which we want to export respondents.
- **Include Security Add in Data** do we include activated security add in data. E.g.: The token security add in will generate extra data in the export to export the token information.

SPSS export

Using the SPSS export tool we are able to export all our respondent information to the native .SAV format of SPSS. The export features full support for variables and labels.

As this option requires to generate temporary files we must grant write access to the SAVTempExport directory.

Note that this directory name / location can be changed from the web.config FeedbackServerSAVTempPath element.

Data Export

- **Respondent Export Filter** is the filter that will be applied to respondents. We can create a filter using the report filter creator
- **Export From / To Date** is the date interval from which we want to export respondents.
- **Include Security Add in Data** do we include activated security add in data. E.g.: The token security add in will generate extra data in the export to export the token information.

Export Formatting

- **Export Following Answers** we can either export all types of answers or select to export only selection or field based answers.

- **User Reporting Aliases** we can export the column names using the reporting aliases instead of using the full answers / questions label text.

This feature is not working on 64bit installations running Internet Information Services in 64bit. To run this features under a 64bit environment we must force Internet Information Services to run in 32bit mode.

Reports

FeedbackServer's reporting builder allows us to create reports and results analysis in an easy way.

Each report is composed of report items.

We can compose these items together to build a report that will suit our analysis need. We can have any number of report items we want per report, however it is recommended to use different paging depending on the load of data to analysis.

Depending on the report item capabilities we can either choose elements from questions, answers or attributes of panels. Note that for panels attributes results will be based on the actual value available for the panelist who participated the survey.

Report settings

From here we can create new reports and manage all report settings.

Report Options

- **Clone** will make an exact copy of the report and all report items that have been assigned to it.

Report Information

- **Report Name** is the name of the report.
- **Generate Template Report** will automatically generate a report based on our survey questions and answers when we first create the report.
- **Public Report** if our report is shared across other Feedback Server® users. Note that a report must also be public in order to make it publicly available to respondents of a survey.

Answer Filters

- **Filter Start Date** is the start date interval on which the results of the Report Items will be calculated.
- **Filter End Date** is the end date interval on which the results of the Report Items will be calculated.
- **Assign A Filter** using the report filter editor we can create specific filters for example if we only want to display the results of respondent who have chosen answer B to the question Z.
- **Language Filter** we can filter the results of the Report Items by the language chosen by the respondent. This feature is only available if we have turned on Feedback Server's Multi-Language features.



Report items are able to have their own filter settings and override the main report filters settings .

Report builder

The report builder is the heart of FeedbackServer's reporting features as it allows us create and edit the report we are going to analyze. A report can be build around report items, we can combine different report items together as we need it.

By default FeedbackServer offers already many types of report items to build our reports but it is very easy to develop and add our own specific report items inside FeedbackServer's reporting engine.

Page Options

- ▲ will move the page up until it reaches the next page break above.
- ▼ will move the page down until it reaches the next page break below.
- **Delete** will delete the page. Report items on the deleted page will be moved to the remaining page in the report.
- **Insert Report Item** inserts a new report item at the end of the page. If we want to insert a report item before another one we can use the insert report item link at the report item options level.

Report Item Options

- ▲ will move a report item's position up.
- ▼ will move a report item's position down.
- **Edit** the report item using the Report Item Editor.

- **Delete** the report item.
- **Clone** makes an exact copy of the question.
- **Insert Report Item** inserts a new report item before the current report item. If we want to insert a report item at the end of the page we can use the insert report item link at the page options level.
- **Insert Report Page Break** inserts a page break before the current report item.

Insert report item

We can insert as many report items per report as we want. FeedbackServer provides out of the box following report items :

- Graphics charts
- Web graphics
- Free text report
- Cross tabulation report
- Respondent entries list
- Individual Reach For Multiple Answers (IRMA)
- Matrix grid report
- Radar rating charts
- Constant sum
- Ranking
- Static text

Extended filters

Extended filters are unique feature that will allow us to apply filters based on text entries answers given by respondents.

Extended filters are available when you have marked at least one answer item to act as an extended filter.

Once you have marked an answer item you will be able to select, from a drop down list that contains all respondent text answers, the one that you want to use as a filter for your results.

As an example we could use the extended filters features along with the Xml - Country type to create geographic / demographic filters that will allows us to filter our report based on countries selected by the respondents.

Here is how the extended filters would look like with the example above.

As these filters are based on live users data we recommend sufficient amount of data.

Country : CH

Region : [select a text filter]

[select a text filter]

Geneva

Update Delete Clone

As you can see we can choose to set a filter based on "live" respondent's answers. Here we have set country and region to act as an extended filter. If we would set these filters the report would show all respondents answers from whom answered CH to country and Geneva to region.

We could have as many extended filters as we want. Each answer filter will be populated according to the other selections, the group of all filters must represent a valid combination of at least one respondent.

As extended filters are based on respondents answers it is highly recommended to create and apply the filters once you have finished the survey as this will give us the widest choice of filters available.

Report analysis

The report analysis will let us see the results of our reports in real time.

Filters

We can also apply filters to narrow down our results within our main report.

- **Start Date** is the start date that will be used to filter the data.
- **End Date** is the end date that will be used to filter the data.
- **Filter** is the filter that will be applied to the data.

File manager

The file manager lets us manage all the files that have been uploaded by the respondent using the File - Upload answer type.

Uploaded Files

We can see the list of uploaded files and download locally the files. Note that we can also see the respondent answers details by clicking on the "Details" link on the right of the "Download" link.

Export All Files To Server's Directory

If we have a huge number of files, downloading each file one by one can quickly become a very cumbersome task.

That's why we can also export all files that were uploaded to a directory on the server. Note that we need to have "write" rights on the server directory to which we will export the files.

- **Create File Group Directories**
 - No sub-directory groups will copy all the files inside the same directory.
 - Voter Id group will group the files uploaded by the same respondent under the same directory.
 - File-GUID sub-directory will group the files uploaded on the same answer item inside the same directory.
- **Full Path On Server** is the complete path of the directory on the server to which the files are going to be exported.

Report filters

Report filter allow us to setup logical rules based on respondent's answers or panel membership. Once we have created and grouped these rule in a filter we are able to apply this filter to any of our report or report item to show only a subset of results matching the filter.

Each condition is based on a set of rules that you can define. You can have as many conditions as you wish and order or re-order them at any time. A condition can also be based on the membership of the panel or one of the panel view if you want for example to show only results for people who are member of a given panel.

Rating groups

Rating groups is a feature that let you group questions that have their rating feature enable into a single group. Once grouped we can use FeedbackServer's Radar Rating Chart to display the average rating of a group and compare against several other group of questions on a single radar graphic. This feature is only available for questions that has been setup as rating enabled questions.

Rating Groups Options

- **Group Name** is the name of our group.
- **Questions** are the questions that are in the group. A single question can belong to any number of groups.

Report printer

Feedback Server's reporting printer allows us to generate printable version of any report in Microsoft Word Format or in a web friendly based format.

Report printer also allows us to print out the complete list of respondents along with their answers.

If you want to generate PDF's from your word document we suggest using the free CutePDF printer for Windows.

Filters

We can also apply filters to narrow down our results on our printed version.

- **Start Date** is the start date that will be used to filter the data.
- **End Date** is the end date that will be used to filter the data.
- **Filter** is the filter that will be applied to the data.

Data import

Data import let us import new respondent data in our survey. We can import any Comma Separated Value based text file that has been either exported from external tools, other survey packages or FeedbackServer itself.

The first line of the imported CSV file will always been used by Feedback Server® to determine the column. No data can be used on the first line.

Data Import Step 1

- **Import File** is the respondent file in CSV format that we want to import.
- **Column Delimiter** is the char that will act as a delimiter between the imported answers columns inside a CSV.
- **Text Delimiter** is the char that will act as a delimiter between the text entries inside a CSV.

- **Selection Answers Select Mark** is the char that is used within the CSV answers to set a Selection Answer type to selected.
- **Selection Answers Un-Selected Mark** is the char that is used within the CSV answers to set a Selection Answer type to un-selected.
- **Display Reporting Alias** will show reporting aliases in the mapping answer import step instead of the question, answer texts.
- **Map Header Against Alias** will try to make a direct mapping between the current survey reporting aliases and the CSV header columns text description.

Data Import Step 2 - The mapping

The step 2 of the process let us map each CSV column to one of our survey answer attribute. We can also map voter attributes to CSV columns.

Tips & Tricks To Import / Export Using FeedbackServer

Its of course possible to import CSV information from another FeedbackServer installation.

The easiest way to go if we want to import existing data from another survey. Is to set all reporting aliases for each answer properly.

Once this is done we can use the standard CSV format to export the data. The only thing to change is to set the "Columns Header Template" to {0} and check the "Use reporting alias". This will generate a CSV with all our reporting aliases as column description.

Once we have generated our CSV file we can then re-use it directly in the import tool by checking the "Map Header Against Alias". FeedbackServer will then automatically map all CSV columns to the current survey answers.

Data generator

The data generator allows us to generate automatically a set of random generated test data. This can be useful to quickly fill our survey results and build our reports based on this data to make sure that everything is working as expected.

- **Number of respondents** is the number of respondents that we want to generate.
- **From date** is the start date from which the test respondent have started the survey.

- **To date** is the max response date that can be generated.
- **Language** let us choose which languages we want to assign randomly to our test data respondents.
- **Response type** is the type of response that we want to generate. We can generate a set of completed responses or also generate a set of partial responses which simulates unfinished survey entries.

Respondent Analysis

The respondent analysis screen let us manage in a easy way all respondents that filled our forms.

- **Search Text Entries** we can use this features to select an answer and filter our respondent based on our text search and their entry for this answer.
- **Voter Status** let us choose if we want to display only respondent who completed the survey (validated), respondent who didn't complete the survey yet or archived respondents.
- **Filters** is the filter we want to apply to our respondent. Filters can be created using the Report Filter features.
- **Extra answers to show** let us select two answers that we want to display within the grid as extra information about the respondent.

Archiving respondents

Archiving of respondent allows us to put each respondent in an archive state. The main advantage of archiving respondent is that these respondent are not anymore taken into account by security add ins. As such we can re-allow respondent to fill the form, this is particularly useful when we want to survey the same respondents on different time frames during the year.

Security

Security add ins

IP protection

This security add in will protect our form against multiple submissions by recording the IP of the respondent and not allow him to fill the form a second time after he submits his answers.

- **IP Expires After** is the number of minutes after a respondent IP that was recorded will be allowed to submit answers again .

If you respondent are behind firewalls or proxies some respondents might have the same IP address. FeedbackServer does it best to get the real IP but depending on the mode activated on the firewalls or proxies it is sometime not possible to get a unique IP per respondent.

Cookie protection

This security add in will protect our form against multiple submissions by setting a cookie on the client's browser.

- **Cookie Expires After** is the number of minutes after a cookie will expire and the respondent will be allowed to submit answers again.

There is no way to prevent the respondent from clearing its cookies once he has filled out a form.

Password protection

This security add in will protect our survey using a password. Only respondent knowing the right password will be able to access the survey.

- **Password** is the password that protects the form access.

Email code protection

This security add in will protect our survey using the email invitation code provided to the respondent when he received his invitation message .

- **Only Invited Emails Can Participate** we can open the survey to respondent who didn't not receive any invitation code. If this option is activated respondent without code will be able to take the survey several times while respondent with the invitation code will only be able to submit once their answers.

Its important to keep in mind that the email code protection add in will also link the respondent answers to the current email filling out the form.

Token protection

This security addin will protect our survey using unique token generated using the token generator.

- **Available Tokens** is the number of token available for use.
- **Used Tokens** is the number of token that have been already used. A token can only be used once.
- **Allow Access To Valid Tokens Only** we can open the survey to respondent who don't have any token. If this option is activated respondent without token will be able to take the survey several times while respondent with token will only be able to submit once their answers.
- **Token Source Variable Name** is the variable name from which FeedbackServer will look after in the querystring and session to see if a token has been set. If nothing is set or no value is found inside the variable an interface will be shown to the user to enter his token.

This feature is generally used along with the Emailing Invitation Features as you can include a link in your invitation that would have following parameters :
`http://www.mydomain.com/fs.aspx?surveyid=[--surveyid-]&token=[--invitationtoken-]`

In this case the source variable name is "token".

Asp.net security context

This security addin will protect the survey based on the ASP.NET security context . It will check the Context.User.Identity.Name object of ASP.NET and store the value of it to prevent multiple submissions.

- **Allow Multiple Submissions** is the same user allowed to take the survey twice.

FeedbackServer security context

This security add in will protect the survey based on the FeedbackServer security context. It will check the current FeedbackServer User that is logged in and store its user name to prevent multiple submissions. If no context is found the security add in will automatically show a login interface that will allow the user to enter his FeedbackServer credential to access the survey.

- **Allow Multiple Submissions** is the same user allowed to take the survey twice.

Panel security context

This security add in will let us allow access only to panelist . The other purpose of the panel security add in is that it will allow FeedbackServer to detect and eventually map the current panelist based on the current respondent of the survey.

Form Security Options

- Give survey access to let us choose to whom we're going to give access.

Everyone lets any person panelist or not take the survey. Detected panelist will have their answers linked to them once the respondent posts his answers.

Only panelists lets only valid detected panelists take the survey. These can be panelists from any available panel.

Invited panelists lets only invited panelists take the survey. Only panelists that have received a valid invitation from the panel survey mailing tool will be able to take the survey.

- **Allow multiple submission** let us allow logged panelist to take the survey several times.
- **Hide Login Screen** in case we don't want to allow respondent to enter their panelist ids manually we can disable the login screen if FeedbackServer doesn't find automatically any panelist id.

Its important to keep in mind that the panel security context will also link the respondent answers to the current panelist taking the survey.

Panelist Mapping Options

- **Panelist id source** is the variable name that will holds the panelist id. Querystring, cookie and session object will be checked to see if they contains a valid panelist id that FeedbackServer can use to either map the respondent

answers at the end of the survey or in case of a linked panel to a survey to show the panelist information to the respondent.

- **Id source type** is panelist attribute against the panelist source will be matched. Very useful if we have existing unique ids within a panelist attribute that we want to use to show the panelist information.

In case we use an attribute we must make sure that the attribute value is unique across all panelists. In case of multiple values FeedbackServer will always and only match the first one found.

Active directory security

This security add in will protect the survey and allow only respondent who have a valid Active Directory account to access it . It will also prevent multiple submissions from the same Active Directory User to take the survey twice.

- **Allow Multiple Submissions** is the text that will be shown next to the selection item (radio, check box) or inside the drop down list.

In case FeedbackServer is running in a mixed anonymous / active directory access mode we strongly suggest for security reason to enable SSL on the public part where the user will enter his Active Directory credentials to enter the survey.

Entry number limitation

This security add in will allow us to set a maximum of respondent who can fill a form .

- **Entry Count** is the current number of respondents of the form.
- **Max. Entries Allowed** is the maximum respondent entries allowed for the form
- **Quota Reached Message** is the text that will be shown instead of the form once the maximum entries have been reached.

Image password

This type will render a an image with a random generated code that the respondent must type manually to enter the form. This system is to protect the survey form against automatic bots.

Email code confirmation

The email confirmation add in allows us to use an existing field in our form that collects an email from the respondent and use that email to send out a validation code to it. The respondent will need to validate his answers using the code that was send out to this email.

- **Source Email Answer Question** is the question from which we want to select the email related field.
- **Answer Email Recipient** is the field that will be used as a recipient to send out the confirmation.
- **From Email** is email that will be used to send out the confirmation email to the recipient.
- **Subject** is the subject of the confirmation email.
- **Confirmation Email** is the body of the confirmation email. The confirmation link will replace the the [--confirmationlink-] at run time.
- **Confirm Action** is the action that will happen once a respondent has confirmed his answers.
- **Confirmation thanks Page Message** is the text that will be displayed to the respondent once he has confirmed his answers.

Token Security

Security tokens

Token security will allow us to create unique tokens and protect our survey using these unique tokens. Token security is the most secure method to prevent multiple submissions as it is only possible to take the survey if we have a valid token.

Another advantage is that we can link a user identity for each token created, this way we know exactly who did take or didn't take the survey and as token is just a text we can distribute them using any electronic or non electronic media we want from standard mail, phone to emails.

Token security has already been used on major projects to conduct large ballots with great success.

Token generator

FeedbackServer provides out of the box a tool to generate automatically, create or import authentication tokens for our forms.

Token Generator

- **Number Of Tokens To Generate** is the number of token we want to generate. Note that a token can only be once in the token list unless its deleted after being created.
- **Token Type** is the type of token we want to generate.
- **Token Length** is the length of the generated tokens for non GUID type token

Add Single Token

- **Token Value** the value of the token to store in the token database.
- **First Name** first name of the identity linked to the token.
- **Last Name** last name of the identity linked to the token.
- **Email** email linked to the token.

If we are using the Email Invitation Features FeedbackServer will automatically try to lookup targets distribution email and try to match a token email identity to send out the matching token to the email.

Import Tokens

Let us import an existing list of tokens in the following format :

TokenValue, FirstName, LastName, Email
TokenValue, FirstName, LastName, Email
TokenValue, FirstName, LastName, Email

...

If a token value is missing it will be replaced by GUID type token.

Token status

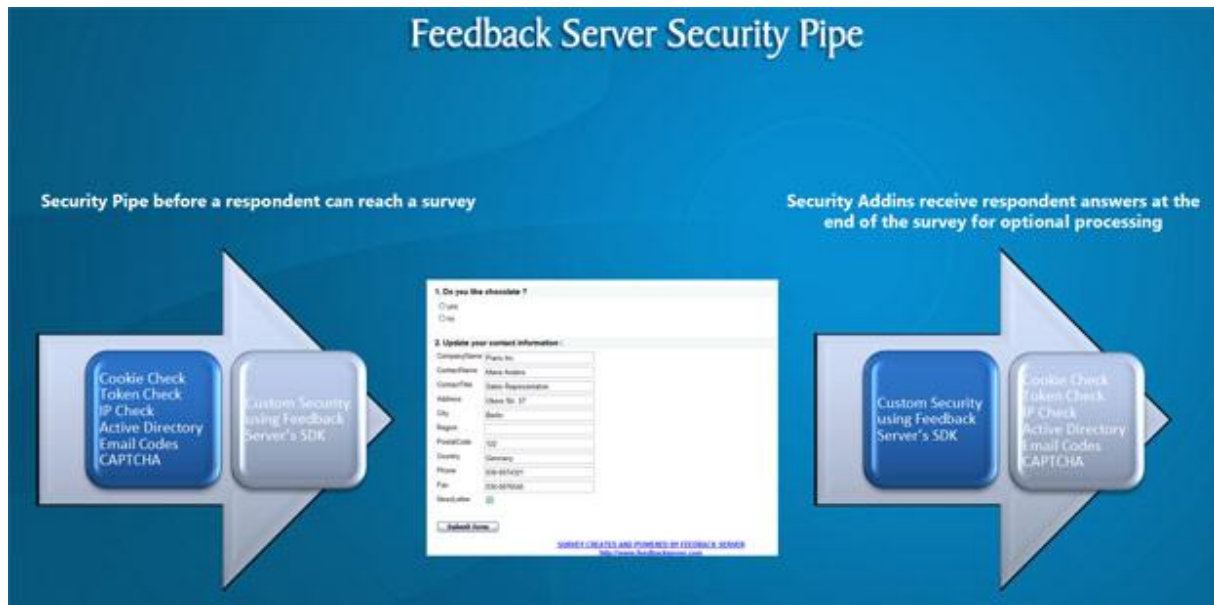
The token status page will allows us to see what is happening with our tokens.

- **Used tokens** is the list of tokens that have already been used by respondents.
- **Available Tokens** is the list of tokens that were created but not yet used by any respondent.

Security

FeedbackServer provides with us with a very rich add in system in order to protect our surveys.

Security in FeedbackServer is based on a "security pipe". You can put as many security add in as you want in the pipe. The respondent will only reach the survey once he has been authenticated by all the security add ins that have been put in the "security pipe" using the form security page.



As you can see in the security pipe architecture, we could easily add our own security add in using FeedbackServer's SDK to match our enterprise's security system.

Following security add ins are provided out of the box to build our security pipe :

- IP Protection
- Cookie Protection
- Password Protection
- EMail Code Protection
- Token Protection
- ASP.NET Security Context
- FeedbackServer Security Context
- Active Directory Security
- Entry Number Limitation
- Image Password
- Email Confirmation Code

Form security

The form security allows us to build our "security pipe" with the security add ins the respondent will be authenticated against.

Security Options

- **User Has Not Been Authenticated By All Addins** lets us choose if we want to show the survey if authentication with the security add ins has failed. Note that if one of the security add ins in the pipe shows an interface (e.g.: password add in), this interface will still be shown.
- **Add In Which Provide Unique Keys** lets us choose which of the security add ins that supports it will replace the default FeedbackServer unique Key Provider for the resume and change modes on the survey.

Following Security Add ins support the key provider architecture and can be used to replace the default FeedbackServer unique key provider :

1. Email Code Protection
2. Active Directory Security Add In
3. Token Protection

Security Add In Options

- ▲ will move the security add in up in the security pipe.
- ▼ will move the security add in down in the security pipe.
- **Disable** will disable the security add in from the pipe. By disabling the security add in it won't try to authenticate the respondent, the advantage is that if the security add in has stored specific information related to respondents (tokens, email codes etc...) these information won't be lost.
- **Delete** will delete the security add in from the pipe including all information it might have saved about respondents.
- **Check Archived** if we want that the security add in checks also the archived respondents for authentication, multiple submission etc ..
- **Insert Security Add** in inserts a new security add in inside the pipe.

Insert security addin

Following security add ins can be added to the security pipe :

- IP Protection
- Cookie Protection
- Password Protection
- Email Code Protection
- Token Protection
- ASP.NET Security Context
- FeedbackServer Security Context

- Active Directory Security
- Entry Number Limitation
- Image Password
- Email Confirmation Code

A security add in can only be added once in the security pipe.

Key provider

This key provider model allows us to have full control on the way unique codes are generated in FeedbackServer. Functions like save / resume progress or answers changes requires to generate unique ids in order to allow the respondent to restore his session in a secure way.

By default FeedbackServer is generating a unique code that will be delivered to the respondent if he choose for example to save his progress during a survey. But what happens if we have invited the respondent using an email invitation that has already a unique code for each email or the respondent has already received a token ?

That's where the key provider model comes in, it will allow us to specify which security add in will take the responsibility to provide and manage the unique keys required to restore the respondent sessions and answers.

Following security add in can act as a key provider for a form :

- Email Code Protection
- Active Directory Security Add In
- Token Protection
- Panel Security Context
- ASP.NET Security Context

Coming back to our email invitation example, if we want to allow the respondent to resume a session later on using the same code than the one that is used by email code protection we could set the email code protection security add in to act as a key provider and the respondent will have only one code to use for all action related to his session.

Some providers like the Active Directory Security goes even a bit further as they automatically detect if an Active Directory user has saved a session and will resume it without having the user to enter any code.

Multi-Language

Multi language forms

There are often times when we have a worldwide audience of respondent who don't speak the same language and are not always knowing English. We could create for each language separate surveys but this would result in big complications while compiling and aggregating the final results.

FeedbackServer has the flexibility to translate a single survey into multiple languages and let the respondent choose in which language he wants to take the survey.

At the end of the survey FeedbackServer's reporting tools will provide us a way to either have a global look at our results or a more granular look per respondent's languages.

FeedbackServer engine handles also languages written from right to left, like Arabic.

Multi language settings

Using the multi-language settings we can enable our forms to support multiple languages.

- **Enable Multi-Languages** enable our survey to support multiple languages.
- **Multi-Languages Mode** is the way the language of the survey is going to be selected by the respondent. FeedbackServer provides several ways of choosing the language of a survey at runtime.
 - **User List Selection**
The respondent will be able to choose the language of the survey from a drop down list.
 - **User Flags Selection**
The respondent will be able to choose the language of the survey from a list of graphical country flags.
 - **Detect From Browser**
FeedbackServer will get the language of the browser and will switch to it if its language is available in the survey.
 - **Get From QueryString**
FeedbackServer will get the language of the query-string variable and will switch to it if its language is available in the survey.

- **Get From Cookie**
FeedbackServer will get the language of the cookie variable and will switch to it if its language is available in the survey.
- **Get From Session**
FeedbackServer will get the language of the session variable and will switch to it if its language is available in the survey.
- **Enabled Languages** is the language that will be available to the respondent and to the survey administrator at edit time.
- **System Language** allows the administrator to see in which language FeedbackServer runs on the server. If you want to change the default language of the survey we may do so from the form settings page where we can switch the default language of the form.
- **Introduction text** let us write some text that will be shown on top of the language selection.

Multi language export / import

In order to make the exchange between the form administrators and translators easy, FeedbackServer provides us with XLIFF format export and import features.

XLIFF being a standard format based on XML it is supported today by most professional translation tools used by translators. If you want to learn more about the XLIFF format you can check the OASIS XLIFF v1.1 specifications.

Translation XLIFF Export

Export From is the language from which you want to export the texts of the survey. Note that if you have started to translate a survey but didn't finish it untranslated texts will be exported from the the default language one.

Translation XLIFF Export

Import Language File is the XLIFF file that contains the <source> and <target> translations.

Import As is the target language to which we want to associate the imported translations from the XLIFF file.

System messages

All system messages or texts (administration / web controls) can be very easily translated as these messages are stored inside an external Xml file located by default in the "XmlData/Languages/" directory. Each language has its own Xml file with the messages in that language. By default FeedbackServer comes with English and French Xml files.

We can make a new language Xml either manually by copying the template native.xml file, renaming the copy to our local language code and translate the values inside it. We can also use the System Messages Manager for a more user friendly experience.

As the Xml file contains also the survey web control's system messages like error messages or submit buttons texts (xx is required, next page etc...) we could change these messages by changing the current language Xml. However its is important to note that these changes will affect all FeedbackServer users.

Messages Manager

- **Translate Messages Into** is the target language in which we want to translate the messages.
- **Web Server's Application Language** is the current language of the web application in which FeedbackServer is running in.
- **Percent Of Messages Translated** is the percent of messages that were already translated in the target language.

Messages Translation

- **Show All** messages.
- **Show Translated** all messages that were already translated.
- **Show UnTranslated** all messages that are not translated.
- **Control Messages** are the message that appear either in the survey or report web control.
- **Admin Messages** are the messages on the Feedback Server's administration interface.
- **Orginal Default Text** is the original text.
- **xxx's Text** is the translated text.
- **Update Language Message** will update the Xml language file of the target language, if no Xml file exists for the target language a new one will be created. If we want to update the messages of the current web application language we have to select the same target language as the web server's application language.
- **Sync 2 Letter Code Language** some languages can have regional codes like "en-US" is for the US English. If FeedbackServer cannot find a matching 2

letter code for a given language it will try to match the single letter code like "en" for English codes.

User Management

User management

Each FeedbackServer installation can be used by multiple users at the same time and any number of users can connect simultaneously to the administration interface.

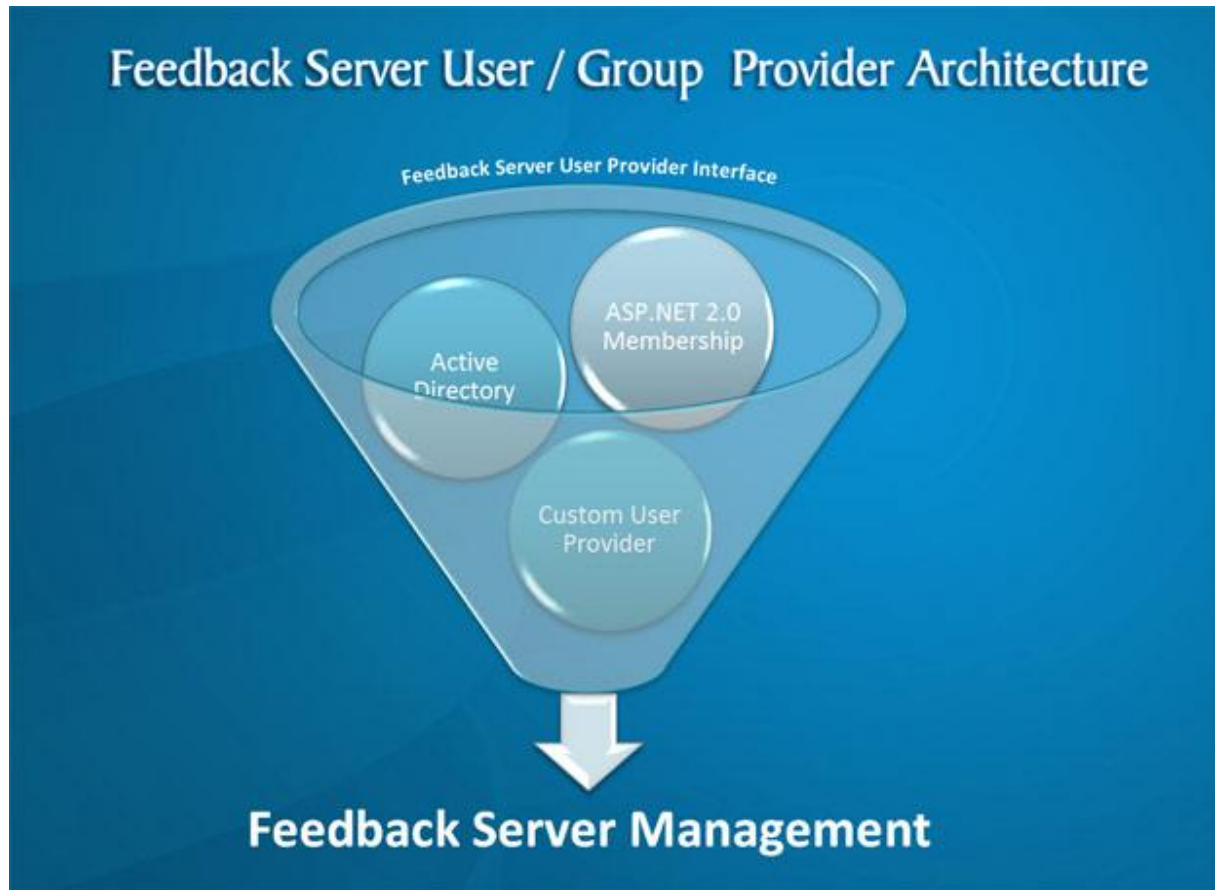
We can create / edit users and setup granular access rights for each user. We can also use the multi-users features to create logins where user can only answers given forms and prevent multiple submissions using the FeedbackServer Security Context add in.

Each user's forms, custom answer types, styles or panels will be isolated from the other users. Only administrator are allowed to see all information that is available inside the system without any restrictions.

Following hierarchy is applied in Feedback Server's users management :

1. Groups belongs to roles.
2. Rights belongs to roles.

FeedbackServer comes out of the box with a very flexible user architecture letting us "plug" and reuse almost any user store to manage our users and groups against our surveys, panel or administrative rights.



FeedbackServer user management is very flexible and comes by default with the three user providers :

- FeedbackServer's User Provider
- ASP.NET Membership Provider
- Active Directory Provider

It is also very easy to develop and plug into FeedbackServer your own user provider using the FeedbackServer's SDK and unleash your existing user and group base.

User manager

This page will let use edit our users in the system. Depending on the user provider we might no be able to edit the user but only assign roles and surveys to him.

- **User Name** is the user name to log into FeedbackServer.
- **User Password** is the password to log into FeedbackServer. This password is encrypted in the database and cannot be recovered.
- **First Name** is the first name of the account's user.

- **Last Name** is the last name of the account's user.
- **Email** is the email of the account's user.
- **Administrator** gives administrator rights to the user.
- **Assign All Surveys** user will be able to access all forms. Note that if he is not administrator the roles will apply for what he can do or not do on the surveys.
- **Assigned Surveys / Folders** what surveys, folders or report does the user have access to.
- **Assigned Panels** what panels does the user have access to.
- **Roles** what roles / rights has the user.
- **Delete User** deletes the user. Note that user's surveys are not deleted with the user.

User creation

This page will let use create our users in the system. Depending on the user provider we might no be able to edit or create users but only assign roles and forms to him.

- **User Name** is the user name to log into FeedbackServer.
- **User Password** is the password to log into FeedbackServer. This password is encrypted in the database and cannot be recovered.
- **First Name** is the first name of the account's user.
- **Last Name** is the last name of the account's user.
- **Email** is the email of the account's user.
- **Administrator** gives administrator rights to the user.
- **Assign All Surveys** user will be able to access all forms. Note that if he is not administrator the roles will apply for what he can do or not do on the surveys.
- **Assigned Surveys / Folders** what surveys, folders or report does the user have access to.
- **Assigned Panels** what panels does the user have access to.
- **Roles** what roles / rights has the user.

Group manager

This page will let use create or edit our groups in the system. Depending on the user provider we might no be able to edit the group but only assign roles and surveys to it.

- **Group Name** is the name of the group.
- **Members Are Administrator** makes all members administrators.
- **Assign All Surveys To Group** members will be able to access all surveys.
Note that if he is not administrator the roles will apply for what he can do or not do on the surveys.
- **Assigned Users** allows us to assign users to the group.
- **Assigned To Group** allows us to assign surveys to the group.
- **Assigned Surveys / Folders** what surveys, folders or reports does the user have access to.
- **Assigned To Panel** allows us to assign panels to the group.
- **Roles** allows us to assign roles / rights to the group.

Roles manager

This page will let use create roles that have specific rights.

- **Role Name** is the name of our role.
- **Role's Rights** is the rights that will be assigned to the role.

User import

This page will let us import existing users.

- **Import CSV Data** we can cut & past a list of users we want to import into FeedbackServer.
- **Administrator** gives administrator rights to the users.

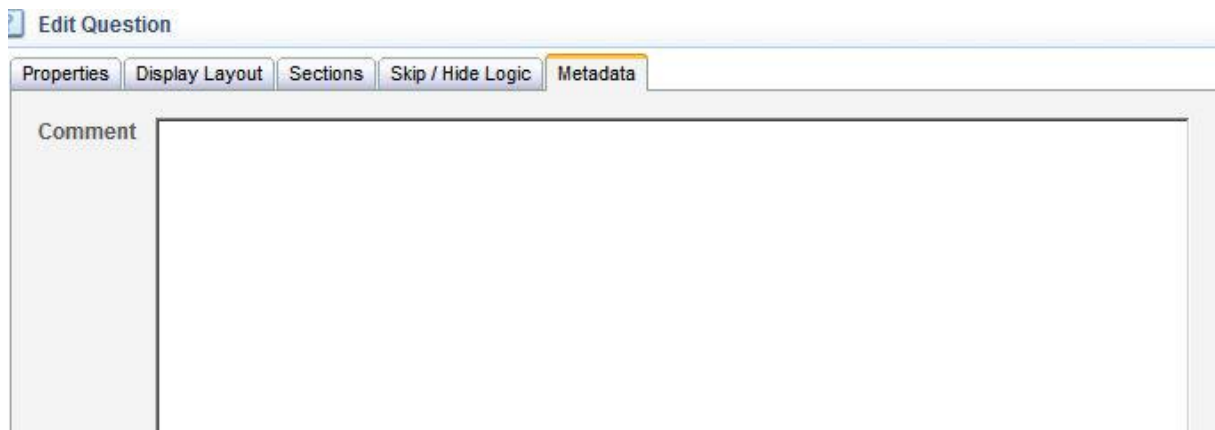
- **Assign All Surveys** users will be able to access all forms. Note that if he is not administrator the roles will apply for what they can do or not do on the surveys.
- **Assigned Surveys** what surveys does the users have access to.
- **Roles** what roles / rights has the user.

Metadata

Metadata

FeedbackServer's metadata system allows us to extend globally parts of the software with our own custom defined attributes. Using metadata we can extend and add information to existing system entities like forms, questions, answers or respondents details. Note that extending entities global and will affect all items and users within the system.

If for example we wanted to collect a comment for our questions we could using metadata add a new comment attribute to our question entity which will act as if it has been built right into the system.

The screenshot shows a web application window titled 'Edit Question'. It has a tabbed interface with five tabs: 'Properties', 'Display Layout', 'Sections', 'Skip / Hide Logic', and 'Metadata'. The 'Metadata' tab is currently selected and highlighted in yellow. Below the tabs, there is a large text area. On the left side of this area, the word 'Comment' is displayed in a light gray font, indicating the type of metadata being added. The rest of the text area is empty, providing space for the user to enter the comment text.

We can use any built in answer type to collect information or we could also create our own answer type using the type editor or SDK.

Metadata Editor

- **Alias** is an internal alias that we can use for reporting purposes.
- **Label** is the label that will be displayed in the entity metadata properties tab.
- **Type** is the answer type that will be used by the attribute to collect information.

Index

1	
1.79E	69
2	
2008 Edition	5
3	
3.40E	69
32bit	5
3D	47
3d matrix	47
6	
64bit	5
A	
active directory	18, 20
Active Directory Username	20
AD	20
ADController	20
ADDomain	20
Admin	62
ADPassword	20
ADSearcherFilter	20
ADToDBSyncTimeOut	20
ADUserName	20
Again	50
Ajax	40
alert	36
Alert Triggers	36
Alias	52, 81
Reporting	52, 81
Allow	57, 83, 95
check	83
Allow Panelist Creation	88
Allow Respondents To Download Their Answers	40
Almost	83
Always Display Submit Button	40
Anonymous Respondents	40
Another powerful feature	62
FeedbackServer	62
answer	36, 40, 44, 45, 46, 47, 51, 52, 56, 57, 61, 62, 63, 65, 69, 75, 77, 81, 84, 88
Answer Pipe Value	84
Answer Piping	62
Answer publisher its	62
Answer Text	52
Answer type creator	61
Answer Types	46, 81
number	46
Answers Only Alert	36
Answers Pipe	84
Anuary	69
Arabic, Hebrew	77
ASP	5, 91
Attachements	36
authentication	157
Avoid creating them again	50
B	
Base its	53
Base their	52
Based	57
number	57
Based language	83
string	83
Been designed	16
users	16
behaviors	77
Beside	69
Big	69
Bind	62
birthday	66
Bit	69
bool	65
Bot	59
Bound	61
branching	75, 103
Built In	95
Buttons Row Display	40
C	
cache	24
Calculation Mode FeedbackServer	81
calendar	57
captcha	59
capture	58
Change	20
LDAP	20
Check	83
allows	83
check box	52, 53, 65, 81, 95
checkbox	46, 47
Checkboxes	46
Chosen	52, 53
Clear Branched	40
cluster	24
collection	63

- Collection Editor.....63
- Collection types let.....63
- Column.....46
- Column Header.....81
- Comes.....88
- comment.....56
- Complete control.....77
- Completion alerts.....36
- Completion Message.....40
- Composed.....46, 47
 - multiple sub.....47
 - rows.....46
- condition rules.....103
- config.....20, 24, 62
- Configuration.....5, 81, 88
 - run.....5
- Configure.....16
 - SQL.....16
- Confirm.....59
- ConnectionString.....20
- constant.....58
- Constraints.....81, 88
 - parent question.....81
 - question.....88
- contacts.....119
- content.....75, 84, 107
- Cookie Variable Value.....84
- copy question.....45
- country.....46, 61, 62, 63
- Countrysalias.....62
- create 36, 40, 46, 50, 56, 57, 61, 62, 63, 77, 83, 88, 91, 95
- Create our own Xml bound type.....61
 - possible.....61
- Created using.....40, 83, 95
 - WYSIWYG Style Editor.....40
- CSS.....95
 - export.....95
- csv.....136
- Current value based.....84
 - respondent state.....84
- Custom.....63
- Customerid.....62
- Customername.....62
- Customize our form features.....40
- D**
- data.....136, 148
- Datasource.....62, 63
- date.....40, 57, 66, 69
 - pick.....57
- date range.....143
- Day.....66
- December 31.....69
- Decimal.....69
- decrypt.....24
- Default.....40
 - FeedbackServer key.....40
- Default Text Value.....61
- Deploy our form.....91
 - FeedbackServer provides several ways.....91
- Deploy our form through invitation popup windows.....92
- Deploy our survey.....93
- deployment.....40, 93, 95
- DHTML.....92
 - using.....92
- Different tag sources.....84
- Disable Response Timer.....83
- Display Header.....83
- Display Next Page Button.....40
- Display Previous Page Button.....40
- Display Restore Button.....40
- Display Resume Progress Button.....40
- Display Save Progress Button.....40
- Displayed.....50
 - respondent.....50
- Drop down lists.....66
- dropdown.....46, 93
- Dropdownlist based.....62
 - value posted.....62
- dynamic.....107
- dynamic ranking.....57
- E**
- Easy.....63, 84
 - use administration interface.....63
 - use interface.....84
- Edition Language.....63, 92
- Eg.....62, 63, 77
- email.....36, 40, 57, 97, 100
- Email From.....36
- Email From Name.....36
- email template.....101
- Emails.....36, 83, 91
- Enable.....51
 - respondent.....51
- Enable / Disable Undo.....77
- Enable integrated windows authentication.....20
 - web server.....20

FeedbackServer

encryption	24	Fixed Display Order	52
entry validation	54	flash.....	96
Error Message.....	83	Flexibility.....	16
Existing.....	16, 18	maximum.....	16
Feedback Server.....	18	Float	69
SQL Server.....	16, 18	Following.....	18
expert mode.....	45	URL.....	18
Export.....	95, 136	Footer	40, 83
CSS.....	95	Footer Progress Display	40
Export CSS	95	Footers	95
Export our form	44	Force https.....	40
Export page lets	44	form.....	5, 20, 36, 40, 44, 45, 50, 52, 59, 62, 63, 75, 77, 83, 84, 91, 92, 95
Expression.....	83	form using	40
given text matches	83	questions being part	50
extended.....	145	form builder.....	40, 62, 63, 77
Extensive ways	69	Form Builder Functions	77
F		Form builder using	40
Feedback Server....	16, 18, 20, 63, 77, 83	question.....	40
existing.....	18	Form Elements.....	77
Feedback Server image	77	Form metadata	44
Feedback ServerAdminADUsername	20	Form using	40
FeedbackServer5, 18, 20, 36, 40, 45, 46, 50, 51, 62, 77, 81, 83, 91, 92, 95		form	40
another powerful feature.....	62	free form	77, 83
installation	18	free text.....	45
FeedbackServer allows	50, 83, 92	friendly link.....	91
FeedbackServer forms	77	Fs_install.....	16, 18
FeedbackServer key	40	Full Entry Report Alert	36
default	40	G	
FeedbackServer Multi.....	88, 92	Gather	69
FeedbackServer provides	91	highest possible data quality	69
FeedbackServer provides several ways ..	91	gender.....	62
deploy our form.....	91	General settings allows	40
FeedbackServer requires	5	Get Language From.....	92
FeedbackServer requires NET 3.5 Service Pack	16	Gif.....	52
FeedbackServer user.....	20	Given text matches	83
FeedbackServer's Scoring	52	expression.....	83
FeedbackServer's SDK	69	Great way.....	36
field	46, 51, 54, 56, 57, 58, 81, 83	grid.....	51
Fields together	57	Group	69
file upload	64	Group multiple.....	57
filter.....	143	H	
filtering	143	Handle all addition	45
filters	103, 145, 147	He needs	51
Finished	36	hidden.....	50, 58
survey	36	Hide.....	69
Firefox	77	language	69
FireFox V2.....	5	Highest possible data quality	69
First step after.....	40	gather	69
		HTML	36, 45, 77, 91

- HTML links91
- Html pages91
- html question45
- HTTP_USER_AGENT58
- HTTPs40
- Human respondent59
 - its59
- I**
- image96
- image code59
- Image URL52
- images40, 44, 45, 52, 59, 77, 95
- import148
- Import our form into another
 - FeedbackServer installation44
- import question45
- Include Media44
- Include Metadata44
- Include Respondent Answers44
- Insert45
 - new question inside our form45
- install16, 40
- Install Feedback Server16
- Installation18
 - FeedbackServer18
- Installed FeedbackServer40
- Integer69
- Internet Explorer77
- Internet Explorer V75
- Internet Information Services16
- Internet Information Services 6.05
- invitation97, 100
- Ip 20, 40
- IP Protection Addin40
- Its59
 - human respondent59
- J**
- January69
- June69
- K**
- Key Provider40
- Keycode40
- Known Issues77
- L**
- Language69, 88, 92
 - hide69
- large46, 56
- Later on re44
- LDAP20
 - change20
- Let45, 84
- library96
- Lic file18
- link40, 57, 88, 91, 92
- Link Text92
- linked panel88
- Linked panel structure88
- Linked using57
- list20, 36, 47, 52, 53, 61, 62, 63, 66, 83, 84, 93
- Load62
- Located61
- login157
- Look like77
- M**
- mailing100
- mandatory56, 57, 62, 81
- Mandatory option56
- Many times51
- master child62
- matrix46, 47, 81
- Matrix groups47
- Max Ranking Count57
- Maximum16
 - flexibility16
- media96
- message101
- Metadata40, 171
 - setup using40
- Metatdata40
- Method57
- Microsoft5, 16
- Min81
- Money69
- moss93
- Most basic question type available45
- MSSQL69
- multiple choice46, 81
- Multiple sub47
 - composed47
- multiple submission157
- Mydomain52
- Myimage52
- N**
- Name84
 - querystring84
- NET 3.5SP15
- NET Version 3.5 Service Pack5
- NEVER18
- New form40

New question inside our form.....	45
insert.....	45
Notepad	20
notification	36
Notification Subject.....	36
Notification To	36
Number.....	46, 57
answer types	46
based.....	57
O	
Of Progress	40
Open In New Window	92
Order	69
other .. 40, 50, 52, 53, 57, 62, 77, 91, 93	
Other piping	62
Other ranking.....	57
Our customers.....	91
Our form	77, 83
Our matrix based questions	81
Our question.....	57
Our survey	91
P	
page.. 16, 20, 40, 44, 46, 52, 62, 75, 83, 84, 91, 92, 93, 95	
Page Disabled.....	83
Page Header Image	40
Page let	93
Page Navigation Type.....	40
Page Title.....	83
Pages using	62
panel.....	84, 88
Panel Attribute Value	84
panels	119
Parent question	81
constraints	81
password	20, 59
pdf.....	96
Pick	57
date.....	57
Pipe source selector.....	84
Pipealias.....	62
popup.....	91, 92
Popup Every	92
Popup Message.....	92
Popup Title.....	92
Possible.....	20, 61
create our own Xml bound type.....	61
Previous / Next Page Navigation.....	40
Previous Pages.....	84
Provides	69

Public Style	95
Q	
Querystring.....	69, 84
name	84
Querystring Variable Value.....	84
question .. 40, 45, 46, 47, 50, 51, 52, 53, 57, 58, 62, 63, 75, 81, 83, 84, 88	
constraints.....	88
form builder using	40
Question composed	46
Question editor	46, 50, 52, 53
Question editor let	81, 88
Question Example	46
Question library	50
Question manually.....	45
Question Numbers.....	40
Question Options	81, 88
Question Row Width	81
Question wizard creation tool allows	45
Questions being part	50
form	50
R	
radio	46, 47, 52, 53, 81, 95
radio button	46, 47, 52, 53, 81
Random generated code	59
show	59
Randomize Row Order	81
ranking.....	57
rating	52
Weight.....	52
Re.....	44, 50
use	44
Read	59
Real	69
Registered License Upgrade	18
regular expression	57, 83
Render	56, 57, 66
set	66
standard	56, 57
repeatable	51, 81
Replaced	84
report	143, 144
report items	144
Reporting	52, 81
Alias	52, 81
required..... 5, 16, 18, 20, 40, 51, 56, 57, 62, 81, 91	
requirement	5
Respondent..... 50, 51, 57, 58, 59, 69	
displayed	50

enable.....	51	sharepoint.....	93
shown.....	58	Short Message Alert	36
specific.....	57	Show	59, 61
Respondent state	84	random generated code.....	59
current value based	84	XmlData directory	61
Respondents entries.....	69	Show Summary Of Respondent's Answers	
Responsents.....	40	40
results	143, 144	Shown.....	58
Rich answer types system FeedbackServer		respondent	58
.....	69	side by side.....	47
Role.....	91	single choice	46
Rows	46	Single question	46
composed	46	skip logic.....	69, 75, 103
rules.....	103	Skip logic conditions allow	69
Run	5	Skipped.....	40
configuration.....	5	Small	69
Runtime	50, 84	Small date	69
S		Sometimes its	20
Saved.....	66	Sometimes its just easier	77
score	39, 40, 52	Specific	57
Score Introduction.....	40	respondent	57
Score Point.....	52	SPS site.....	93
scoring	39	sql	16, 20, 62
SDK	77	configure	16
sections.....	18, 51, 62, 63, 81	Sql based query answer types	62
Sections allows	51	SQL Installer	18
security	157	SQL Server.....	16, 18
Select.....	84	existing.....	16, 18
Select month.....	66	SQL Server 2005.....	5
selection.....	46, 47, 52, 53, 62, 81, 95	SQL Server Express 2005	5
Selection Allowed	81	SQL Server Installation.....	16
Selection check	62	SqlBasedAnswerTypesAllowed	62
Selection mode	52, 53	setting	62
Selection type	52, 53	Standard.....	56, 57
Send email based notifications.....	36	render	56, 57
Servervariablename	62	start	40
Session.....	69	static question	45
Session Variable Value	84	static text.....	45
Set.....	62, 66	Store frequently used	50
render.....	66	String.....	83
SqlBasedAnswerTypesAllowed	62	based language	83
settings....	16, 24, 40, 45, 52, 56, 57, 92, 95	Style builder	95
setup..	16, 20, 40, 57, 69, 81, 83, 84, 88	Style Name.....	95
Setup following.....	20	Style templates.....	95
Setup logical rules based on respondent's		submit.....	40, 63, 75, 77, 83
answers	69	Submit Button.....	83
Setup specific features	83	subscriber	62
Setup using.....	40	sum.....	58
metadata	40	survey36, 40, 44, 52, 62, 83, 91, 92, 93,	95

finished	36	Using	92
Survey ASP	91	DHTML	92
Survey Description	40	V	
Survey Language	40	validation	56, 57, 69, 81, 83
Survey using	95	Validation Asterix	81
surveybox	93	Validation depending	69
swf	96	Validator	69
Swiss	57	Value	63
T		Value items	63
Tbcustomer	62	Value posted	62
Tbcustomers	62	dropdownlist based	62
template	36, 40, 44, 48, 95, 101	Variable Name	84
test	39, 83, 92	Very straightforward	18
Text	63	W	
Text / Html	77	We	40, 50
text box	54, 56, 57, 59, 81	Web	16
Their installation process	16	web control	93
Tiny	69	Web server	20
To Test	83	enable integrated windows	
true / false	65	authentication	20
Turkish	5	Web Server Installation	16
Type	56, 57, 58, 59, 61, 62, 65, 66	Web Server Variable Value	84
Types we	57	web.config	24
U		Website HTML pages	91
UID	40	Weight	52
Undo / Redo	77	Rating	52
United States	62	Which	50
upload	63, 64, 96	Which text cannot	59
URL	16, 18, 40, 52	Windows	5
following	18	Windows 2003	5
Urls	91	Windows Vista	5
US	62	Windows XP Professional	5
us states	61	Wish	16
Use	44	WSS	93
re 44		WYSIWG style editor	95
Use administration interface	63	WYSIWYG Style Editor	40
easy	63	created using	40
Use Default Page Navigator	83	X	
Use interface	84	xml	40, 44, 46, 60, 61, 62, 63, 95
easy	84	Xml file	62
Use them	50	Xml format which we	44
User login	20	Xml List	62
User To Change His Answers Once	40	XmlData directory	61
user	40	show	61
Username	20	Y	
users . 16, 18, 20, 36, 40, 50, 59, 62, 63,		Year	66
66, 69, 91, 95		Z	
been designed	16	ZIP file	16
User To Change His Answers Once ...	40		
Users trying	40		

