

DotNetNuke 4.0 Module Developers Guide (Part 2)

Michael Washington



Version 1.0.0

Last Updated: January 13, 2007

Category: DotNetNuke v4.0.5



DotNetNuke 4.0 Module Development

Information in this document, including URL and other Internet Web site references, is subject to change without notice. The entire risk of the use or the results of the use of this document remains with the user.

The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, places, or events is intended or should be inferred.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Perpetual Motion Interactive Systems, Inc. Perpetual Motion Interactive Systems may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Perpetual Motion, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Copyright © 2005, Perpetual Motion Interactive Systems, Inc. All Rights Reserved.

DotNetNuke® and the DotNetNuke logo are either registered trademarks or trademarks of Perpetual Motion Interactive Systems, Inc. in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.



DotNetNuke 4.0 Module Development

Abstract

This guide covers module development for the DotNetNuke framework, covering DotNetNuke 4 (ASP.NET 2.0)

Contents

Survey Module Source Code..... 1

This guide contains the source code for the Survey module covered in *DotNetNuke 4.0 Module Developers Guide (Part 1)* available at DotNetNuke.com 1

Additional Information.....133

Appendix A: Document History 134

Survey Module Source Code

This guide contains the source code for the Survey module covered in *DotNetNuke 4.0 Module Developers Guide (Part 1)* available at DotNetNuke.com

DNN4 Module Developers Guide

DataProvider.cs

```
//
// DotNetNuke - http://www.dotnetnuke.com
// Copyright (c) 2002-2005
// by Perpetual Motion Interactive Systems Inc. (
http://www.perpetualmotion.ca )
//
// Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated
// documentation files (the "Software"), to deal in the Software
without restriction, including without limitation
// the rights to use, copy, modify, merge, publish, distribute,
sublicense, and/or sell copies of the Software, and
// to permit persons to whom the Software is furnished to do so,
subject to the following conditions:
//
// The above copyright notice and this permission notice shall be
included in all copies or substantial portions
// of the Software.
//
// THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED
// TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR
PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
// THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR
OTHER LIABILITY, WHETHER IN AN ACTION OF
// CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER
// DEALINGS IN THE SOFTWARE.
//

using System;
using System.Data;
using DotNetNuke;

namespace DotNetNuke.Modules.Survey
{
    public abstract class DataProvider
    {
        // singleton reference to the instantiated object
        private static DataProvider objProvider = null;

        // constructor
        static DataProvider()
        {
            CreateProvider();
        }

        // dynamically create provider
        private static void CreateProvider()
```

DNN4 Module Developers Guide

```
{
    objProvider =
((DataProvider)(Framework.Reflection.CreateObject("data",
"DotNetNuke.Modules.Survey", "")));
}

// return the provider
public static DataProvider Instance()
{
    return objProvider;
}

// all core methods defined below
public abstract IDataReader GetSurveys(int ModuleId);
public abstract IDataReader GetSurvey(int SurveyId, int
ModuleId);
    public abstract int AddSurvey(int ModuleId, string Question,
int ViewOrder, string OptionType, int UserId);
    public abstract void UpdateSurvey(int SurveyId, string
Question, int ViewOrder, string OptionType, int UserId, int ModuleId);
    public abstract void DeleteSurvey(int SurveyID, int ModuleId);
    public abstract IDataReader GetSurveyOptions(int SurveyId);
    public abstract int AddSurveyOption(int SurveyId, string
OptionName, int ViewOrder, bool IsCorrect);
    public abstract void UpdateSurveyOption(int SurveyOptionId,
string OptionName, int ViewOrder, bool IsCorrect);
    public abstract void DeleteSurveyOption(int SurveyOptionID);
    public abstract void AddSurveyResult(int SurveyOptionId, int
UserId);
}
}
```

DNN4 Module Developers Guide

DataProvider.vb

```
'
' DotNetNuke - http://www.dotnetnuke.com
' Copyright (c) 2002-2005
' by Perpetual Motion Interactive Systems Inc. (
http://www.perpetualmotion.ca )
'
' Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated
' documentation files (the "Software"), to deal in the Software without
restriction, including without limitation
' the rights to use, copy, modify, merge, publish, distribute,
sublicense, and/or sell copies of the Software, and
' to permit persons to whom the Software is furnished to do so, subject
to the following conditions:
'
' The above copyright notice and this permission notice shall be
included in all copies or substantial portions
' of the Software.
'
' THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED
' TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR
PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
' THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR
OTHER LIABILITY, WHETHER IN AN ACTION OF
' CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER
' DEALINGS IN THE SOFTWARE.
'

Imports System
Imports DotNetNuke

Namespace DotNetNuke.Modules.Survey

    Public MustInherit Class DataProvider

        ' singleton reference to the instantiated object
        Private Shared objProvider As DataProvider = Nothing

        ' constructor
        Shared Sub New()
            CreateProvider()
        End Sub

        ' dynamically create provider
        Private Shared Sub CreateProvider()
```


DNN4 Module Developers Guide

```
        objProvider =
CType(Framework.Reflection.CreateObject("data",
"DotNetNuke.Modules.Survey", ""), DataProvider)
    End Sub

    ' return the provider
    Public Shared Shadows Function Instance() As DataProvider
        Return objProvider
    End Function

    ' all core methods defined below

    Public MustOverride Function GetSurveys(ByVal ModuleId As
Integer) As IDataReader
    Public MustOverride Function GetSurvey(ByVal SurveyID As
Integer, ByVal ModuleId As Integer) As IDataReader
    Public MustOverride Function AddSurvey(ByVal ModuleId As
Integer, ByVal Question As String, ByVal ViewOrder As Integer, ByVal
OptionType As String, ByVal UserId As Integer) As Integer
    Public MustOverride Sub UpdateSurvey(ByVal SurveyId As Integer,
ByVal Question As String, ByVal ViewOrder As Integer, ByVal OptionType
As String, ByVal UserId As Integer, ByVal ModuleId As Integer)
    Public MustOverride Sub DeleteSurvey(ByVal SurveyID As Integer,
ByVal ModuleId As Integer)
    Public MustOverride Function GetSurveyOptions(ByVal SurveyId As
Integer) As IDataReader
    Public MustOverride Function AddSurveyOption(ByVal SurveyId As
Integer, ByVal OptionName As String, ByVal ViewOrder As Integer, ByVal
IsCorrect As Boolean) As Integer
    Public MustOverride Sub UpdateSurveyOption(ByVal SurveyOptionId
As Integer, ByVal OptionName As String, ByVal ViewOrder As Integer,
ByVal IsCorrect As Boolean)
    Public MustOverride Sub DeleteSurveyOption(ByVal SurveyOptionID
As Integer)
    Public MustOverride Sub AddSurveyResult(ByVal SurveyOptionId As
Integer, ByVal UserId As Integer)

    End Class

End Namespace
```

DNN4 Module Developers Guide

SurveyController.cs

```
//
// DotNetNuke - http://www.dotnetnuke.com
// Copyright (c) 2002-2005
// by Perpetual Motion Interactive Systems Inc. (
http://www.perpetualmotion.ca )
//
// Permission is hereby granted, free of charge, to any person
// obtaining a copy of this software and associated
// documentation files (the "Software"), to deal in the Software
// without restriction, including without limitation
// the rights to use, copy, modify, merge, publish, distribute,
// sublicense, and/or sell copies of the Software, and
// to permit persons to whom the Software is furnished to do so,
// subject to the following conditions:
//
// The above copyright notice and this permission notice shall be
// included in all copies or substantial portions
// of the Software.
//
// THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
// EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED
// TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR
// PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
// THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR
// OTHER LIABILITY, WHETHER IN AN ACTION OF
// CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
// WITH THE SOFTWARE OR THE USE OR OTHER
// DEALINGS IN THE SOFTWARE.
//

using System;
using System.Xml;
using System.Xml.XPath;
using System.Data;
using System.Collections.Generic;
using DotNetNuke;
using DotNetNuke.Services.Search;
using DotNetNuke.Common.Utilities;
using System.Text;

namespace DotNetNuke.Modules.Survey
{
    public class SurveyController : Entities.Modules.ISearchable,
        Entities.Modules.IPortable, DotNetNuke.Entities.Modules.IUpgradeable
    {
        static public List<SurveyInfo> GetSurveys(int ModuleId)
        {
            List<SurveyInfo> SurveyInfolist = new List<SurveyInfo>();

```

DNN4 Module Developers Guide

```
        using (IDataReader dr =
DataProvider.Instance().GetSurveys(ModuleId))
        {
            while (dr.Read())
            {
                SurveyInfo colSurveyInfo = new SurveyInfo();
                colSurveyInfo.SurveyId =
Convert.ToInt32(dr["SurveyId"]);
                colSurveyInfo.Question =
Convert.ToString(dr["Question"]);
                colSurveyInfo.OptionType =
Convert.ToString(dr["OptionType"]);
                colSurveyInfo.ViewOrder =
Convert.ToInt32(ConvertNullInteger(dr["ViewOrder"]));
                colSurveyInfo.CreatedByUser =
Convert.ToInt32(dr["CreatedByUser"]);
                colSurveyInfo.CreatedDate =
Convert.ToDateTime(dr["CreatedDate"]);
                SurveyInfolist.Add(colSurveyInfo);
            }
        }
        return SurveyInfolist;
    }

    static public SurveyInfo GetSurvey(int SurveyID, int ModuleId)
    {
        SurveyInfo colSurveyInfo = new SurveyInfo();
        using (IDataReader dr =
DataProvider.Instance().GetSurvey(SurveyID, ModuleId))
        {
            while (dr.Read())
            {
                colSurveyInfo.SurveyId =
Convert.ToInt32(dr["SurveyId"]);
                colSurveyInfo.ModuleId =
Convert.ToInt32(dr["ModuleID"]);
                colSurveyInfo.Question =
Convert.ToString(dr["Question"]);
                colSurveyInfo.OptionType =
Convert.ToString(dr["OptionType"]);
                colSurveyInfo.ViewOrder =
Convert.ToInt32(ConvertNullInteger(dr["ViewOrder"]));
                colSurveyInfo.Votes =
Convert.ToInt32(ConvertNullInteger(dr["Votes"]));
                colSurveyInfo.CreatedByUser =
Convert.ToInt32(dr["CreatedByUser"]);
                colSurveyInfo.CreatedDate =
Convert.ToDateTime(dr["CreatedDate"]);
            }
        }
        return colSurveyInfo;
    }
}
```

DNN4 Module Developers Guide

```
    }

    public static void DeleteSurvey(SurveyInfo objSurvey)
    {
        DataProvider.Instance().DeleteSurvey(objSurvey.SurveyId,
objSurvey.ModuleId);
    }

    public static int AddSurvey(SurveyInfo objSurvey)
    {
        return
((int)(DataProvider.Instance().AddSurvey(objSurvey.ModuleId,
objSurvey.Question, objSurvey.ViewOrder, objSurvey.OptionType,
objSurvey.CreatedByUser)));
    }

    public static void UpdateSurvey(SurveyInfo objSurvey)
    {
        DataProvider.Instance().UpdateSurvey(objSurvey.SurveyId,
objSurvey.Question, objSurvey.ViewOrder, objSurvey.OptionType,
objSurvey.CreatedByUser, objSurvey.ModuleId);
    }

    #region Utility Functions
    public static int ConvertNullInteger(object Field)
    {
        if (Field == DBNull.Value)
        {
            return 0;
        }
        else
        {
            return Convert.ToInt32(Field);
        }
    }
    #endregion

    #region IUpgradable

    public string UpgradeModule(string Version)
    {
        return "Custom upgrade code goes here for Version: " +
Version;
    }
    #endregion

    #region IPortable Members

    string DotNetNuke.Entities.Modules.IPortable.ExportModule(int
ModuleID)
    {
```

DNN4 Module Developers Guide

```
StringBuilder strXML = new StringBuilder();
XmlWriterSettings settings = new XmlWriterSettings();
settings.Indent = true;
settings.OmitXmlDeclaration = true;
XmlWriter Writer = XmlWriter.Create(strXML, settings);

// Outer Loop - To build the Surveys
List<SurveyInfo> colSurveys = GetSurveys(ModuleID);
if ((colSurveys.Count > 0))
{
    Writer.WriteStartElement("surveys");
    foreach (SurveyInfo colSurveyInfo in colSurveys)
    {
        Writer.WriteStartElement("survey");
        Writer.WriteElementString("question",
colSurveyInfo.Question);
        Writer.WriteElementString("vieworder",
colSurveyInfo.ViewOrder.ToString());
        Writer.WriteElementString("createdbyuser",
colSurveyInfo.CreatedByUser.ToString());
        Writer.WriteElementString("createddate",
colSurveyInfo.CreatedDate.ToShortDateString());
        Writer.WriteElementString("optiontype",
colSurveyInfo.OptionType.ToString());
        // Inner Loop - To build the Options for each
Survey
        List<SurveyOptionInfo> colSurveyOptions =
SurveyOptionController.GetSurveyOptions(colSurveyInfo.SurveyId);
        if ((colSurveyOptions.Count > 0))
        {
            Writer.WriteStartElement("surveyoptions");
            foreach (SurveyOptionInfo colSurveyOptionInfo
in colSurveyOptions)
            {
                Writer.WriteStartElement("surveyoption");
                Writer.WriteElementString("optionname",
colSurveyOptionInfo.OptionName);
                Writer.WriteElementString("isincorrect",
colSurveyOptionInfo.IsCorrect.ToString());
                Writer.WriteElementString("vieworder",
colSurveyOptionInfo.ViewOrder.ToString());
                Writer.WriteEndElement();
            }
            // Retrieve the next SurveyOption
            Writer.WriteEndElement();
        }
        Writer.WriteEndElement();
    }
    Writer.WriteEndElement();
    Writer.Close();
}
```

DNN4 Module Developers Guide

```
        else
        {
            // There is nothing to export
            return String.Empty;
        }
        return strXML.ToString();
    }

    void DotNetNuke.Entities.Modules.IPortable.ImportModule(int
ModuleID, string Content, string Version, int UserID)
    {
        //Import the Surveys
        //Outer Loop - To insert the Surveys
        int intCurrentSurvey;
        XmlNode xmlSurveys =
DotNetNuke.Common.Globals.GetContent(Content, "surveys");
        foreach (XmlNode xmlSurvey in xmlSurveys)
        {
            SurveyInfo SurveyInfo = new SurveyInfo();
            SurveyInfo.ModuleId = ModuleID;
            SurveyInfo.Question = xmlSurvey["question"].InnerText;
            SurveyInfo.ViewOrder =
Convert.ToInt32(xmlSurvey["vieworder"].InnerText);
            SurveyInfo.CreatedByUser =
Convert.ToInt32(xmlSurvey["createdbyuser"].InnerText);
            SurveyInfo.CreatedDate =
Convert.ToDateTime(xmlSurvey["createddate"].InnerText);
            SurveyInfo.OptionType =
xmlSurvey["optiontype"].InnerText;

            // Add the Survey to the database
            intCurrentSurvey = AddSurvey(SurveyInfo);

            //Inner Loop - To insert the Survey Options
            XmlNode xmlSurveyOptions =
xmlSurvey.SelectSingleNode("surveyoptions");
            foreach (XmlNode xmlSurveyOption in xmlSurveyOptions)
            {
                SurveyOptionInfo SurveyOptionInfo = new
SurveyOptionInfo();
                SurveyOptionInfo.SurveyId = intCurrentSurvey;
                SurveyOptionInfo.OptionName =
xmlSurveyOption["optionname"].InnerText;
                SurveyOptionInfo.IsCorrect =
Convert.ToBoolean(xmlSurveyOption["isincorrect"].InnerText);
                SurveyOptionInfo.ViewOrder =
Convert.ToInt32(xmlSurveyOption["vieworder"].InnerText);

                // Add the Survey to the database

                SurveyOptionController.AddSurveyOption(SurveyOptionInfo);
            }
        }
    }
}
```

DNN4 Module Developers Guide

```
        }
    }
}

#endregion

#region ISearchable Members

public SearchItemInfoCollection
GetSearchItems(DotNetNuke.Entities.Modules.ModuleInfo ModInfo)
{
    // Get the Surveys for this Module instance
    List<SurveyInfo> colSurveys = GetSurveys(ModInfo.ModuleID);
    SearchItemInfoCollection SearchItemCollection = new
SearchItemInfoCollection();
    foreach (SurveyInfo SurveyInfo in colSurveys)
    {
        SearchItemInfo SearchItem;
        SearchItem = new SearchItemInfo
        (ModInfo.ModuleTitle + " - " + SurveyInfo.Question,
        SurveyInfo.Question,
        SurveyInfo.CreatedByUser,
        SurveyInfo.CreatedDate, ModInfo.ModuleID,
        Convert.ToString(SurveyInfo.SurveyId),
        SurveyInfo.Question);
        SearchItemCollection.Add(SearchItem);
    }
    return SearchItemCollection;
}

#endregion
}
```

DNN4 Module Developers Guide

SurveyController.vb

```
'
' DotNetNuke - http://www.dotnetnuke.com
' Copyright (c) 2002-2005
' by Perpetual Motion Interactive Systems Inc. (
http://www.perpetualmotion.ca )
'
' Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated
' documentation files (the "Software"), to deal in the Software without
restriction, including without limitation
' the rights to use, copy, modify, merge, publish, distribute,
sublicense, and/or sell copies of the Software, and
' to permit persons to whom the Software is furnished to do so, subject
to the following conditions:
'
' The above copyright notice and this permission notice shall be
included in all copies or substantial portions
' of the Software.
'
' THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED
' TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR
PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
' THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR
OTHER LIABILITY, WHETHER IN AN ACTION OF
' CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER
' DEALINGS IN THE SOFTWARE.
'

Imports System
Imports System.Xml
Imports System.Xml.XPath
Imports System.Data
Imports System.Collections.Generic
Imports DotNetNuke
Imports DotNetNuke.Services.Search
Imports DotNetNuke.Common.Utilities.XmlUtils

Namespace DotNetNuke.Modules.Survey

    Public Class SurveyController
        Implements Entities.Modules.ISearchable

        Implements Entities.Modules.IPortable
        Implements DotNetNuke.Entities.Modules.IUpgradeable

        Public Shared Function GetSurveys(ByVal ModuleId As Integer) As
List(Of SurveyInfo)
```


DNN4 Module Developers Guide

```
        Dim SurveyInfolist As List(Of SurveyInfo) = New List(Of
SurveyInfo)
        Using dr As IDataReader =
DataProvider.Instance().GetSurveys(ModuleId)
            While dr.Read
                Dim SurveyInfo As SurveyInfo = New SurveyInfo
                SurveyInfo.SurveyId =
Convert.ToInt32(dr("SurveyId"))
                SurveyInfo.Question =
Convert.ToString(dr("Question"))
                SurveyInfo.OptionType =
Convert.ToString(dr("OptionType"))
                SurveyInfo.ViewOrder =
Convert.ToInt32(ConvertNullInteger(dr("ViewOrder")))
                SurveyInfo.CreatedByUser =
Convert.ToInt32(dr("CreatedByUser"))
                SurveyInfo.CreatedDate =
Convert.ToDateTime(dr("CreatedDate"))
                SurveyInfolist.Add(SurveyInfo)
            End While
        End Using
        Return SurveyInfolist
    End Function

    Public Shared Function GetSurvey(ByVal SurveyID As Integer,
ByVal ModuleId As Integer) As SurveyInfo
        Dim SurveyInfo As SurveyInfo = New SurveyInfo
        Using dr As IDataReader =
DataProvider.Instance().GetSurvey(SurveyID, ModuleId)
            While dr.Read
                SurveyInfo.SurveyId =
Convert.ToInt32(dr("SurveyId"))
                SurveyInfo.ModuleId =
Convert.ToInt32(dr("ModuleID"))
                SurveyInfo.Question =
Convert.ToString(dr("Question"))
                SurveyInfo.OptionType =
Convert.ToString(dr("OptionType"))
                SurveyInfo.ViewOrder =
Convert.ToInt32(ConvertNullInteger(dr("ViewOrder")))
                SurveyInfo.Votes =
Convert.ToInt32(ConvertNullInteger(dr("Votes")))
                SurveyInfo.CreatedByUser =
Convert.ToInt32(dr("CreatedByUser"))
                SurveyInfo.CreatedDate =
Convert.ToDateTime(dr("CreatedDate"))
            End While
        End Using
        Return SurveyInfo
    End Function

    Public Shared Sub DeleteSurvey(ByVal objSurvey As SurveyInfo)
```

DNN4 Module Developers Guide

```
        DataProvider.Instance().DeleteSurvey(objSurvey.SurveyId,
objSurvey.ModuleId)
    End Sub
    Public Shared Function AddSurvey(ByVal objSurvey As SurveyInfo)
As Integer
        Return
CType(DataProvider.Instance().AddSurvey(objSurvey.ModuleId,
objSurvey.Question, objSurvey.ViewOrder, objSurvey.OptionType,
objSurvey.CreatedByUser), Integer)
    End Function
    Public Shared Sub UpdateSurvey(ByVal objSurvey As SurveyInfo)
        DataProvider.Instance().UpdateSurvey(objSurvey.SurveyId,
objSurvey.Question, objSurvey.ViewOrder, objSurvey.OptionType,
objSurvey.CreatedByUser, objSurvey.ModuleId)
    End Sub

#Region " Utility Functions "
    Public Shared Function ConvertNullInteger(ByVal Field As
Object) As Integer
        If Field Is DBNull.Value Then
            Return 0
        Else
            Return Convert.ToInt32(Field)
        End If
    End Function
#End Region

#Region " Optional Interfaces "
    Public Function UpgradeModule(ByVal Version As String) As
String Implements Entities.Modules.IUpgradeable.UpgradeModule
        Return "Custom upgrade code goes here for Version: " &
Version
    End Function
    Public Function ExportModule(ByVal ModuleID As Integer) As
String Implements DotNetNuke.Entities.Modules.IPortable.ExportModule
        Dim strXML As New StringBuilder()

        Dim settings As New XmlWriterSettings()
        settings.Indent = True
        settings.OmitXmlDeclaration = True

        Dim Writer As XmlWriter = XmlWriter.Create(strXML,
settings)
        '----
        'Outer Loop - To build the Surveys
        Dim colSurveys As List(Of SurveyInfo) =
GetSurveys(ModuleID)
        If colSurveys.Count > 0 Then
            Writer.WriteStartElement("surveys")
            Dim SurveyInfo As SurveyInfo
            For Each SurveyInfo In colSurveys
```

DNN4 Module Developers Guide

```

        Writer.WriteStartElement("survey")
        Writer.WriteElementString("question",
SurveyInfo.Question)
        Writer.WriteElementString("vieworder",
SurveyInfo.ViewOrder)
        Writer.WriteElementString("createdbyuser",
SurveyInfo.CreatedByUser)
        Writer.WriteElementString("createddate",
SurveyInfo.CreatedDate)
        Writer.WriteElementString("optiontype",
SurveyInfo.OptionType.ToString)

        'Inner Loop - To build the Options for each Survey
        Dim colSurveyOptions As List(Of SurveyOptionInfo) =
SurveyOptionController.GetSurveyOptions(SurveyInfo.SurveyId)
        If colSurveyOptions.Count > 0 Then
            Writer.WriteStartElement("surveyoptions")
            Dim SurveyOptionInfo As SurveyOptionInfo
            For Each SurveyOptionInfo In colSurveyOptions
                Writer.WriteStartElement("surveyoption")
                Writer.WriteElementString("optionname",
SurveyOptionInfo.OptionName)
                Writer.WriteElementString("isincorrect",
SurveyOptionInfo.IsCorrect)
                Writer.WriteElementString("vieworder",
SurveyOptionInfo.ViewOrder)
                Writer.WriteEndElement()
            Next ' Retrieve the next SurveyOption
            Writer.WriteEndElement()
        End If
        Writer.WriteEndElement()

    Next
    Writer.WriteEndElement()
    Writer.Close()
Else
    ' There is nothing to export
    Return String.Empty
End If
Return strXML.ToString()
End Function
```

```
Public Sub ImportModule(ByVal ModuleID As Integer, ByVal
Content As String, ByVal Version As String, ByVal UserID As Integer)
Implements DotNetNuke.Entities.Modules.IPortable.ImportModule
```

```
    'Import the Surveys
    'Outer Loop - To insert the Surveys
    Dim intCurrentSurvey As Integer
    Dim xmlSurvey As XmlNode
```

DNN4 Module Developers Guide

```
Dim xmlSurveys As XmlNode = GetContent(Content, "surveys")

For Each xmlSurvey In xmlSurveys

    Dim SurveyInfo As New SurveyInfo
    SurveyInfo.ModuleId = ModuleID
    SurveyInfo.Question =
xmlSurvey.Item("question").InnerText
    SurveyInfo.ViewOrder =
xmlSurvey.Item("vieworder").InnerText
    SurveyInfo.CreatedByUser =
xmlSurvey.Item("createdbyuser").InnerText
    SurveyInfo.CreatedDate =
xmlSurvey.Item("createddate").InnerText
    SurveyInfo.OptionType =
xmlSurvey.Item("optiontype").InnerText

    'Add the Survey to the database
    intCurrentSurvey = AddSurvey(SurveyInfo)

    'Inner Loop - To insert the Survey Options
    Dim xmlSurveyOption As XmlNode
    Dim xmlSurveyOptions As XmlNode =
xmlSurvey.SelectSingleNode("surveyoptions")

    For Each xmlSurveyOption In xmlSurveyOptions
        Dim SurveyOptionInfo As New SurveyOptionInfo
        SurveyOptionInfo.SurveyId = intCurrentSurvey
        SurveyOptionInfo.OptionName =
xmlSurveyOption.Item("optionname").InnerText
        SurveyOptionInfo.IsCorrect =
xmlSurveyOption.Item("isincorrect").InnerText
        SurveyOptionInfo.ViewOrder =
xmlSurveyOption.Item("vieworder").InnerText

        'Add the Survey to the database

        SurveyOptionController.AddSurveyOption(SurveyOptionInfo)
    Next

Next ' Retrieve the next Survey
End Sub

#End Region

Public Function GetSearchItems(ByVal ModInfo As
Entities.Modules.ModuleInfo) _
    As Services.Search.SearchItemInfoCollection Implements
Entities.Modules.ISearchable.GetSearchItems
    ' Get the Surveys for this Module instance
```

DNN4 Module Developers Guide

```
        Dim colSurveys As List(Of SurveyInfo) =  
GetSurveys(ModInfo.ModuleID)  
  
        Dim SearchItemCollection As New SearchItemInfoCollection  
        Dim SurveyInfo As SurveyInfo  
  
        For Each SurveyInfo In colSurveys  
            Dim SearchItem As SearchItemInfo  
            SearchItem = New SearchItemInfo _  
                (ModInfo.ModuleTitle & " - " & SurveyInfo.Question, _  
                SurveyInfo.Question, _  
                SurveyInfo.CreatedByUser, _  
                SurveyInfo.CreatedDate, ModInfo.ModuleID, _  
                SurveyInfo.SurveyId, _  
                SurveyInfo.Question)  
            SearchItemCollection.Add(SearchItem)  
        Next  
  
        Return SearchItemCollection  
    End Function  
End Class  
End Namespace
```

DNN4 Module Developers Guide

SurveyInfo.cs

```
//
// DotNetNuke - http://www.dotnetnuke.com
// Copyright (c) 2002-2005
// by Perpetual Motion Interactive Systems Inc. (
http://www.perpetualmotion.ca )
//
// Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated
// documentation files (the "Software"), to deal in the Software
without restriction, including without limitation
// the rights to use, copy, modify, merge, publish, distribute,
sublicense, and/or sell copies of the Software, and
// to permit persons to whom the Software is furnished to do so,
subject to the following conditions:
//
// The above copyright notice and this permission notice shall be
included in all copies or substantial portions
// of the Software.
//
// THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED
// TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR
PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
// THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR
OTHER LIABILITY, WHETHER IN AN ACTION OF
// CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER
// DEALINGS IN THE SOFTWARE.
//
namespace DotNetNuke.Modules.Survey
{
    public class SurveyInfo
    {
        private int _SurveyId;
        private int _ModuleId;
        private string _Question;
        private int _ViewOrder;
        private string _OptionType;
        private int _CreatedByUser;
        private System.DateTime _CreatedDate;
        private int _Votes;

        public SurveyInfo()
        {
        }

        public int SurveyId
        {
            get
```

DNN4 Module Developers Guide

```
        {
            return _SurveyId;
        }
        set
        {
            _SurveyId = value;
        }
    }

    public int ModuleId
    {
        get
        {
            return _ModuleId;
        }
        set
        {
            _ModuleId = value;
        }
    }

    public string Question
    {
        get
        {
            return _Question;
        }
        set
        {
            _Question = value;
        }
    }

    public int ViewOrder
    {
        get
        {
            return _ViewOrder;
        }
        set
        {
            _ViewOrder = value;
        }
    }

    public string OptionType
    {
        get
        {
            return _OptionType;
        }
    }
```

DNN4 Module Developers Guide

```
        set
        {
            _OptionType = value;
        }
    }

    public int CreatedByUser
    {
        get
        {
            return _CreatedByUser;
        }
        set
        {
            _CreatedByUser = value;
        }
    }

    public System.DateTime CreatedDate
    {
        get
        {
            return _CreatedDate;
        }
        set
        {
            _CreatedDate = value;
        }
    }

    public int Votes
    {
        get
        {
            return _Votes;
        }
        set
        {
            _Votes = value;
        }
    }
    }
}
```


DNN4 Module Developers Guide

SurveyInfo.vb

```
'  
' DotNetNuke - http://www.dotnetnuke.com  
' Copyright (c) 2002-2005  
' by Perpetual Motion Interactive Systems Inc. (  
http://www.perpetualmotion.ca )  
'  
' Permission is hereby granted, free of charge, to any person obtaining  
a copy of this software and associated  
' documentation files (the "Software"), to deal in the Software without  
restriction, including without limitation  
' the rights to use, copy, modify, merge, publish, distribute,  
sublicense, and/or sell copies of the Software, and  
' to permit persons to whom the Software is furnished to do so, subject  
to the following conditions:  
'  
' The above copyright notice and this permission notice shall be  
included in all copies or substantial portions  
' of the Software.  
'  
' THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,  
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED  
' TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR  
PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL  
' THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR  
OTHER LIABILITY, WHETHER IN AN ACTION OF  
' CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION  
WITH THE SOFTWARE OR THE USE OR OTHER  
' DEALINGS IN THE SOFTWARE.  
'
```

```
Imports System  
Imports System.Data  
Imports DotNetNuke
```

```
Namespace DotNetNuke.Modules.Survey
```

```
    Public Class SurveyInfo
```

```
        ' local property declarations  
        Private _SurveyId As Integer  
        Private _ModuleId As Integer  
        Private _Question As String  
        Private _ViewOrder As Integer  
        Private _OptionType As String  
        Private _CreatedByUser As Integer  
        Private _CreatedDate As Date  
        Private _Votes As Integer  
  
        ' initialization
```

DNN4 Module Developers Guide

```
Public Sub New()  
End Sub  
  
' public properties  
Public Property SurveyId() As Integer  
    Get  
        Return _SurveyId  
    End Get  
    Set(ByVal Value As Integer)  
        _SurveyId = Value  
    End Set  
End Property  
  
Public Property ModuleId() As Integer  
    Get  
        Return _ModuleId  
    End Get  
    Set(ByVal Value As Integer)  
        _ModuleId = Value  
    End Set  
End Property  
  
Public Property Question() As String  
    Get  
        Return _Question  
    End Get  
    Set(ByVal Value As String)  
        _Question = Value  
    End Set  
End Property  
  
Public Property ViewOrder() As Integer  
    Get  
        Return _ViewOrder  
    End Get  
    Set(ByVal Value As Integer)  
        _ViewOrder = Value  
    End Set  
End Property  
  
Public Property OptionType() As String  
    Get  
        Return _OptionType  
    End Get  
    Set(ByVal Value As String)  
        _OptionType = Value  
    End Set  
End Property  
  
Public Property CreatedByUser() As Integer  
    Get
```

DNN4 Module Developers Guide

```
        Return _CreatedByUser
    End Get
    Set(ByVal Value As Integer)
        _CreatedByUser = Value
    End Set
End Property

Public Property CreatedDate() As Date
    Get
        Return _CreatedDate
    End Get
    Set(ByVal Value As Date)
        _CreatedDate = Value
    End Set
End Property

Public Property Votes() As Integer
    Get
        Return _Votes
    End Get
    Set(ByVal Value As Integer)
        _Votes = Value
    End Set
End Property

End Class

End Namespace
```

SurveyOptionController.cs

```
//
// DotNetNuke - http://www.dotnetnuke.com
// Copyright (c) 2002-2005
// by Perpetual Motion Interactive Systems Inc. (
http://www.perpetualmotion.ca )
//
// Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated
// documentation files (the "Software"), to deal in the Software
without restriction, including without limitation
// the rights to use, copy, modify, merge, publish, distribute,
sublicense, and/or sell copies of the Software, and
// to permit persons to whom the Software is furnished to do so,
subject to the following conditions:
//
// The above copyright notice and this permission notice shall be
included in all copies or substantial portions
// of the Software.
//
// THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED
// TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR
PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
// THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR
OTHER LIABILITY, WHETHER IN AN ACTION OF
// CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER
// DEALINGS IN THE SOFTWARE.
//

using System;
using System.Collections.Generic;
using System.Data;
using DotNetNuke;

namespace DotNetNuke.Modules.Survey
{
    public class SurveyOptionController
    {
        static public List<SurveyOptionInfo> GetSurveyOptions(int
SurveyId)
        {
            List<SurveyOptionInfo> SurveyOptionInfolist = new
List<SurveyOptionInfo>();
            using (IDataReader dr =
DataProvider.Instance().GetSurveyOptions(SurveyId))
            {
                while (dr.Read())
                {
```

DNN4 Module Developers Guide

```
        SurveyOptionInfo colSurveyOptionInfo = new
SurveyOptionInfo();
        colSurveyOptionInfo.SurveyOptionId =
Convert.ToInt32(dr["SurveyOptionID"]);
        colSurveyOptionInfo.OptionName =
Convert.ToString(dr["OptionName"]);
        colSurveyOptionInfo.IsCorrect =
Convert.ToBoolean(dr["IsCorrect"]);
        colSurveyOptionInfo.Votes =
Convert.ToInt32(SurveyController.ConvertNullInteger(dr["Votes"]));
        colSurveyOptionInfo.ViewOrder =
Convert.ToInt32(SurveyController.ConvertNullInteger(dr["ViewOrder"]));
        SurveyOptionInfolist.Add(colSurveyOptionInfo);
    }
}
return SurveyOptionInfolist;
}
public static void DeleteSurveyOption(SurveyOptionInfo
objSurveyOption)
{
    DataProvider.Instance().DeleteSurveyOption(objSurveyOption.SurveyOption
Id);
}
public static int AddSurveyOption(SurveyOptionInfo
objSurveyOption)
{
    return
((int)(DataProvider.Instance().AddSurveyOption(objSurveyOption.SurveyId
, objSurveyOption.OptionName, objSurveyOption.ViewOrder,
objSurveyOption.IsCorrect)));
}

public static void UpdateSurveyOption(SurveyOptionInfo
objSurveyOption)
{
    DataProvider.Instance().UpdateSurveyOption(objSurveyOption.SurveyOption
Id, objSurveyOption.OptionName, objSurveyOption.ViewOrder,
objSurveyOption.IsCorrect);
}

public static void AddSurveyResult(SurveyOptionInfo
objSurveyOption, int UserID)
{
    DataProvider.Instance().AddSurveyResult(objSurveyOption.SurveyOptionId,
UserID);
}
}
```


DNN4 Module Developers Guide

SurveyOptionController.vb

```
'
' DotNetNuke - http://www.dotnetnuke.com
' Copyright (c) 2002-2005
' by Perpetual Motion Interactive Systems Inc. (
http://www.perpetualmotion.ca )
'
' Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated
' documentation files (the "Software"), to deal in the Software without
restriction, including without limitation
' the rights to use, copy, modify, merge, publish, distribute,
sublicense, and/or sell copies of the Software, and
' to permit persons to whom the Software is furnished to do so, subject
to the following conditions:
'
' The above copyright notice and this permission notice shall be
included in all copies or substantial portions
' of the Software.
'
' THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED
' TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR
PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
' THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR
OTHER LIABILITY, WHETHER IN AN ACTION OF
' CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER
' DEALINGS IN THE SOFTWARE.
'

Imports System
Imports System.Collections.Generic
Imports System.Data
Imports DotNetNuke

Namespace DotNetNuke.Modules.Survey

    Public Class SurveyOptionController
        Public Shared Function GetSurveyOptions(ByVal SurveyId As
Integer) As List(Of SurveyOptionInfo)
            Dim SurveyOptionInfolist As List(Of SurveyOptionInfo) = New
List(Of SurveyOptionInfo)
            Using dr As IDataReader =
DataProvider.Instance().GetSurveyOptions(SurveyId)
                While dr.Read
                    Dim SurveyOptionInfo As SurveyOptionInfo = New
SurveyOptionInfo
                    SurveyOptionInfo.SurveyOptionId =
Convert.ToInt32(dr("SurveyOptionID"))
```

DNN4 Module Developers Guide

```
        SurveyOptionInfo.OptionName =  
Convert.ToString(dr("OptionName"))  
        SurveyOptionInfo.IsCorrect =  
Convert.ToString(dr("IsCorrect"))  
        SurveyOptionInfo.Votes =  
Convert.ToInt32(SurveyController.ConvertNullInteger(dr("Votes")))  
        SurveyOptionInfo.ViewOrder =  
Convert.ToInt32(SurveyController.ConvertNullInteger(dr("ViewOrder")))  
        SurveyOptionInfolist.Add(SurveyOptionInfo)  
    End While  
End Using  
Return SurveyOptionInfolist  
End Function  
Public Shared Sub DeleteSurveyOption(ByVal objSurveyOption As  
SurveyOptionInfo)  
  
DataProvider.Instance().DeleteSurveyOption(objSurveyOption.SurveyOption  
Id)  
End Sub  
Public Shared Function AddSurveyOption(ByVal objSurveyOption As  
SurveyOptionInfo) As Integer  
Return  
CType(DataProvider.Instance().AddSurveyOption(objSurveyOption.SurveyId,  
objSurveyOption.OptionName, objSurveyOption.ViewOrder,  
objSurveyOption.IsCorrect), Integer)  
End Function  
Public Shared Sub UpdateSurveyOption(ByVal objSurveyOption As  
SurveyOptionInfo)  
  
DataProvider.Instance().UpdateSurveyOption(objSurveyOption.SurveyOption  
Id, objSurveyOption.OptionName, objSurveyOption.ViewOrder,  
objSurveyOption.IsCorrect)  
End Sub  
Public Shared Sub AddSurveyResult(ByVal objSurveyOption As  
SurveyOptionInfo, ByVal UserID As Integer)  
  
DataProvider.Instance().AddSurveyResult(objSurveyOption.SurveyOptionId,  
UserID)  
End Sub  
  
End Class  
  
End Namespace
```


DNN4 Module Developers Guide

SurveyOptionInfo.cs

```
//
// DotNetNuke - http://www.dotnetnuke.com
// Copyright (c) 2002-2005
// by Perpetual Motion Interactive Systems Inc. (
http://www.perpetualmotion.ca )
//
// Permission is hereby granted, free of charge, to any person
// obtaining a copy of this software and associated
// documentation files (the "Software"), to deal in the Software
// without restriction, including without limitation
// the rights to use, copy, modify, merge, publish, distribute,
// sublicense, and/or sell copies of the Software, and
// to permit persons to whom the Software is furnished to do so,
// subject to the following conditions:
//
// The above copyright notice and this permission notice shall be
// included in all copies or substantial portions
// of the Software.
//
// THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
// EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED
// TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR
// PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
// THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR
// OTHER LIABILITY, WHETHER IN AN ACTION OF
// CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
// WITH THE SOFTWARE OR THE USE OR OTHER
// DEALINGS IN THE SOFTWARE.
//

namespace DotNetNuke.Modules.Survey
{
    public class SurveyOptionInfo
    {
        private int _SurveyOptionId;
        private int _SurveyId;
        private int _ViewOrder;
        private string _OptionName;
        private int _Votes;
        private bool _IsCorrect;

        public SurveyOptionInfo()
        {
        }

        public int SurveyOptionId
        {
            get
            {

```

DNN4 Module Developers Guide

```
        return _SurveyOptionId;
    }
    set
    {
        _SurveyOptionId = value;
    }
}

public int SurveyId
{
    get
    {
        return _SurveyId;
    }
    set
    {
        _SurveyId = value;
    }
}

public int ViewOrder
{
    get
    {
        return _ViewOrder;
    }
    set
    {
        _ViewOrder = value;
    }
}

public string OptionName
{
    get
    {
        return _OptionName;
    }
    set
    {
        _OptionName = value;
    }
}

public int Votes
{
    get
    {
        return _Votes;
    }
    set
```

DNN4 Module Developers Guide

```
        {
            _Votes = value;
        }
    }

    public bool IsCorrect
    {
        get
        {
            return _IsCorrect;
        }
        set
        {
            _IsCorrect = value;
        }
    }
}
```

DNN4 Module Developers Guide

SurveyOptionInfo.vb

```
'
' DotNetNuke - http://www.dotnetnuke.com
' Copyright (c) 2002-2005
' by Perpetual Motion Interactive Systems Inc. (
http://www.perpetualmotion.ca )
'
' Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated
' documentation files (the "Software"), to deal in the Software without
restriction, including without limitation
' the rights to use, copy, modify, merge, publish, distribute,
sublicense, and/or sell copies of the Software, and
' to permit persons to whom the Software is furnished to do so, subject
to the following conditions:
'
' The above copyright notice and this permission notice shall be
included in all copies or substantial portions
' of the Software.
'
' THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED
' TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR
PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
' THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR
OTHER LIABILITY, WHETHER IN AN ACTION OF
' CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER
' DEALINGS IN THE SOFTWARE.
'
```

```
Imports System
Imports System.Data
Imports DotNetNuke
```

```
Namespace DotNetNuke.Modules.Survey
```

```
    Public Class SurveyOptionInfo
```

```
        ' local property declarations
        Private _SurveyOptionId As Integer
        Private _SurveyId As Integer
        Private _ViewOrder As Integer
        Private _OptionName As String
        Private _Votes As Integer
        Private _IsCorrect As Boolean
```

```
        ' initialization
        Public Sub New()
        End Sub
```

DNN4 Module Developers Guide

```
' public properties
Public Property SurveyOptionId() As Integer
    Get
        Return _SurveyOptionId
    End Get
    Set(ByVal Value As Integer)
        _SurveyOptionId = Value
    End Set
End Property

Public Property SurveyId() As Integer
    Get
        Return _SurveyId
    End Get
    Set(ByVal Value As Integer)
        _SurveyId = Value
    End Set
End Property

Public Property ViewOrder() As Integer
    Get
        Return _ViewOrder
    End Get
    Set(ByVal Value As Integer)
        _ViewOrder = Value
    End Set
End Property

Public Property OptionName() As String
    Get
        Return _OptionName
    End Get
    Set(ByVal Value As String)
        _OptionName = Value
    End Set
End Property

Public Property Votes() As Integer
    Get
        Return _Votes
    End Get
    Set(ByVal Value As Integer)
        _Votes = Value
    End Set
End Property

Public Property IsCorrect() As Boolean
    Get
        Return _IsCorrect
    End Get
```

DNN4 Module Developers Guide

```
        Set(ByVal Value As Boolean)
            _IsCorrect = Value
        End Set
    End Property

End Class

End Namespace
```

DNN4 Module Developers Guide

SqlDataProvider.cs

```
//
// DotNetNuke - http://www.dotnetnuke.com
// Copyright (c) 2002-2005
// by Perpetual Motion Interactive Systems Inc. (
http://www.perpetualmotion.ca )
//
// Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated
// documentation files (the "Software"), to deal in the Software
without restriction, including without limitation
// the rights to use, copy, modify, merge, publish, distribute,
sublicense, and/or sell copies of the Software, and
// to permit persons to whom the Software is furnished to do so,
subject to the following conditions:
//
// The above copyright notice and this permission notice shall be
included in all copies or substantial portions
// of the Software.
//
// THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED
// TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR
PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
// THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR
OTHER LIABILITY, WHETHER IN AN ACTION OF
// CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER
// DEALINGS IN THE SOFTWARE.
//
using System;
using System.Data;
using System.Data.SqlClient;
using Microsoft.ApplicationBlocks.Data;
using DotNetNuke;
using DotNetNuke.Common.Utilities;
namespace DotNetNuke.Modules.Survey
{
    public SqlDataProvider()
    {
        Framework.Providers.Provider objProvider =
        ((Framework.Providers.Provider)(_providerConfiguration.Providers[_provi
derConfiguration.DefaultProvider]));

        // This code handles getting the connection string from
either the connectionString / appsetting section and uses the
connectionstring section by default if it exists.
        // Get Connection string from web.config
        _connectionString = Config.GetConnectionString();
    }
}
```

DNN4 Module Developers Guide

```
// If above funtion does not return anything then
connectionString must be set in the dataprovider section.
    if ((_connectionString == ""))
    {
        // Use connection string specified in provider
        _connectionString =
objProvider.Attributes["connectionString"];
    }

    _providerPath = objProvider.Attributes["providerPath"];
    _objectQualifier =
objProvider.Attributes["objectQualifier"];
    if (_objectQualifier != "" & _objectQualifier.EndsWith("_")
== false)
    {
        _objectQualifier += "_";
    }
    _databaseOwner = objProvider.Attributes["databaseOwner"];
    if (_databaseOwner != "" & _databaseOwner.EndsWith(".") ==
false)
    {
        _databaseOwner += ".";
    }
}

public string ConnectionString
{
    get
    {
        return _connectionString;
    }
}

public string ProviderPath
{
    get
    {
        return _providerPath;
    }
}

public string ObjectQualifier
{
    get
    {
        return _objectQualifier;
    }
}

public string DatabaseOwner
{
    get
    {
        return _databaseOwner;
    }
}
```


DNN4 Module Developers Guide

```
        get
        {
            return _databaseOwner;
        }
    }

    private object GetNull(object Field)
    {
        return DotNetNuke.Common.Utilities.Null.GetNull(Field,
DBNull.Value);
    }

    public override IDataReader GetSurveys(int ModuleId)
    {
        return
        ((IDataReader)(SqlHelper.ExecuteReader(ConnectionString, DatabaseOwner
+ ObjectQualifier + "GetSurveys", ModuleId)));
    }

    public override IDataReader GetSurvey(int SurveyID, int
ModuleId)
    {
        return
        ((IDataReader)(SqlHelper.ExecuteReader(ConnectionString, DatabaseOwner
+ ObjectQualifier + "GetSurvey", SurveyID, ModuleId)));
    }

    public override int AddSurvey(int ModuleId, string Question,
int ViewOrder, string OptionType, int UserId)
    {
        return
        (Convert.ToInt32((SqlHelper.ExecuteScalar(ConnectionString,
DatabaseOwner + ObjectQualifier + "AddSurvey", ModuleId, Question,
GetNull(ViewOrder), OptionType, UserId))));
    }

    public override void UpdateSurvey(int SurveyId, string
Question, int ViewOrder, string OptionType, int UserId, int ModuleId)
    {
        SqlHelper.ExecuteNonQuery(ConnectionString, DatabaseOwner +
ObjectQualifier + "UpdateSurvey", SurveyId, Question,
GetNull(ViewOrder), OptionType, UserId, ModuleId);
    }

    public override void DeleteSurvey(int SurveyID, int ModuleId)
    {
        SqlHelper.ExecuteNonQuery(ConnectionString, DatabaseOwner +
ObjectQualifier + "DeleteSurvey", SurveyID, ModuleId);
    }

    public override IDataReader GetSurveyOptions(int SurveyId)
```

DNN4 Module Developers Guide

```
        {
            return
            ((IDataReader)(SqlHelper.ExecuteReader(ConnectionString, DatabaseOwner
            + ObjectQualifier + "GetSurveyOptions", SurveyId)));
        }

        public override int AddSurveyOption(int SurveyId, string
        OptionName, int ViewOrder, bool IsCorrect)
        {
            return
            (Convert.ToInt32((SqlHelper.ExecuteScalar(ConnectionString,
            DatabaseOwner + ObjectQualifier + "AddSurveyOption", SurveyId,
            OptionName, GetNull(ViewOrder), IsCorrect))));
        }

        public override void UpdateSurveyOption(int SurveyOptionId,
        string OptionName, int ViewOrder, bool IsCorrect)
        {
            SqlHelper.ExecuteNonQuery(ConnectionString, DatabaseOwner +
            ObjectQualifier + "UpdateSurveyOption", SurveyOptionId, OptionName,
            GetNull(ViewOrder), IsCorrect);
        }

        public override void DeleteSurveyOption(int SurveyOptionID)
        {
            SqlHelper.ExecuteNonQuery(ConnectionString, DatabaseOwner +
            ObjectQualifier + "DeleteSurveyOption", SurveyOptionID);
        }

        public override void AddSurveyResult(int SurveyOptionId, int
        UserId)
        {
            SqlHelper.ExecuteNonQuery(ConnectionString, DatabaseOwner +
            ObjectQualifier + "AddSurveyResult", SurveyOptionId, UserId);
        }
    }
}
```

DNN4 Module Developers Guide

SqlDataProvider.vb

```
'
' DotNetNuke - http://www.dotnetnuke.com
' Copyright (c) 2002-2005
' by Perpetual Motion Interactive Systems Inc. (
http://www.perpetualmotion.ca )
'
' Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated
' documentation files (the "Software"), to deal in the Software without
restriction, including without limitation
' the rights to use, copy, modify, merge, publish, distribute,
sublicense, and/or sell copies of the Software, and
' to permit persons to whom the Software is furnished to do so, subject
to the following conditions:
'
' The above copyright notice and this permission notice shall be
included in all copies or substantial portions
' of the Software.
'
' THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED
' TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR
PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
' THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR
OTHER LIABILITY, WHETHER IN AN ACTION OF
' CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER
' DEALINGS IN THE SOFTWARE.
'

Imports System
Imports System.Data
Imports System.Data.SqlClient
Imports Microsoft.ApplicationBlocks.Data
Imports DotNetNuke

Namespace DotNetNuke.Modules.Survey

    Public Class SqlDataProvider

        Inherits DataProvider

    #Region "Private Members"

        Private Const ProviderType As String = "data"

        Private _providerConfiguration As
Framework.Providers.ProviderConfiguration =
```

DNN4 Module Developers Guide

```
Framework.Providers.ProviderConfiguration.GetProviderConfiguration(ProviderType)
    Private _connectionString As String
    Private _providerPath As String
    Private _objectQualifier As String
    Private _databaseOwner As String

#End Region

#Region "Constructors"

    Public Sub New()

        ' Read the configuration specific information for this
provider
        Dim objProvider As Framework.Providers.Provider =
CType(_providerConfiguration.Providers(_providerConfiguration.DefaultProvider), Framework.Providers.Provider)

        ' This code handles getting the connection string from
either the connectionString / appsetting section and uses the
connectionstring section by default if it exists.
        ' Get Connection string from web.config
        _connectionString = Config.GetConnectionString()

        ' If above function does not return anything then
connectionString must be set in the dataprovider section.
        If _connectionString = "" Then
            ' Use connection string specified in provider
            _connectionString =
objProvider.Attributes("connectionString")
        End If

        _providerPath = objProvider.Attributes("providerPath")

        _objectQualifier =
objProvider.Attributes("objectQualifier")
        If _objectQualifier <> "" And
_objectQualifier.EndsWith("_") = False Then
            _objectQualifier += "_"
        End If

        _databaseOwner = objProvider.Attributes("databaseOwner")
        If _databaseOwner <> "" And _databaseOwner.EndsWith(".") =
False Then
            _databaseOwner += "."
        End If

    End Sub
```

DNN4 Module Developers Guide

```
#End Region

#Region "Properties"

    Public ReadOnly Property ConnectionString() As String
        Get
            Return _connectionString
        End Get
    End Property

    Public ReadOnly Property ProviderPath() As String
        Get
            Return _providerPath
        End Get
    End Property

    Public ReadOnly Property ObjectQualifier() As String
        Get
            Return _objectQualifier
        End Get
    End Property

    Public ReadOnly Property DatabaseOwner() As String
        Get
            Return _databaseOwner
        End Get
    End Property

#End Region

    ' general
    Private Function GetNull(ByVal Field As Object) As Object
        Return DotNetNuke.Common.Utilities.Null.GetNull(Field,
DBNull.Value)
    End Function

    Public Overrides Function GetSurveys(ByVal ModuleId As Integer)
As IDataReader
        Return CType(SqlHelper.ExecuteReader(ConnectionString,
DatabaseOwner & ObjectQualifier & "GetSurveys", ModuleId), IDataReader)
    End Function

    Public Overrides Function GetSurvey(ByVal SurveyID As Integer,
ByVal ModuleId As Integer) As IDataReader
        Return CType(SqlHelper.ExecuteReader(ConnectionString,
DatabaseOwner & ObjectQualifier & "GetSurvey", SurveyID, ModuleId),
IDataReader)
    End Function

    Public Overrides Function AddSurvey(ByVal ModuleId As Integer,
ByVal Question As String, ByVal ViewOrder As Integer, ByVal OptionType
As String, ByVal UserId As Integer) As Integer
```

DNN4 Module Developers Guide

```
        Return CType(SqlHelper.ExecuteScalar(ConnectionString,
DatabaseOwner & ObjectQualifier & "AddSurvey", ModuleId, Question,
GetNull(ViewOrder), OptionType, UserId), Integer)
    End Function
    Public Overrides Sub UpdateSurvey(ByVal SurveyId As Integer,
ByVal Question As String, ByVal ViewOrder As Integer, ByVal OptionType
As String, ByVal UserId As Integer, ByVal ModuleId As Integer)
        SqlHelper.ExecuteNonQuery(ConnectionString, DatabaseOwner &
ObjectQualifier & "UpdateSurvey", SurveyId, Question,
GetNull(ViewOrder), OptionType, UserId, ModuleId)
    End Sub
    Public Overrides Sub DeleteSurvey(ByVal SurveyID As Integer,
ByVal ModuleId As Integer)
        SqlHelper.ExecuteNonQuery(ConnectionString, DatabaseOwner &
ObjectQualifier & "DeleteSurvey", SurveyID, ModuleId)
    End Sub
    Public Overrides Function GetSurveyOptions(ByVal SurveyId As
Integer) As IDataReader
        Return CType(SqlHelper.ExecuteReader(ConnectionString,
DatabaseOwner & ObjectQualifier & "GetSurveyOptions", SurveyId),
IDataReader)
    End Function
    Public Overrides Function AddSurveyOption(ByVal SurveyId As
Integer, ByVal OptionName As String, ByVal ViewOrder As Integer, ByVal
IsCorrect As Boolean) As Integer
        Return CType(SqlHelper.ExecuteScalar(ConnectionString,
DatabaseOwner & ObjectQualifier & "AddSurveyOption", SurveyId,
OptionName, GetNull(ViewOrder), IsCorrect), Integer)
    End Function
    Public Overrides Sub UpdateSurveyOption(ByVal SurveyOptionId As
Integer, ByVal OptionName As String, ByVal ViewOrder As Integer, ByVal
IsCorrect As Boolean)
        SqlHelper.ExecuteNonQuery(ConnectionString, DatabaseOwner &
ObjectQualifier & "UpdateSurveyOption", SurveyOptionId, OptionName,
GetNull(ViewOrder), IsCorrect)
    End Sub
    Public Overrides Sub DeleteSurveyOption(ByVal SurveyOptionID As
Integer)
        SqlHelper.ExecuteNonQuery(ConnectionString, DatabaseOwner &
ObjectQualifier & "DeleteSurveyOption", SurveyOptionID)
    End Sub
    Public Overrides Sub AddSurveyResult(ByVal SurveyOptionId As
Integer, ByVal UserId As Integer)
        SqlHelper.ExecuteNonQuery(ConnectionString, DatabaseOwner &
ObjectQualifier & "AddSurveyResult", SurveyOptionId, UserId)
    End Sub

End Class

End Namespace
```


04.00.10.txt

```
bin\DotNetNuke.Modules.Survey.dll  
bin\DotNetNuke.Modules.Survey.SqlDataProvider.dll
```


DNN_Survey.dnn [C#]

```
<dotnetnuke version="3.0" type="Module">
  <folders>
    <folder>
      <name>DNN_Survey</name>
      <friendlyname>Survey</friendlyname>
      <foldername>Survey</foldername>
      <modulename>DNN_Survey</modulename>
      <description>Survey allows you to create custom surveys to obtain
public feedback</description>
      <version>04.00.10</version>

    <businesscontrollerclass>DotNetNuke.Modules.Survey.SurveyController</bu
sinesscontrollerclass>
    <modules>
      <module>
        <friendlyname>Survey</friendlyname>
        <cachetime>0</cachetime>
        <controls>
          <control>
            <src>DesktopModules/Survey/Survey.ascx</src>
            <type>View</type>

            <helpurl>http://www.dotnetnuke.com/default.aspx?tabid=787</helpurl>
          </control>
          <control>
            <key>Edit</key>
            <title>Create Survey</title>
            <src>DesktopModules/Survey/EditSurvey.ascx</src>
            <iconfile>icon_survey_32px.gif</iconfile>
            <type>Edit</type>

            <helpurl>http://www.dotnetnuke.com/default.aspx?tabid=787</helpurl>
          </control>
          <control>
            <key>Settings</key>
            <title>Survey Settings</title>
            <src>DesktopModules/Survey/Settings.ascx</src>
            <iconfile>icon_survey_32px.gif</iconfile>
            <type>Edit</type>

            <helpurl>http://www.dotnetnuke.com/default.aspx?tabid=787</helpurl>
          </control>
        </controls>
      </module>
    </modules>
    <files>
      <file>
        <path>App_LocalResources</path>
        <name>EditSurvey.ascx.resx</name>
      </file>
    </files>
  </folders>
</dotnetnuke>
```

DNN4 Module Developers Guide

```
</file>
<file>
  <path>App_LocalResources</path>
  <name>Settings.ascx.resx</name>
</file>
<file>
  <path>App_LocalResources</path>
  <name>Survey.ascx.resx</name>
</file>
<file>
  <path>Providers\DataProviders\SqlDataProvider</path>
  <name>03.01.00.SqlDataProvider</name>
</file>
<file>
  <path>Providers\DataProviders\SqlDataProvider</path>
  <name>03.03.00.SqlDataProvider</name>
</file>
<file>
  <path>Providers\DataProviders\SqlDataProvider</path>
  <name>04.00.10.SqlDataProvider</name>
</file>
<file>
  <name>04.00.10.txt</name>
</file>
<file>
  <name>DNN_Survey.dnn.config</name>
</file>
<file>
  <name>EditSurvey.ascx</name>
</file>
<file>
  <name>EditSurvey.ascx.cs</name>
</file>
<file>
  <name>icon_survey_32px.gif</name>
</file>
<file>
  <name>module.css</name>
</file>
<file>
  <name>red.gif</name>
</file>
<file>
  <name>Settings.ascx</name>
</file>
<file>
  <name>Settings.ascx.cs</name>
</file>
<file>
  <name>Survey.ascx</name>
</file>
```

DNN4 Module Developers Guide

```
<file>
  <name>Survey.ascx.cs</name>
</file>
<file>
  <name>Uninstall.SqlDataProvider</name>
</file>
<file>
  <path>[app_code]Providers</path>
  <name>SqlDataProvider.cs</name>
</file>
<file>
  <path>[app_code]</path>
  <name>DataProvider.cs</name>
</file>
<file>
  <path>[app_code]</path>
  <name>SurveyController.cs</name>
</file>
<file>
  <path>[app_code]</path>
  <name>SurveyInfo.cs</name>
</file>
<file>
  <path>[app_code]</path>
  <name>SurveyOptionController.cs</name>
</file>
<file>
  <path>[app_code]</path>
  <name>SurveyOptionInfo.cs</name>
</file>
</files>
</folder>
</folders>
</dotnetnuke>
```

DNN_Survey.dnn [VB]

```
<dotnetnuke version="3.0" type="Module">
  <folders>
    <folder>
      <name>DNN_Survey</name>
      <friendlyname>Survey</friendlyname>
      <foldername>Survey</foldername>
      <modulename>DNN_Survey</modulename>
      <description>Survey allows you to create custom surveys to obtain
public feedback</description>
      <version>04.00.10</version>

    <businesscontrollerclass>DotNetNuke.Modules.Survey.SurveyController</bu
sinesscontrollerclass>
      <modules>
        <module>
          <friendlyname>Survey</friendlyname>
          <cachetime>0</cachetime>
          <controls>
            <control>
              <src>DesktopModules/Survey/Survey.ascx</src>
              <type>View</type>

            <helpurl>http://www.dotnetnuke.com/default.aspx?tabid=787</helpurl>
            </control>
            <control>
              <key>Edit</key>
              <title>Create Survey</title>
              <src>DesktopModules/Survey/EditSurvey.ascx</src>
              <iconfile>icon_survey_32px.gif</iconfile>
              <type>Edit</type>

            <helpurl>http://www.dotnetnuke.com/default.aspx?tabid=787</helpurl>
            </control>
            <control>
              <key>Settings</key>
              <title>Survey Settings</title>
              <src>DesktopModules/Survey/Settings.ascx</src>
              <iconfile>icon_survey_32px.gif</iconfile>
              <type>Edit</type>

            <helpurl>http://www.dotnetnuke.com/default.aspx?tabid=787</helpurl>
            </control>
          </controls>
        </module>
      </modules>
      <files>
        <file>
          <path>App_LocalResources</path>
          <name>EditSurvey.ascx.resx</name>
        </file>
      </files>
    </folder>
  </folders>
</dotnetnuke>
```

DNN4 Module Developers Guide

```
</file>
<file>
  <path>App_LocalResources</path>
  <name>Settings.ascx.resx</name>
</file>
<file>
  <path>App_LocalResources</path>
  <name>Survey.ascx.resx</name>
</file>
<file>
  <path>Providers\DataProviders\SqlDataProvider</path>
  <name>03.01.00.SqlDataProvider</name>
</file>
<file>
  <path>Providers\DataProviders\SqlDataProvider</path>
  <name>03.03.00.SqlDataProvider</name>
</file>
<file>
  <path>Providers\DataProviders\SqlDataProvider</path>
  <name>04.00.10.SqlDataProvider</name>
</file>
<file>
  <name>04.00.10.txt</name>
</file>
<file>
  <name>DNN_Survey.dnn.config</name>
</file>
<file>
  <name>EditSurvey.ascx</name>
</file>
<file>
  <name>EditSurvey.ascx.vb</name>
</file>
<file>
  <name>icon_survey_32px.gif</name>
</file>
<file>
  <name>module.css</name>
</file>
<file>
  <name>red.gif</name>
</file>
<file>
  <name>Settings.ascx</name>
</file>
<file>
  <name>Settings.ascx.vb</name>
</file>
<file>
  <name>Survey.ascx</name>
</file>
```

DNN4 Module Developers Guide

```
<file>
  <name>Survey.ascx.vb</name>
</file>
<file>
  <name>Uninstall.SqlDataProvider</name>
</file>
<file>

<path>[app_code]Providers\DataProviders\SqlDataProvider</path>
  <name>SqlDataProvider.vb</name>
</file>
<file>
  <path>[app_code]</path>
  <name>DataProvider.vb</name>
</file>
<file>
  <path>[app_code]</path>
  <name>SurveyController.vb</name>
</file>
<file>
  <path>[app_code]</path>
  <name>SurveyInfo.vb</name>
</file>
<file>
  <path>[app_code]</path>
  <name>SurveyOptionController.vb</name>
</file>
<file>
  <path>[app_code]</path>
  <name>SurveyOptionInfo.vb</name>
</file>
</files>
</folder>
</folders>
</dotnetnuke>
```

DNN_Survey.dnn.config [C#]

```
<dotnetnuke version="3.0" type="Module">
  <folders>
    <folder>
      <name>DNN_Survey</name>
      <friendlyname>Survey</friendlyname>
      <foldername>Survey</foldername>
      <modulename>DNN_Survey</modulename>
      <description>Survey allows you to create custom surveys to obtain
public feedback</description>
      <version>04.00.10</version>

    <businesscontrollerclass>DotNetNuke.Modules.Survey.SurveyController</bu
sinesscontrollerclass>
    <modules>
      <module>
        <friendlyname>Survey</friendlyname>
        <cachetime>0</cachetime>
        <controls>
          <control>
            <src>DesktopModules/Survey/Survey.ascx</src>
            <type>View</type>

            <helpurl>http://www.dotnetnuke.com/default.aspx?tabid=787</helpurl>
          </control>
          <control>
            <key>Edit</key>
            <title>Create Survey</title>
            <src>DesktopModules/Survey/EditSurvey.ascx</src>
            <iconfile>icon_survey_32px.gif</iconfile>
            <type>Edit</type>

            <helpurl>http://www.dotnetnuke.com/default.aspx?tabid=787</helpurl>
          </control>
          <control>
            <key>Settings</key>
            <title>Survey Settings</title>
            <src>DesktopModules/Survey/Settings.ascx</src>
            <iconfile>icon_survey_32px.gif</iconfile>
            <type>Edit</type>

            <helpurl>http://www.dotnetnuke.com/default.aspx?tabid=787</helpurl>
          </control>
        </controls>
      </module>
    </modules>
    <files>
      <file>
        <path>App_LocalResources</path>
        <name>EditSurvey.ascx.resx</name>
      </file>
    </files>
  </folders>
</dotnetnuke>
```

DNN4 Module Developers Guide

```
</file>
<file>
  <path>App_LocalResources</path>
  <name>Settings.ascx.resx</name>
</file>
<file>
  <path>App_LocalResources</path>
  <name>Survey.ascx.resx</name>
</file>
<file>
  <path>Providers\DataProviders\SqlDataProvider</path>
  <name>03.01.00.SqlDataProvider</name>
</file>
<file>
  <path>Providers\DataProviders\SqlDataProvider</path>
  <name>03.03.00.SqlDataProvider</name>
</file>
<file>
  <path>Providers\DataProviders\SqlDataProvider</path>
  <name>04.00.10.SqlDataProvider</name>
</file>
<file>
  <name>04.00.10.txt</name>
</file>
<file>
  <name>DNN_Survey.dnn.config</name>
</file>
<file>
  <name>EditSurvey.ascx</name>
</file>
<file>
  <name>EditSurvey.ascx.cs</name>
</file>
<file>
  <name>icon_survey_32px.gif</name>
</file>
<file>
  <name>module.css</name>
</file>
<file>
  <name>red.gif</name>
</file>
<file>
  <name>Settings.ascx</name>
</file>
<file>
  <name>Settings.ascx.cs</name>
</file>
<file>
  <name>Survey.ascx</name>
</file>
```


DNN4 Module Developers Guide

```
<file>
  <name>Survey.ascx.cs</name>
</file>
<file>
  <name>Uninstall.SqlDataProvider</name>
</file>
<file>

<path>[app_code]Providers\DataProviders\SqlDataProvider</path>
  <name>SqlDataProvider.cs</name>
</file>
<file>
  <path>[app_code]</path>
  <name>DataProvider.cs</name>
</file>
<file>
  <path>[app_code]</path>
  <name>SurveyController.cs</name>
</file>
<file>
  <path>[app_code]</path>
  <name>SurveyInfo.cs</name>
</file>
<file>
  <path>[app_code]</path>
  <name>SurveyOptionController.cs</name>
</file>
<file>
  <path>[app_code]</path>
  <name>SurveyOptionInfo.cs</name>
</file>
</files>
</folder>
</folders>
</dotnetnuke>
```

DNN_Survey.dnn.config [VB]

```
<dotnetnuke version="3.0" type="Module">
  <folders>
    <folder>
      <name>DNN_Survey</name>
      <friendlyname>Survey</friendlyname>
      <foldername>Survey</foldername>
      <modulename>DNN_Survey</modulename>
      <description>Survey allows you to create custom surveys to obtain
public feedback</description>
      <version>04.00.10</version>

    <businesscontrollerclass>DotNetNuke.Modules.Survey.SurveyController</bu
sinesscontrollerclass>
      <modules>
        <module>
          <friendlyname>Survey</friendlyname>
          <cachetime>0</cachetime>
          <controls>
            <control>
              <src>DesktopModules/Survey/Survey.ascx</src>
              <type>View</type>

            <helpurl>http://www.dotnetnuke.com/default.aspx?tabid=787</helpurl>
            </control>
            <control>
              <key>Edit</key>
              <title>Create Survey</title>
              <src>DesktopModules/Survey/EditSurvey.ascx</src>
              <iconfile>icon_survey_32px.gif</iconfile>
              <type>Edit</type>

            <helpurl>http://www.dotnetnuke.com/default.aspx?tabid=787</helpurl>
            </control>
            <control>
              <key>Settings</key>
              <title>Survey Settings</title>
              <src>DesktopModules/Survey/Settings.ascx</src>
              <iconfile>icon_survey_32px.gif</iconfile>
              <type>Edit</type>

            <helpurl>http://www.dotnetnuke.com/default.aspx?tabid=787</helpurl>
            </control>
          </controls>
        </module>
      </modules>
      <files>
        <file>
          <path>App_LocalResources</path>
          <name>EditSurvey.ascx.resx</name>
        </file>
      </files>
    </folder>
  </folders>
</dotnetnuke>
```

DNN4 Module Developers Guide

```
</file>
<file>
  <path>App_LocalResources</path>
  <name>Settings.ascx.resx</name>
</file>
<file>
  <path>App_LocalResources</path>
  <name>Survey.ascx.resx</name>
</file>
<file>
  <path>Providers\DataProviders\SqlDataProvider</path>
  <name>03.01.00.SqlDataProvider</name>
</file>
<file>
  <path>Providers\DataProviders\SqlDataProvider</path>
  <name>03.03.00.SqlDataProvider</name>
</file>
<file>
  <path>Providers\DataProviders\SqlDataProvider</path>
  <name>04.00.10.SqlDataProvider</name>
</file>
<file>
  <name>04.00.10.txt</name>
</file>
<file>
  <name>DNN_Survey.dnn.config</name>
</file>
<file>
  <name>EditSurvey.ascx</name>
</file>
<file>
  <name>EditSurvey.ascx.vb</name>
</file>
<file>
  <name>icon_survey_32px.gif</name>
</file>
<file>
  <name>module.css</name>
</file>
<file>
  <name>red.gif</name>
</file>
<file>
  <name>Settings.ascx</name>
</file>
<file>
  <name>Settings.ascx.vb</name>
</file>
<file>
  <name>Survey.ascx</name>
</file>
```

DNN4 Module Developers Guide

```
<file>
  <name>Survey.ascx.vb</name>
</file>
<file>
  <name>Uninstall.SqlDataProvider</name>
</file>
<file>

<path>[app_code]Providers\DataProviders\SqlDataProvider</path>
  <name>SqlDataProvider.vb</name>
</file>
<file>
  <path>[app_code]</path>
  <name>DataProvider.vb</name>
</file>
<file>
  <path>[app_code]</path>
  <name>SurveyController.vb</name>
</file>
<file>
  <path>[app_code]</path>
  <name>SurveyInfo.vb</name>
</file>
<file>
  <path>[app_code]</path>
  <name>SurveyOptionController.vb</name>
</file>
<file>
  <path>[app_code]</path>
  <name>SurveyOptionInfo.vb</name>
</file>
</files>
</folder>
</folders>
</dotnetnuke>
```

EditSurvey.ascx [C#]

```
<%@ Register TagPrefix="dnn" TagName="Label" Src="~/controls/LabelControl.ascx" %>
<%@ Control Inherits="DotNetNuke.Modules.Survey.EditSurvey"
CodeFile="EditSurvey.ascx.cs" language="C#" AutoEventWireup="true" Explicit="True"
%>
<%@ Register TagPrefix="Portal" TagName="Audit"
Src="~/controls/ModuleAuditControl.ascx" %>
<br>
<table class="Settings" cellpadding="2" cellspacing="0" border="0">
<tr>
<td class="SubHead" width="165"><dnn:label id="plQuestion" text="Question:"
controlname="txtQuestion" runat="server" /></td>
<td><asp:textbox id="txtQuestion" runat="server" cssclass="NormalTextBox"
textmode="MultiLine" rows="3"
maxlength="500" width="300"></asp:textbox></td>
</tr>
<tr>
<td class="SubHead" width="165"><dnn:label id="plOptionType" text="Type:"
controlname="cboOptionType" runat="server" /></td>
<td valign="top">
<asp:dropdownlist id="cboOptionType" runat="server" cssclass="NormalTextBox"
width="300">
<asp:listitem resourcekey="SingleSelection" value="R">Single Selection</asp:listitem>
<asp:listitem resourcekey="MultipleSelection" value="C">Multiple
Selection</asp:listitem>
</asp:dropdownlist>
</td>
</tr>
<tr>
<td class="SubHead" width="165"><dnn:label id="plViewOrder" text="View Order:"
controlname="txtViewOrder" runat="server" /></td>
<td><asp:textbox id="txtViewOrder" runat="server" cssclass="NormalTextBox"
maxlength="128" width="44px"></asp:textbox></td>
</tr>
<tr>
<td colspan="2">&nbsp;</td>
</tr>
<tr>
<td class="SubHead" width="165"><dnn:label id="plOption" text="New Option:"
controlname="txtOption" runat="server" /></td>
<td>
<asp:textbox id="txtOption" runat="server" cssclass="NormalTextBox"
textmode="MultiLine" rows="3"
maxlength="500" width="300px"></asp:textbox>
```



Copyright © 2003-2005 Perpetual Motion Interactive Systems, Inc. All Rights Reserved.

DNN4 Module Developers Guide

```
imageurl="~/images/delete.gif"
OnClick="cmdDeleteOption_Click"></asp:imagebutton></td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
<p>
<asp:linkbutton id="cmdUpdate" resourcekey="cmdUpdate" runat="server"
cssclass="CommandButton" text="Update"
borderstyle="none" OnClick="Update_Click"></asp:linkbutton>&nbsp;
<asp:linkbutton id="cmdCancel" resourcekey="cmdCancel" runat="server"
cssclass="CommandButton" text="Cancel"
borderstyle="none" causesvalidation="False"
OnClick="cmdCancel_Click"></asp:linkbutton>&nbsp;
<asp:linkbutton id="cmdDelete" resourcekey="cmdDelete" runat="server"
cssclass="CommandButton" text="Delete"
borderstyle="none" causesvalidation="False" OnClick="Delete_Click"></asp:linkbutton>
</p>
<portal:audit id="ctlAudit" runat="server" />
```

EditSurvey.ascx [VB]

```
<%@ Register TagPrefix="dnn" TagName="Label" Src="~/controls/LabelControl.ascx" %>
<%@ Control Inherits="DotNetNuke.Modules.Survey.EditSurvey"
CodeFile="EditSurvey.ascx.vb" language="vb" AutoEventWireup="false" Explicit="True"
%>
<%@ Register TagPrefix="Portal" TagName="Audit"
Src="~/controls/ModuleAuditControl.ascx" %>
<br>
<table class="Settings" cellpadding="2" cellspacing="0" border="0">
<tr>
<td class="SubHead" width="165"><dnn:label id="plQuestion" text="Question:"
controlname="txtQuestion" runat="server" /></td>
<td><asp:textbox id="txtQuestion" runat="server" cssclass="NormalTextBox"
textmode="MultiLine" rows="3"
maxlength="500" width="300"></asp:textbox></td>
</tr>
<tr>
<td class="SubHead" width="165"><dnn:label id="plOptionType" text="Type:"
controlname="cboOptionType" runat="server" /></td>
<td valign="top">
<asp:dropdownlist id="cboOptionType" runat="server" cssclass="NormalTextBox"
width="300">
<asp:listitem resourcekey="SingleSelection" value="R">Single Selection</asp:listitem>
<asp:listitem resourcekey="MultipleSelection" value="C">Multiple
Selection</asp:listitem>
</asp:dropdownlist>
</td>
</tr>
<tr>
<td class="SubHead" width="165"><dnn:label id="plViewOrder" text="View Order:"
controlname="txtViewOrder" runat="server" /></td>
<td><asp:textbox id="txtViewOrder" runat="server" cssclass="NormalTextBox"
maxlength="128" width="44px"></asp:textbox></td>
</tr>
<tr>
<td colspan="2">&nbsp;</td>
</tr>
<tr>
<td class="SubHead" width="165"><dnn:label id="plOption" text="New Option:"
controlname="txtOption" runat="server" /></td>
<td>
<asp:textbox id="txtOption" runat="server" cssclass="NormalTextBox"
textmode="MultiLine" rows="3"
maxlength="500" width="300px"></asp:textbox>
```




DNN4 Module Developers Guide

[illegible]

DNN4 Module Developers Guide

```
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
<p>
<asp:linkbutton id="cmdUpdate" resourcekey="cmdUpdate" runat="server"
cssclass="CommandButton" text="Update"
borderstyle="none"></asp:linkbutton>&nbsp;
<asp:linkbutton id="cmdCancel" resourcekey="cmdCancel" runat="server"
cssclass="CommandButton" text="Cancel"
borderstyle="none" causesvalidation="False"></asp:linkbutton>&nbsp;
<asp:linkbutton id="cmdDelete" resourcekey="cmdDelete" runat="server"
cssclass="CommandButton" text="Delete"
borderstyle="none" causesvalidation="False"></asp:linkbutton>
</p>
<portal:audit id="ctlAudit" runat="server" />
```

DNN4 Module Developers Guide

EditSurvey.ascx.cs

```
//
// DotNetNuke - http://www.dotnetnuke.com
// Copyright (c) 2002-2005
// by Perpetual Motion Interactive Systems Inc. ( http://www.perpetualmotion.ca )
//
// Permission is hereby granted, free of charge, to any person obtaining a copy of this
// software and associated
// documentation files (the "Software"), to deal in the Software without restriction,
// including without limitation
// the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
// copies of the Software, and
// to permit persons to whom the Software is furnished to do so, subject to the
// following conditions:
//
// The above copyright notice and this permission notice shall be included in all copies
// or substantial portions
// of the Software.
//
// THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
// OR IMPLIED, INCLUDING BUT NOT LIMITED
// TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
// AND NONINFRINGEMENT. IN NO EVENT SHALL
// THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR
// OTHER LIABILITY, WHETHER IN AN ACTION OF
// CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH
// THE SOFTWARE OR THE USE OR OTHER
// DEALINGS IN THE SOFTWARE.
//

using DotNetNuke;
using DotNetNuke.Services.Localization;
using System.Collections.Generic;
using DotNetNuke.Common.Utilities;
using DotNetNuke.Common;
using System.Web.UI.WebControls;
namespace DotNetNuke.Modules.Survey
{
    public partial class EditSurvey : DotNetNuke.Entities.Modules.PortalModuleBase
    {
        protected System.Web.UI.WebControls.Label Label5;
        private int SurveyId = -1;

        private void Page_Load(object sender, System.EventArgs e)
```

DNN4 Module Developers Guide

```
{
    // get parameter
    if (!(Request.QueryString["SurveyId"] == null))
    {
        SurveyId = int.Parse(Request.QueryString["SurveyId"]);
    }
    else
    {
        SurveyId = Null.NullInteger;
    }
    if ((Page.IsPostBack == false))
    {
        cmdDelete.Attributes.Add("onClick", ("javascript:return confirm(\"
            + (Localization.GetString("DeleteItem") + "\");"));
        if (!Null.IsNull(SurveyId))
        {
            SurveyInfo objSurvey = SurveyController.GetSurvey(SurveyId, ModuleId);
            if (!(objSurvey == null))
            {
                txtQuestion.Text = objSurvey.Question;
                cboOptionType.Items.FindByValue(objSurvey.OptionType).Selected =
true;
                if (!Null.IsNull(objSurvey.ViewOrder))
                {
                    txtViewOrder.Text = objSurvey.ViewOrder.ToString();
                }
                List<SurveyOptionInfo> arrSurveyOptions =
SurveyOptionController.GetSurveyOptions(SurveyId);
                SurveyOptionInfo objSurveyOption;
                int intIndex;
                for (intIndex = 0; (intIndex <= (arrSurveyOptions.Count - 1));
intIndex++)
                {
                    objSurveyOption = ((SurveyOptionInfo)(arrSurveyOptions[intIndex]));
                    objSurveyOption.OptionName =
FormatSurveyOption(objSurveyOption.OptionName, objSurveyOption.IsCorrect);
                    arrSurveyOptions[intIndex] = objSurveyOption;
                }
                lstOptions.DataSource = arrSurveyOptions;
                lstOptions.DataBind();
                ctlAudit.CreatedByUser = objSurvey.CreatedByUser.ToString();
                ctlAudit.CreatedDate = objSurvey.CreatedDate.ToShortDateString();
            }
            else
            {

```

DNN4 Module Developers Guide

```
Module    // security violation attempt to access item not related to this
    Response.Redirect(Globals.NavigateURL());
    }
    }
    else
    {
        // new item
        cmdDelete.Visible = false;
        ctlAudit.Visible = false;
    }
}

protected void Update_Click(object sender, System.EventArgs e)
{
    SurveyInfo objSurvey = new SurveyInfo();
    objSurvey.SurveyId = SurveyId;
    objSurvey.ModuleId = ModuleId;
    objSurvey.Question = txtQuestion.Text;
    if ((txtViewOrder.Text != ""))
    {
        objSurvey.ViewOrder = int.Parse(txtViewOrder.Text);
    }
    else
    {
        objSurvey.ViewOrder = Null.NullInteger;
    }
    objSurvey.OptionType = cboOptionType.SelectedItem.Value;
    objSurvey.CreatedByUser = UserId;
    if (Null.IsNull(SurveyId))
    {
        SurveyId = SurveyController.AddSurvey(objSurvey);
    }
    else
    {
        SurveyController.UpdateSurvey(objSurvey);
    }
    int intOption;
    for (intOption = 0; (intOption <= (lstOptions.Items.Count - 1)); intOption++)
    {
        SurveyOptionInfo objSurveyOption = new SurveyOptionInfo();
        objSurveyOption.SurveyOptionId =
int.Parse(lstOptions.Items[intOption].Value);
        objSurveyOption.SurveyId = SurveyId;
```

DNN4 Module Developers Guide

```
        objSurveyOption.OptionName = lstOptions.Items[intOption].Text;
        objSurveyOption.ViewOrder = intOption;
        objSurveyOption.Votes = 0;
        objSurveyOption.IsCorrect = false;
        if ((objSurveyOption.OptionName ==
FormatSurveyOption(objSurveyOption.OptionName, true)))
        {
            objSurveyOption.OptionName =
FormatSurveyOption(objSurveyOption.OptionName, false);
            objSurveyOption.IsCorrect = true;
        }
        if (Null.IsNull(objSurveyOption.SurveyOptionId))
        {
            SurveyOptionController.AddSurveyOption(objSurveyOption);
        }
        else
        {
            SurveyOptionController.UpdateSurveyOption(objSurveyOption);
        }
    }
    // Redirect back to the portal
    Response.Redirect(Globals.NavigateURL());
}

protected void Delete_Click(object sender, System.EventArgs e)
{
    if (!Null.IsNull(SurveyId))
    {
        SurveyInfo objSurvey = new SurveyInfo();
        objSurvey.SurveyId = SurveyId;
        objSurvey.ModuleId = ModuleId;
        SurveyController.DeleteSurvey(objSurvey);
    }
    // Redirect back to the portal
    Response.Redirect(Globals.NavigateURL());
}

protected void cmdCancel_Click(object sender, System.EventArgs e)
{
    // Redirect back to the portal
    Response.Redirect(Globals.NavigateURL());
}

protected void cmdAddOption_Click(object sender, System.EventArgs e)
{

```

DNN4 Module Developers Guide

```
        if ((txtOption.Text != ""))
        {
            bool blnIsUnique = true;

            int intItem;
            for (intItem = 0; (intItem <= (lstOptions.Items.Count - 1)); intItem++)
            {
                if ((lstOptions.Items[intItem].Text.ToLower() ==
txtOption.Text.ToLower()))
                {
                    blnIsUnique = false;
                }
            }
            if (blnIsUnique)
            {
                string strOption = FormatSurveyOption(txtOption.Text,
chkCorrect.Checked);
                lstOptions.Items.Add(new ListItem(strOption, "-1"));
                chkCorrect.Checked = false;
            }
            else
            {
                DotNetNuke.UI.Skins.Skin.AddModuleMessage(this,
Localization.GetString("OptionExists", this.LocalResourceFile),
DotNetNuke.UI.Skins.Controls.ModuleMessage.ModuleMessageType.YellowWarning);
            }
            txtOption.Text = "";
        }
    }

    protected void cmdDeleteOption_Click(object sender,
System.Web.UI.ImageClickEventArgs e)
    {
        SurveyOptionInfo objSurveyOptionInfo = new SurveyOptionInfo();
        if ((lstOptions.SelectedIndex != -1))
        {
            if ((int.Parse(lstOptions.SelectedItem.Value) != -1))
            {
                objSurveyOptionInfo.SurveyOptionId =
int.Parse(lstOptions.SelectedItem.Value);
                SurveyOptionController.DeleteSurveyOption(objSurveyOptionInfo);
            }
            lstOptions.Items.RemoveAt(lstOptions.SelectedIndex);
        }
    }
}
```

DNN4 Module Developers Guide

```
e) protected void cmdUp_Click(object sender, System.Web.UI.ImageClickEventArgs
{
    if ((lstOptions.SelectedIndex > 0))
    {
        // save
        int intSelectedIndex = lstOptions.SelectedIndex;
        ListItem objListItem = lstOptions.Items[lstOptions.SelectedIndex];
        // remove
        lstOptions.Items.Remove(objListItem);
        // move up
        lstOptions.Items.Insert((intSelectedIndex - 1), objListItem);
    }
}

protected void cmdDown_Click(object sender,
System.Web.UI.ImageClickEventArgs e)
{
    if ((lstOptions.SelectedIndex < (lstOptions.Items.Count - 1)))
    {
        // save
        int intSelectedIndex = lstOptions.SelectedIndex;
        ListItem objListItem = lstOptions.Items[lstOptions.SelectedIndex];
        // remove
        lstOptions.Items.Remove(objListItem);
        // move up
        lstOptions.Items.Insert((intSelectedIndex + 1), objListItem);
    }
}

private string FormatSurveyOption(string OptionName, bool IsCorrect)
{
    if (IsCorrect)
    {
        if (((OptionName.StartsWith(">") == true)
            && (OptionName.EndsWith("<") == true)))
        {
            return OptionName;
        }
        else
        {
            return ">" + OptionName + "<";
        }
    }
}
```


DNN4 Module Developers Guide

```
        else if (((OptionName.StartsWith(">") == true)
                  && (OptionName.EndsWith("<") == true)))
        {
            return OptionName.Substring(1, (OptionName.Length - 2));
        }
        else
        {
            return OptionName;
        }
    }
}
```

EditSurvey.ascx.vb

```
'  
' DotNetNuke - http://www.dotnetnuke.com  
' Copyright (c) 2002-2006  
' by Perpetual Motion Interactive Systems Inc. ( http://www.perpetualmotion.ca )  
'  
' Permission is hereby granted, free of charge, to any person obtaining a copy of this  
software and associated  
' documentation files (the "Software"), to deal in the Software without restriction,  
including without limitation  
' the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell  
copies of the Software, and  
' to permit persons to whom the Software is furnished to do so, subject to the  
following conditions:  
'  
' The above copyright notice and this permission notice shall be included in all copies  
or substantial portions  
' of the Software.  
'  
' THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR  
IMPLIED, INCLUDING BUT NOT LIMITED  
' TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE  
AND NONINFRINGEMENT. IN NO EVENT SHALL  
' THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR  
OTHER LIABILITY, WHETHER IN AN ACTION OF  
' CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH  
THE SOFTWARE OR THE USE OR OTHER  
' DEALINGS IN THE SOFTWARE.  
'
```

```
Imports DotNetNuke  
Imports DotNetNuke.Services.Localization  
Imports System.Collections.Generic
```

```
Namespace DotNetNuke.Modules.Survey
```

```
    Partial Class EditSurvey
```

```
        Inherits DotNetNuke.Entities.Modules.PortalModuleBase  
        Protected Label5 As System.Web.UI.WebControls.Label
```

```
        Dim SurveyId As Integer = -1
```

DNN4 Module Developers Guide

```
Private Sub Page_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

    ' get parameter
    If Not (Request.QueryString("SurveyId") Is Nothing) Then
        SurveyId = Integer.Parse(Request.QueryString("SurveyId"))
    Else
        SurveyId = Null.NullInteger
    End If

    If Page.IsPostBack = False Then

        cmdDelete.Attributes.Add("onClick", "javascript:return confirm("'" &
Localization.GetString("DeleteItem") & "'");")

        If Not Null.IsNull(SurveyId) Then

            Dim objSurvey As SurveyInfo = SurveyController.GetSurvey(SurveyId,
ModuleId)

            If Not objSurvey Is Nothing Then

                txtQuestion.Text = objSurvey.Question
                cboOptionType.Items.FindByValue(objSurvey.OptionType).Selected =
True

                If Not Null.IsNull(objSurvey.ViewOrder) Then
                    txtViewOrder.Text = objSurvey.ViewOrder.ToString
                End If

                Dim arrSurveyOptions As List(Of SurveyOptionInfo) =
SurveyOptionController.GetSurveyOptions(SurveyId)
                Dim objSurveyOption As SurveyOptionInfo
                Dim intIndex As Integer
                For intIndex = 0 To arrSurveyOptions.Count - 1
                    objSurveyOption = CType(arrSurveyOptions(intIndex),
SurveyOptionInfo)
                    objSurveyOption.OptionName =
FormatSurveyOption(objSurveyOption.OptionName, objSurveyOption.IsCorrect)
                    arrSurveyOptions(intIndex) = objSurveyOption
                Next
                lstOptions.DataSource = arrSurveyOptions
                lstOptions.DataBind()

                ctlAudit.CreatedByUser = objSurvey.CreatedByUser.ToString
                ctlAudit.CreatedDate = objSurvey.CreatedDate.ToShortDateString()
```

DNN4 Module Developers Guide

```
Else ' security violation attempt to access item not related to this Module

    Response.Redirect(NavigateURL())

End If

Else ' new item

    cmdDelete.Visible = False
    ctlAudit.Visible = False

End If

End If

End Sub

Private Sub Update_Click(ByVal sender As Object, ByVal e As EventArgs) Handles
cmdUpdate.Click

    Dim objSurvey As New SurveyInfo

    objSurvey.SurveyId = SurveyId
    objSurvey.ModuleId = ModuleId
    objSurvey.Question = txtQuestion.Text
    If txtViewOrder.Text <> "" Then
        objSurvey.ViewOrder = Integer.Parse(txtViewOrder.Text)
    Else
        objSurvey.ViewOrder = Null.NullInteger
    End If
    objSurvey.OptionType = cboOptionType.SelectedItem.Value
    objSurvey.CreatedByUser = UserId

    If Null.IsNull(SurveyId) Then
        SurveyId = SurveyController.AddSurvey(objSurvey)
    Else
        SurveyController.UpdateSurvey(objSurvey)
    End If

    Dim intOption As Integer
    For intOption = 0 To lstOptions.Items.Count - 1

        Dim objSurveyOption As New SurveyOptionInfo
```

DNN4 Module Developers Guide

```
        objSurveyOption.SurveyOptionId =  
Integer.Parse(lstOptions.Items(intOption).Value)  
        objSurveyOption.SurveyId = SurveyId  
        objSurveyOption.OptionName = lstOptions.Items(intOption).Text  
        objSurveyOption.ViewOrder = intOption  
        objSurveyOption.Votes = 0  
        objSurveyOption.IsCorrect = False  
        If objSurveyOption.OptionName =  
FormatSurveyOption(objSurveyOption.OptionName, True) Then  
            objSurveyOption.OptionName =  
FormatSurveyOption(objSurveyOption.OptionName, False)  
            objSurveyOption.IsCorrect = True  
        End If  
  
        If Null.IsNull(objSurveyOption.SurveyOptionId) Then  
            SurveyOptionController.AddSurveyOption(objSurveyOption)  
        Else  
            SurveyOptionController.UpdateSurveyOption(objSurveyOption)  
        End If  
  
    Next  
  
    ' Redirect back to the portal  
    Response.Redirect(NavigateURL())  
  
End Sub  
  
Private Sub Delete_Click(ByVal sender As Object, ByVal e As EventArgs) Handles  
cmdDelete.Click  
  
    If Not Null.IsNull(SurveyId) Then  
        Dim objSurvey As New SurveyInfo  
        objSurvey.SurveyId = SurveyId  
        objSurvey.ModuleId = ModuleId  
        SurveyController.DeleteSurvey(objSurvey)  
    End If  
  
    ' Redirect back to the portal  
    Response.Redirect(NavigateURL())  
  
End Sub  
  
Private Sub cmdCancel_Click(ByVal sender As Object, ByVal e As  
System.EventArgs) Handles cmdCancel.Click
```

DNN4 Module Developers Guide

```
' Redirect back to the portal
Response.Redirect(NavigateURL())

End Sub

Private Sub cmdAddOption_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdAddOption.Click

    If txtOption.Text <> "" Then
        Dim blIsUnique As Boolean = True

        Dim intItem As Integer
        For intItem = 0 To lstOptions.Items.Count - 1
            If lstOptions.Items(intItem).Text.ToLower = txtOption.Text.ToLower
Then
                blIsUnique = False
            End If
        Next

        If blIsUnique Then
            Dim strOption As String = FormatSurveyOption(txtOption.Text,
chkCorrect.Checked)
            lstOptions.Items.Add(New ListItem(strOption, "-1"))
            chkCorrect.Checked = False
        Else
            DotNetNuke.UI.Skins.Skin.AddModuleMessage(Me,
Localization.GetString("OptionExists", Me.LocalResourceFile),
DotNetNuke.UI.Skins.Controls.ModuleMessage.ModuleMessageType.YellowWarning)
        End If

        txtOption.Text = ""
    End If

End Sub

Private Sub cmdDeleteOption_Click(ByVal sender As Object, ByVal e As
System.Web.UI.ImageClickEventArgs) Handles cmdDeleteOption.Click

    Dim objSurveyOptionInfo As New SurveyOptionInfo
    If lstOptions.SelectedIndex <> -1 Then
        If Integer.Parse(lstOptions.SelectedItem.Value) <> -1 Then
            objSurveyOptionInfo.SurveyOptionId =
Integer.Parse(lstOptions.SelectedItem.Value)
            SurveyOptionController.DeleteSurveyOption(objSurveyOptionInfo)
        End If
    End If
```

DNN4 Module Developers Guide

```
IstOptions.Items.RemoveAt(IstOptions.SelectedIndex)
End If

End Sub

Private Sub cmdUp_Click(ByVal sender As Object, ByVal e As
System.Web.UI.ImageClickEventArgs) Handles cmdUp.Click

    If IstOptions.SelectedIndex > 0 Then
        ' save
        Dim intSelectedIndex As Integer = IstOptions.SelectedIndex
        Dim objListItem As ListItem = IstOptions.Items(IstOptions.SelectedIndex)
        ' remove
        IstOptions.Items.Remove(objListItem)
        ' move up
        IstOptions.Items.Insert(intSelectedIndex - 1, objListItem)
    End If

End Sub

Private Sub cmdDown_Click(ByVal sender As Object, ByVal e As
System.Web.UI.ImageClickEventArgs) Handles cmdDown.Click

    If IstOptions.SelectedIndex < IstOptions.Items.Count - 1 Then
        ' save
        Dim intSelectedIndex As Integer = IstOptions.SelectedIndex
        Dim objListItem As ListItem = IstOptions.Items(IstOptions.SelectedIndex)
        ' remove
        IstOptions.Items.Remove(objListItem)
        ' move up
        IstOptions.Items.Insert(intSelectedIndex + 1, objListItem)
    End If

End Sub

Private Function FormatSurveyOption(ByVal OptionName As String, ByVal
IsCorrect As Boolean) As String
    If IsCorrect Then
        If OptionName.StartsWith(">") = True And OptionName.EndsWith("<") = True
Then
            Return OptionName
        Else
            Return ">" & OptionName & "<"
        End If
    Else
```

DNN4 Module Developers Guide

```

    Then
        If OptionName.StartsWith(">") = True And OptionName.EndsWith("<") = True
            Return OptionName.Substring(1, OptionName.Length - 2)
        Else
            Return OptionName
        End If
    End If

End Function

End Class

End Namespace
```


module.css

```
/* Survey Custom Style */  
  
.YourCompanyNameSurveyResults  
{  
  
    font-family: Tahoma, Arial, Helvetica;  
  
    font-size: 12px;  
  
    font-weight: normal;  
  
}
```

Settings.ascx [C#]

```
<% @ Control Inherits="DotNetNuke.Modules.Survey.Settings"
CodeFile="Settings.ascx.cs" language="C#" AutoEventWireup="false" Explicit="true"
%>
<% @ Register TagPrefix="dnn" TagName="Label" Src="~/controls/LabelControl.ascx"
%>
<table cellSpacing="0" cellPadding="4" border="0">
<tr>
<td class="SubHead" width="200"><dnn:label id="plClosingDate" text="Survey Closing
Date:" controlname="txtClosingDate" runat="server" /></td>
<td valign="top">
<asp:TextBox id="txtClosingDate" Runat="server"
CssClass="NormalTextBox"></asp:TextBox>
<asp:HyperLink id=cmdCalendar resourcekey="Calendar" CssClass="CommandButton"
Runat="server">Calendar</asp:HyperLink>
<asp:CompareValidator id=valClosingDate runat="server" CssClass="NormalRed"
resourcekey="valClosingDate.ErrorMessage" ControlToValidate="txtClosingDate"
ErrorMessage="<br>Invalid Closing Date!" Operator="DataTypeCheck" Type="Date"
Display="Dynamic"></asp:CompareValidator>
</td>
</tr>
<tr>
<td class="SubHead" width="200"><dnn:label id="plGraphWidth" text="Maximum Bar
Graph Width:" controlname="txtGraphWidth" runat="server" /></td>
<td valign="top"><asp:TextBox id="txtGraphWidth" Runat="server"
CssClass="NormalTextBox"></asp:TextBox></td>
</tr>
<tr>
<td class="SubHead" width="200"><dnn:label id="plPersonal" text="Vote Tracking:"
controlname="rblstPersonal" runat="server" /></td>
<td vAlign="top" height="36">
<asp:RadioButtonList id="rblstPersonal" CssClass="NormalTextBox" runat="server"
Width="216px">
<asp:ListItem Value="0" Selected="True">Vote tracking via cookie</asp:ListItem>
<asp:ListItem Value="1">1 Vote/Registered User</asp:ListItem>
</asp:RadioButtonList>
</td>
</tr>
<tr>
<td class="SubHead" width="200"><dnn:label id="plSurveyResults" text="Survey
Results:" controlname="rblstSurveyResults" runat="server" /></td>
```

DNN4 Module Developers Guide

```
<td vAlign="top" height="36">
<asp:RadioButtonList id="rblstSurveyResults" CssClass="NormalTextBox"
Width="216px" runat="server" repeatdirection="Horizontal">
<asp:ListItem Value="0" Selected="True">Public</asp:ListItem>
<asp:ListItem Value="1">Private</asp:ListItem>
</asp:RadioButtonList>
</td>
</tr>
</table>
```

Settings.ascx [VB]

```

<% @ Control Inherits="DotNetNuke.Modules.Survey.Settings"
CodeFile="Settings.ascx.vb" language="vb" AutoEventWireup="false" Explicit="true"
%>
<% @ Register TagPrefix="dnn" TagName="Label" Src="~/controls/LabelControl.ascx"
%>
<table cellSpacing="0" cellPadding="4" border="0">
<tr>
<td class="SubHead" width="200"><dnn:label id="plClosingDate" text="Survey Closing
Date:" controlname="txtClosingDate" runat="server" /></td>
<td valign="top">
<asp:TextBox id="txtClosingDate" Runat="server"
CssClass="NormalTextBox"></asp:TextBox>
<asp:HyperLink id=cmdCalendar resourcekey="Calendar" CssClass="CommandButton"
Runat="server">Calendar</asp:HyperLink>
<asp:CompareValidator id=valClosingDate runat="server" CssClass="NormalRed"
resourcekey="valClosingDate.ErrorMessage" ControlToValidate="txtClosingDate"
ErrorMessage="<br>Invalid Closing Date!" Operator="DataTypeCheck" Type="Date"
Display="Dynamic"></asp:CompareValidator>
</td>
</tr>
<tr>
<td class="SubHead" width="200"><dnn:label id="plGraphWidth" text="Maximum Bar
Graph Width:" controlname="txtGraphWidth" runat="server" /></td>
<td valign="top"><asp:TextBox id="txtGraphWidth" Runat="server"
CssClass="NormalTextBox"></asp:TextBox></td>
</tr>
<tr>
<td class="SubHead" width="200"><dnn:label id="plPersonal" text="Vote Tracking:"
controlname="rblstPersonal" runat="server" /></td>
<td vAlign="top" height="36">
<asp:RadioButtonList id="rblstPersonal" CssClass="NormalTextBox" runat="server"
Width="216px">
<asp:ListItem resourcekey="No" Value="0" Selected="true">Vote tracking via
cookie</asp:ListItem>
<asp:ListItem resourcekey="Yes" Value="1">1 Vote/Registered User</asp:ListItem>
</asp:RadioButtonList>
</td>
</tr>
<tr>
<td class="SubHead" width="200"><dnn:label id="plSurveyResults" text="Survey

```

DNN4 Module Developers Guide

```
Results:" controlname="rblstSurveyResults" runat="server" /></td>
<td vAlign="top" height="36">
<asp:RadioButtonList id="rblstSurveyResults" CssClass="NormalTextBox"
Width="216px" runat="server" repeatdirection="Horizontal">
<asp:ListItem resourcekey="PublicResults" Value="0"
Selected="true">Public</asp:ListItem>
<asp:ListItem resourcekey="PrivateResults" Value="1">Private</asp:ListItem>
</asp:RadioButtonList>
</td>
</tr>
</table>
```

Settings.ascx.cs

```
//
// DotNetNuke - http://www.dotnetnuke.com
// Copyright (c) 2002-2005
// by Perpetual Motion Interactive Systems Inc. ( http://www.perpetualmotion.ca )
//
// Permission is hereby granted, free of charge, to any person obtaining a copy of this
// software and associated
// documentation files (the "Software"), to deal in the Software without restriction,
// including without limitation
// the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
// copies of the Software, and
// to permit persons to whom the Software is furnished to do so, subject to the
// following conditions:
//
// The above copyright notice and this permission notice shall be included in all copies
// or substantial portions
// of the Software.
//
// THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
// OR IMPLIED, INCLUDING BUT NOT LIMITED
// TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
// AND NONINFRINGEMENT. IN NO EVENT SHALL
// THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR
// OTHER LIABILITY, WHETHER IN AN ACTION OF
// CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH
// THE SOFTWARE OR THE USE OR OTHER
// DEALINGS IN THE SOFTWARE.
//
using System.IO;
using System.Runtime.Serialization.Formatters;
using DotNetNuke;
using System;
using DotNetNuke.Services.Exceptions;
using DotNetNuke.Common.Utilities;

namespace DotNetNuke.Modules.Survey
{
    public partial class Settings : DotNetNuke.Entities.Modules.ModuleSettingsBase
    {
        public override void LoadSettings()
        {
            try
            {

```

DNN4 Module Developers Guide

```
// this needs to execute always to the client script code is registered in
InvokePopupCal
    cmdCalendar.NavigateUrl =
DotNetNuke.Common.Utilities.Calendar.InvokePopupCal(txtClosingDate);
    if ((Page.IsPostBack == false))
    {
        if ((ModuleSettings["surveyclosingdate"]) == null)
        {
            ModuleSettings["surveyclosingdate"] =
DateTime.Now.AddYears(1).ToShortDateString();
        }
        if ((ModuleSettings["surveytracking"]) == null)
        {
            ModuleSettings["surveytracking"] = 0;
        }
        if ((ModuleSettings["surveyresultstype"]) == null)
        {
            ModuleSettings["surveyresultstype"] = 0;
        }

        txtClosingDate.Text =
Convert.ToDateTime(ModuleSettings["surveyclosingdate"]).ToShortDateString();
        txtGraphWidth.Text = ((string)(TabModuleSettings["surveygraphwidth"]));
        rblstPersonal.SelectedIndex =
(Convert.ToInt32((ModuleSettings["surveytracking"])));
        rblstSurveyResults.SelectedIndex =
(Convert.ToInt32((ModuleSettings["surveyresultstype"])));
    }
}
catch (Exception exc)
{
    // Module failed to load
    Exceptions.ProcessModuleLoadException(this, exc);
}

public override void UpdateSettings()
{
    try
    {
        DotNetNuke.Entities.Modules.ModuleController objModules = new
DotNetNuke.Entities.Modules.ModuleController();
        DateTime datClosingDate = Null.NullDate;
        if ((txtClosingDate.ToString().Trim().Length > 0))
        {
```

DNN4 Module Developers Guide

```
        datClosingDate = Convert.ToDateTime(txtClosingDate.Text.Trim());
    }
    // Update Module Settings
    objModules.UpdateModuleSetting(ModuleId, "surveyclosingdate",
DotNetNuke.Common.Globals.DateToString(datClosingDate));
    objModules.UpdateModuleSetting(ModuleId, "surveytracking",
rblstPersonal.SelectedIndex.ToString());
    objModules.UpdateModuleSetting(ModuleId, "surveyresultstype",
rblstSurveyResults.SelectedIndex.ToString());
    // Update Tab Module Settings
    objModules.UpdateTabModuleSetting(TabModuleId, "surveygraphwidth",
txtGraphWidth.Text);
    }
    catch (Exception exc)
    {
        // Module failed to load
        Exceptions.ProcessModuleLoadException(this, exc);
    }
}
}
```


Settings.ascx.vb

```
'
' DotNetNuke - http://www.dotnetnuke.com
' Copyright (c) 2002-2006
' by Perpetual Motion Interactive Systems Inc. ( http://www.perpetualmotion.ca )
'
' Permission is hereby granted, free of charge, to any person obtaining a copy of this
' software and associated
' documentation files (the "Software"), to deal in the Software without restriction,
' including without limitation
' the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
' copies of the Software, and
' to permit persons to whom the Software is furnished to do so, subject to the
' following conditions:
'
' The above copyright notice and this permission notice shall be included in all copies
' or substantial portions
' of the Software.
'
' THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
' IMPLIED, INCLUDING BUT NOT LIMITED
' TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
' AND NONINFRINGEMENT. IN NO EVENT SHALL
' THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR
' OTHER LIABILITY, WHETHER IN AN ACTION OF
' CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH
' THE SOFTWARE OR THE USE OR OTHER
' DEALINGS IN THE SOFTWARE.
'

Imports System.IO
Imports System.Runtime.Serialization.Formatters
Imports DotNetNuke

Namespace DotNetNuke.Modules.Survey

    Partial Public Class Settings
        Inherits DotNetNuke.Entities.Modules.ModuleSettingsBase

    #Region "Base Method Implementations"

        Public Overrides Sub LoadSettings()
            Try
```

DNN4 Module Developers Guide

```
'this needs to execute always to the client script code is registered in
InvokePopupCal
    cmdCalendar.NavigateUrl =
DotNetNuke.Common.Utilities.Calendar.InvokePopupCal(txtClosingDate)

If (Page.IsPostBack = False) Then

    If CType(ModuleSettings("surveyclosingdate"), String) <> "" Then
        txtClosingDate.Text = CType(ModuleSettings("surveyclosingdate"),
Date).ToShortDateString
    End If
    txtGraphWidth.Text = CType(TabModuleSettings("surveygraphwidth"),
String)
    rblstPersonal.SelectedIndex = CType(ModuleSettings("surveytracking"),
Integer)
    rblstSurveyResults.SelectedIndex =
CType(ModuleSettings("surveyresultstype"), Integer)
    End If
    Catch exc As Exception 'Module failed to load
        ProcessModuleLoadException(Me, exc)
    End Try
End Sub

Public Overrides Sub UpdateSettings()
    Try
        Dim objModules As New DotNetNuke.Entities.Modules.ModuleController

        Dim datClosingDate As Date = Null.NullDate
        If txtClosingDate.Text.Trim.Length > 0 Then
            datClosingDate = Convert.ToDateTime(txtClosingDate.Text.Trim)
        End If

        'Update Module Settings
        objModules.UpdateModuleSetting(ModuleId, "surveyclosingdate",
DotNetNuke.Common.Globals.DateToString(datClosingDate))
        objModules.UpdateModuleSetting(ModuleId, "surveytracking",
(rblstPersonal.SelectedIndex).ToString)
        objModules.UpdateModuleSetting(ModuleId, "surveyresultstype",
(rblstSurveyResults.SelectedIndex).ToString)

        'Update Tab Module Settings
        objModules.UpdateTabModuleSetting(TabModuleId, "surveygraphwidth",
txtGraphWidth.Text)

        Catch exc As Exception 'Module failed to load
```

DNN4 Module Developers Guide

```
        ProcessModuleLoadException(Me, exc)
    End Try
End Sub

#End Region

End Class

End Namespace
```

Survey.ascx [C#]

```
<%@ Control Language="C#" AutoEventWireup="true"
Inherits="DotNetNuke.Modules.Survey.Survey" CodeFile="Survey.ascx.cs" %>
<asp:panel id="pnlSurvey" runat="server" visible="False">
<asp:datalist id="lstSurvey" runat="server" cellpadding="4" datakeyfield="SurveyId"
OnItemDataBound="lstSurvey_ItemDataBound">
<ItemTemplate>
<asp:HyperLink id="cmdEdit2" ImageUrl="~/images/edit.gif" NavigateUrl='<##
EditUrl("SurveyId",Eval("SurveyId").ToString()) %>' Visible="<## IsEditable %>"
runat="server" />
    <asp:Label ID="QuestionLabel" runat="server" Text='<## Eval("Question") %>'
Font-Bold="False"></asp:Label><br />
    <asp:RadioButtonList ID="optOptions" runat="server">
</asp:RadioButtonList>
    <asp:CheckBoxList ID="chkOptions" runat="server">
</asp:CheckBoxList><br />
</ItemTemplate>
</asp:datalist>
<asp:linkbutton id="cmdSubmit" runat="server" resourcekey="cmdSubmit"
cssclass="CommandButton" OnClick="cmdSubmit_Click">Submit
Survey</asp:linkbutton>&nbsp;
<asp:linkbutton id="cmdResults" runat="server" resourcekey="cmdResults"
cssclass="CommandButton" OnClick="cmdResults_Click">View
Results</asp:linkbutton></asp:panel>
<asp:Label ID="Message_Label" runat="server" EnableViewState="False"></asp:Label>
<asp:panel id="pnlResults" runat="server" visible="False">
<asp:datalist id="lstResults" runat="server" cellpadding="4" datakeyfield="SurveyId"
OnItemDataBound="lstResults_ItemDataBound">
<itemtemplate>
<asp:HyperLink id="cmdEdit2" ImageUrl="~/images/edit.gif" NavigateUrl='<##
EditUrl("SurveyId",Eval("SurveyId").ToString()) %>' Visible="<## IsEditable %>"
runat="server" />
<asp:Label ID="lblQuestion2" Runat="server" CssClass="NormalBold" Text='<##
FormatQuestion(Eval("Question").ToString()),(lstResults.Items.Count + 1))
%>'></asp:Label><br>
<asp:label id="lblResults" runat="server" cssclass="Normal"></asp:label>
<br>
</itemtemplate>
</asp:datalist>
<asp:linkbutton id="cmdSurvey" runat="server" resourcekey="cmdSurvey"
cssclass="CommandButton" OnClick="cmdSurvey_Click">View
Survey</asp:linkbutton></asp:panel>
&nbsp;
```

Survey.ascx [VB]

```
<%@ Control Language="vb" AutoEventWireup="false"
Inherits="DotNetNuke.Modules.Survey.Survey" CodeFile="Survey.ascx.vb" %>
<asp:panel id="pnlSurvey" runat="server" visible="False">
<asp:datalist id="lstSurvey" runat="server" cellpadding="4" datakeyfield="SurveyId">
<ItemTemplate>
<asp:HyperLink id="cmdEdit2" ImageUrl="~/images/edit.gif" NavigateUrl='<##
EditURL("SurveyId",DataBinder.Eval(Container.DataItem,"SurveyId")) %>' Visible="<##
IsEditable %>" runat="server" />
    <asp:Label ID="QuestionLabel" runat="server" Text='<## Eval("Question") %>'
Font-Bold="False"></asp:Label><br />
    <asp:RadioButtonList ID="optOptions" runat="server">
</asp:RadioButtonList>
    <asp:CheckBoxList ID="chkOptions" runat="server">
</asp:CheckBoxList><br />
</ItemTemplate>
</asp:datalist>
<asp:linkbutton id="cmdSubmit" runat="server" resourcekey="cmdSubmit"
cssclass="CommandButton">Submit Survey</asp:linkbutton>&nbsp;
<asp:linkbutton id="cmdResults" runat="server" resourcekey="cmdResults"
cssclass="CommandButton">View Results</asp:linkbutton></asp:panel>
<asp:Label ID="Message_Label" runat="server" EnableViewState="False"></asp:Label>
<asp:panel id="pnlResults" runat="server" visible="False">
<asp:datalist id="lstResults" runat="server" cellpadding="4" datakeyfield="SurveyId">
<itemtemplate>
<asp:HyperLink id="cmdEdit2" ImageUrl="~/images/edit.gif" NavigateUrl='<##
EditURL("SurveyId",DataBinder.Eval(Container.DataItem,"SurveyId")) %>' Visible="<##
IsEditable %>" runat="server" />
<asp:Label ID="lblQuestion2" Runat="server" CssClass="NormalBold" Text='<##
FormatQuestion(DataBinder.Eval(Container.DataItem,"Question"),lstResults.Items.Coun
t + 1) %>'>
</asp:Label><br>
<asp:label id="lblResults" runat="server" cssclass="Normal"></asp:label>
<br>
</itemtemplate>
</asp:datalist>
<asp:linkbutton id="cmdSurvey" runat="server" resourcekey="cmdSurvey"
cssclass="CommandButton">View Survey</asp:linkbutton>
</asp:panel>
```

DNN4 Module Developers Guide

Survey.ascx.cs

```
//
// DotNetNuke - http://www.dotnetnuke.com
// Copyright (c) 2002-2005
// by Perpetual Motion Interactive Systems Inc. ( http://www.perpetualmotion.ca )
//
// Permission is hereby granted, free of charge, to any person obtaining a copy of this
// software and associated
// documentation files (the "Software"), to deal in the Software without restriction,
// including without limitation
// the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
// copies of the Software, and
// to permit persons to whom the Software is furnished to do so, subject to the
// following conditions:
//
// The above copyright notice and this permission notice shall be included in all copies
// or substantial portions
// of the Software.
//
// THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
// OR IMPLIED, INCLUDING BUT NOT LIMITED
// TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
// AND NONINFRINGEMENT. IN NO EVENT SHALL
// THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR
// OTHER LIABILITY, WHETHER IN AN ACTION OF
// CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH
// THE SOFTWARE OR THE USE OR OTHER
// DEALINGS IN THE SOFTWARE.
//
using DotNetNuke;
using DotNetNuke.Security.Roles;
using System.Collections.Generic;
using System.Web.UI.WebControls;
using System;
using DotNetNuke.Services.Localization;
using DotNetNuke.Security;
using DotNetNuke.Services.Exceptions;
using System.Web;
using Microsoft.VisualBasic;
namespace DotNetNuke.Modules.Survey
{
    public partial class Survey : DotNetNuke.Entities.Modules.PortalModuleBase,
    Entities.Modules.IActionable
    {
```

DNN4 Module Developers Guide

```
private string strCookie;
private bool blnSurveyExpired;
private bool blnPersonalVoteTracking;
private bool blnVoted;
private bool blnPrivateResults;

private void Page_Load(object sender, System.EventArgs e)
{
    try
    {
        strCookie = ("_Module" + (ModuleId.ToString() + "_Survey"));
        blnSurveyExpired = SurveyExpired();
        blnPersonalVoteTracking = SurveyTracking();
        blnPrivateResults = SurveyResultsType();
        blnVoted = HasVoted();
        if (!Page.IsPostBack)
        {
            DisplaySurveyPage();
        }
    }
    catch (Exception exc)
    {
        // Module failed to load
        Exceptions.ProcessModuleLoadException(this, exc);
    }
}

private bool SurveyExpired()
{
    if (!(Settings["surveyclosingdate"] == null))
    {
        try
        {
            // Using VB DateAndTime rather than using a C# method to keep the
            code close to the VB version
            if (DateAndTime.DateDiff(DateInterval.Day, DateAndTime.Now,
                DateTime.Parse(Settings["surveyclosingdate"].ToString()),
                Microsoft.VisualBasic.FirstDayOfWeek.System, FirstWeekOfYear.System) > -1)
            {
                return false;
            }
            else
            {
                return true;
            }
        }
    }
}
```

DNN4 Module Developers Guide

```
        }
        catch (System.Exception)
        {
            return false;
        }
    }
    else
    {
        return false;
    }
}

private bool SurveyTracking()
{
    // Check the settings to see if this module is using Personalization for vote
tracking
    if (!(Settings["surveytracking"] == null))
    {
        return
(Convert.ToBoolean(int.Parse(Settings["surveytracking"].ToString())));
    }
    else
    {
        return false;
    }
}

private bool SurveyResultsType()
{
    // If the user is able to edit the module then they are always able to see the
results
    if
(DotNetNuke.Security.PortalSecurity.IsInRoles(this.ModuleConfiguration.AuthorizedEdi
tRoles))
    {
        return false;
    }
    // Check the settings to see if this module is using Personalization for vote
tracking
    if (!(Settings["surveyresultstype"] == null))
    {
        return
(Convert.ToBoolean(int.Parse(Settings["surveyresultstype"].ToString())));
    }
    else
```


DNN4 Module Developers Guide

```
        {
            return false;
        }
    }

    private bool HasVoted()
    {
        if ((blnPersonalVoteTracking == true))
        {
            if ((UserId != -1))
            {
                // Check if the user has voted before
                if
                (!DotNetNuke.Services.Personalization.Personalization.GetProfile(ModuleId.ToString()
                , "Voted") == null))
                {
                    return
                    ((bool)(DotNetNuke.Services.Personalization.Personalization.GetProfile(ModuleId.ToSt
                    ring(), "Voted")));
                }
                else
                {
                    return false;
                }
            }
            else
            {
                // The module is using Personalization but the user is not logged in
                return true;
            }
        }
        else
        {
            // The Module is not using personalization so look for a cookie
            if ((Request.Cookies[strCookie] == null))
            {
                return false;
            }
            else
            {
                return true;
            }
        }
    }
}
```

DNN4 Module Developers Guide

```
private void DisplaySurveyPage()
{
    // Only show Survey if it is not expired
    if (!blnSurveyExpired)
    {
        if ((blnPersonalVoteTracking && (UserId == -1)))
        {
            // The User is not logged in and the module is using personalization
            DisplaySurvey();
            cmdSubmit.Visible = false;
            cmdResults.Visible = false;
            Message_Label.Text = Localization.GetString("Survey_MustBeSignedIn",
this.LocalResourceFile);
        }
        else if ((blnVoted == false))
        {
            // They haven't voted so show the Survey
            DisplaySurvey();
            // If the results are private, hide the results link
            if (blnPrivateResults)
            {
                cmdResults.Visible = false;
            }
        }
        else
        {
            // They have voted so display the results
            DisplayResults();
        }
    }
    else
    {
        // Survey is expired so display the results
        DisplayResults();
    }
}

private void DisplaySurvey()
{
    pnlResults.Visible = false;
    List<SurveyInfo> Surveylist = SurveyController.GetSurveys(ModuleId);
    if ((Surveylist.Count > 0))
    {
        lstSurvey.DataSource = Surveylist;
        lstSurvey.DataBind();
    }
}
```

DNN4 Module Developers Guide

```
        pnlSurvey.Visible = true;
    }
    else
    {
        pnlSurvey.Visible = false;
    }
}

private void DisplayResults()
{
    lstResults.DataSource = SurveyController.GetSurveys(ModuleId);
    lstResults.DataBind();
    // Hide the survey items, show the results
    pnlSurvey.Visible = false;
    pnlResults.Visible = true;
    if
((DotNetNuke.Security.PortalSecurity.IsInRoles(this.ModuleConfiguration.AuthorizedEditRoles) && (UserId != -1)))
    {
        cmdSurvey.Visible = true;
    }
    else
    {
        // If the results are private, hide the results and the results link
        if (blnPrivateResults)
        {
            lstResults.Visible = false;
            cmdResults.Visible = false;
            Message_Label.Text = Localization.GetString("AlreadyVoted",
this.LocalResourceFile);
        }
        // If they have voted or the Survey has expired, hide the button to show
the Survey items
        if ((!blnVoted && !blnSurveyExpired))
        {
            cmdSurvey.Visible = true;
        }
        else
        {
            cmdSurvey.Visible = false;
        }
    }
}

public string FormatQuestion(string strQuestion, int ItemNumber)
```

DNN4 Module Developers Guide

```
{
    return (ItemNumber.ToString() + ("." + strQuestion));
}

private string FormatSurveyOption(string OptionName, bool IsCorrect)
{
    if (IsCorrect)
    {
        if (((OptionName.StartsWith(">") == true) && (OptionName.EndsWith("<") ==
true)))
        {
            return OptionName;
        }
        else
        {
            return (">" + OptionName + "<");
        }
    }
    else if (((OptionName.StartsWith(">") == true)
        && (OptionName.EndsWith("<") == true)))
    {
        return OptionName.Substring(1, (OptionName.Length - 2));
    }
    else
    {
        return OptionName;
    }
}

protected void lstSurvey_ItemDataBound(object sender, DataListItemEventArgs e)
{
    SurveyInfo objSurvey =
SurveyController.GetSurvey(Int32.Parse(lstSurvey.DataKeys[e.Item.ItemIndex].ToString()), ModuleId);
    if (!(objSurvey == null))
    {
        List<SurveyOptionInfo> SurveyOptionList = new List<SurveyOptionInfo>();
        SurveyOptionList =
SurveyOptionController.GetSurveyOptions(objSurvey.SurveyId);
        switch (objSurvey.OptionType)
        {
            case "R":
                RadioButtonList optOptions =
((RadioButtonList)(e.Item.FindControl("optOptions")));
                optOptions.DataSource = SurveyOptionList;
```

DNN4 Module Developers Guide

```
        optOptions.DataTextField = "OptionName";
        optOptions.DataValueField = "SurveyOptionId";
        optOptions.DataBind();
        optOptions.SelectedValue =
SurveyOptionList[0].SurveyOptionId.ToString();
        optOptions.Visible = true;
        break;
    case "C":
        CheckBoxList chkOptions =
((CheckBoxList)(e.Item.FindControl("chkOptions")));
        chkOptions.DataSource = SurveyOptionList;
        chkOptions.DataTextField = "OptionName";
        chkOptions.DataValueField = "SurveyOptionId";
        chkOptions.DataBind();
        chkOptions.Visible = true;
        break;
    }
}
}

protected void cmdSubmit_Click(object sender, System.EventArgs e)
{
    SurveyInfo objSurvey;
    int intSurvey;
    SurveyOptionInfo objSurveyOption = new SurveyOptionInfo();
    int intSurveyOption;
    int intQuestion;
    int intOption;
    bool blnValid = true;
    intQuestion = -1;
    List<SurveyInfo> arrSurveys = SurveyController.GetSurveys(ModuleId);
    for (intSurvey = 0; (intSurvey <= (arrSurveys.Count - 1)); intSurvey++)
    {
        objSurvey = arrSurveys[intSurvey];
        intQuestion++;
        switch (objSurvey.OptionType)
        {
            case "R":
                if ((Request.Form[(intSurvey.UniqueId + (":_ctl" +
(intQuestion.ToString() + ":optOptions")))] == ""))
                {
                    blnValid = false;
                }
                break;
            case "C":
```

DNN4 Module Developers Guide

```

        blnValid = false;
        intOption = -1;
        List<SurveyOptionInfo> arrSurveyOptions =
SurveyOptionController.GetSurveyOptions(objSurvey.SurveyId);
        for (intSurveyOption = 0; (intSurveyOption <= (arrSurveyOptions.Count
- 1)); intSurveyOption++)
        {
            intOption++;
            if ((Request.Form[(IstSurvey.UniqueId + (":_ctl" +
(intQuestion.ToString() + ":chkOptions:" + intOption.ToString())))] != ""))
            {
                blnValid = true;
            }
        }
        break;
    }
}
if (blnValid)
{
    intQuestion = -1;
    for (intSurvey = 0; (intSurvey <= (arrSurveys.Count - 1)); intSurvey++)
    {
        objSurvey = ((SurveyInfo)(arrSurveys[intSurvey]));
        intQuestion++;
        switch (objSurvey.OptionType)
        {
            case "R":
                if ((Request.Form[(IstSurvey.UniqueId + (":_ctl" +
(intQuestion.ToString() + ":optOptions")))] != ""))
                {
                    objSurveyOption.SurveyOptionId =
Int32.Parse(Request.Form[(IstSurvey.UniqueId + (":_ctl" + (intQuestion.ToString() +
":optOptions")))]);
                    SurveyOptionController.AddSurveyResult(objSurveyOption,
UserId);
                }
                break;
            case "C":
                intOption = -1;
                List<SurveyOptionInfo> arrSurveyOptions =
SurveyOptionController.GetSurveyOptions(objSurvey.SurveyId);
                for (intSurveyOption = 0; (intSurveyOption <=
(arrSurveyOptions.Count - 1)); intSurveyOption++)
                {

```

DNN4 Module Developers Guide

```
                objSurveyOption =
((SurveyOptionInfo)(arrSurveyOptions[intSurveyOption]));
                intOption++;
                if ((Request.Form[(IstSurvey.UniqueID + (":_ctl" +
(intQuestion.ToString() + (":chkOptions:" + intOption.ToString())))] != null))
                {
                    SurveyOptionController.AddSurveyResult(objSurveyOption,
UserId);
                }
            }
            break;
        }
    }
    Message_Label.Text = Localization.GetString("SurveyComplete",
this.LocalResourceFile);
    if ((bInPersonalVoteTracking == true))
    {
        // This means the module vote tracking is using personalization, so set
the profile to show they have voted
DotNetNuke.Services.Personalization.Personalization.SetProfile(ModuleId.ToString(),
"Voted", true);
        bInVoted = true;
        DisplayResults();
    }
    else
    {
        // Store a cookie to show the chart after the submit
        HttpCookie objCookie = new HttpCookie(("_Module" +
(ModuleId.ToString() + "_Survey")));
        objCookie.Value = "True";
        objCookie.Expires = DateTime.MaxValue;
        // never expires
        Response.AppendCookie(objCookie);
        bInVoted = true;
        DisplayResults();
    }
}
else
{
    Message_Label.Text = Localization.GetString("SurveyIncomplete",
this.LocalResourceFile);
}
}
```

DNN4 Module Developers Guide

```
protected void cmdResults_Click(object sender, System.EventArgs e)
{
    DisplayResults();
}

protected void lstResults_ItemDataBound(object sender,
System.Web.UI.WebControls.DataListItemEventArgs e)
{
    SurveyInfo objSurvey;
    SurveyOptionInfo objSurveyOption;
    int intSurveyOption;
    string strHTML;
    int intGraphWidth = 100;
    if ((Settings["surveygraphwidth"] != null)
    {
        try
        {
            intGraphWidth = Int32.Parse((string)(Settings["surveygraphwidth"]));
        }
        catch
        {
            intGraphWidth = 100;
        }
    }
    strHTML = "<BR><TABLE BORDER=\"0\" CELLPADDING=\"2\" CELLSPACING=\"0\"
WIDTH=\"100%\">";
    objSurvey =
SurveyController.GetSurvey(Int32.Parse(lstResults.DataKeys[e.Item.ItemIndex].ToString()), ModuleId);
    if (!(objSurvey == null))
    {
        List<SurveyOptionInfo> arrSurveyOptions =
SurveyOptionController.GetSurveyOptions(objSurvey.SurveyId);
        for (intSurveyOption = 0; (intSurveyOption <= (arrSurveyOptions.Count - 1));
intSurveyOption++)
        {
            objSurveyOption = arrSurveyOptions[intSurveyOption];
            float dblPercent = 0;
            int intPercent = 0;
            if ((objSurvey.Votes != 0))
            {
                dblPercent = ((float)objSurveyOption.Votes / (float)objSurvey.Votes);
                intPercent = Convert.ToInt16(dblPercent * 100);
            }
            strHTML += "<TR>";
```


DNN4 Module Developers Guide

```

        strHTML = (strHTML + ("<TD VALIGN=\"top\"
CLASS=\"YourCompanyNameSurveyResults\">"
        + (FormatSurveyOption(objSurveyOption.OptionName,
objSurveyOption.IsCorrect) + (" ("
        + (objSurveyOption.Votes.ToString() + "</TD>"))));
        strHTML = (strHTML + ("<TD ALIGN=\"left\" VALIGN=\"top\"
CLASS=\"Normal\" NOWRAP><IMG SRC=\"\"
        + (this.TemplateSourceDirectory + ("/red.gif\" WIDTH=\"\"
        + ((intGraphWidth * dblPercent) + ("\" BORDER=\"0\"
HEIGHT=\"15\"> "
        + (intPercent.ToString() + "%</TD>"))))));
        strHTML += "</TR>";
    }
}
strHTML += "</TABLE>";
Label lblResults = ((Label)(e.Item.FindControl("lblResults")));
lblResults.Text = strHTML;
}

protected void cmdSurvey_Click(object sender, System.EventArgs e)
{
    DisplaySurvey();
}

public DotNetNuke.Entities.Modules.Actions.ModuleActionCollection
ModuleActions
{
    get
    {
        Entities.Modules.Actions.ModuleActionCollection Actions = new
Entities.Modules.Actions.ModuleActionCollection();
        Actions.Add(GetNextActionID(),
Localization.GetString(Entities.Modules.Actions.ModuleActionType.AddContent,
LocalResourceFile), Entities.Modules.Actions.ModuleActionType.AddContent, "", "",
EditUrl(), false, SecurityAccessLevel.Edit, true, false);
        return Actions;
    }
}
}
}

```

Survey.ascx.vb

```
'  
' DotNetNuke - http://www.dotnetnuke.com  
' Copyright (c) 2002-2006  
' by Perpetual Motion Interactive Systems Inc. ( http://www.perpetualmotion.ca )  
'  
' Permission is hereby granted, free of charge, to any person obtaining a copy of this  
software and associated  
' documentation files (the "Software"), to deal in the Software without restriction,  
including without limitation  
' the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell  
copies of the Software, and  
' to permit persons to whom the Software is furnished to do so, subject to the  
following conditions:  
'  
' The above copyright notice and this permission notice shall be included in all copies  
or substantial portions  
' of the Software.  
'  
' THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR  
IMPLIED, INCLUDING BUT NOT LIMITED  
' TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE  
AND NONINFRINGEMENT. IN NO EVENT SHALL  
' THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR  
OTHER LIABILITY, WHETHER IN AN ACTION OF  
' CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH  
THE SOFTWARE OR THE USE OR OTHER  
' DEALINGS IN THE SOFTWARE.  
'
```

```
Imports DotNetNuke  
Imports DotNetNuke.Security.Roles  
Imports System.Collections.Generic  
Namespace DotNetNuke.Modules.Survey
```

```
    Partial Class Survey
```

```
        Inherits DotNetNuke.Entities.Modules.PortalModuleBase  
        Implements Entities.Modules.IActionable
```

```
        Private strCookie As String  
        Private blnSurveyExpired As Boolean  
        Private blnPersonalVoteTracking As Boolean  
        Private blnVoted As Boolean  
        Private blnPrivateResults As Boolean
```

DNN4 Module Developers Guide

```
Protected Sub Page_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Try
        strCookie = "_Module" & ModuleId.ToString & "_Survey"
        blnSurveyExpired = SurveyExpired()
        blnPersonalVoteTracking = SurveyTracking()
        blnPrivateResults = SurveyResultsType()
        blnVoted = HasVoted()

        If Not Page.IsPostBack Then
            DisplaySurveyPage()
        End If

        Catch exc As Exception      'Module failed to load
            ProcessModuleLoadException(Me, exc)
        End Try
    End Sub

#Region "Survey Closing Date"
    Private Function SurveyExpired() As Boolean
        If Not Settings("surveyclosingdate") Is Nothing Then
            Try
                If (DateDiff(DateInterval.Day, Now, CDate(Settings("surveyclosingdate"))
> -1) Then
                    Return False
                Else
                    Return True
                End If
            Catch
                Return False
            End Try
        Else
            Return False
        End If
    End Function
#End Region

#Region "Survey Tracking"
    Private Function SurveyTracking() As Boolean
        ' Check the settings to see if this module is using Personalization for vote
tracking
        If Not Settings("surveytracking") Is Nothing Then
            Return CType(Settings("surveytracking"), Boolean)
        Else
```

DNN4 Module Developers Guide

```
        Return False
    End If
End Function
#End Region

#Region "SurveyResultsType"
    Private Function SurveyResultsType() As Boolean
        'If the user is able to edit the module then they are always able to see the
        results
        If
            DotNetNuke.Security.PortalSecurity.IsInRoles(Me.ModuleConfiguration.AuthorizedEditRoles) Then
                Return False
            End If

            ' Check the settings to see if this module is using Personalization for vote
            tracking
            If Not Settings("surveyresultstype") Is Nothing Then
                Return CType(Settings("surveyresultstype"), Boolean)
            Else
                Return False
            End If
        End Function
    #End Region

    #Region "HasVoted"
        Private Function HasVoted() As Boolean
            If blnPersonalVoteTracking = True Then
                If UserId <> -1 Then
                    ' Check if the user has voted before
                    If Not
                        DotNetNuke.Services.Personalization.Personalization.GetProfile(ModuleId.ToString,
                        "Voted") Is Nothing Then
                            Return
                            CType(DotNetNuke.Services.Personalization.Personalization.GetProfile(ModuleId.ToString, "Voted"), Boolean)
                        Else
                            Return False
                        End If
                    Else
                        ' The module is using Personalization but the user is not logged in
                        Return True
                    End If
                Else
                    ' The Module is not using personalization so look for a cookie
```

DNN4 Module Developers Guide

```
        If Request.Cookies(strCookie) Is Nothing Then
            Return False
        Else
            Return True
        End If
    End If

End Function
#End Region

#Region "Display Survey Page"
Private Sub DisplaySurveyPage()
    ' Only show Survey if it is not expired
    If Not blnSurveyExpired Then
        If (blnPersonalVoteTracking) And (UserId = -1) Then
            ' The User is not logged in and the module is using personalization
            DisplaySurvey()
            cmdSubmit.Visible = False
            cmdResults.Visible = False
            Message_Label.Text = Localization.GetString("Survey_MustBeSignedIn",
Me.LocalResourceFile)
        Else
            If blnVoted = False Then
                'They haven't voted so show the Survey
                DisplaySurvey()
                ' If the results are private, hide the results link
                If blnPrivateResults Then
                    cmdResults.Visible = False
                End If
            Else
                'They have voted so display the results
                DisplayResults()
            End If
        End If
    Else
        ' Survey is expired so display the results
        DisplayResults()
    End If
End Sub
#End Region

#Region "Display Survey"
Private Sub DisplaySurvey()
    pnlResults.Visible = False
```

DNN4 Module Developers Guide

```
Dim Surveylist As List(Of SurveyInfo)
Surveylist = SurveyController.GetSurveys(ModuleId)
If Surveylist.Count > 0 Then
    IstSurvey.DataSource = Surveylist
    IstSurvey.DataBind()
    pnlSurvey.Visible = True
Else
    pnlSurvey.Visible = False
End If

End Sub
#End Region

#Region "Display Results"
Private Sub DisplayResults()

    IstResults.DataSource = SurveyController.GetSurveys(ModuleId)
    IstResults.DataBind()

    ' Hide the survey items, show the results
    pnlSurvey.Visible = False
    pnlResults.Visible = True

    ' If they can edit the module then they can always see the show survey button
    If
        (DotNetNuke.Security.PortalSecurity.IsInRoles(Me.ModuleConfiguration.AuthorizedEdit
        Roles)) And (UserId <> -1) Then
        cmdSurvey.Visible = True
    Else
        ' If the results are private, hide the results and the results link
        If blnPrivateResults Then
            IstResults.Visible = False
            cmdResults.Visible = False
            Message_Label.Text = Localization.GetString("AlreadyVoted",
            Me.LocalResourceFile)
        End If

        ' If they have voted or the Survey has expired, hide the button to show the
        Survey items
        If Not blnVoted And Not blnSurveyExpired Then
            cmdSurvey.Visible = True
        Else
            cmdSurvey.Visible = False
        End If
    End If
End Sub
```

DNN4 Module Developers Guide

```
End Sub
#End Region

#Region "Utility Functions"
Public Function FormatQuestion(ByVal strQuestion As String, ByVal ItemNumber
As Integer) As String
    Return ItemNumber.ToString & ". " & strQuestion
End Function

Private Function FormatSurveyOption(ByVal OptionName As String, ByVal
IsCorrect As Boolean) As String
    If IsCorrect Then
        If OptionName.StartsWith(">") = True And OptionName.EndsWith("<") = True
Then
            Return OptionName
        Else
            Return ">" & OptionName & "<"
        End If
    Else
        If OptionName.StartsWith(">") = True And OptionName.EndsWith("<") = True
Then
            Return OptionName.Substring(1, OptionName.Length - 2)
        Else
            Return OptionName
        End If
    End If
End Function
#End Region

#Region "Form Events"
Private Sub IstSurvey_ItemDataBound(ByVal sender As Object, ByVal e As
System.Web.UI.WebControls.DataListItemEventArgs) Handles IstSurvey.ItemDataBound

    Dim objSurvey As SurveyInfo =
SurveyController.GetSurvey(Int32.Parse(IstSurvey.DataKeys(e.Item.ItemIndex).ToStrin
g), ModuleId)
    If Not objSurvey Is Nothing Then
        Dim SurveyOptionList As New List(Of SurveyOptionInfo)
        SurveyOptionList =
SurveyOptionController.GetSurveyOptions(objSurvey.SurveyId)

        Select Case objSurvey.OptionType
            Case "R"
```

DNN4 Module Developers Guide

```
        Dim optOptions As RadioButtonList =
CType(e.Item.FindControl("optOptions"), RadioButtonList)
        optOptions.DataSource = SurveyOptionList
        optOptions.DataTextField = "OptionName"
        optOptions.DataValueField = "SurveyOptionId"
        optOptions.DataBind()
        optOptions.SelectedValue =
SurveyOptionList.Item(0).SurveyOptionId.ToString
        optOptions.Visible = True
    Case "C"
        Dim chkOptions As CheckBoxLayout =
CType(e.Item.FindControl("chkOptions"), CheckBoxLayout)
        chkOptions.DataSource = SurveyOptionList
        chkOptions.DataTextField = "OptionName"
        chkOptions.DataValueField = "SurveyOptionId"
        chkOptions.DataBind()
        chkOptions.Visible = True
    End Select
End If

End Sub

Private Sub cmdSubmit_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles cmdSubmit.Click

    Dim objSurvey As SurveyInfo
    Dim intSurvey As Integer
    Dim objSurveyOption As New SurveyOptionInfo
    Dim intSurveyOption As Integer
    Dim intQuestion As Integer
    Dim intOption As Integer
    Dim blnValid As Boolean = True

    intQuestion = -1
    Dim arrSurveys As List(Of SurveyInfo) = SurveyController.GetSurveys(ModuleId)
    For intSurvey = 0 To arrSurveys.Count - 1
        objSurvey = arrSurveys(intSurvey)
        intQuestion += 1
        Select Case objSurvey.OptionType
            Case "R"
                If Request.Form(IstSurvey.UniqueID & ":_ctl" & intQuestion.ToString &
":optOptions") = "" Then
                    blnValid = False
                End If
            Case "C"
```


DNN4 Module Developers Guide

```
        blnValid = False
        intOption = -1
        Dim arrSurveyOptions As List(Of SurveyOptionInfo) =
SurveyOptionController.GetSurveyOptions(objSurvey.SurveyId)
        For intSurveyOption = 0 To arrSurveyOptions.Count - 1
            intOption += 1
            If Request.Form(IstSurvey.UniqueID & ":_ctl" & intQuestion.ToString
& " :chkOptions:" & intOption.ToString) <> "" Then
                blnValid = True
            End If
        Next
    End Select
Next

If blnValid Then
    intQuestion = -1
    For intSurvey = 0 To arrSurveys.Count - 1
        objSurvey = CType(arrSurveys(intSurvey), SurveyInfo)
        intQuestion += 1
        Select Case objSurvey.OptionType
            Case "R"
                If Request.Form(IstSurvey.UniqueID & ":_ctl" & intQuestion.ToString
& " :optOptions") <> "" Then
                    objSurveyOption.SurveyOptionId =
Int32.Parse(Request.Form(IstSurvey.UniqueID & ":_ctl" & intQuestion.ToString &
":optOptions"))
                    SurveyOptionController.AddSurveyResult(objSurveyOption,
UserId)
                End If
            Case "C"
                intOption = -1
                Dim arrSurveyOptions As List(Of SurveyOptionInfo) =
SurveyOptionController.GetSurveyOptions(objSurvey.SurveyId)
                For intSurveyOption = 0 To arrSurveyOptions.Count - 1
                    objSurveyOption = CType(arrSurveyOptions(intSurveyOption),
SurveyOptionInfo)
                    intOption += 1
                    If Request.Form(IstSurvey.UniqueID & ":_ctl" &
intQuestion.ToString & " :chkOptions:" & intOption.ToString) <> "" Then
                        SurveyOptionController.AddSurveyResult(objSurveyOption,
UserId)
                    End If
                Next
            End Select
        Next
    End Select
Next
```

DNN4 Module Developers Guide

```
        Message_Label.Text = Localization.GetString("SurveyComplete",
Me.LocalResourceFile)

        If blnPersonalVoteTracking = True Then
            ' This means the module vote tracking is using personalization, so set the
            profile to show they have voted

            DotNetNuke.Services.Personalization.Personalization.SetProfile(ModuleId.ToString,
            "Voted", True)
                blnVoted = True
                DisplayResults()
            Else
                ' Store a cookie to show the chart after the submit
                Dim objCookie As HttpCookie = New HttpCookie("_Module" &
ModuleId.ToString & "_Survey")
                objCookie.Value = "True"
                objCookie.Expires = DateTime.MaxValue ' never expires
                Response.AppendCookie(objCookie)
                blnVoted = True
                DisplayResults()
            End If
        Else
            Message_Label.Text = Localization.GetString("SurveyIncomplete",
Me.LocalResourceFile)
        End If
    End Sub

    Private Sub cmdResults_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles cmdResults.Click
        DisplayResults()
    End Sub

    Private Sub lstResults_ItemDataBound(ByVal sender As Object, ByVal e As
System.Web.UI.WebControls.DataListItemEventArgs) Handles lstResults.ItemDataBound

        Dim objSurvey As SurveyInfo
        Dim objSurveyOption As SurveyOptionInfo
        Dim intSurveyOption As Integer
        Dim strHTML As String
        Dim intGraphWidth As Integer = 400

        If CType(Settings("surveygraphwidth"), String) <> "" Then
            intGraphWidth = Int32.Parse(CType(Settings("surveygraphwidth"), String))
        End If
```

DNN4 Module Developers Guide

```

        strHTML = "<BR><TABLE BORDER=""0"" CELLPADDING=""2"" CELLSPACING=""0""
WIDTH=""100%"">"

        objSurvey =
SurveyController.GetSurvey(Int32.Parse(lstResults.DataKeys(e.Item.ItemIndex).ToString), ModuleId)
        If Not objSurvey Is Nothing Then
            Dim arrSurveyOptions As List(Of SurveyOptionInfo) =
SurveyOptionController.GetSurveyOptions(objSurvey.SurveyId)
            For intSurveyOption = 0 To arrSurveyOptions.Count - 1
                objSurveyOption = arrSurveyOptions(intSurveyOption)

                Dim dblPercent As Double = 0
                If objSurvey.Votes <> 0 Then
                    dblPercent = Convert.ToDouble(objSurveyOption.Votes /
objSurvey.Votes)
                End If
                strHTML += "<TR>"
                strHTML += "<TD VALIGN=""top""
CLASS=""YourCompanyNameSurveyResults"">" &
FormatSurveyOption(objSurveyOption.OptionName, objSurveyOption.IsCorrect) &
"&nbsp;" & objSurveyOption.Votes.ToString & "</TD>"
                strHTML += "<TD ALIGN=""left"" VALIGN=""top"" CLASS=""Normal""
NOWRAP><IMG SRC="" & Me.TemplateSourceDirectory & "/red.gif"" WIDTH="" &
(intGraphWidth * dblPercent) & """" BORDER=""0"" HEIGHT=""15"">&nbsp;" &
CInt(dblPercent * 100).ToString & "%</TD>"
                strHTML += "</TR>"
            Next
        End If

        strHTML += "</TABLE>"

        Dim lblResults As Label = CType(e.Item.FindControl("lblResults"), Label)
        lblResults.Text = strHTML
    End Sub

    Private Sub cmdSurvey_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles cmdSurvey.Click
        DisplaySurvey()
    End Sub

#End Region

#Region "Optional Interfaces"

```

DNN4 Module Developers Guide

```
Public ReadOnly Property ModuleActions() As
DotNetNuke.Entities.Modules.Actions.ModuleActionCollection Implements
DotNetNuke.Entities.Modules.IActionable.ModuleActions
    Get
        Dim Actions As New Entities.Modules.Actions.ModuleActionCollection
        Actions.Add(GetNextActionID,
Localization.GetString(Entities.Modules.Actions.ModuleActionType.AddContent,
LocalResourceFile), Entities.Modules.Actions.ModuleActionType.AddContent, "", "",
EditUrl(), False, SecurityAccessLevel.Edit, True, False)
        Return Actions
    End Get
End Property

#End Region

End Class

End Namespace
```

Uninstall.SqlDataProvider

```
drop procedure {databaseOwner}{objectQualifier}GetSurveys  
GO
```

```
drop procedure {databaseOwner}{objectQualifier}GetSurvey  
GO
```

```
drop procedure {databaseOwner}{objectQualifier}AddSurvey  
GO
```

```
drop procedure {databaseOwner}{objectQualifier}UpdateSurvey  
GO
```

```
drop procedure {databaseOwner}{objectQualifier>DeleteSurvey  
GO
```

```
drop procedure {databaseOwner}{objectQualifier}GetSurveyOptions  
GO
```

```
drop procedure {databaseOwner}{objectQualifier}AddSurveyOption  
GO
```

```
drop procedure {databaseOwner}{objectQualifier}UpdateSurveyOption  
GO
```

```
drop procedure {databaseOwner}{objectQualifier>DeleteSurveyOption  
GO
```

```
drop procedure {databaseOwner}{objectQualifier}AddSurveyResult  
GO
```

```
ALTER TABLE {databaseOwner}{objectQualifier}Surveys  
    DROP CONSTRAINT FK_{objectQualifier}Surveys_{objectQualifier}Modules  
GO
```

```
ALTER TABLE {databaseOwner}{objectQualifier}SurveyOptions  
    DROP CONSTRAINT FK_{objectQualifier}SurveyOptions_{objectQualifier}Surveys  
GO
```

```
ALTER TABLE {databaseOwner}{objectQualifier}SurveyOptions  
    DROP CONSTRAINT  
    FK_{objectQualifier}SurveyResults_{objectQualifier}SurveyOptions  
GO
```

DNN4 Module Developers Guide

```
DROP TABLE {databaseOwner}{objectQualifier}SurveyResults  
GO
```

```
DROP TABLE {databaseOwner}{objectQualifier}SurveyOptions  
GO
```

```
DROP TABLE {databaseOwner}{objectQualifier}Surveys  
GO
```

DNN4 Module Developers Guide

EditSurvey.ascx.resx

	Name	Value	Comment
	chkCorrect.Text	Option Is Correct Answer?	
	cmdAddOption.Text	Add Option	
	cmdDeleteOption.Text	Delete Option	
	cmdDown.Text	Move Option Down	
	cmdUp.Text	Move Option Up	
	ModuleHelp.Text	<h1>About the Survey Module</h1><p>The Survey Module generates an easy-to-complete survey to which website visitors can add their input. Survey results are rendered as a bar chart and include the number of individual responses.</p>	
	MultipleSelection.Text	Multiple Selection	
▶	pOption.Help	Enter a possible Answer for the Question. If it is the correct Answer, select the checkbox. Then select Add Option.	
	pOption.Text	New Option:	
	pOptions.Help	The options available to the user	
	pOptions.Text	Options:	
	pOptionType.Help	Select the type of Survey	
	pOptionType.Text	Type:	
	pQuestion.Help	Enter the question for your Survey	
	pQuestion.Text	Question:	
	pViewOrder.Help	Enter the View Order for this question	
	pViewOrder.Text	View Order:	
	SingleSelection.Text	Single Selection	
*			

Settings.ascx.resx

	Name	Value	Comment
▶	ControlTitle_settings.Text	Survey Settings	
	ModuleHelp.Text	Settings Help	
	No.Text	Vote tracking via cookie	
	plClosingDate.Help	Enter the closing date for the survey. You may use the pop-up calendar to choose the date, by clicking on the Calendar Link.	
	plClosingDate.Text	Survey Closing Date:	
	plGraphWidth.Help	Enter the Maximum width for the results graph	
	plGraphWidth.Text	Maximum Bar Graph Width:	
	plPersonal.Help	Choose how to track votes	
	plPersonal.Text	Vote Tracking:	
	plSurveyResults.Help	Choose whether the results are made public or are only available to Administrators	
	plSurveyResults.Text	Survey Results:	
	PrivateResults.Text	Private	
	PublicResults.Text	Public	
	valClosingDate.ErrorMessage	 Invalid Closing Date!	
	Yes.Text	1 Vote/Registered User	
*			

Survey.ascx.resx

	Name	Value	Comment
►	AddContent.Action	Add Question	
	AlreadyVoted.Text	You have already participated	
	cmdResults.Text	View Results	
	cmdSubmit.Text	Submit Survey	
	cmdSurvey.Text	View Survey	
	ModuleHelp.Text	<h1>About the Survey Module</h1><p>The Survey Module generates an easy-to-complete survey to which website visitors can add their input. Survey results are rendered as a bar chart and include the number of individual responses.</p>	
	OptionExists.Text	Option Already Exists.	
	Survey_MustBeSignedIn.Text	You must sign in to vote in this survey.	
	SurveyComplete.Text	Thank You for Participating	
	SurveyIncomplete.Text	The survey is incomplete. Please complete and resubmit.	
*			

03.01.00.SqlDataProvider

```

/*****
/*****          SqlDataProvider          *****/
/*****
/*****
/***** Note: To manually execute this script you must *****/
/*****          perform a search and replace operation *****/
/*****          for {databaseOwner} and {objectQualifier} *****/
/*****
/*****
/*****/

if exists (select * from dbo.sysobjects where id =
object_id(N'{databaseOwner}{objectQualifier}GetSurveys') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure {databaseOwner}{objectQualifier}GetSurveys
GO

if exists (select * from dbo.sysobjects where id =
object_id(N'{databaseOwner}{objectQualifier}GetSurvey') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure {databaseOwner}{objectQualifier}GetSurvey
GO

if exists (select * from dbo.sysobjects where id =
object_id(N'{databaseOwner}{objectQualifier}AddSurvey') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure {databaseOwner}{objectQualifier}AddSurvey
GO

if exists (select * from dbo.sysobjects where id =
object_id(N'{databaseOwner}{objectQualifier}UpdateSurvey') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure {databaseOwner}{objectQualifier}UpdateSurvey
GO

if exists (select * from dbo.sysobjects where id =
object_id(N'{databaseOwner}{objectQualifier}DeleteSurvey') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure {databaseOwner}{objectQualifier}DeleteSurvey
GO

if exists (select * from dbo.sysobjects where id =
object_id(N'{databaseOwner}{objectQualifier}GetSurveyOptions') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure {databaseOwner}{objectQualifier}GetSurveyOptions
GO

if exists (select * from dbo.sysobjects where id =
object_id(N'{databaseOwner}{objectQualifier}AddSurveyOption') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)

```

DNN4 Module Developers Guide

```
drop procedure {databaseOwner}{objectQualifier}AddSurveyOption
GO

if exists (select * from dbo.sysobjects where id =
object_id(N'{databaseOwner}{objectQualifier}UpdateSurveyOption') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure {databaseOwner}{objectQualifier}UpdateSurveyOption
GO

if exists (select * from dbo.sysobjects where id =
object_id(N'{databaseOwner}{objectQualifier>DeleteSurveyOption') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure {databaseOwner}{objectQualifier>DeleteSurveyOption
GO

if exists (select * from dbo.sysobjects where id =
object_id(N'{databaseOwner}{objectQualifier}AddSurveyResult') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure {databaseOwner}{objectQualifier}AddSurveyResult
GO

if not exists (select * from dbo.sysobjects where id =
object_id(N'{databaseOwner}{objectQualifier}SurveyOptions') and
OBJECTPROPERTY(id, N'IsTable') = 1)
CREATE TABLE {databaseOwner}{objectQualifier}SurveyOptions (
    [SurveyOptionID] [int] IDENTITY (1, 1) NOT NULL ,
    [SurveyID] [int] NOT NULL ,
    [ViewOrder] [int] NOT NULL ,
    [OptionName] [nvarchar] (500) NOT NULL ,
    [Votes] [int] NOT NULL
) ON [PRIMARY]

GO

if not exists (select * from dbo.sysobjects where id =
object_id(N'{databaseOwner}{objectQualifier}Surveys') and
OBJECTPROPERTY(id, N'IsTable') = 1)
CREATE TABLE {databaseOwner}{objectQualifier}Surveys (
    [SurveyID] [int] IDENTITY (1, 1) NOT NULL ,
    [ModuleID] [int] NOT NULL ,
    [Question] [nvarchar] (500) NOT NULL ,
    [ViewOrder] [int] NULL ,
    [OptionType] [char] (1) NOT NULL ,
    [CreatedByUser] [nvarchar] (100) NOT NULL ,
    [CreateDate] [datetime] NOT NULL
) ON [PRIMARY]

GO
```

DNN4 Module Developers Guide

```
if not exists (select * from dbo.sysobjects where id =
object_id(N'PK_{objectQualifier}Surveys') and OBJECTPROPERTY(id,
N'IsPrimaryKey') = 1)
ALTER TABLE {databaseOwner}{objectQualifier}Surveys ADD CONSTRAINT
    PK_{objectQualifier}Surveys PRIMARY KEY CLUSTERED
    (
        SurveyID
    ) ON [PRIMARY]
```

GO

```
if not exists (select * from dbo.sysobjects where id =
object_id(N'PK_{objectQualifier}SurveyOptions') and OBJECTPROPERTY(id,
N'IsPrimaryKey') = 1)
ALTER TABLE {databaseOwner}{objectQualifier}SurveyOptions ADD
CONSTRAINT
    PK_{objectQualifier}SurveyOptions PRIMARY KEY CLUSTERED
    (
        SurveyOptionID
    ) ON [PRIMARY]
```

GO

```
if not exists (select * from dbo.sysobjects where id =
object_id(N'FK_{objectQualifier}SurveyOptions_{objectQualifier}Surveys'
) and OBJECTPROPERTY(id, N'IsForeignKey') = 1)
ALTER TABLE {databaseOwner}{objectQualifier}SurveyOptions ADD
CONSTRAINT
    FK_{objectQualifier}SurveyOptions_{objectQualifier}Surveys
FOREIGN KEY
    (
        SurveyID
    ) REFERENCES {databaseOwner}{objectQualifier}Surveys
    (
        SurveyID
    ) ON DELETE CASCADE
    NOT FOR REPLICATION
```

GO

```
if not exists (select * from dbo.sysobjects where id =
object_id(N'FK_{objectQualifier}Surveys_{objectQualifier}Modules') and
OBJECTPROPERTY(id, N'IsForeignKey') = 1)
ALTER TABLE {databaseOwner}{objectQualifier}Surveys ADD CONSTRAINT
    FK_{objectQualifier}Surveys_{objectQualifier}Modules FOREIGN KEY
    (
        ModuleID
    ) REFERENCES {databaseOwner}{objectQualifier}Modules
    (
        ModuleID
    ) ON DELETE CASCADE
```

DNN4 Module Developers Guide

NOT FOR REPLICATION

GO

```
create procedure {databaseOwner}{objectQualifier}GetSurveys
```

```
@ModuleID int
```

```
as
```

```
select SurveyID,
       Question,
       ViewOrder,
       OptionType,
       CreatedByUser,
       CreatedDate
from   {objectQualifier}Surveys
where  ModuleID = @ModuleID
order by ViewOrder
```

GO

```
create procedure {databaseOwner}{objectQualifier}GetSurvey
```

```
@SurveyID int,
```

```
@ModuleID int
```

```
as
```

```
select SurveyID,
       ModuleID,
       Question,
       ViewOrder,
       OptionType,
       'CreatedByUser' = {objectQualifier}Users.FirstName + ' ' +
{objectQualifier}Users.LastName,
       CreatedDate,
       'Votes' = ( select sum(Votes) from
{objectQualifier}SurveyOptions where
{objectQualifier}SurveyOptions.SurveyID =
{objectQualifier}Surveys.SurveyID )
from   {objectQualifier}Surveys
left outer join {objectQualifier}Users on
{objectQualifier}Surveys.CreatedByUser = {objectQualifier}Users.UserID
where  SurveyID = @SurveyID
and    ModuleID = @ModuleID
```

GO

```
create procedure {databaseOwner}{objectQualifier}AddSurvey
```

DNN4 Module Developers Guide

```
@ModuleID      int,
@Question      nvarchar(500),
@ViewOrder     int,
@OptionType    char(1),
@UserName      nvarchar(100)

as

insert into {objectQualifier}Surveys (
    ModuleID,
    Question,
    ViewOrder,
    OptionType,
    CreatedByUser,
    CreatedDate
)
values (
    @ModuleID,
    @Question,
    @ViewOrder,
    @OptionType,
    @userName,
    getdate()
)

select SCOPE_IDENTITY()

GO

create procedure {databaseOwner}{objectQualifier}UpdateSurvey

@SurveyID      int,
@Question      nvarchar(500),
@ViewOrder     int,
@OptionType    char(1),
@UserName      nvarchar(100)

as

update {objectQualifier}Surveys
set     Question = @Question,
        ViewOrder = @ViewOrder,
        OptionType = @OptionType,
        CreatedByUser = @UserName,
        CreatedDate = getdate()
where   SurveyID = @SurveyID

GO

create procedure {databaseOwner}{objectQualifier}DeleteSurvey
```

DNN4 Module Developers Guide

```
@SurveyID int

as

delete
from {objectQualifier}Surveys
where SurveyID = @SurveyID

GO

create procedure {databaseOwner}{objectQualifier}GetSurveyOptions

@SurveyID int

as

select SurveyOptionID,
       ViewOrder,
       OptionName,
       Votes
from {objectQualifier}SurveyOptions
where SurveyID = @SurveyID
order by ViewOrder

GO

create procedure {databaseOwner}{objectQualifier}AddSurveyOption

@SurveyID int,
@OptionName nvarchar(500),
@ViewOrder int

as

insert into {objectQualifier}SurveyOptions (
    SurveyID,
    OptionName,
    ViewOrder,
    Votes
)
values (
    @SurveyID,
    @OptionName,
    @ViewOrder,
    0
)

select SCOPE_IDENTITY()

GO
```

DNN4 Module Developers Guide

```
create procedure {databaseOwner}{objectQualifier}UpdateSurveyOption

@SurveyOptionID int,
@OptionName      nvarchar(500),
@ViewOrder       int

as

update {objectQualifier}SurveyOptions
set    OptionName = @OptionName,
       ViewOrder  = @ViewOrder
where  SurveyOptionID = @SurveyOptionID

GO

create procedure {databaseOwner}{objectQualifier}DeleteSurveyOption

@SurveyOptionID int

as

delete
from  {objectQualifier}SurveyOptions
where SurveyOptionID = @SurveyOptionID

GO

create procedure {databaseOwner}{objectQualifier}AddSurveyResult

@SurveyOptionID int

as

update {objectQualifier}SurveyOptions
set    Votes = Votes + 1
where  SurveyOptionID = @SurveyOptionID

GO

/*****
/*****      SqlDataProvider      *****/
/*****/
```


03.03.00.SqlDataProvider

```

/*****
/*****          SqlDataProvider          *****/
/*****          *****/
/*****          *****/
/***** Note: To manually execute this script you must *****/
/*****          perform a search and replace operation *****/
/*****          for {databaseOwner} and {objectQualifier} *****/
/*****          *****/
/*****          *****/
/*****          *****/

/** Change CreatedByUser column to an int for performance **/

ALTER TABLE {databaseOwner}{objectQualifier}Surveys ADD
    CreatedByUserID int NULL
GO

update {databaseOwner}{objectQualifier}Surveys
set    CreatedByUserID = convert(int,CreatedByUser)
GO

ALTER TABLE {databaseOwner}{objectQualifier}Surveys
    DROP COLUMN CreatedByUser
GO

ALTER TABLE {databaseOwner}{objectQualifier}Surveys ADD
    CreatedByUser int NOT NULL CONSTRAINT
DF_{objectQualifier}Surveys_CreatedByUser DEFAULT 0
GO

update {databaseOwner}{objectQualifier}Surveys
set    CreatedByUser = CreatedByUserID
GO

ALTER TABLE {databaseOwner}{objectQualifier}Surveys
    DROP COLUMN CreatedByUserID
GO

ALTER TABLE {databaseOwner}{objectQualifier}Surveys
    DROP CONSTRAINT DF_{objectQualifier}Surveys_CreatedByUser
GO

/** Drop Existing Stored Procedures **/

if exists (select * from dbo.sysobjects where id =
object_id(N'{databaseOwner} [{objectQualifier}AddSurvey]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
    drop procedure {databaseOwner}{objectQualifier}AddSurvey
GO
```

DNN4 Module Developers Guide

```
if exists (select * from {databaseOwner}sysobjects where id =
object_id(N'{databaseOwner} [{objectQualifier}GetSurvey]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
    drop procedure {databaseOwner}{objectQualifier}GetSurvey
GO

if exists (select * from dbo.sysobjects where id =
object_id(N'{databaseOwner} [{objectQualifier}UpdateSurvey]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
    drop procedure {databaseOwner}{objectQualifier}UpdateSurvey
GO

/** Create Optimized Stored Procedures **/

create procedure {databaseOwner}{objectQualifier}AddSurvey

@ModuleID          int,
@Question           nvarchar(500),
@ViewOrder          int,
@OptionType         char(1),
@UserID             int

as

insert into {objectQualifier}Surveys (
    ModuleID,
    Question,
    ViewOrder,
    OptionType,
    CreatedByUser,
    CreatedDate
)
values (
    @ModuleID,
    @Question,
    @ViewOrder,
    @OptionType,
    @UserID,
    getdate()
)

select SCOPE_IDENTITY()

GO

create procedure {databaseOwner}{objectQualifier}GetSurvey

@SurveyID int,
@ModuleID int

as
```

DNN4 Module Developers Guide

```
select SurveyID,
       ModuleID,
       Question,
       ViewOrder,
       OptionType,
       CreatedByUser,
       CreatedDate,
       'Votes' = ( select sum(Votes) from
 {objectQualifier}SurveyOptions where
 {objectQualifier}SurveyOptions.SurveyID =
 {objectQualifier}Surveys.SurveyID )
from   {objectQualifier}Surveys
where  SurveyID = @SurveyID
and    ModuleID = @ModuleID

GO

create procedure {databaseOwner}{objectQualifier}UpdateSurvey

@SurveyID          int,
@Question          nvarchar(500),
@ViewOrder         int,
@OptionType        char(1),
@UserID            int

as

update {objectQualifier}Surveys
set    Question = @Question,
       ViewOrder = @ViewOrder,
       OptionType = @OptionType,
       CreatedByUser = @UserID,
       CreatedDate = getdate()
where  SurveyID = @SurveyID

GO

ALTER TABLE {databaseOwner}{objectQualifier}SurveyOptions ADD
       IsCorrect bit NOT NULL CONSTRAINT
DF_{objectQualifier}SurveyOptions_IsCorrect DEFAULT 0
GO

if exists (select * from dbo.sysobjects where id =
object_id(N'{databaseOwner}{objectQualifier}GetSurveyOptions') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure {databaseOwner}{objectQualifier}GetSurveyOptions
GO
```

DNN4 Module Developers Guide

```
if exists (select * from dbo.sysobjects where id =
object_id(N'{databaseOwner}{objectQualifier}AddSurveyOption') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure {databaseOwner}{objectQualifier}AddSurveyOption
GO

if exists (select * from dbo.sysobjects where id =
object_id(N'{databaseOwner}{objectQualifier}UpdateSurveyOption') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure {databaseOwner}{objectQualifier}UpdateSurveyOption
GO

create procedure {databaseOwner}{objectQualifier}GetSurveyOptions

@SurveyID int

as

select SurveyOptionID,
       ViewOrder,
       OptionName,
       Votes,
       IsCorrect
from   {objectQualifier}SurveyOptions
where  SurveyID = @SurveyID
order by ViewOrder

GO

create procedure {databaseOwner}{objectQualifier}AddSurveyOption

@SurveyID      int,
@OptionName    nvarchar(500),
@ViewOrder     int,
@IsCorrect     bit

as

insert into {objectQualifier}SurveyOptions (
    SurveyID,
    OptionName,
    ViewOrder,
    Votes,
    IsCorrect
)
values (
    @SurveyID,
    @OptionName,
    @ViewOrder,
    0,
    @IsCorrect
)
```

DNN4 Module Developers Guide

```
)

select SCOPE_IDENTITY()

GO

create procedure {databaseOwner}{objectQualifier}UpdateSurveyOption

@SurveyOptionID int,
@OptionName      nvarchar(500),
@ViewOrder      int,
@IsCorrect       bit

as

update {objectQualifier}SurveyOptions
set    OptionName = @OptionName,
        ViewOrder = @ViewOrder,
        IsCorrect = @IsCorrect
where  SurveyOptionID = @SurveyOptionID

GO

if exists (select * from dbo.sysobjects where id =
object_id(N'{databaseOwner}{objectQualifier}AddSurveyResult') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure {databaseOwner}{objectQualifier}AddSurveyResult
GO

if not exists (select * from dbo.sysobjects where id =
object_id(N'{databaseOwner}{objectQualifier}SurveyResults') and
OBJECTPROPERTY(id, N'IsTable') = 1)
CREATE TABLE {databaseOwner}{objectQualifier}SurveyResults (
    [SurveyResultID] [int] IDENTITY (1, 1) NOT NULL ,
    [SurveyOptionID] [int] NOT NULL ,
    [UserID] [int] NOT NULL
) ON [PRIMARY]

GO

if not exists (select * from dbo.sysobjects where id =
object_id(N'PK_{objectQualifier}SurveyResults') and OBJECTPROPERTY(id,
N'IsPrimaryKey') = 1)
ALTER TABLE {databaseOwner}{objectQualifier}SurveyResults ADD
CONSTRAINT
    PK_{objectQualifier}SurveyResults PRIMARY KEY CLUSTERED
(
    SurveyResultID
) ON [PRIMARY]
```

DNN4 Module Developers Guide

GO

```
if not exists (select * from dbo.sysobjects where id =
object_id(N'FK_{objectQualifier}SurveyResults_{objectQualifier}SurveyOptions') and OBJECTPROPERTY(id, N'IsForeignKey') = 1)
ALTER TABLE {databaseOwner}{objectQualifier}SurveyResults ADD
CONSTRAINT
    FK_{objectQualifier}SurveyResults_{objectQualifier}SurveyOptions
FOREIGN KEY
    (
        SurveyOptionID
    ) REFERENCES {databaseOwner}{objectQualifier}SurveyOptions
    (
        SurveyOptionID
    ) ON DELETE CASCADE
    NOT FOR REPLICATION
```

GO

```
create procedure {databaseOwner}{objectQualifier}AddSurveyResult

@SurveyOptionID int,
@UserID          int

as

update {objectQualifier}SurveyOptions
set    Votes = Votes + 1
where  SurveyOptionID = @SurveyOptionID

insert into {objectQualifier}SurveyResults ( SurveyOptionID, UserID )
values ( @SurveyOptionID, @UserID )
```

GO

```
/*
*****
*****          SqlDataProvider          *****
*****
*/
```

04.00.10.SqlDataProvider

```

/*****
/*****          SqlDataProvider          *****/
/*****
/*****          *****/
/***** Note: To manually execute this script you must *****/
/*****          perform a search and replace operation *****/
/*****          for {databaseOwner} and {objectQualifier} *****/
/*****          *****/
/*****
/*****

if exists (select * from dbo.sysobjects where id =
object_id(N'{databaseOwner} [{objectQualifier}]UpdateSurvey') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
    drop procedure {databaseOwner}{objectQualifier}UpdateSurvey
GO

if exists (select * from dbo.sysobjects where id =
object_id(N'{databaseOwner} [{objectQualifier}]DeleteSurvey') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
    drop procedure {databaseOwner}{objectQualifier}DeleteSurvey
GO

/** Create Stored Procedures **/

create procedure {databaseOwner}{objectQualifier}UpdateSurvey

@SurveyID          int,
@Question           nvarchar(500),
@ViewOrder         int,
@OptionType        char(1),
@UserID            int,
@ModuleID int

as

update {objectQualifier}Surveys
set    Question = @Question,
        ViewOrder = @ViewOrder,
        OptionType = @OptionType,
        CreatedByUser = @UserID,
        CreatedDate = getdate()
where  SurveyID = @SurveyID and ModuleID = @ModuleID

GO

create procedure {databaseOwner}{objectQualifier}DeleteSurvey

@SurveyID int,
@ModuleID int
```

DNN4 Module Developers Guide

as

```
delete
from   {objectQualifier}Surveys
where  SurveyID = @SurveyID and ModuleID = @ModuleID
```

```
/*****
/*****          SqlDataProvider          *****/
/*****/
```


Additional Information

The DotNetNuke Portal Application Framework is constantly being revised and improved. To ensure that you have the most recent version of the software and this document, please visit the DotNetNuke website at:

<http://www.dotnetnuke.com>

The following additional websites provide helpful information about technologies and concepts related to DotNetNuke:

DotNetNuke Community Forums

<http://forums.asp.net/90/ShowForum.aspx>

Microsoft® ASP.Net

<http://www.asp.net>

Open Source

<http://www.opensource.org/>

W3C Cascading Style Sheets, level 1

<http://www.w3.org/TR/CSS1>

Errors and Omissions

If you discover any errors or omissions in this document, please email marketing@dotnetnuke.com. Please provide the title of the document, the page number of the error and the corrected content along with any additional information that will help us in correcting the error.

Appendix A: Document History

Version	Last Update	Author(s)	Changes
1.0.0	Nov 1, 2006	Michael Washington	First Draft