



PNMsoft Knowledge Base

Sequence Administrator Guides

Extending Form Templates

Oct. 2013

Product Version 7.5 and above

© 2013 PNMsoft All Rights Reserved

This document, including any supporting materials, is owned by PNMsoft Ltd and/or its affiliates and is for the sole use of the PNMsoft customers, PNMsoft official business partners, or other authorized recipients. This document may contain information that is confidential, proprietary or otherwise legally protected, and it may not be further copied, distributed or publicly displayed without the express written permission of PNMsoft Ltd. or its affiliates.

PNMsoft UK 38 Clarendon Road Watford Hertfordshire WD17 1JJ

Tel: +44(0)192 381 3420 • Email: info@pnmsoft.com • Website: www.pnmsoft.com

Microsoft Partner

Gold Application Development

TABLE OF CONTENTS

General Document Information	1
Purpose.....	1
Prerequisites.....	1
Resources	1
Introduction	2
Creating New Templates	3
Understanding the Elements of Form Templates	4
TemplateInfo.config	5
Content.ascx.sqtemplate	6
Shared Files.....	7
OneTableDataModel.sqddl.sqtemplate	8
OneTableDataModel.sqeml.sqtemplate	9
An Example: Creating a Simple Template	10
Limitations and Best Practices	16

General Document Information

Purpose

The purpose of this document is to enable Sequence Administrators and Developers to create and save new Form templates.

This guide will explain the concepts behind creating a new Form template, and show a step by step example of how to do so.

Prerequisites

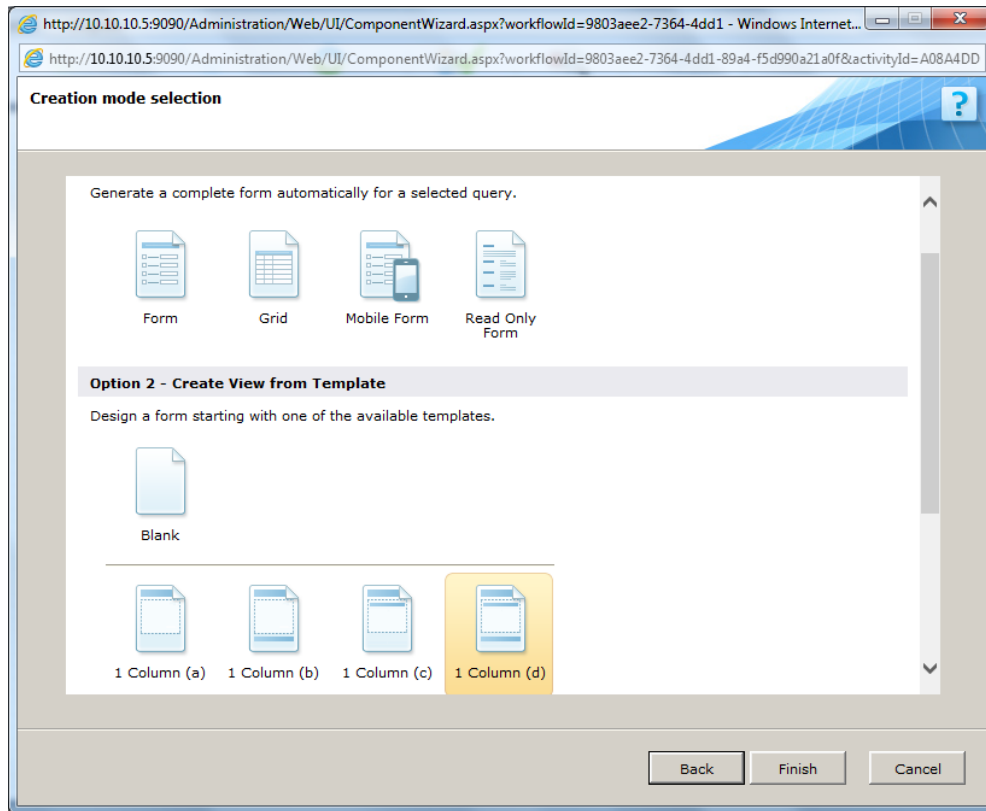
- Knowledge of Sequence.
- Familiarity with Forms, Form markup, Form Data Model and Views.

Resources

See the [UX Studio page](#) on our Knowledge Centre to attain familiarity with the above concepts.

Introduction

Sequence enables you to create Forms quickly, offering a variety of useful Form templates which you can choose from in the *Creation Mode Selection* screen:



Form Creation Mode Selection

Creating a form using a template can save you time styling and designing the form, or fitting it for a particular type of device (e.g. Mobile). Here is a form created using the template selected in the screen above:

Request Account

Form description...

Section Header	
Section description...	
Label	<input type="text"/>
Label	
Label	
Label	

* Mandatory Fields

Submit

Form Created Using a Template

Creating New Templates

But what if your team requires a form template which is not offered out-of-the-box?

Sequence enables you to create and save such templates easily. Once you've created and saved a new template, you can add it to the list of Templates in the *Creation Mode Selection* screen.

This guide will demonstrate how to do this.

Understanding the Elements of Form Templates

Form templates are stored in the *C:\Program Files\PNMsoft\Shared Resources\Components\Forms\ControlTemplates\ViewTemplates* folder. This folder contains both **System** and **Custom** templates:

Program Files > PNMsoft > Shared Resources > Components > Forms > ControlTemplates > ViewTemplates		
Include in library Share with New folder		
Name ^	Date modified	Type
Custom	9/24/2013 2:14 PM	File folder
System	9/24/2013 2:14 PM	File folder

System templates are those which are available out-of-the-box, and Custom templates are those which you create for your specific needs.

Here are some of the Templates in the System folder:

PNMsoft > Shared Resources > Components > Forms > ControlTemplates > ViewTemplates > System		
Include in library Share with New folder		
Name ^	Date modified	Type
_Shared	9/24/2013 2:14 PM	File folder
Blank	9/24/2013 2:14 PM	File folder
Form	9/24/2013 2:14 PM	File folder
Grid	9/24/2013 2:14 PM	File folder
MobileForm	9/24/2013 2:14 PM	File folder
ReadOnlyForm	9/24/2013 2:14 PM	File folder
Section-1C-H	9/24/2013 2:14 PM	File folder
Section-1C-H-F	9/24/2013 2:14 PM	File folder
Section-1C-H-S	9/24/2013 2:14 PM	File folder

Each Form Template is defined in a folder containing the following files:

PNMsoft > Shared Resources > Components > Forms > ControlTemplates > ViewTemplates > System > S		
Include in library Share with New folder		
Name ^	Date modified	Type
Content.ascx.sqtemplate	9/2/2013 2:34 PM	SQTEMPLATE File
Section-1C-H-S.png	8/14/2013 9:36 AM	PNG image
TemplateInfo.config	8/14/2013 9:36 AM	CONFIG File

TemplateInfo.config

This file is the configuration file of the template which includes the paths to all the files that define the template's properties. This is its format:

```
<?xml version="1.0" encoding="utf-8"?>
<FormTemplate xmlns="http://schemas.pnmsoft.com/sequence/2013/07/ux/formtemplates">
  <TemplateData>
    <ID>b1d38f8a-4155-4b4a-b996-e6dc12a4c705</ID>
    <Name>Section-1C-H-S</Name>
    <DisplayName>1 Column (c)</DisplayName>
    <Description>One Column, Header, Sub-Header</Description>
    <LargelconPath>Section-1C-H-S.png</LargelconPath>
    <ContentTemplatePath>Content.ascx.sqtemplate</ContentTemplatePath>
    <DataModelPath>..\_Shared\OneTableDataModel.sqddl.sqtemplate</DataModelPath>
    <EntityModelPath>..\_Shared\OneTableDataModel.sqeml.sqtemplate</EntityModelPath>
    <Sequence>23</Sequence>
  </TemplateData>
</FormTemplate>
```

The fields above define the template as follows:

- **ID:** The GUID of the template.
- **Name:** the Name of the template in Sequence.
- **DisplayName:** the name of the Template as displayed in the Creation Mode Selection screen.
- **Description:** the tooltip for the Template in the Creation Mode Selection screen.
- **LargelconPath:** the path to the image of the template as displayed in the Creation Mode Selection screen.
- **ContentTemplatePath:** the path to a file containing the markup of the Form view which is generated when the user selects to create a form using this Template.
- **DataModelPath:** the path to a file containing the physical data model of the form. (Not included in the Blank template.)
- **EntityModelPath:** the path to a file containing the logical data model of the form. (Not included in the Blank template.)
- **Sequence:** a sequential number defining the order that the Template is displayed among the other templates. Generally, you should place custom templates last (e.g. 1000).

Note: all paths above are relative to the template folder.

Content.ascx.sqtemplate

This file is the markup of the Form view which is generated when the user selects to create a form using a Template.

```
<%@ Control %>

<sq:Form runat="server" ID="Form1" DataSourceID="Form1DataSource" Height="0px"
DataKeyNames="fldId" Width="100%" CellPadding="0" CellSpacing="0">
<ContentTemplate>
<table class="sqf-main-table" cellpadding="0" cellspacing="0">
<tr>
<td class="sqf-header">
<h1><sq:Text runat="server">$formname$</sq:Text></h1>
<sq:Text runat="server">Form description...</sq:Text>
</td>
</tr>
<tr>
<td>
<table class="sqf-section-table" cellpadding="0" cellspacing="0">
<tr>
<td class="sqf-section-label-col">
<sq:Label runat="server" Text="Label" />
</td>
<td>
<sq:TextBox runat="server" ID="TextBox1" />
</td>
</tr>
<tr>
<td>
<sq:Label runat="server" Text="Label" />
</td>
<td>&nbsp;</td>
</tr>
<tr>
<td>
<sq:Label runat="server" Text="Label" />
</td>
<td>&nbsp;</td>
</tr>
<tr>
<td>
<sq:Label runat="server" Text="Label" />
</td>
<td>&nbsp;</td>
</tr>
<tr>
<td>
</td>
</tr>
</table>
</td>
</tr>
</table>
```

```

        <sq:Label runat="server" Text="Label" />
    </td>
    <td>&nbsp;</td>
</tr>
</table>
</td>
</tr>
</table>
</ContentTemplate>
</sq:Form>
<sq:DataSource runat="server" ID="Form1DataSource"
QueryName="$queryname$"></sq:DataSource>
<sq:BoundControl runat="server" TargetControlID="Form1"></sq:BoundControl>

```

When you create a new Template, you customise this markup according to your needs (see the example below).

The Content.ascx.sqtemplate file includes fields such as \$formname\$ which are given actual values when the form is generated. These fields are:

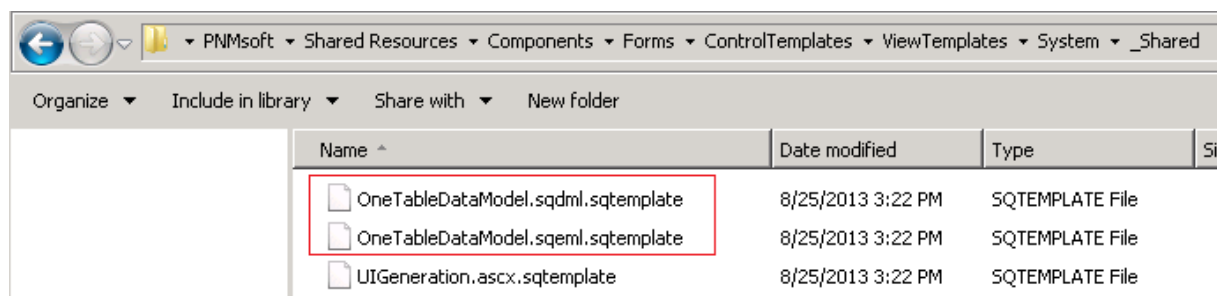
- **formname**: The name of the Form.
- **viewname**: The name of the Form view.
- **queryname**: The name of the query of the Form control.
- **A set of 10 GUIDs: guid1, guid2,** These GUIDs are given to data source elements such as UACT tables.
- **A new GUID: newguid**: This field enables the creation of a new GUID if necessary.

Shared Files

The files:

- OneTableDataModel.sqddl.sqtemplate
- OneTableDataModel.sqeml.sqtemplate

are shared for all System templates:



OneTableDataModel.sqddl.sqtemplate

This file contains an xml representation of the physical data model of the Form, as generated by the Template:

```
<?xml version="1.0" encoding="utf-8" ?>
<DataModel xmlns="http://pmnsoft.com/sequence/2008/03/metadata"
type="PNMsoft.Sequence.Data.EntityDataModel, PNMsoft.Sequence.Extensions, Version=7.0.0.0,
Culture=neutral, PublicKeyToken=0a1a1b90c1c5dca1">
  <DataSources type="PNMsoft.Sequence.Data.DataSourceDefinitionCollection,
PNMsoft.Sequence, Version=7.0.0.0, Culture=neutral, PublicKeyToken=0a1a1b90c1c5dca1">
    <LinqDataSourceDefinition type="PNMsoft.Sequence.Linq.LinqDataSourceDefinition,
PNMsoft.Sequence.Extensions, Version=7.0.0.0, Culture=neutral,
PublicKeyToken=0a1a1b90c1c5dca1">
      <Id type="System.Guid">$guid1$</Id>
      <KeyFields type="System.String[], mscorlib, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089" length="1">
        <Item>fldlActId</Item>
      </KeyFields>
      <KeyValues type="PNMsoft.Sequence.Expressions.SqExpression[], PNMsoft.Sequence,
Version=7.0.0.0, Culture=neutral, PublicKeyToken=0a1a1b90c1c5dca1" length="1">
        <Item type="PNMsoft.Sequence.Expressions.UnParsedExpression, PNMsoft.Sequence,
Version=7.0.0.0, Culture=neutral, PublicKeyToken=0a1a1b90c1c5dca1">
          <Body type="System.String" />
          <Type type="PNMsoft.Sequence.Expressions.SqExpressionFormat, PNMsoft.Sequence,
Version=7.0.0.0, Culture=neutral, PublicKeyToken=0a1a1b90c1c5dca1">Object</Type>
        </Item>
      </KeyValues>
      <Name type="System.String">$queryname$</Name>
      <SchemaName type="System.String">dbo</SchemaName>
      <SecondaryKeyFields type="System.String[], mscorlib, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089" length="1">
        <Item />
      </SecondaryKeyFields>
      <SecondaryKeyValues type="PNMsoft.Sequence.Expressions.SqExpression[],
PNMsoft.Sequence, Version=7.0.0.0, Culture=neutral, PublicKeyToken=0a1a1b90c1c5dca1"
length="1">
        <Item type="PNMsoft.Sequence.Expressions.UnParsedExpression, PNMsoft.Sequence,
Version=7.0.0.0, Culture=neutral, PublicKeyToken=0a1a1b90c1c5dca1">
          <Body type="System.String" />
          <Type type="PNMsoft.Sequence.Expressions.SqExpressionFormat, PNMsoft.Sequence,
Version=7.0.0.0, Culture=neutral, PublicKeyToken=0a1a1b90c1c5dca1">Object</Type>
        </Item>
      </SecondaryKeyValues>
      <TableName type="System.String">UACT$guid1$</TableName>
```

```
<TableType type="PNMsoft.Sequence.Linq.LinqDataSourceDefinitionType,
PNMsoft.Sequence.Extensions, Version=7.0.0.0, Culture=neutral,
PublicKeyToken=0a1a1b90c1c5dca1">Table</TableType>

  <UseDefaultConnection type="System.Boolean">true</UseDefaultConnection>

</LinqDataSourceDefinition>

</DataSources>

</DataModel>
```

OneTableDataModel.sqeml.sqtemplate

This file contains an xml representation of the logical data model of the Form, as generated by the Template:

```
<?xml version="1.0" encoding="utf-8"?>

<EntityModelFragment Version="1.0"
xmlns="http://pmnsoft.com/sequence/2010/11/data/modeling">

  <EntityContainer Namespace="ref:Self">

    <EntityQuery Name="$queryname$" ElementType="$queryname$" DataSource="$queryname$"
/>

    <EntityDataSource Name="$queryname$" />

    <EntityType Name="$queryname$" DataSource="$queryname$">

      <EntityProperty Name="fldId" IsPrimaryKey="true" IsIdentity="true" Nullable="false"
TypeKind="Int32" />

      <EntityProperty Name="fldWfld" IsSystem="true" Nullable="false" TypeKind="Int32" />

      <EntityProperty Name="fldActId" IsSystem="true" Nullable="false" TypeKind="Int32" />

      <EntityProperty Name="fldAllId" IsSystem="true" Nullable="true" TypeKind="Int32" />

    </EntityType>

  </EntityContainer>

</EntityModelFragment>
```

This file contains the query, including its default data fields.

In order to create a custom Template (as we will see in the example below), you copy these files and modify them, creating a customized Data Model.

An Example: Creating a Simple Template

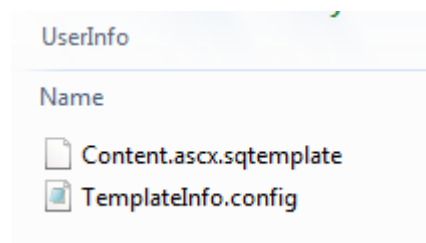
The following examples demonstrates how to create a simple Template called **UserInfo**: a one column Form with two fields: First Name and Last Name.

Request Service	
Form description...	
First Name	<input type="text"/>
Last Name	<input type="text"/>
* Mandatory Fields Submit	

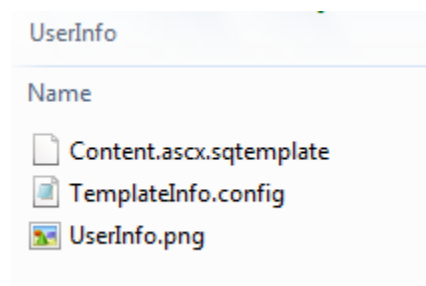
UserInfo

To create the Template:

1. Under the C:\Program Files\PNMsoft\Shared Resources\Components\Forms\ControlTemplates\ViewTemplates\Custom folder, create the folder: UserInfo.
2. Copy the following files from a System Template folder (e.g. C:\Program Files\PNMsoft\Shared Resources\Components\Forms\ControlTemplates\ViewTemplates\System\Section-1C-H-S) into the UserInfo folder:



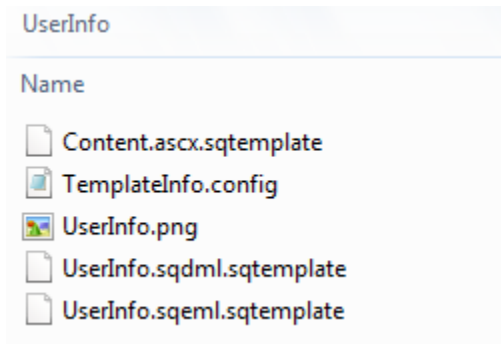
3. Create a 70 x 70 pixel png image file for the Template and copy it into the UserInfo folder.



For example:



4. Copy the OneTableDataModel.sqddl.sqtemplate and OneTableDataModel.sqeml.sqtemplate files from *C:\Program Files\PNMsoft\Shared Resources\Components\Forms\ControlTemplates\ViewTemplates\System_Shared* into the UserInfo folder and rename them: UserInfo.sqddl.sqtemplate and UserInfo.sqeml.sqtemplate

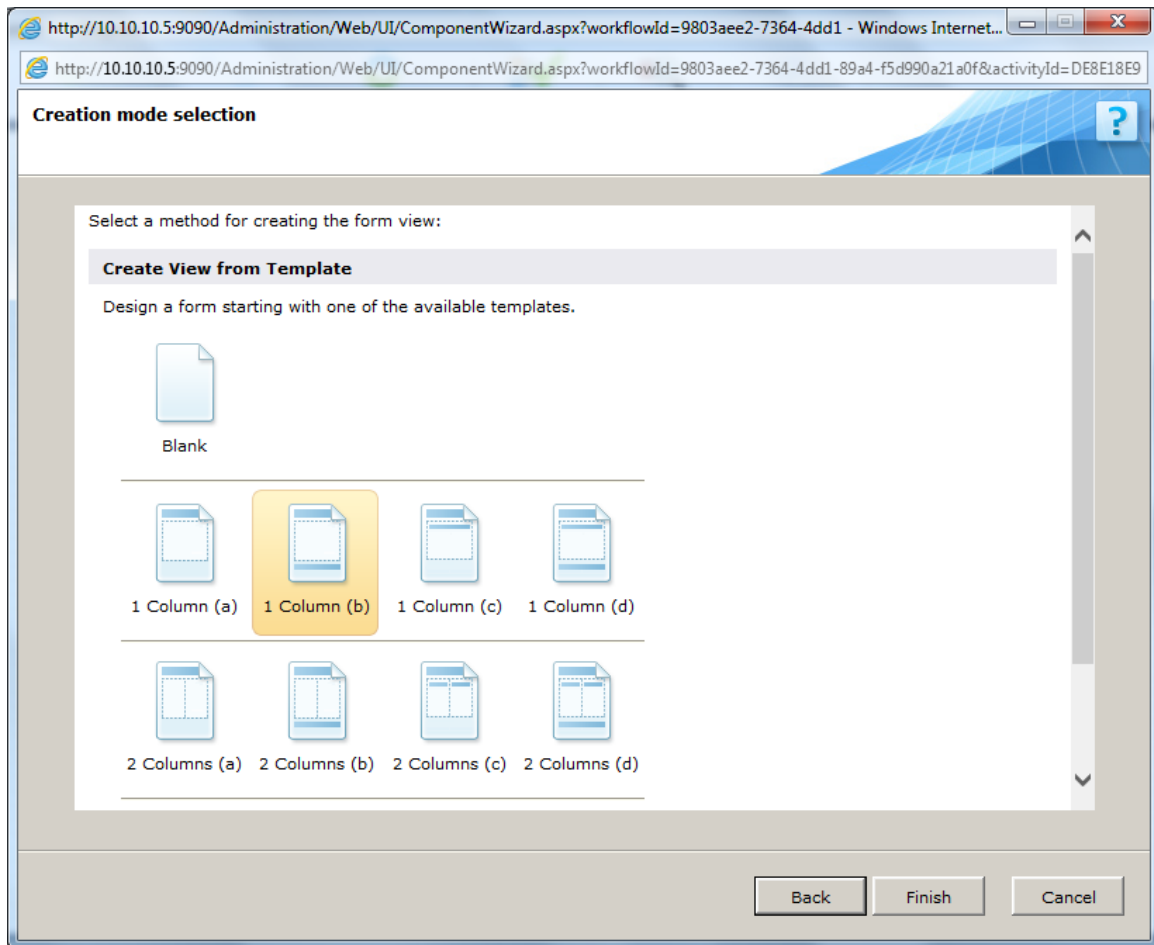


5. Edit the file Content.ascx.sqtemplate as below (generate a unique GUID instead of the one below for the ID field):

```
<?xml version="1.0" encoding="utf-8"?>
<FormTemplate xmlns="http://schemas.pnmsoft.com/sequence/2013/07/ux/formtemplates">
  <TemplateData>
    <ID>097b252e-f6c5-4263-9aa4-ecf539816bcc</ID>
    <Name>UserInfo</Name>
    <DisplayName>User Info</DisplayName>
    <Description>User Info Template</Description>
    <LargeIconPath>UserInfo.png</LargeIconPath>
    <ContentTemplatePath>Content.ascx.sqtemplate</ContentTemplatePath>
    <DataModelPath>UserInfo.sqddl.sqtemplate</DataModelPath>
    <EntityModelPath>UserInfo.sqeml.sqtemplate</EntityModelPath>
    <Sequence>1000</Sequence>
  </TemplateData>
</FormTemplate>
```

- In Sequence, create a new Form using the View First option and selecting the 1 Column (b) form template:

Note: If you are comfortable editing the Form markup, you can skip steps 6-8.



Creation Mode Selection

- Add First Name and Last Name text box fields to this form and define data fields for them in the Data Model:

Form5

Form description...

First Name	<input type="text"/>	<div> <div>TextBox Tasks</div> <div> <div>Data Field:</div> <div> <div>(None)</div> <div>fidId</div> <div>fidIWfid</div> <div>fidIActId</div> <div>fidAIId</div> <div><New field...></div> </div> </div> </div>
Last Name	<input type="text"/>	
Label		
Label		

* Mandatory Fields

Submit

Add Fields to Form

Define Data Fields

- Copy the markup (Source) of the Form and paste it into the Content.ascx.sqtemplate file.

Note: If you are comfortable editing the Form markup, you can edit Content.ascx.sqtemplate directly.

- Replace the form name and query name in the Content.ascx.sqtemplate file with:

- \$formname\$
- \$queryname\$

```
<%@ Control %>

<sq:Form runat="server" ID="Form1" DataSourceID="Form1DataSource" Height="0px" DataKeyNames="fldId" Width="100%" CellPadding="0"

<table class="sqf-main-table" cellpadding="0" cellspacing="0">
<tr>
<td class="sqf-header">
<h1><sq:Text runat="server">$formname$</sq:Text></h1>
<sq:Text runat="server">Form description...</sq:Text>
</td>
</tr>
<tr>
<td>
<table class="sqf-section-table" cellpadding="0" cellspacing="0">
<tr>
<td class="sqf-section-label-col">
<sq:Label runat="server" Text="First Name" />
</td>
<td>
<sq:TextBox runat="server" ID="FirstName" />
<sq:BindableControl runat="server" TargetControlID="FirstName" DataField="FirstName"></sq:BindableControl>
</td>
</tr>
<tr>
<td>
<sq:Label runat="server" Text="Last Name" />
</td>
</tr>
</table>
</td>
</tr>
</table>
```

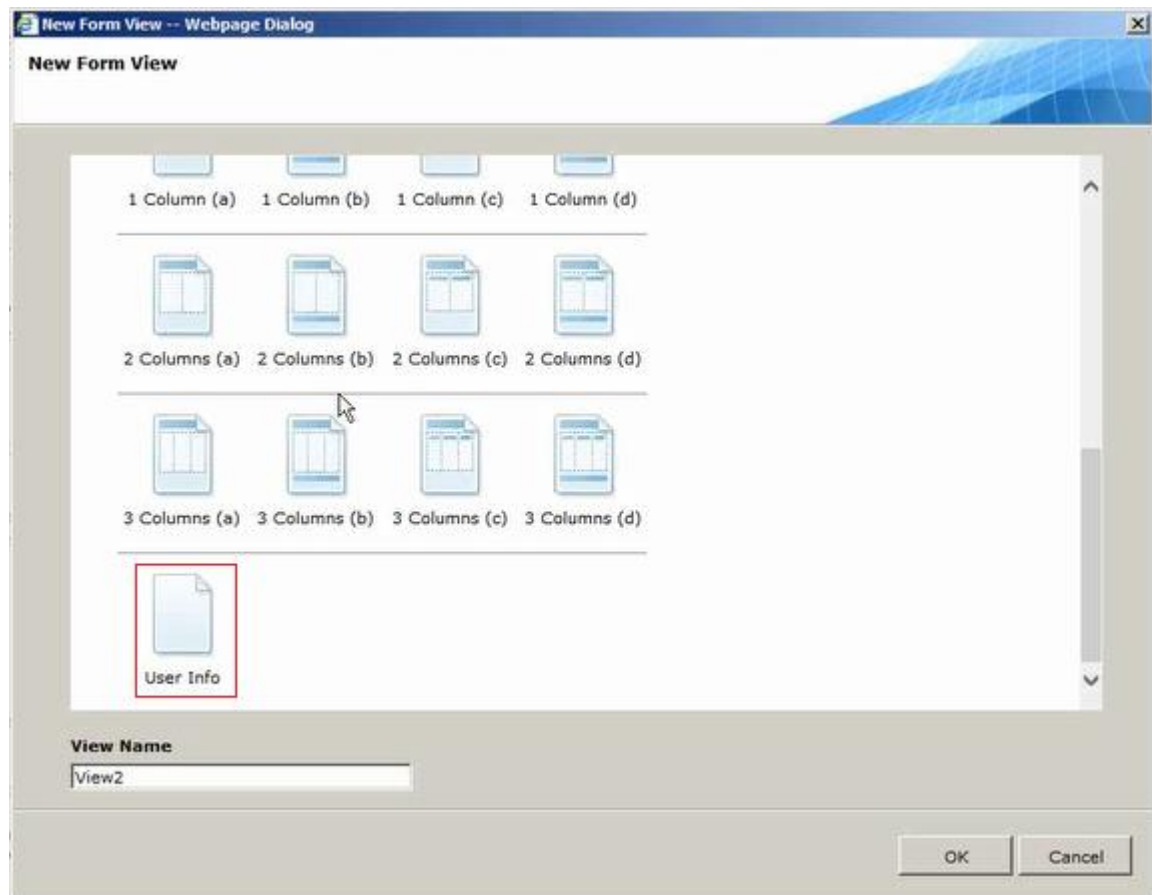
Content.ascx.sqtemplate

10. In the file UserInfo.sqeml.sqtemplate, add two new lines for the new Data Fields (First Name and Last Name) and define them as TypeKind: String:

```
<?xml version="1.0" encoding="utf-8"?>
<EntityModelFragment Version="1.0"
xmlns="http://pmnsoft.com/sequence/2010/11/data/modeling">
  <EntityContainer Namespace="ref:Self">
    <EntityQuery Name="$queryname$" ElementType="$queryname$" DataSource="$queryname$"
/>
    <EntityDataSource Name="$queryname$" />
    <EntityType Name="$queryname$" DataSource="$queryname$">
      <EntityProperty Name="fldId" IsPrimaryKey="true" IsIdentity="true" Nullable="false"
TypeKind="Int32" />
      <EntityProperty Name="fldWfld" IsSystem="true" Nullable="false" TypeKind="Int32" />
      <EntityProperty Name="fldActId" IsSystem="true" Nullable="false" TypeKind="Int32" />
      <EntityProperty Name="fldAllId" IsSystem="true" Nullable="true" TypeKind="Int32" />
      <EntityProperty Name="FirstName" Nullable="true" TypeKind="String" />
      <EntityProperty Name="LastName" Nullable="true" TypeKind="String" />
    </EntityType>
  </EntityContainer>
</EntityModelFragment>
```

11. Perform IISreset.
12. Create a New Form in Sequence.

The template should now appear in the Creation Mode Selection screen:



UserInfo Template Created

When you select this Template, the Form is created with the View and Data Model set up as you defined:

Request Service	
Form description...	
First Name	<input type="text"/>
Last Name	<input type="text"/>
* Mandatory Fields	
<input type="button" value="Submit"/>	

Form Created Based on UserInfo Template

Note: Template folders beginning with the character **_** are hidden, i.e. they are not displayed in the Creation Mode Selection screen (e.g. **_HiddenTemplate**).

Limitations and Best Practices

Currently the Form Template extensions module supports templates with one Data Table.

When customizing the Form Template markup, it is advisable to use the existing naming conventions for class names.