

Q-flow

Guía para Upgrade a 3.05 SQL Server

Versión del documento revisada

Julio 2010

Índice

Tareas previas	2
Actualización	2
Consideraciones para formularios personalizados en migración a 3.04 o posterior	3
Uso de Ajax	3
Consideraciones a tener en cuenta (SOLO PARA Q-flow 3.01).....	4
Formularios personalizados	4

Tareas previas

1. Leer atentamente el manual de instalación de Q-flow 3.05, en él encontrará indicaciones precisas de cómo actualizar Q-flow. Este documento es complementario.
2. Leer atentamente el manual para el diseño de formularios personalizados de Q-flow 3.05, capítulo: “Compatibilidad con formularios de versiones anteriores”.
3. Bajar todos los servicios de Q-flow
4. Backup de la base de datos
 - a. Qflow3 – puede ser que se llame distinto, por ejemplo Qflow3e
 - b. QflowPersonalization – según como se haya configurado puede llamarse diferente, por ejemplo aspnetdb.
5. Backup de las carpetas de servicios y herramientas:
 - a. <%ProgramFiles%>\Urudata, donde <%ProgramFiles%> debe sustituirse por la carpeta donde se instalan los programas en la computadora que tiene instalado Q-flow.
 - b. <%wwwroot%>\QflowWebSite, donde <%wwwroot%> es la carpeta donde se instalan por defecto las aplicaciones web. Esta carpeta se encuentra normalmente en: “c:\inetpub\wwwroot”.

Actualización

1. En el Panel de control de Windows, abra la ventana Agregar y quitar programas.
 - a. Desinstale todos los componentes de Q-flow
 - i. Q-flow Backend services
 - ii. Q-flow BPA (este componente está disponible a partir de 3.03)
 - iii. Q-flow BPM
 - iv. Q-flow OMM
 - v. Q-flow Web Services (este componente es opcional)
 - vi. Q-flow Web Site
2. Acceder al instalador de Q-flow 3.05 y ejecutar Setup.exe
 - a. Seleccionar Standard Setup
 - b. Ejecutar instalador de la base de datos
 - i. Seleccionar **Update** Q-flow database
 - ii. Asegúrese de indicar correctamente los parámetros solicitados
 - c. Instalar los servicios de backend

- d. Instalar el sitio web de Q-flow. No será necesario instalar la base de datos de personalización para el sitio web, debido a que es posible utilizar la instalada previamente.
 - e. Por último instale las herramientas de diseño de procesos, administración del organigrama y el nuevo administrador de procesos.
3. Si es necesario instale los Web Services de Q-flow. Los mismos se encuentran en un instalador separado.

Consideraciones para formularios personalizados en migración a 3.04 o posterior

A partir de 3.04 se incorpora un nuevo recurso “master page” al sitio de Q-flow llamado CustomFormMaster.master; este a su vez hereda de ContentMaster.master que ya existía anteriormente.

Para el correcto funcionamiento de sus formularios personalizados, los mismos deben utilizar la nueva master page (CustomFormMaster.master).

Uso de Ajax

Si usted desea migrar de 3.03 o anterior a 3.04 o posterior y utiliza los controles de AJAX en sus formularios personalizados debe tener en cuenta este capítulo.

El problema de compatibilidad se genera debido a que se agregó a ContentMaster.master el control **ScriptManager** de Ajax para que todas las páginas del sitio web de Q-flow puedan utilizarlo. De esta forma, si usted tiene formularios personalizados que definen este control, el mismo estará repetido y generará un error de tiempo de ejecución.

Solución: si simplemente utiliza el control ScriptManager para trabajar con controles UpdatePanel, simplemente debe borrar de su página el ScriptManager. Si a su vez utiliza otras características Ajax, tales como registrar js o un web services, reemplace el control ScriptManager por el control ScriptManagerProxy.

Consideraciones a tener en cuenta (SOLO PARA Q-flow 3.01)

Formularios personalizados

Si sus procesos utilizan formularios personalizados deberá realizar las siguientes modificaciones para que estos funcionen adecuadamente en Q-flow 3.05.

Un cambio importante en la nueva versión de Q-flow es el refactor que se realizó a nivel de proyectos. Este refactor permite separar aquellos componentes que antes eran exclusivos de Q-flow a proyectos separados que son reutilizados por diferentes aplicaciones, como por ejemplo Q-expeditive. Estos componentes pertenecen ahora al namespace “Qframework”, lo que implica algunos cambios en las importaciones realizadas en formularios personalizados. Algunos de los cambios detectados son los siguientes:

- El namespace Qflow.Common.Exceptions cambió a Qframework.Common.Exceptions, por lo que si se lo incluye en el code behind es necesario cambiar la sentencia de importación.
- Los controles estándar de Q-flow pasaron de la dll Qflow.Web (namespace Qflow.Web.Controls) a Qframework.Web (namespace Qframework.Web.Controls), por lo que si se hacía uso de estos controles en el markup, es necesario cambiar las referencias.
- Un cambio importante es que las referencias en el code behind a la propiedad Interaction se deben cambiar por referencias a la propiedad FlowInteraction. Donde se utilice esta nueva propiedad es necesario importar el namespace Qframework.Web.Interaction.
- Los mensajes fueron movidos al namespace Qframework.BusinessLayer.Messages.Interaction. Si se hace referencia a esos mensajes es necesario agregar la importación de ese namespace.
- La gran mayoría de los enumerados fueron movidos del namespace Qflow.Common a Qframework.Common, como es el caso de ItemScope.

Esta lista de cambios no está cerrada, puede ser que existan otros cambios de namespace según las clases que fueran utilizadas en formularios personalizados. En caso de utilizar alguna clase que no se encuentra dentro de los namespaces aquí sugeridos se recomienda como pauta general buscar una clase con el mismo nombre en un namespace similar cuyo nombre comience por “Qframework”, pues es el lugar más probable donde encontrarla.

Adicionalmente, debido a una mejora del comportamiento del control “Submit” de Q-flow, surgen algunos cambios en lo que respecta a validaciones client-side en formularios personalizados. En la versión anterior era posible agregar rutinas de validación javascript a eventos como el “onsubmit” del formulario o el “onclick” del botón “Submit” de Q-flow, mediante snippets como los siguientes:

```
document.forms[0].attachEvent("onsubmit",validarForm)
GetSubmitElement().attachEvent("onclick",validarForm)
```

A partir de la nueva versión estas validaciones, si bien se ejecutarán, no detendrán el postback de la página. Sin embargo, es posible mediante poco esfuerzo lograr que estas validaciones escritas en javascript sean ejecutadas normalmente dentro del ciclo de la página. La clave es utilizar los controles de

validación de ASP.NET especificando que se utilizará una rutina de validación client-side. Esto se hace de la siguiente manera:

- En el markup de la página se agrega el control de validación ASP.NET con una definición como la siguiente:

```
<asp:CustomValidator ID="CustomValidator1" runat="server"  
ClientValidationFunction="validarForm" EnableClientScript="true"  
ValidationGroup="QCommandButtonValidationGroup"></asp:CustomValidator>
```
- La rutina de validación javascript se modifica para recibir argumentos necesarios para el framework de validación de ASP.NET. En el caso de ejemplo que estamos viendo la firma sería la siguiente:

```
function validarForm(source, clientside_arguments)
```
- Dentro de la rutina de validación javascript se utiliza la propiedad booleana `clientside_arguments.IsValid` para especificar si la validación fue exitosa o no. Si la validación no es exitosa no se realiza el postback.

Debe verificar que todos los elementos de los Templates que contienen script compilen debido a los cambios de namespace y algunas propiedades que cambiaron su nombre. En general los cambios de nombres se dan en propiedades terminadas en "ID" por "Id", por ejemplo: "FlowID" ahora se llama "FlowId". Los elementos que debe verificar son:

- Pasos de código
- Pasos de evaluación por código
- Integraciones (las operaciones de cada integración, al menos la que se encuentra en producción).
- Manejadores de eventos