Cómo actualizar la definición de una instancia de flujo de trabajo en ejecución

**.NET Framework (current version)**

Publicada: abril de 2016

La actualización dinámica proporciona un mecanismo para que los desarrolladores de aplicaciones de flujo de trabajo actualicen la definición de flujo de trabajo de una instancia de flujo de trabajo persistente.El cambio necesario puede ser implementar una corrección de errores, nuevos requisitos o dar cabida a cambios inesperados.En este paso del tutorial se muestra cómo usar la actualización dinámica para modificar las instancias persistentes del flujo de trabajo de adivinación del número de v1 para que coincida con la nueva funcionalidad presentada en [Cómo hospedar varias versiones de un flujo de trabajo en paralelo](https://msdn.microsoft.com/es-es/library/jj205425(v=vs.110).aspx).

|  |
| --- |
| **System_CAPS_noteNota** |
| Para descargar una versión completada o consultar una descripción en vídeo del tutorial, vea [Windows Workflow Foundation (WF45): tutorial de introducción](http://go.microsoft.com/fwlink/?LinkID=248976). |

[En este tema](javascript:void(0))

* [Para crear el proyecto de CreateUpdateMaps](https://msdn.microsoft.com/es-es/library/jj205427(v=vs.110).aspx#BKMK_CreateProject)
* [Para actualizar StateMachineNumberGuessWorkflow](https://msdn.microsoft.com/es-es/library/jj205427(v=vs.110).aspx#BKMK_StateMachine)
* [Para actualizar FlowchartNumberGuessWorkflow](https://msdn.microsoft.com/es-es/library/jj205427(v=vs.110).aspx#BKMK_Flowchart)
* [Para actualizar SequentialNumberGuessWorkflow](https://msdn.microsoft.com/es-es/library/jj205427(v=vs.110).aspx#BKMK_Sequential)
* [Para compilar y ejecutar la aplicación CreateUpdateMaps](https://msdn.microsoft.com/es-es/library/jj205427(v=vs.110).aspx#BKMK_CreateUpdateMaps)
* [Para compilar el ensamblado de flujo de trabajo actualizado](https://msdn.microsoft.com/es-es/library/jj205427(v=vs.110).aspx#BKMK_BuildAssembly)
* [Para actualizar WorkflowVersionMap con las nuevas versiones](https://msdn.microsoft.com/es-es/library/jj205427(v=vs.110).aspx#BKMK_UpdateWorkflowVersionMap)
* [Para aplicar las actualizaciones dinámicas](https://msdn.microsoft.com/es-es/library/jj205427(v=vs.110).aspx#BKMK_ApplyUpdate)
* [Para ejecutar la aplicación con los flujos de trabajo actualizados](https://msdn.microsoft.com/es-es/library/jj205427(v=vs.110).aspx#BKMK_BuildAndRun)
* [Para habilitar la opción de iniciar versiones anteriores del flujo de trabajo](https://msdn.microsoft.com/es-es/library/jj205427(v=vs.110).aspx#BKMK_StartPreviousVersions)

[Para crear el proyecto de CreateUpdateMaps](javascript:void(0))

1. Haga clic con el botón secundario en **WF45GettingStartedTutorial** en el **Explorador de soluciones** y elija **Agregar**, **Nuevo proyecto**.
2. En el nodo **Instalado**, seleccione **Visual C#**, **Windows** (o **Windows**, **Visual Basic**).

|  |
| --- |
| **System_CAPS_noteNota** |
| En función del lenguaje de programación que se configure como lenguaje principal en Visual Studio, el nodo **Visual C#** o **Visual Basic** puede estar bajo el nodo **Otros lenguajes** en el nodo **Instalado**. |

1. Asegúrese de que se haya seleccionado **.NET Framework 4.5** en la lista desplegable correspondiente a la versión de .NET Framework.Seleccione **Aplicación de consola** en la lista **Windows**.Escriba **CreateUpdateMaps** en el cuadro **Nombre** y haga clic en**Aceptar**.
2. Haga clic con el botón secundario en **CreateUpdateMaps** en **Explorador de soluciones** y elija **Agregar referencia**.
3. Seleccione **Framework** del nodo **Ensamblados** en la lista **Agregar referencia**.Escriba **System.Activities** en el cuadro **Buscar ensamblados** para filtrar los ensamblados y hacer que las referencias deseadas sean más fáciles de seleccionar.
4. Active la casilla situada junto a **System.Activities** en la lista **Resultados de la búsqueda**.
5. Escriba **Serialización** en el cuadro **Buscar ensamblados** y active la casilla situada junto a **System.Runtime.Serialization** de la lista**Resultados de la búsqueda**.
6. Escriba **System.Xaml** en el cuadro **Buscar ensamblados** y active la casilla situada junto a **System.Xaml** de la lista **Resultados de la búsqueda**.
7. Haga clic en **Aceptar** para cerrar **Administrador de referencias** y agregar las referencias.
8. Agregue las siguientes instrucciones **using** (o **Imports**) al principio del archivo con las demás instrucciones **using** (o **Imports**).

C#

[**VB**](https://msdn.microsoft.com/es-es/library/jj205427(v=vs.110).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-1)

using System.Activities;

using System.Activities.Statements;

using System.IO;

using System.Xaml;

using System.Reflection;

using System.Activities.XamlIntegration;

using System.Activities.DynamicUpdate;

using System.Runtime.Serialization;

using Microsoft.CSharp.Activities;

1. Agregue los dos miembros de cadena siguientes a la clase Program (o Module1).

C#

[**VB**](https://msdn.microsoft.com/es-es/library/jj205427(v=vs.110).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-2)

const string mapPath = @"..\..\..\PreviousVersions";

const string definitionPath = @"..\..\..\NumberGuessWorkflowActivities\_du";

1. Agregue el siguiente método StartUpdate a la clase Program (o Module1).Este método carga la definición de flujo de trabajo de xaml especificada en **ActivityBuilder** y, a continuación, llama a **DynamicUpdate.PrepareForUpdate**. **PrepareForUpdate** realiza una copia de la definición de flujo de trabajo dentro de **ActivityBuilder**.Después de que se modifique la definición de flujo de trabajo, esta copia se usa junto con la definición de flujo de trabajo modificado para crear la asignación de actualización.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/jj205427(v=vs.110).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-3)

private static ActivityBuilder StartUpdate(string name)

{

// Create the XamlXmlReaderSettings.

XamlXmlReaderSettings readerSettings = new XamlXmlReaderSettings()

{

// In the XAML the "local" namespace referes to artifacts that come from

// the same project as the XAML. When loading XAML if the currently executing

// assembly is not the same assembly that was referred to as "local" in the XAML

// LocalAssembly must be set to the assembly containing the artifacts.

// Assembly.LoadFile requires an absolute path so convert this relative path

// to an absolute path.

LocalAssembly = Assembly.LoadFile(

Path.GetFullPath(Path.Combine(mapPath, "NumberGuessWorkflowActivities\_v1.dll")))

};

string path = Path.Combine(definitionPath, name);

XamlXmlReader xamlReader = new XamlXmlReader(path, readerSettings);

// Load the workflow definition into an ActivityBuilder.

ActivityBuilder wf = XamlServices.Load(

ActivityXamlServices.CreateBuilderReader(xamlReader))

as ActivityBuilder;

// PrepareForUpdate makes a copy of the workflow definition in the

// ActivityBuilder that is used for comparison when the update

// map is created.

DynamicUpdateServices.PrepareForUpdate(wf);

return wf;

}

1. A continuación, agregue el siguiente CreateUpdateMethod a la clase Program (o Module1).Esto crea una asignación de actualización dinámica al llamar a DynamicUpdateServices.CreateUpdateMap y, a continuación, guarda la asignación de actualización con el nombre especificado.Esta asignación de actualización contiene la información que necesita el runtime de flujo de trabajo para actualizar una instancia de flujo de trabajo persistente que se inició mediante la definición de flujo de trabajo inicial incluida en **ActivityBuilder**, de modo que se completa con la definición de flujo de trabajo actualizada.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/jj205427(v=vs.110).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-4)

private static void CreateUpdateMaps(ActivityBuilder wf, string name)

{

// Create the UpdateMap.

DynamicUpdateMap map =

DynamicUpdateServices.CreateUpdateMap(wf);

// Serialize it to a file.

string path = Path.Combine(mapPath, name);

DataContractSerializer sz = new DataContractSerializer(typeof(DynamicUpdateMap));

using (FileStream fs = System.IO.File.Open(path, FileMode.Create))

{

sz.WriteObject(fs, map);

}

}

1. Agregue el siguiente método SaveUpdatedDefinition a la clase Program (o Module1).Este método guarda la definición de flujo de trabajo actualizado una vez creada la asignación de actualización.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/jj205427(v=vs.110).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-5)

private static void SaveUpdatedDefinition(ActivityBuilder wf, string name)

{

string xamlPath = Path.Combine(definitionPath, name);

StreamWriter sw = File.CreateText(xamlPath);

XamlWriter xw = ActivityXamlServices.CreateBuilderWriter(

new XamlXmlWriter(sw, new XamlSchemaContext()));

XamlServices.Save(xw, wf);

sw.Close();

}

[Para actualizar StateMachineNumberGuessWorkflow](javascript:void(0))

1. Agregue CreateStateMachineUpdateMap a la clase Program (o Module1).

C#

[**VB**](https://msdn.microsoft.com/es-es/library/jj205427(v=vs.110).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-6)

private static void CreateStateMachineUpdateMap()

{

}

1. Realice una llamada a StartUpdate y, a continuación, obtenga una referencia a la actividad **StateMachine** raíz del flujo de trabajo.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/jj205427(v=vs.110).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-7)

ActivityBuilder wf = StartUpdate("StateMachineNumberGuessWorkflow.xaml");

// Get a reference to the root StateMachine activity.

StateMachine sm = wf.Implementation as StateMachine;

1. A continuación, actualice las expresiones de las dos actividades **WriteLine** que muestran si el intento del usuario es demasiado alto o demasiado bajo para que coincidan con las actualizaciones realizadas en [Cómo hospedar varias versiones de un flujo de trabajo en paralelo](https://msdn.microsoft.com/es-es/library/jj205425(v=vs.110).aspx).

C#

[**VB**](https://msdn.microsoft.com/es-es/library/jj205427(v=vs.110).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-8)

// Update the Text of the two WriteLine activities that write the

// results of the user's guess. They are contained in the workflow as the

// Then and Else action of the If activity in sm.States[1].Transitions[1].Action.

If guessLow = sm.States[1].Transitions[1].Action as If;

// Update the "too low" message.

WriteLine tooLow = guessLow.Then as WriteLine;

tooLow.Text = new CSharpValue<string>("Guess.ToString() + \" is too low.\"");

// Update the "too high" message.

WriteLine tooHigh = guessLow.Else as WriteLine;

tooHigh.Text = new CSharpValue<string>("Guess.ToString() + \" is too high.\"");

1. A continuación, agregue la nueva actividad **WriteLine** que muestra el mensaje de cierre.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/jj205427(v=vs.110).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-9)

// Create the new WriteLine that displays the closing message.

WriteLine wl = new WriteLine

{

Text = new CSharpValue<string>("Guess.ToString() + \" is correct. You guessed it in \" + Turns.ToString() + \" turns.\"")

};

// Add it as the Action for the Guess Correct transition. The Guess Correct

// transition is the first transition of States[1]. The transitions are listed

// at the bottom of the State activity designer.

sm.States[1].Transitions[0].Action = wl;

1. Una vez actualizado el flujo de trabajo, llame a CreateUpdateMaps y a SaveUpdatedDefinition. CreateUpdateMaps crea y guarda**DynamicUpdateMap**, y SaveUpdatedDefinition guarda la definición de flujo de trabajo actualizada.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/jj205427(v=vs.110).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-10)

// Create the update map.

CreateUpdateMaps(wf, "StateMachineNumberGuessWorkflow.map");

// Save the updated workflow definition.

SaveUpdatedDefinition(wf, "StateMachineNumberGuessWorkflow\_du.xaml");

El siguiente ejemplo es el método CreateStateMachineUpdateMap completado.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/jj205427(v=vs.110).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-11)

private static void CreateStateMachineUpdateMap()

{

ActivityBuilder wf = StartUpdate("StateMachineNumberGuessWorkflow.xaml");

// Get a reference to the root StateMachine activity.

StateMachine sm = wf.Implementation as StateMachine;

// Update the Text of the two WriteLine activities that write the

// results of the user's guess. They are contained in the workflow as the

// Then and Else action of the If activity in sm.States[1].Transitions[1].Action.

If guessLow = sm.States[1].Transitions[1].Action as If;

// Update the "too low" message.

WriteLine tooLow = guessLow.Then as WriteLine;

tooLow.Text = new CSharpValue<string>("Guess.ToString() + \" is too low.\"");

// Update the "too high" message.

WriteLine tooHigh = guessLow.Else as WriteLine;

tooHigh.Text = new CSharpValue<string>("Guess.ToString() + \" is too high.\"");

// Create the new WriteLine that displays the closing message.

WriteLine wl = new WriteLine

{

Text = new CSharpValue<string>("Guess.ToString() + \" is correct. You guessed it in \" + Turns.ToString() + \" turns.\"")

};

// Add it as the Action for the Guess Correct transition. The Guess Correct

// transition is the first transition of States[1]. The transitions are listed

// at the bottom of the State activity designer.

sm.States[1].Transitions[0].Action = wl;

// Create the update map.

CreateUpdateMaps(wf, "StateMachineNumberGuessWorkflow.map");

// Save the updated workflow definition.

SaveUpdatedDefinition(wf, "StateMachineNumberGuessWorkflow\_du.xaml");

}

[Para actualizar FlowchartNumberGuessWorkflow](javascript:void(0))

1. Agregue el siguiente CreateFlowchartUpdateMethod a la clase Program (o Module1).Este método es parecido aCreateStateMachineUpdateMap.Comienza con una llamada a StartUpdate, actualiza la definición de flujo de trabajo del diagrama de flujo y finaliza guardando la asignación de actualización y la definición de flujo de trabajo actualizada.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/jj205427(v=vs.110).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-12)

private static void CreateFlowchartUpdateMap()

{

ActivityBuilder wf = StartUpdate("FlowchartNumberGuessWorkflow.xaml");

// Get a reference to the root Flowchart activity.

Flowchart fc = wf.Implementation as Flowchart;

// Update the Text of the two WriteLine activities that write the

// results of the user's guess. They are contained in the workflow as the

// True and False action of the "Guess < Target" FlowDecision, which is

// Nodes[4].

FlowDecision guessLow = fc.Nodes[4] as FlowDecision;

// Update the "too low" message.

FlowStep trueStep = guessLow.True as FlowStep;

WriteLine tooLow = trueStep.Action as WriteLine;

tooLow.Text = new CSharpValue<string>("Guess.ToString() + \" is too low.\"");

// Update the "too high" message.

FlowStep falseStep = guessLow.False as FlowStep;

WriteLine tooHigh = falseStep.Action as WriteLine;

tooHigh.Text = new CSharpValue<string>("Guess.ToString() + \" is too high.\"");

// Add the new WriteLine that displays the closing message.

WriteLine wl = new WriteLine

{

Text = new CSharpValue<string>("Guess.ToString() + \" is correct. You guessed it in \" + Turns.ToString() + \" turns.\"")

};

// Create a FlowStep to hold the WriteLine.

FlowStep closingStep = new FlowStep

{

Action = wl

};

// Add this new FlowStep to the True action of the

// "Guess == Guess" FlowDecision

FlowDecision guessCorrect = fc.Nodes[3] as FlowDecision;

guessCorrect.True = closingStep;

// Add the new FlowStep to the Nodes collection.

// If closingStep was replacing an existing node then

// we would need to remove that Step from the collection.

// In this example there was no existing True step to remove.

fc.Nodes.Add(closingStep);

// Create the update map.

CreateUpdateMaps(wf, "FlowchartNumberGuessWorkflow.map");

// Save the updated workflow definition.

SaveUpdatedDefinition(wf, "FlowchartNumberGuessWorkflow\_du.xaml");

}

[Para actualizar SequentialNumberGuessWorkflow](javascript:void(0))

1. Agregue el siguiente CreateSequentialUpdateMethod a la clase Program (o Module1).Este método es similar a los otros dos métodos.Comienza con una llamada a StartUpdate, actualiza la definición de flujo de trabajo secuencial y finaliza guardando la asignación de actualización y la definición de flujo de trabajo actualizada.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/jj205427(v=vs.110).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-13)

private static void CreateSequentialUpdateMap()

{

ActivityBuilder wf = StartUpdate("SequentialNumberGuessWorkflow.xaml");

// Get a reference to the root activity in the workflow.

Sequence rootSequence = wf.Implementation as Sequence;

// Update the Text of the two WriteLine activities that write the

// results of the user's guess. They are contained in the workflow as the

// Then and Else action of the "Guess < Target" If activity.

// Sequence[1]->DoWhile->Body->Sequence[2]->If->Then->If

DoWhile gameLoop = rootSequence.Activities[1] as DoWhile;

Sequence gameBody = gameLoop.Body as Sequence;

If guessCorrect = gameBody.Activities[2] as If;

If guessLow = guessCorrect.Then as If;

WriteLine tooLow = guessLow.Then as WriteLine;

tooLow.Text = new CSharpValue<string>("Guess.ToString() + \" is too low.\"");

WriteLine tooHigh = guessLow.Else as WriteLine;

tooHigh.Text = new CSharpValue<string>("Guess.ToString() + \" is too high.\"");

// Add the new WriteLine that displays the closing message.

WriteLine wl = new WriteLine

{

Text = new CSharpValue<string>("Guess.ToString() + \" is correct. You guessed it in \" + Turns.ToString() + \" turns.\"")

};

// Insert it as the third activity in the root sequence

rootSequence.Activities.Insert(2, wl);

// Create the update map.

CreateUpdateMaps(wf, "SequentialNumberGuessWorkflow.map");

// Save the updated workflow definition.

SaveUpdatedDefinition(wf, "SequentialNumberGuessWorkflow\_du.xaml");

}

[Para compilar y ejecutar la aplicación CreateUpdateMaps](javascript:void(0))

1. Actualice el método Main y agregue las tres llamadas de método siguientes.Estos métodos se agregan en las secciones siguientes.Cada método actualiza el flujo de trabajo de acierto de números correspondiente y crea un elemento **DynamicUpdateMap** que describe las actualizaciones.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/jj205427(v=vs.110).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-14)

static void Main(string[] args)

{

// Create the update maps for the changes needed to the v1 activities

// so they match the v2 activities.

CreateSequentialUpdateMap();

CreateFlowchartUpdateMap();

CreateStateMachineUpdateMap();

}

1. Haga clic con el botón secundario en **CreateUpdateMaps** en el **Explorador de soluciones** y elija **Establecer como proyecto de inicio**.
2. Presione CTRL+MAYÚS+B para compilar la solución y, a continuación, CTRL+F5 para ejecutar la aplicación **CreateUpdateMaps**.

|  |
| --- |
| **System_CAPS_noteNota** |
| La aplicación **CreateUpdateMaps** no muestra ninguna información de estado mientras se ejecuta, pero si se examinan las carpetas**NumberGuessWorkflowActivities\_du** y **PreviousVersions**, se verán los archivos de la definición de flujo de trabajo actualizados y las asignaciones de actualización. |

1. Una vez creadas las asignaciones de actualización y actualizadas las definiciones de flujo de trabajo, el siguiente paso es compilar un ensamblado de flujo de trabajo actualizado que contenga las definiciones actualizadas.

[Para compilar el ensamblado de flujo de trabajo actualizado](javascript:void(0))

1. Abra una segunda instancia de Visual Studio 2012.
2. Elija **Abrir**, **Proyecto/Solución** en el menú **Archivo**.
3. Navegue a la carpeta **NumberGuessWorkflowActivities\_du** que creó en [Cómo hospedar varias versiones de un flujo de trabajo en paralelo](https://msdn.microsoft.com/es-es/library/jj205425(v=vs.110).aspx), seleccione **NumberGuessWorkflowActivities.csproj** (o **vbproj**) y haga clic en **Abrir**.
4. En el **Explorador de soluciones**, haga clic con el botón secundario en **SequentialNumberGuessWorkflow.xaml** y elija **Excluir del proyecto**.Haga lo mismo para **FlowchartNumberGuessWorkflow.xaml** y **StateMachineNumberGuessWorkflow.xaml**.Este paso quita las versiones anteriores de las definiciones de flujo de trabajo del proyecto.
5. Elija **Agregar elemento existente** en el menú **Proyecto**.
6. Navegue a la carpeta **NumberGuessWorkflowActivities\_du** que creó en [Cómo hospedar varias versiones de un flujo de trabajo en paralelo](https://msdn.microsoft.com/es-es/library/jj205425(v=vs.110).aspx).
7. Elija **Archivos XAML (\*.xaml; \*.xoml)** en la lista desplegable **Archivos de tipo**.
8. Seleccione **SequentialNumberGuessWorkflow\_du.xaml**, **FlowchartNumberGuessWorkflow\_du.xaml** y**StateMachineNumberGuessWorkflow\_du.xaml**, y haga clic en **Agregar**.

|  |
| --- |
| **System_CAPS_noteNota** |
| Presione CTRL y haga clic para seleccionar varios elementos a la vez. |

1. Este paso agrega las versiones actualizadas de las definiciones de flujo de trabajo al proyecto.
2. Presione Ctrl+Mayús+B para compilar el proyecto.
3. Elija **Cerrar solución** en el menú **Archivo**.No se requiere un archivo de solución para el proyecto, por lo tanto, haga clic en **No** para cerrar Visual Studio sin guardar un archivo de solución.Elija **Salir** en el menú **Archivo** para cerrar Visual Studio.
4. Abra el Explorador de Windows y navegue a la carpeta **NumberGuessWorkflowActivities\_du\bin\Debug** (o a **bin\Release** según la configuración del proyecto).
5. Cambie el nombre de **NumberGuessWorkflowActivities.dll** por **NumberGuessWorkflowActivities\_v15.dll** y cópielo en la carpeta**PreviousVersions** que creó en [Cómo hospedar varias versiones de un flujo de trabajo en paralelo](https://msdn.microsoft.com/es-es/library/jj205425(v=vs.110).aspx).

[Para actualizar WorkflowVersionMap con las nuevas versiones](javascript:void(0))

1. Vuelva a la instancia inicial de Visual Studio 2012.
2. Haga doble clic en **WorkflowVersionMap.cs** (o **WorkflowVersionMap.vb**) del proyecto **NumberGuessWorkflowHost** para abrirlo.
3. Agregue tres nuevas identidades de flujo de trabajo justo debajo de las seis declaraciones de identidad de flujo de trabajo existentes.En este tutorial, 1.5.0.0 se usa como **WorkflowIdentity.Version** para las identidades de actualización dinámica.Estas nuevas identidades de flujo de trabajo de v15 se usarán para proporcionar la definición de flujo de trabajo correcta para las instancias de flujo de trabajo persistentes y actualizadas dinámicamente.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/jj205427(v=vs.110).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-15)

// Current version identities.

static public WorkflowIdentity StateMachineNumberGuessIdentity;

static public WorkflowIdentity FlowchartNumberGuessIdentity;

static public WorkflowIdentity SequentialNumberGuessIdentity;

// v1 identities.

static public WorkflowIdentity StateMachineNumberGuessIdentity\_v1;

static public WorkflowIdentity FlowchartNumberGuessIdentity\_v1;

static public WorkflowIdentity SequentialNumberGuessIdentity\_v1;

// v1.5 (Dynamic Update) identities.

static public WorkflowIdentity StateMachineNumberGuessIdentity\_v15;

static public WorkflowIdentity FlowchartNumberGuessIdentity\_v15;

static public WorkflowIdentity SequentialNumberGuessIdentity\_v15;

1. Al final del constructor agregue el siguiente código.Este código inicializa las identidades de flujo de trabajo de actualización dinámica, carga las definiciones de flujo de trabajo correspondientes y las agrega al diccionario de versiones de flujo de trabajo.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/jj205427(v=vs.110).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-16)

// Initialize the dynamic update workflow identities.

StateMachineNumberGuessIdentity\_v15 = new WorkflowIdentity

{

Name = "StateMachineNumberGuessWorkflow",

Version = new Version(1, 5, 0, 0)

};

FlowchartNumberGuessIdentity\_v15 = new WorkflowIdentity

{

Name = "FlowchartNumberGuessWorkflow",

Version = new Version(1, 5, 0, 0)

};

SequentialNumberGuessIdentity\_v15 = new WorkflowIdentity

{

Name = "SequentialNumberGuessWorkflow",

Version = new Version(1, 5, 0, 0)

};

// Add the dynamic update workflow identities to the dictionary along with

// the corresponding workflow definitions loaded from the v15 assembly.

// Assembly.LoadFile requires an absolute path so convert this relative path

// to an absolute path.

string v15AssemblyPath = @"..\..\..\PreviousVersions\NumberGuessWorkflowActivities\_v15.dll";

v15AssemblyPath = Path.GetFullPath(v15AssemblyPath);

Assembly v15Assembly = Assembly.LoadFile(v15AssemblyPath);

map.Add(StateMachineNumberGuessIdentity\_v15,

v15Assembly.CreateInstance("NumberGuessWorkflowActivities.StateMachineNumberGuessWorkflow") as Activity);

map.Add(SequentialNumberGuessIdentity\_v15,

v15Assembly.CreateInstance("NumberGuessWorkflowActivities.SequentialNumberGuessWorkflow") as Activity);

map.Add(FlowchartNumberGuessIdentity\_v15,

v15Assembly.CreateInstance("NumberGuessWorkflowActivities.FlowchartNumberGuessWorkflow") as Activity);

El ejemplo siguiente es la clase WorkflowVersionMap completada.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/jj205427(v=vs.110).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-17)

public static class WorkflowVersionMap

{

static Dictionary<WorkflowIdentity, Activity> map;

// Current version identities.

static public WorkflowIdentity StateMachineNumberGuessIdentity;

static public WorkflowIdentity FlowchartNumberGuessIdentity;

static public WorkflowIdentity SequentialNumberGuessIdentity;

// v1 identities.

static public WorkflowIdentity StateMachineNumberGuessIdentity\_v1;

static public WorkflowIdentity FlowchartNumberGuessIdentity\_v1;

static public WorkflowIdentity SequentialNumberGuessIdentity\_v1;

// v1.5 (Dynamic Update) identities.

static public WorkflowIdentity StateMachineNumberGuessIdentity\_v15;

static public WorkflowIdentity FlowchartNumberGuessIdentity\_v15;

static public WorkflowIdentity SequentialNumberGuessIdentity\_v15;

static WorkflowVersionMap()

{

map = new Dictionary<WorkflowIdentity, Activity>();

// Add the current workflow version identities.

StateMachineNumberGuessIdentity = new WorkflowIdentity

{

Name = "StateMachineNumberGuessWorkflow",

// Version = new Version(1, 0, 0, 0),

Version = new Version(2, 0, 0, 0)

};

FlowchartNumberGuessIdentity = new WorkflowIdentity

{

Name = "FlowchartNumberGuessWorkflow",

// Version = new Version(1, 0, 0, 0),

Version = new Version(2, 0, 0, 0)

};

SequentialNumberGuessIdentity = new WorkflowIdentity

{

Name = "SequentialNumberGuessWorkflow",

// Version = new Version(1, 0, 0, 0),

Version = new Version(2, 0, 0, 0)

};

map.Add(StateMachineNumberGuessIdentity, new StateMachineNumberGuessWorkflow());

map.Add(FlowchartNumberGuessIdentity, new FlowchartNumberGuessWorkflow());

map.Add(SequentialNumberGuessIdentity, new SequentialNumberGuessWorkflow());

// Initialize the previous workflow version identities.

StateMachineNumberGuessIdentity\_v1 = new WorkflowIdentity

{

Name = "StateMachineNumberGuessWorkflow",

Version = new Version(1, 0, 0, 0)

};

FlowchartNumberGuessIdentity\_v1 = new WorkflowIdentity

{

Name = "FlowchartNumberGuessWorkflow",

Version = new Version(1, 0, 0, 0)

};

SequentialNumberGuessIdentity\_v1 = new WorkflowIdentity

{

Name = "SequentialNumberGuessWorkflow",

Version = new Version(1, 0, 0, 0)

};

// Add the previous version workflow identities to the dictionary along with

// the corresponding workflow definitions loaded from the v1 assembly.

// Assembly.LoadFile requires an absolute path so convert this relative path

// to an absolute path.

string v1AssemblyPath = @"..\..\..\PreviousVersions\NumberGuessWorkflowActivities\_v1.dll";

v1AssemblyPath = Path.GetFullPath(v1AssemblyPath);

Assembly v1Assembly = Assembly.LoadFile(v1AssemblyPath);

map.Add(StateMachineNumberGuessIdentity\_v1,

v1Assembly.CreateInstance("NumberGuessWorkflowActivities.StateMachineNumberGuessWorkflow") as Activity);

map.Add(SequentialNumberGuessIdentity\_v1,

v1Assembly.CreateInstance("NumberGuessWorkflowActivities.SequentialNumberGuessWorkflow") as Activity);

map.Add(FlowchartNumberGuessIdentity\_v1,

v1Assembly.CreateInstance("NumberGuessWorkflowActivities.FlowchartNumberGuessWorkflow") as Activity);

// Initialize the dynamic update workflow identities.

StateMachineNumberGuessIdentity\_v15 = new WorkflowIdentity

{

Name = "StateMachineNumberGuessWorkflow",

Version = new Version(1, 5, 0, 0)

};

FlowchartNumberGuessIdentity\_v15 = new WorkflowIdentity

{

Name = "FlowchartNumberGuessWorkflow",

Version = new Version(1, 5, 0, 0)

};

SequentialNumberGuessIdentity\_v15 = new WorkflowIdentity

{

Name = "SequentialNumberGuessWorkflow",

Version = new Version(1, 5, 0, 0)

};

// Add the dynamic update workflow identities to the dictionary along with

// the corresponding workflow definitions loaded from the v15 assembly.

// Assembly.LoadFile requires an absolute path so convert this relative path

// to an absolute path.

string v15AssemblyPath = @"..\..\..\PreviousVersions\NumberGuessWorkflowActivities\_v15.dll";

v15AssemblyPath = Path.GetFullPath(v15AssemblyPath);

Assembly v15Assembly = Assembly.LoadFile(v15AssemblyPath);

map.Add(StateMachineNumberGuessIdentity\_v15,

v15Assembly.CreateInstance("NumberGuessWorkflowActivities.StateMachineNumberGuessWorkflow") as Activity);

map.Add(SequentialNumberGuessIdentity\_v15,

v15Assembly.CreateInstance("NumberGuessWorkflowActivities.SequentialNumberGuessWorkflow") as Activity);

map.Add(FlowchartNumberGuessIdentity\_v15,

v15Assembly.CreateInstance("NumberGuessWorkflowActivities.FlowchartNumberGuessWorkflow") as Activity);

}

public static Activity GetWorkflowDefinition(WorkflowIdentity identity)

{

return map[identity];

}

public static string GetIdentityDescription(WorkflowIdentity identity)

{

return identity.ToString();

}

}

1. Presione Ctrl+Mayús+B para compilar el proyecto.

[Para aplicar las actualizaciones dinámicas](javascript:void(0))

1. Haga clic con el botón secundario en **WF45GettingStartedTutorial** en el **Explorador de soluciones** y elija **Agregar**, **Nuevo proyecto**.
2. En el nodo **Instalado**, seleccione **Visual C#**, **Windows** (o **Windows**, **Visual Basic**).

|  |
| --- |
| **System_CAPS_noteNota** |
| En función del lenguaje de programación que se configure como lenguaje principal en Visual Studio, el nodo **Visual C#** o **Visual Basic** puede estar bajo el nodo **Otros lenguajes** en el nodo **Instalado**. |

1. Asegúrese de que se haya seleccionado **.NET Framework 4.5** en la lista desplegable correspondiente a la versión de .NET Framework.Seleccione **Aplicación de consola** en la lista **Windows**.Escriba **ApplyDynamicUpdate** en el cuadro **Nombre** y haga clic en**Aceptar**.
2. Haga clic con el botón secundario en **ApplyDynamicUpdate** en **Explorador de soluciones** y elija **Agregar referencia**.
3. Haga clic en **Solución** y active la casilla situada junto a **NumberGuessWorkflowHost**.Esta referencia es necesaria para queApplyDynamicUpdate pueda usar la clase NumberGuessWorkflowHost.WorkflowVersionMap.
4. Seleccione **Framework** del nodo **Ensamblados** en la lista **Agregar referencia**.Escriba **System.Activities** en el cuadro **Buscar ensamblados**.Esto filtrará los ensamblados y simplificará la selección de las referencias deseadas.
5. Active la casilla situada junto a **System.Activities** en la lista **Resultados de la búsqueda**.
6. Escriba **Serialización** en el cuadro **Buscar ensamblados** y active la casilla situada junto a **System.Runtime.Serialization** de la lista**Resultados de la búsqueda**.
7. Escriba **DurableInstancing** en el cuadro **Buscar ensamblados** y active la casilla situada junto a **System.Activities.DurableInstancing** y**System.Runtime.DurableInstancing** de la lista **Resultados de la búsqueda**.
8. Haga clic en **Aceptar** para cerrar **Administrador de referencias** y agregar las referencias.
9. Haga clic con el botón secundario en **ApplyDynamicUpdate** en Explorador de soluciones y elija **Agregar**, **Clase**.Escriba**DynamicUpdateInfo** en el cuadro **Nombre** y haga clic en **Agregar**.
10. Agregue los dos miembros siguientes a la clase DynamicUpdateInfo.El ejemplo siguiente es la clase DynamicUpdateInfocompletada.Esta clase contiene información sobre la asignación de actualización y la nueva identidad de flujo de trabajo usada cuando se actualiza una instancia de flujo de trabajo.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/jj205427(v=vs.110).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-18)

class DynamicUpdateInfo

{

public DynamicUpdateMap updateMap;

public WorkflowIdentity newIdentity;

}

1. Agregue las siguientes instrucciones **using** (o **Imports**) al principio del archivo con las demás instrucciones **using** (o **Imports**).

C#

[**VB**](https://msdn.microsoft.com/es-es/library/jj205427(v=vs.110).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-19)

using System.Activities;

using System.Activities.DynamicUpdate;

1. Haga doble clic en **Program.cs** (o en **Module1.vb**) en el Explorador de soluciones.
2. Agregue las siguientes instrucciones **using** (o **Imports**) al principio del archivo con las demás instrucciones **using** (o **Imports**).

C#

[**VB**](https://msdn.microsoft.com/es-es/library/jj205427(v=vs.110).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-20)

using NumberGuessWorkflowHost;

using System.Data;

using System.Data.SqlClient;

using System.Activities;

using System.Activities.DynamicUpdate;

using System.IO;

using System.Runtime.Serialization;

using System.Activities.DurableInstancing;

1. Agregue el siguiente miembro de cadena de conexión a la clase Program (o Module1).

C#

[**VB**](https://msdn.microsoft.com/es-es/library/jj205427(v=vs.110).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-21)

const string connectionString = "Server=.\\SQLEXPRESS;Initial Catalog=WF45GettingStartedTutorial;Integrated Security=SSPI";

|  |
| --- |
| **System_CAPS_noteNota** |
| Según la edición de SQL Server, el nombre de servidor de cadena de conexión puede ser diferente. |

1. Agregue el siguiente método GetIDs a la clase Program (o Module1).Este método devuelve una lista de identificadores de instancia de flujo de trabajo persistente.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/jj205427(v=vs.110).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-22)

static IList<Guid> GetIds()

{

List<Guid> Ids = new List<Guid>();

string localCmd = string.Format("Select [InstanceId] from [System.Activities.DurableInstancing].[Instances] Order By [CreationTime]");

using (SqlConnection localCon = new SqlConnection(connectionString))

{

SqlCommand cmd = localCon.CreateCommand();

cmd.CommandText = localCmd;

localCon.Open();

using (SqlDataReader reader = cmd.ExecuteReader(CommandBehavior.CloseConnection))

{

while (reader.Read())

{

// Get the InstanceId of the persisted Workflow

Guid id = Guid.Parse(reader[0].ToString());

// Add it to the list.

Ids.Add(id);

}

}

}

return Ids;

}

1. Agregue el siguiente método LoadMap a la clase Program (o Module1).Este método crea un diccionario que asigna las identidades de flujo de trabajo de v1 a las asignaciones de actualización y a las nuevas identidades de flujo de trabajo usadas para actualizar las instancias de flujo de trabajo persistentes correspondientes.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/jj205427(v=vs.110).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-23)

static DynamicUpdateMap LoadMap(string mapName)

{

string path = Path.Combine(@"..\..\..\PreviousVersions", mapName);

DynamicUpdateMap map;

using (FileStream fs = File.Open(path, FileMode.Open))

{

DataContractSerializer serializer = new DataContractSerializer(typeof(DynamicUpdateMap));

object updateMap = serializer.ReadObject(fs);

if (updateMap == null)

{

throw new ApplicationException("DynamicUpdateMap is null.");

}

map = updateMap as DynamicUpdateMap;

}

return map;

}

1. Agregue el siguiente método LoadMaps a la clase Program (o Module1).Este método carga las tres asignaciones de actualización y crea un diccionario que asigna las identidades de flujo de trabajo de v1 a las asignaciones de actualización.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/jj205427(v=vs.110).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-24)

static IDictionary<WorkflowIdentity, DynamicUpdateInfo> LoadMaps()

{

// There are 3 update maps to describe the changes to update v1 workflows,

// one for reach of the 3 workflow types in the tutorial.

Dictionary<WorkflowIdentity, DynamicUpdateInfo> maps =

new Dictionary<WorkflowIdentity, DynamicUpdateInfo>();

DynamicUpdateMap sequentialMap = LoadMap("SequentialNumberGuessWorkflow.map");

DynamicUpdateInfo sequentialInfo = new DynamicUpdateInfo

{

updateMap = sequentialMap,

newIdentity = WorkflowVersionMap.SequentialNumberGuessIdentity\_v15

};

maps.Add(WorkflowVersionMap.SequentialNumberGuessIdentity\_v1, sequentialInfo);

DynamicUpdateMap stateMap = LoadMap("StateMachineNumberGuessWorkflow.map");

DynamicUpdateInfo stateInfo = new DynamicUpdateInfo

{

updateMap = stateMap,

newIdentity = WorkflowVersionMap.StateMachineNumberGuessIdentity\_v15

};

maps.Add(WorkflowVersionMap.StateMachineNumberGuessIdentity\_v1, stateInfo);

DynamicUpdateMap flowchartMap = LoadMap("FlowchartNumberGuessWorkflow.map");

DynamicUpdateInfo flowchartInfo = new DynamicUpdateInfo

{

updateMap = flowchartMap,

newIdentity = WorkflowVersionMap.FlowchartNumberGuessIdentity\_v15

};

maps.Add(WorkflowVersionMap.FlowchartNumberGuessIdentity\_v1, flowchartInfo);

return maps;

}

1. Agregue el código siguiente a Main.Este código repite las instancias de flujo de trabajo persistentes y examina cada **WorkflowIdentity**.Si**WorkflowIdentity** se asigna a una instancia de flujo de trabajo de v1, se configura un **WorkflowApplication** con la definición de flujo de trabajo actualizada y una identidad de flujo de trabajo actualizada.A continuación, se llama a **WorkflowApplication.Load** con la instancia y la asignación de actualización, que aplica la asignación de actualización dinámica.Una vez aplicada la actualización, la instancia actualizada es persistente con una llamada a **Unload**.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/jj205427(v=vs.110).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-25)

SqlWorkflowInstanceStore store = new SqlWorkflowInstanceStore(connectionString);

WorkflowApplication.CreateDefaultInstanceOwner(store, null, WorkflowIdentityFilter.Any);

IDictionary<WorkflowIdentity, DynamicUpdateInfo> updateMaps = LoadMaps();

foreach (Guid id in GetIds())

{

// Get a proxy to the instance.

WorkflowApplicationInstance instance =

WorkflowApplication.GetInstance(id, store);

Console.WriteLine("Inspecting: {0}", instance.DefinitionIdentity);

// Only update v1 workflows.

if (instance.DefinitionIdentity != null &&

instance.DefinitionIdentity.Version.Equals(new Version(1, 0, 0, 0)))

{

DynamicUpdateInfo info = updateMaps[instance.DefinitionIdentity];

// Associate the persisted WorkflowApplicationInstance with

// a WorkflowApplication that is configured with the updated

// definition and updated WorkflowIdentity.

Activity wf = WorkflowVersionMap.GetWorkflowDefinition(info.newIdentity);

WorkflowApplication wfApp =

new WorkflowApplication(wf, info.newIdentity);

// Apply the Dynamic Update.

wfApp.Load(instance, info.updateMap);

// Persist the updated instance.

wfApp.Unload();

Console.WriteLine("Updated to: {0}", info.newIdentity);

}

else

{

// Not updating this instance, so unload it.

instance.Abandon();

}

}

1. Haga clic con el botón secundario en **ApplyDynamicUpdate** en el **Explorador de soluciones** y elija **Establecer como proyecto de inicio**.
2. Presione CTRL+MAYÚS+B para compilar la solución y, a continuación, presione CTRL+F5 para ejecutar la aplicación**ApplyDynamicUpdate** y actualizar las instancias de flujo de trabajo persistentes.Debería ver un resultado parecido al siguiente.Los flujos de trabajo de la versión 1.0.0.0 se actualizan a la versión 1.5.0.0, mientras que los flujos de trabajo de la versión 2.0.0.0 no se actualizan.

Inspeccionando: StateMachineNumberGuessWorkflow; versión=1.0.0.0Actualizado a: StateMachineNumberGuessWorkflow; versión=1.5.0.0Inspeccionando: StateMachineNumberGuessWorkflow; versión=1.0.0.0Actualizado a: StateMachineNumberGuessWorkflow; versión=1.5.0.0Inspeccionando: FlowchartNumberGuessWorkflow; versión=1.0.0.0Actualizado a: FlowchartNumberGuessWorkflow; versión=1.5.0.0Inspeccionando: FlowchartNumberGuessWorkflow; versión=1.0.0.0Actualizado a: FlowchartNumberGuessWorkflow; versión=1.5.0.0Inspeccionando: SequentialNumberGuessWorkflow; versión=1.0.0.0Actualizado a: SequentialNumberGuessWorkflow; versión=1.5.0.0Inspeccionando: SequentialNumberGuessWorkflow; versión=1.0.0.0Actualizado a: SequentialNumberGuessWorkflow; versión=1.5.0.0Inspeccionando: SequentialNumberGuessWorkflow; versión=1.0.0.0Actualizado a: SequentialNumberGuessWorkflow; versión=1.5.0.0Inspeccionando: StateMachineNumberGuessWorkflow; versión=1.0.0.0Actualizado a: StateMachineNumberGuessWorkflow; versión=1.5.0.0Inspeccionando: FlowchartNumberGuessWorkflow; versión=1.0.0.0Actualizado a: FlowchartNumberGuessWorkflow; versión=1.5.0.0Inspeccionando: StateMachineNumberGuessWorkflow; versión=2.0.0.0Inspeccionando: StateMachineNumberGuessWorkflow; versión=2.0.0.0Inspeccionando: FlowchartNumberGuessWorkflow; versión=2.0.0.0Inspeccionando: FlowchartNumberGuessWorkflow; versión=2.0.0.0Inspeccionando: SequentialNumberGuessWorkflow; versión=2.0.0.0Inspeccionando: SequentialNumberGuessWorkflow; versión=2.0.0.0Presione cualquier tecla para continuar ...

[Para ejecutar la aplicación con los flujos de trabajo actualizados](javascript:void(0))

1. Haga clic con el botón secundario en **NumberGuessWorkflowHost** en el **Explorador de soluciones** y elija **Establecer como proyecto de inicio**.
2. Presione CTRL+F5 para ejecutar la aplicación.
3. Haga clic en **New Game** para iniciar un nuevo flujo de trabajo y observe la información de versión que hay debajo de la ventana de estado que indica que es un flujo de trabajo de v2.
4. Seleccione uno de los flujos de trabajo de v1 que inició al principio del tema [Cómo hospedar varias versiones de un flujo de trabajo en paralelo](https://msdn.microsoft.com/es-es/library/jj205425(v=vs.110).aspx).Observe que la información de versión debajo de la ventana de estado indica que es un flujo de trabajo de la versión**1.5.0.0**.Observe que no hay información indicada sobre intentos anteriores aparte de si eran demasiado altos o demasiado bajos.

Escriba un número entre el 1 y el 10Su intento es demasiado bajo.

1. Anote **InstanceId** y, a continuación, escriba números hasta que se complete el flujo de trabajo.La ventana de estado muestra información sobre el contenido del intento porque la actualización dinámica ha actualizado las actividades **WriteLine**.

Escriba un número entre el 1 y el 10Su intento es demasiado bajo. Escriba un número entre el 1 y el 105 es demasiado bajo. Escriba un número entre el 1 y el 107 es demasiado alto. Escriba un número entre el 1 y el 10Enhorabuena, acertó el número en 4 intentos.

1. Abra el Explorador de Windows y navegue a la carpeta **NumberGuessWorkflowHost\bin\debug** (o **bin\release** en función de la configuración del proyecto) y abra el archivo de seguimiento mediante el Bloc de notas que corresponde al flujo de trabajo completado.Si no ha anotado **InstanceId** tal vez pueda identificar el archivo correcto de seguimiento mediante la información de **Fecha de modificación** en el Explorador de Windows.La última línea de la información de seguimiento contiene el resultado de la actividad**WriteLine** agregada recientemente.

Escriba un número entre el 1 y el 10Su intento es demasiado bajo. Escriba un número entre el 1 y el 105 es demasiado bajo. Escriba un número entre el 1 y el 107 es demasiado alto. Escriba un número entre el 1 y el 106 es correcto.Lo acertó en 4 intentos.

[Para habilitar la opción de iniciar versiones anteriores del flujo de trabajo](javascript:void(0))

Si ya no le quedan flujos de trabajo que actualizar, puede modificar la aplicación **NumberGuessWorkflowHost** para habilitar el inicio de versiones anteriores de los flujos de trabajo.

1. Haga doble clic en **WorkflowHostForm** en **Explorador de soluciones** y seleccione el cuadro combinado **WorkflowType**.
2. En la ventana **Propiedades** , seleccione la propiedad **Elementos** y haga clic en el botón de puntos suspensivos para editar la colección**Elementos**.
3. Agregue los siguientes tres elementos a la colección.

visualbasicANDcsharp

StateMachineNumberGuessWorkflow v1

FlowchartNumberGuessWorkflow v1

SequentialNumberGuessWorkflow v1

La colección **Items** completa tendrá seis elementos.

visualbasicANDcsharp

StateMachineNumberGuessWorkflow

FlowchartNumberGuessWorkflow

SequentialNumberGuessWorkflow

StateMachineNumberGuessWorkflow v1

FlowchartNumberGuessWorkflow v1

SequentialNumberGuessWorkflow v1

1. Haga doble clic en **WorkflowHostForm** en el **Explorador de soluciones** y seleccione **Ver código**.
2. Agregue tres nuevos casos a la instrucción **switch** (o **Select Case**) en el controlador de NewGame\_Click para asignar los nuevos elementos en el cuadro combinado **WorkflowType** a las identidades de flujo de trabajo que coincidan.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/jj205427(v=vs.110).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-28)

case "SequentialNumberGuessWorkflow v1":

identity = WorkflowVersionMap.SequentialNumberGuessIdentity\_v1;

break;

case "StateMachineNumberGuessWorkflow v1":

identity = WorkflowVersionMap.StateMachineNumberGuessIdentity\_v1;

break;

case "FlowchartNumberGuessWorkflow v1":

identity = WorkflowVersionMap.FlowchartNumberGuessIdentity\_v1;

break;

El ejemplo siguiente incluye la instrucción **switch** (o **Select Case**) completa.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/jj205427(v=vs.110).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-29)

switch (WorkflowType.SelectedItem.ToString())

{

case "SequentialNumberGuessWorkflow":

identity = WorkflowVersionMap.SequentialNumberGuessIdentity;

break;

case "StateMachineNumberGuessWorkflow":

identity = WorkflowVersionMap.StateMachineNumberGuessIdentity;

break;

case "FlowchartNumberGuessWorkflow":

identity = WorkflowVersionMap.FlowchartNumberGuessIdentity;

break;

case "SequentialNumberGuessWorkflow v1":

identity = WorkflowVersionMap.SequentialNumberGuessIdentity\_v1;

break;

case "StateMachineNumberGuessWorkflow v1":

identity = WorkflowVersionMap.StateMachineNumberGuessIdentity\_v1;

break;

case "FlowchartNumberGuessWorkflow v1":

identity = WorkflowVersionMap.FlowchartNumberGuessIdentity\_v1;

break;

};

1. Presione CTRL+F5 para compilar y ejecutar la aplicación.Ahora puede iniciar las versiones v1 del flujo de trabajo así como las versiones actuales.Para actualizar dinámicamente estas instancias nuevas, ejecute la aplicación **ApplyDynamicUpdate**.