Cómo crear un participante de seguimiento personalizado

**.NET Framework (current version)**

Publicada: abril de 2016

El seguimiento de flujo de trabajo proporciona visibilidad del estado de la ejecución del flujo de trabajo.El runtime de flujo de trabajo emite registros de seguimiento que describen los eventos de ciclo de vida de flujo de trabajo, los eventos de ciclo de vida de actividad, los errores y la reanudación de marcadores.Los participantes de seguimiento usan estos registros de seguimiento.Windows Workflow Foundation (WF) incluye un participante de seguimiento estándar que escribe los registros de seguimiento como eventos de Seguimiento de eventos para Windows (ETW).Si eso no cumple sus requisitos, también puede escribir un participante de seguimiento personalizado.Este paso del tutorial describe cómo crear un participante y un perfil de seguimiento personalizados que capturen la salida de las actividades **WriteLine** para poder mostrarla al usuario.

|  |
| --- |
| **System_CAPS_noteNota** |
| Cada uno de los temas del tutorial de introducción depende de los temas anteriores.Para completar este tema, primero debe finalizar los temas anteriores.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 participante de seguimiento personalizado](https://msdn.microsoft.com/es-es/library/jj205426(v=vs.110).aspx#BKMK_CustomTrackingParticipant)
* [Para crear el perfil de seguimiento y registrar el participante de seguimiento](https://msdn.microsoft.com/es-es/library/jj205426(v=vs.110).aspx#BKMK_TrackingProfile)
* [Para mostrar la información de seguimiento](https://msdn.microsoft.com/es-es/library/jj205426(v=vs.110).aspx#BKMK_DisplayTracking)
* [Para compilar y ejecutar la aplicación](https://msdn.microsoft.com/es-es/library/jj205426(v=vs.110).aspx#BKMK_BuildAndRun)

[Para crear el participante de seguimiento personalizado](javascript:void(0))

1. Haga clic con el botón secundario en **NumberGuessWorkflowHost** en el **Explorador de soluciones** y elija **Agregar**, **Clase**.Escriba**StatusTrackingParticipant** en el cuadro **Nombre** y haga clic en **Agregar**.
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/jj205426(v=vs.110).aspx?cs-save-lang=1&cs-lang=vb" \l "code-snippet-1)**

using System.Activities.Tracking;

using System.IO;

1. Modifique la clase StatusTrackingParticipant para que herede de **TrackingParticipant**.

C#

**[VB](https://msdn.microsoft.com/es-es/library/jj205426(v=vs.110).aspx?cs-save-lang=1&cs-lang=vb" \l "code-snippet-2)**

class StatusTrackingParticipant : TrackingParticipant

{

}

1. Agregue el siguiente remplazo de método **Track**.Hay varios tipos distintos de registros de seguimiento.Estamos interesados en el resultado de las actividades de **WriteLine** , contenidas en registros de seguimiento de actividad.Si **TrackingRecord** es un**ActivityTrackingRecord** para una actividad **WriteLine**, **Text** de **WriteLine** se anexa a un archivo con una nombre según InstanceId del flujo de trabajo.En este tutorial, el archivo se guarda en la carpeta actual de la aplicación de host.

C#

**[VB](https://msdn.microsoft.com/es-es/library/jj205426(v=vs.110).aspx?cs-save-lang=1&cs-lang=vb" \l "code-snippet-3)**

protected override void Track(TrackingRecord record, TimeSpan timeout)

{

ActivityStateRecord asr = record as ActivityStateRecord;

if (asr != null)

{

if (asr.State == ActivityStates.Executing &&

asr.Activity.TypeName == "System.Activities.Statements.WriteLine")

{

// Append the WriteLine output to the tracking

// file for this instance

using (StreamWriter writer = File.AppendText(record.InstanceId.ToString()))

{

writer.WriteLine(asr.Arguments["Text"]);

writer.Close();

}

}

}

}

Cuando no se especifica ningún perfil de seguimiento, se usa el perfil de seguimiento predeterminado.Cuando se usa el perfil de seguimiento predeterminado, los registros de seguimiento se emiten para todos **ActivityStates**.Dado que solo debemos capturar el texto una vez durante el ciclo de vida de la actividad **WriteLine** , solo extraemos el texto del estado **ActivityStates.Executing** .En [Para crear el perfil de seguimiento y registrar el participante de seguimiento](https://msdn.microsoft.com/es-es/library/jj205426(v=vs.110).aspx#BKMK_TrackingProfile), se crea un perfil de seguimiento que especifica que solo se emiten los registros de seguimiento de **WriteLine** **ActivityStates.Executing**.

[Para crear el perfil de seguimiento y registrar el participante de seguimiento](javascript:void(0))

1. Haga clic con el botón secundario en **WorkflowHostForm** en **Explorador de soluciones** y elija **Ver código**.
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/jj205426(v=vs.110).aspx?cs-save-lang=1&cs-lang=vb" \l "code-snippet-4)**

using System.Activities.Tracking;

1. Agregue el código siguiente a ConfigureWorkflowApplication justo después del código que agrega **StringWriter** a las extensiones de flujo de trabajo y justo antes de los controladores de ciclo de vida de flujo de trabajo.

C#

**[VB](https://msdn.microsoft.com/es-es/library/jj205426(v=vs.110).aspx?cs-save-lang=1&cs-lang=vb" \l "code-snippet-5)**

// Add the custom tracking participant with a tracking profile

// that only emits tracking records for WriteLine activities.

StatusTrackingParticipant stp = new StatusTrackingParticipant

{

TrackingProfile = new TrackingProfile

{

Queries =

{

new ActivityStateQuery

{

ActivityName = "WriteLine",

States = { ActivityStates.Executing },

Arguments = { "Text" }

}

}

}

};

wfApp.Extensions.Add(stp);

Este perfil de seguimiento especifica que solo los registros de estado de actividad para actividades **WriteLine** en el estado de **Executing**se emiten al participante de seguimiento personalizado.

Después de agregar el código, el inicio de ConfigureWorkflowApplication será como el ejemplo siguiente.

C#

**[VB](https://msdn.microsoft.com/es-es/library/jj205426(v=vs.110).aspx?cs-save-lang=1&cs-lang=vb" \l "code-snippet-6)**

private void ConfigureWorkflowApplication(WorkflowApplication wfApp)

{

// Configure the persistence store.

wfApp.InstanceStore = store;

// Add a StringWriter to the extensions. This captures the output

// from the WriteLine activities so we can display it in the form.

StringWriter sw = new StringWriter();

wfApp.Extensions.Add(sw);

// Add the custom tracking participant with a tracking profile

// that only emits tracking records for WriteLine activities.

StatusTrackingParticipant stp = new StatusTrackingParticipant

{

TrackingProfile = new TrackingProfile

{

Queries =

{

new ActivityStateQuery

{

ActivityName = "WriteLine",

States = { ActivityStates.Executing },

Arguments = { "Text" }

}

}

}

};

wfApp.Extensions.Add(stp);

// Workflow lifecycle handlers...

[Para mostrar la información de seguimiento](javascript:void(0)" \o "Contraer)

1. Haga clic con el botón secundario en **WorkflowHostForm** en **Explorador de soluciones** y elija **Ver código**.
2. En el controlador de InstanceId\_SelectedIndexChanged , agregue el código siguiente inmediatamente después del código que borra la ventana de estado.

C#

**[VB](https://msdn.microsoft.com/es-es/library/jj205426(v=vs.110).aspx?cs-save-lang=1&cs-lang=vb" \l "code-snippet-7)**

// If there is tracking data for this workflow, display it

// in the status window.

if (File.Exists(WorkflowInstanceId.ToString()))

{

string status = File.ReadAllText(WorkflowInstanceId.ToString());

UpdateStatus(status);

}

Cuando se selecciona un nuevo flujo de trabajo en la lista de flujos de trabajo, los registros de seguimiento para ese flujo de trabajo se cargan y se muestran en la ventana de estado.El siguiente ejemplo es el controlador InstanceId\_SelectedIndexChanged completado.

C#

[**VB**](https://msdn.microsoft.com/es-es/library/jj205426(v=vs.110).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-8)

private void InstanceId\_SelectedIndexChanged(object sender, EventArgs e)

{

if (InstanceId.SelectedIndex == -1)

{

return;

}

// Clear the status window.

WorkflowStatus.Clear();

// If there is tracking data for this workflow, display it

// in the status window.

if (File.Exists(WorkflowInstanceId.ToString()))

{

string status = File.ReadAllText(WorkflowInstanceId.ToString());

UpdateStatus(status);

}

// Get the workflow version and display it.

// If the workflow is just starting then this info will not

// be available in the persistence store so do not try and retrieve it.

if (!WorkflowStarting)

{

WorkflowApplicationInstance instance =

WorkflowApplication.GetInstance(this.WorkflowInstanceId, store);

WorkflowVersion.Text =

WorkflowVersionMap.GetIdentityDescription(instance.DefinitionIdentity);

// Unload the instance.

instance.Abandon();

}

}

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

1. Presione Ctrl+Mayús+B para compilar la aplicación.
2. Presione Ctrl + F5 para iniciar la aplicación.
3. Seleccione un intervalo para el juego de adivinar un número y el tipo de flujo de trabajo que se va a iniciar, y haga clic en **New Game**.Escriba un intento en el cuadro de **Guess** y haga clic en **Go** para enviarlo.Observe que el estado del flujo de trabajo se muestra en la ventana de estado.Este resultado se captura de las actividades de **WriteLine**.Cambie a un flujo de trabajo diferente mediante la selección de uno en el cuadro combinado **Identificador de instancia de flujo de trabajo** y observe que el estado del flujo de trabajo actual se ha quitado.Cambie de nuevo al flujo de trabajo anterior y observe que se restablece el estado, similar al siguiente ejemplo.

|  |
| --- |
| **System_CAPS_noteNota** |
| Si cambia a un flujo de trabajo que se había iniciado antes de que se habilitara el seguimiento, no se mostrará ningún estado.Sin embargo, si hace intentos adicionales, el estado se guarda porque ahora se ha habilitado el seguimiento. |

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

|  |
| --- |
| **System_CAPS_noteNota** |
| Esta información es útil para determinar el intervalo del número aleatorio, pero no contiene información sobre qué intentos se han hecho anteriormente.Esta información está en el siguiente paso, [Cómo hospedar varias versiones de un flujo de trabajo en paralelo](https://msdn.microsoft.com/es-es/library/jj205425(v=vs.110).aspx). |

1. Anote el identificador de instancias de flujo de trabajo, y juegue hasta completar el juego.
2. Abra el Explorador de Windows y navegue a la carpeta **NumberGuessWorkflowHost\bin\debug** (o **bin\release** según la configuración del proyecto).Observe que además de los archivos ejecutables del proyecto hay archivos con nombres de archivo guid.Identifique cuál corresponde al identificador de instancia del flujo de trabajo completado en el paso anterior y ábralo en el Bloc de notas.La información de seguimiento incluye información similar a la siguiente.

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

Además de no contener los intentos del usuario, este dato de seguimiento no contiene información sobre el último intento del flujo de trabajo.Esto se debe a que la información de seguimiento solo consta del resultado de **WriteLine** del flujo de trabajo, y el mensaje final que se muestra se hace así desde el controlador de **Completed** una vez el flujo de trabajo se ha completado.En el siguiente paso del tutorial, [Cómo hospedar varias versiones de un flujo de trabajo en paralelo](https://msdn.microsoft.com/es-es/library/jj205425(v=vs.110).aspx), las actividades **WriteLine** existentes se modifican para mostrar los intentos del usuario y se agrega una actividad **WriteLine** adicional que muestra el resultado final.Una vez que estos cambios se han integrado, [Cómo hospedar varias versiones de un flujo de trabajo en paralelo](https://msdn.microsoft.com/es-es/library/jj205425(v=vs.110).aspx) muestra cómo hospedar varias versiones de un flujo de trabajo simultáneamente.