Itinerario – Creación de un informe anual para el proveedor

Itinerario – Creación de un informe anual para el proveedor 1

Itinerario – Creación de un informe anual para el proveedor

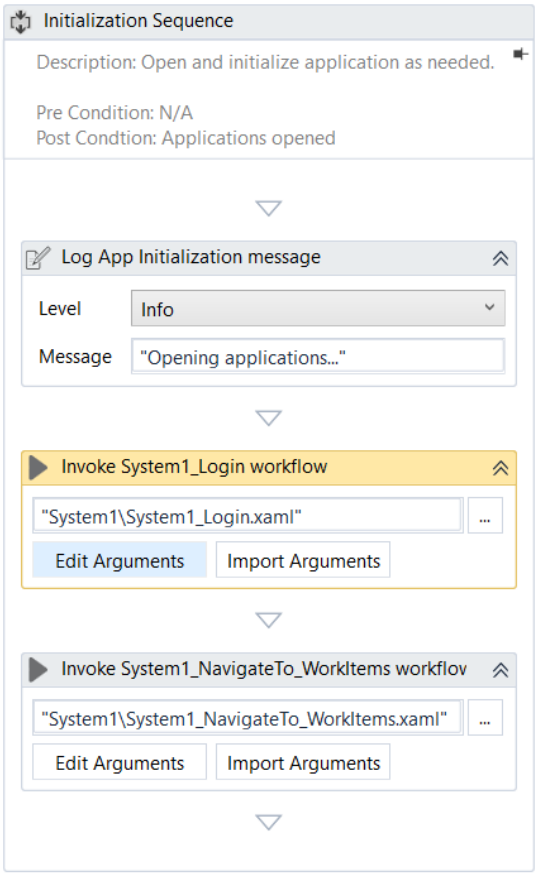
Esta vez utilizaremos Orchestrator Queues (colas) para procesar los elementos de trabajo y entender más a fondo las posibilidades de esta función. Mediante un ejemplo, veremos cómo utilizar varios robots para procesar datos, cómo conseguir que no procesen toda la lista de elementos en cola desde el principio sino que reanuden el trabajo en caso de que se produzca un error del sistema, etc. También dividiremos el proceso usando dos procesos diferentes. Uno crea la cola de elementos y se llama **Dispatcher** (distribuidor). Y el otro, procesa los elementos en cola previamente creados y se denomina **Performer** (ejecutor). Con este enfoque podemos cargar las transacciones utilizando el Dispatcher una sola vez y utilizar a continuación múltiples robots Performer para procesar los elementos en cola creados por el Dispatcher.

El proceso del Dispatcher

- Comience por la plantilla REFramework.
    - El Dispatcher es responsable de cargar los elementos de trabajo en la cola. Deberíamos cargar el **WIID** (ID del elemento de trabajo) en la cola para identificar de forma unívoca cada elemento de transacción.
    - Imaginemos que la flecha para pasar a la página siguiente no está disponible para WI 4, por lo que no podemos recuperar los datos de la tabla para extraer todos los elementos de trabajo. Además, si le sucediera algo a la aplicación System 1 al navegar por las páginas, el Dispatcher solucionará el error y reanudará el trabajo. También reintentará procesar las transacciones fallidas. Consideraremos que una página de la lista de elementos de trabajo es una transacción y que el número de página es el elemento de transacción.
    - El elemento de transacción es una cadena que representa el número de página que se está procesando en ese momento.
- Edite el archivo Config para el proceso actual.
    - En la hoja **Settings**, añada el valor **InHouse_Process4** en el parámetro **QueueName** . La cola se definirá en Orchestrator con el mismo nombre.
    - En la hoja **Settings** , añada los parámetros de configuración para **System1 URL** y **System1 Credential**.
    - En la hoja **Constants**, asigne a **MaxRetryNumber** el valor 2.
- Realice los siguientes cambios en el marco de trabajo:

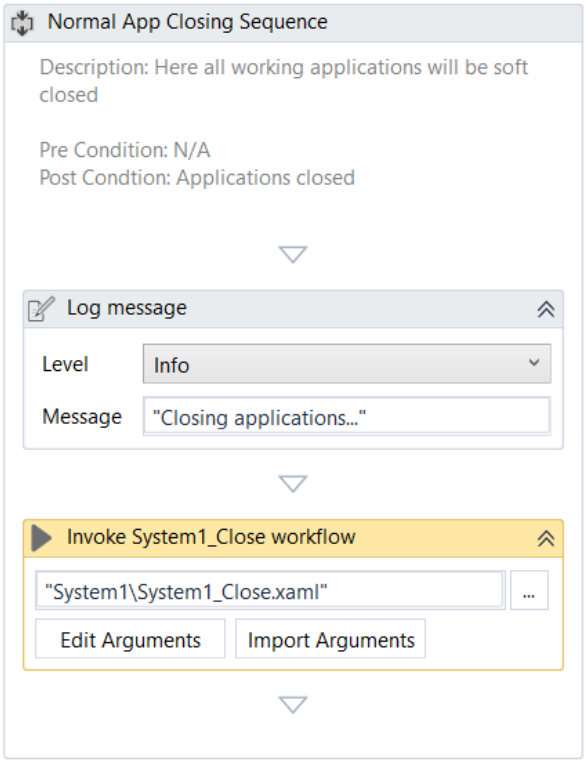
- La variable TransactionItem del archivo Main debe ser de tipo System.String . También debe asegurarse de que los tipos de argumentos en los archivos **GetTransactionData**, **Process** y **SetTransactionStatus** coinciden con el tipo de TransactionItem.

- En este ejercicio solo utilizaremos una aplicación: ACME System1. Cree una carpeta con el nombre **System1** en la carpeta raíz de la solución.
  - Pueden reutilizarse los siguientes componentes del proceso 5.
    - Copie los archivos **System1_Login.xaml**, **System1_Close.xaml** y **System1_NavigateTo_WorkItems.xaml** en la carpeta **System1**.
    - Copie el archivo **SendEmail.xaml** en la carpeta **Common**.
- Abra el archivo **InitAllApplications**.
  - Haga una llamada al archivo **System1\System1_Login.xaml**.
  - Haga una llamada al archivo **System1\System1_NavigateTo_WorkItems.xaml**. Para asegurarse de que se han recuperado todos los datos de la transacción, la página WorkItems debe estar abierta en el navegador.
  - El proyecto **InitAllAplications** debería quedar así:

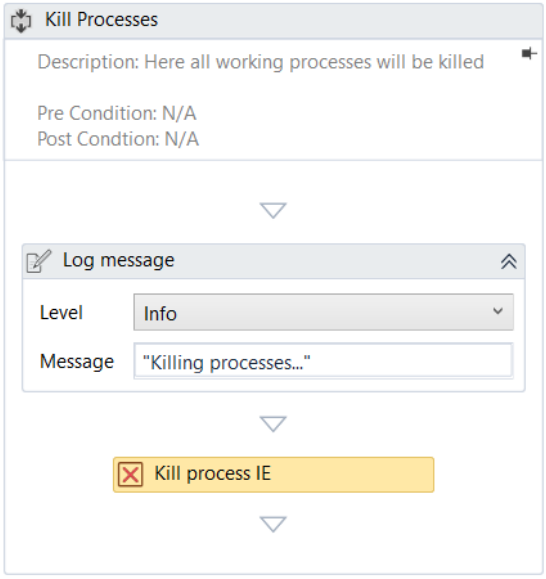


Itinerario – Creación de un informe anual para el proveedor 3

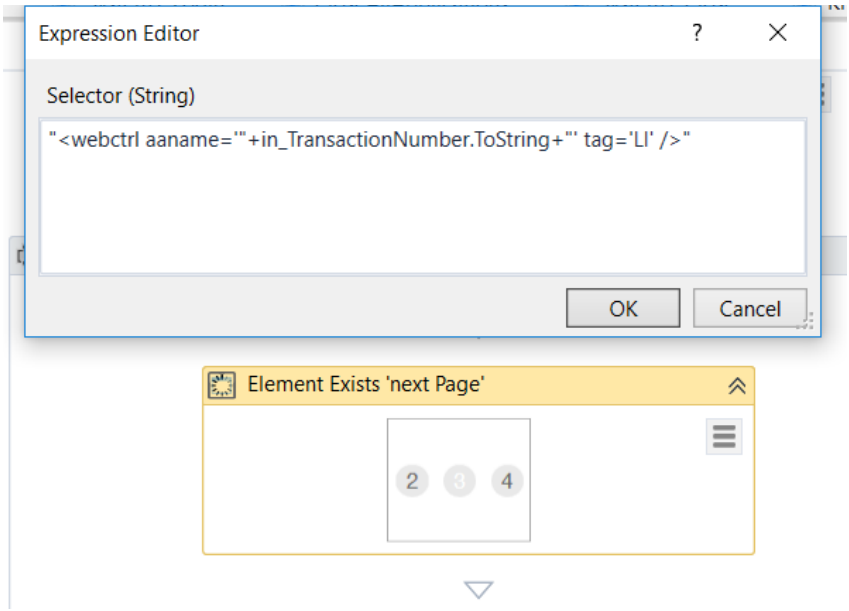
- Abra el archivo **CloseAllApplications**.
    - Haga una llamada al archivo **System1\System1_Close.xaml**.
    - El flujo de trabajo CloseAllApplications debería quedar así:



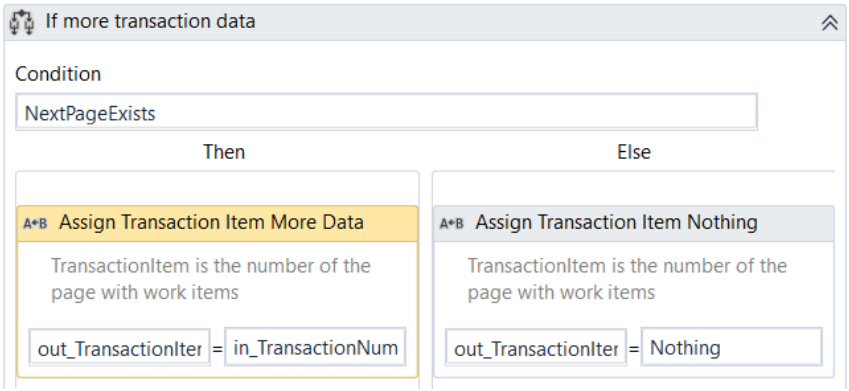
- Abra el archivo **KillAllProcesses.xaml** en la carpeta Framework.
    - Añada una actividad **Kill Process** y renómbrela como «Kill process IE».
    - Defina la propiedad **ProcessName** como «iexplore».
    - El proyecto **KillAllProcesses.xaml** debería quedar así:



- Abra el proyecto **GetTransactionData.xaml** en la carpeta Framework. Se encuentra en el estado **Get Transaction Data**.
    - Como siempre, debemos comenzar con una nota. Añada información sobre TransactionNumber: «TransactionNumber» representa el número de página, tal y como se estipuló anteriormente.
    - Elimine la actividad **Get Transaction Item**, puesto que no se necesita, ya que se utiliza el proceso Dispatcher para cargar datos en la cola.
    - Antes de la secuencia **Write Transaction info in Logging Fields**, añada una actividad **Attach Browser** y adjunte la página WorkItems.
    - Añada una actividad **Element Exists** para comprobar si la página siguiente está disponible. Indique un número de página y modifique el selector para utilizar un atributo relacionado con el número de página (recordatorio: el argumento **in_TransactionNumber** es el número de página para el distribuidor).
    - En el parámetro de salida, cree una variable booleana con el nombre **NextPageExists.**
    - La actividad **Element Exists** debería quedar así:



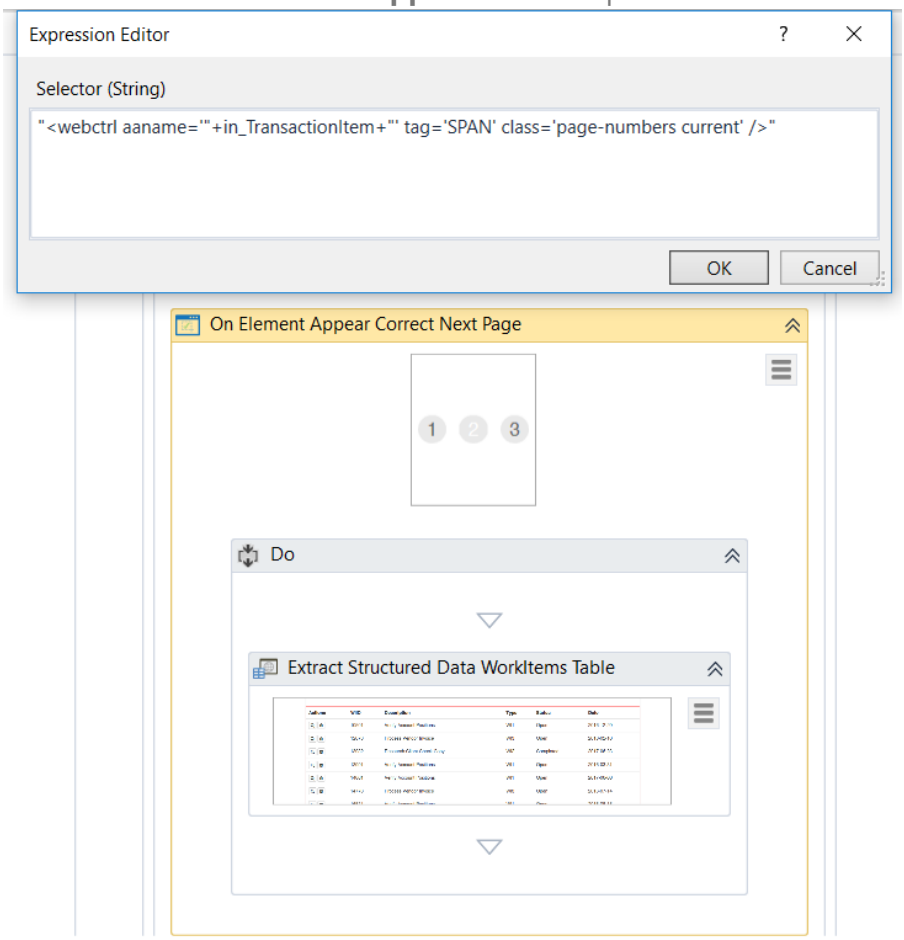
- Utilice una actividad **If** para comprobar si quedan más datos de transacción.
- Si la página siguiente existe, defina el argumento de salida **out_TransactionItem** con el valor de la página actual, es decir, **in_TransactionNumber**.
- Si la página siguiente no existe, asigne a **out_TransactionItem** el valor Nothing.
- La actividad **If** debería quedar así:



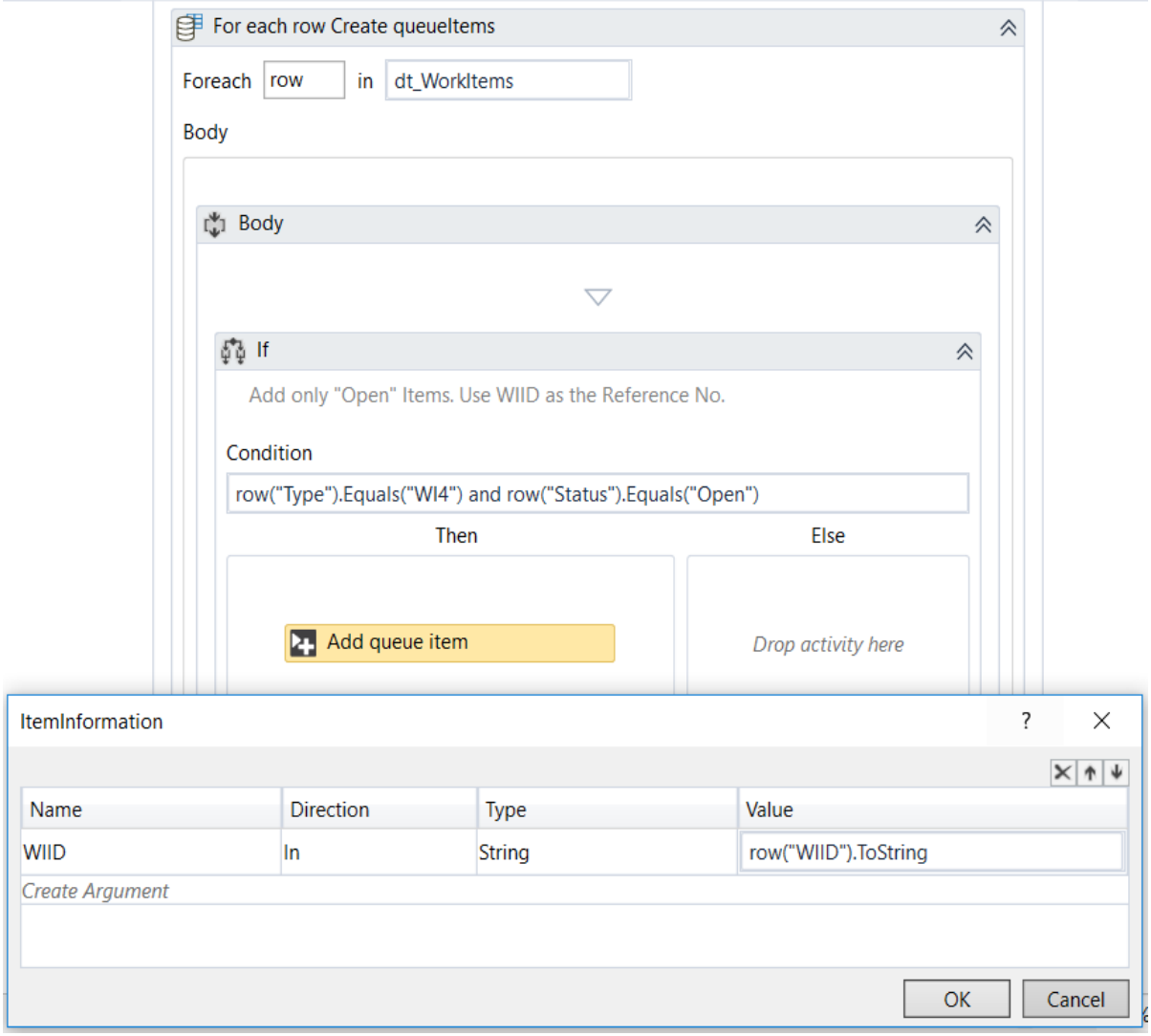
- Abra el archivo **Process.xaml** en la carpeta Framework. Se encuentra en el estado **Process Transaction**.
  - Añada una actividad **Attach Browser** e indique la página **WorkItems** de la aplicación ACME System1.
  - Utilice una actividad **Click** para seleccionar el número de página de procesamiento utilizando el mismo selector dinámico para identificar la página actual (argumento **in_TransactionNumber**).

Itinerario – Creación de un informe anual para el proveedor 6

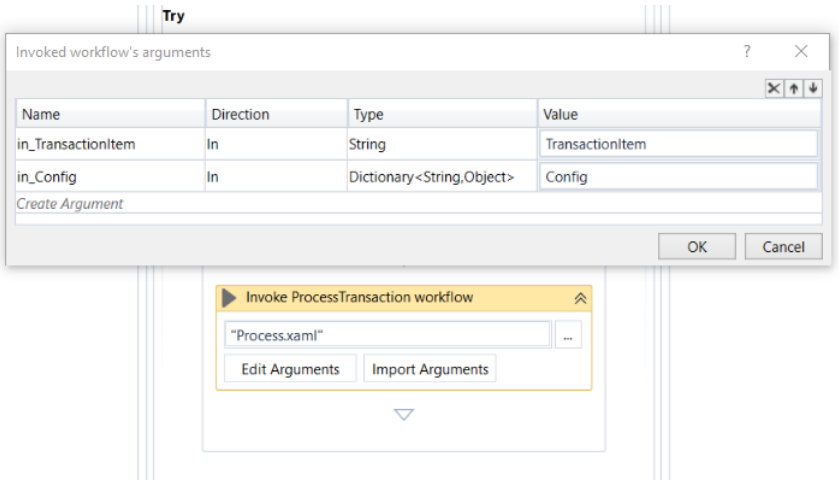
- A continuación, añada una actividad **On Element Appear** para comprobar si la página de procesamiento ya está abierta. Puede utilizar UiExplorer. El atributo de clase se puede utilizar para identificar páginas activas o inactivas, así que también se puede utilizar para crear el selector dinámico. Dentro de la actividad **On Element Appear**, extraiga la tabla que contiene los elementos de trabajo. Cree una variable con el nombre **dt_WorkItems** en la propiedad **Output** para almacenarlos.
- La actividad **On Element Appear** debería quedar así:



- El siguiente paso es cargar el **WIID** en todas las filas dentro de la tabla de datos extraídos que sean de tipo WI4 y el estado Open en la cola. Para cargar el valor en la cola, utilice una actividad **Add Queue Item**. En el panel **Properties**, utilice el valor del diccionario **in_Config** para rellenar el campo **QueueName**. En el campo **ItemInformation**, cree un argumento con el nombre **WIID**. Asigne el valor correspondiente al argumento.
- La actividad **Add Queue Item** debería quedar así:



- Abra el archivo **Main.xaml**.
    - Asegúrese de que los argumentos de la llamada al proyecto **Process.xaml** estén asignados correctamente.
        - Los argumentos deberían quedar así:



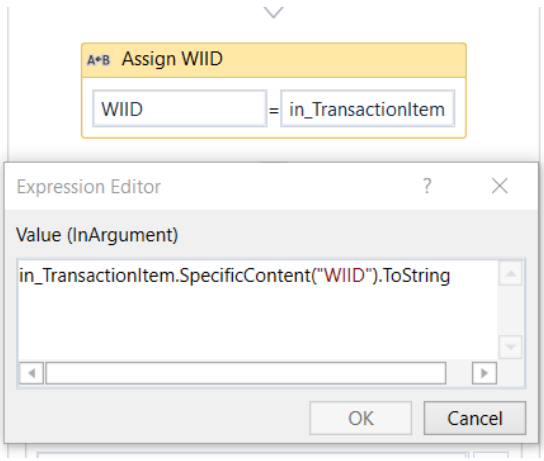
- Ya hemos concluido la implementación del proceso. A continuación debemos probar el proceso y verificar que los valores se cargan en la cola correctamente.

El proceso del Performer

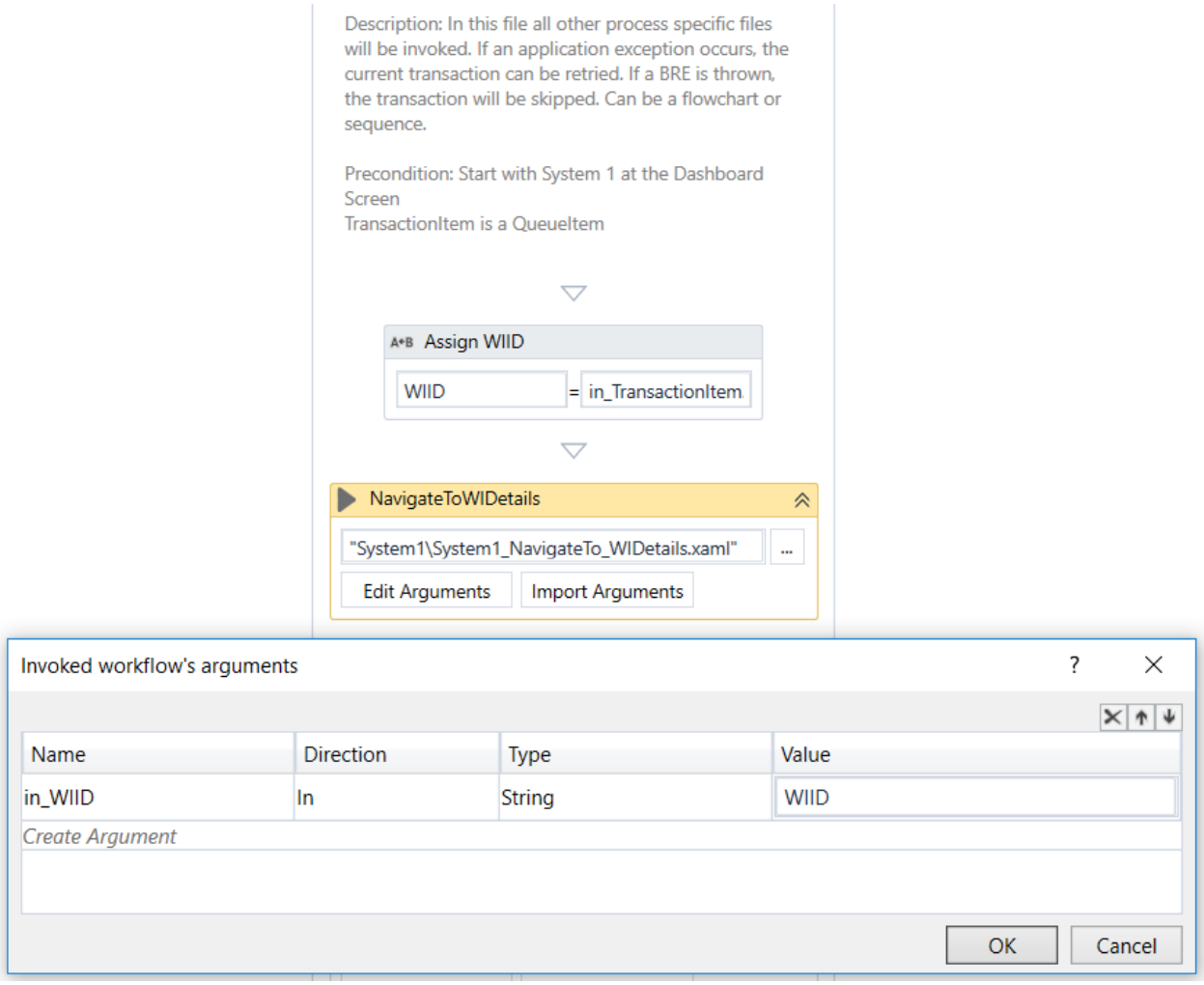
El Performer procesa todas las transacciones cargadas por el Dispatcher en la cola. Por ello, el tipo de TransactionItem debe ser **QueueItem**.

- Comience por la plantilla REFramework.
  - El argumento **TransactionItem** debe ser de tipo **QueueItem**. Este es el tipo predeterminado de TransactionItem en REFramework.
- Edite el archivo **Config** para el proceso actual como se indica a continuación:
  - En la hoja **Settings**, añada el valor «InHouse_Process4» en el parámetro **QueueName**. La cola se definirá en Orchestrator con el mismo nombre.
  - En la hoja **Settings**, añada los parámetros de configuración para **System1 URL** y **System1 Credential**.
  - En la hoja **Constants**, mantenga el valor de **MaxRetryNumber** a 0, porque estamos utilizando elementos en cola en el proceso Performer y el mecanismo de reintento se gestiona en Orchestrator.
- En este ejercicio solo utilizaremos una aplicación: ACME System1. Cree una carpeta con el nombre **System1** en la carpeta raíz de la solución.
  - Pueden reutilizarse los siguientes componentes del proceso **Performer**.
    - Copie los archivos **System1_Login.xaml**, **System1_Close.xaml**, **System1_NavigateTo_WorkItems.xaml**, **System1_NavigateTo_WIDetails.xaml** y **System1_UpdateWorkItem.xaml** en la carpeta **System1**.
    - Copie el archivo **SendEmail.xaml** en la carpeta **Common**.

- Abra el archivo **InitAllApplications**.
    - Haga una llamada al archivo **System1\System1_Login.xaml**.
- Abra el archivo **CloseAllApplications**.
    - Haga una llamada al archivo **System1\System1_Close.xaml**.
- Abra el proyecto **KillAllProcesses.xaml** en la carpeta Framework. Añada una actividad **Kill Process** y renómbrela como **Kill process IE**.
    - Defina la propiedad **ProcessName** como **iexplore**.
- Abra el proyecto **Process.xaml** en la carpeta Framework. Se encuentra en el estado **Process Transaction**.
    - Cree una variable String para definir el ID del elemento de trabajo actual (WIID).
    - Añada una actividad **Assign** y asigne la variable creada anteriormente al valor en la cola. Para ello, podemos utilizar el método SpecificContent in_TransactionItem.SpecificContent("WIID").ToString
    - La actividad **Assign** debería quedar así:

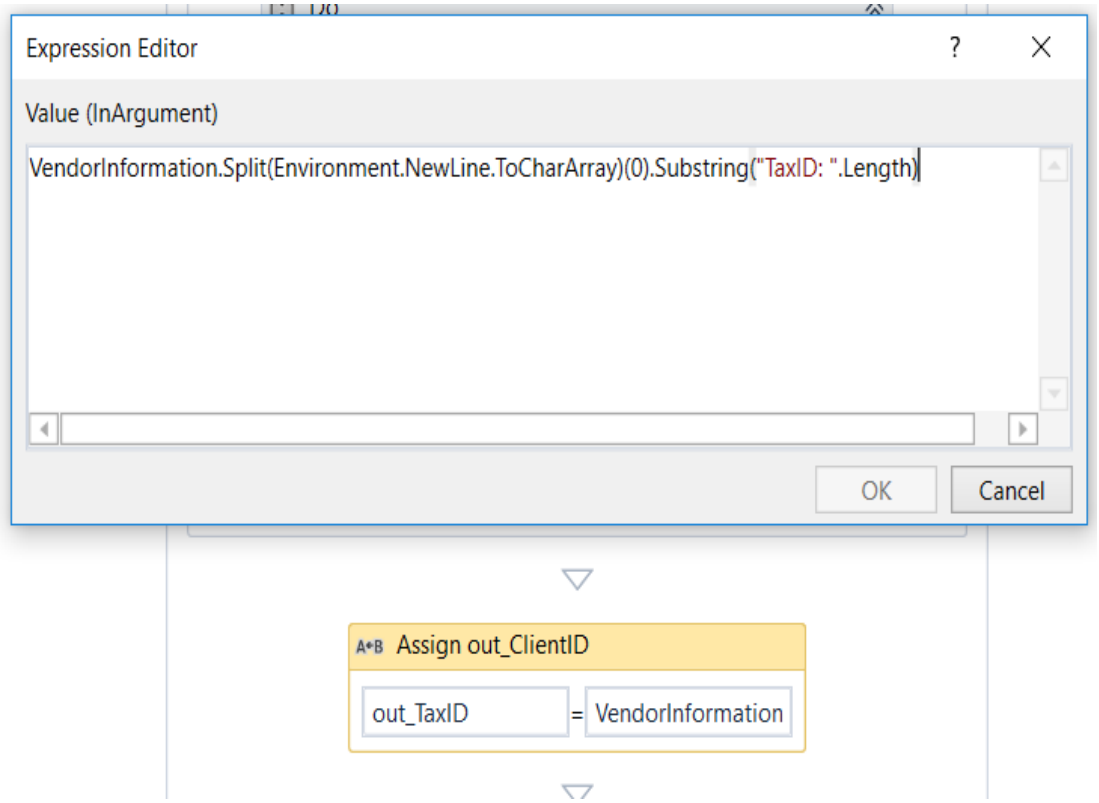


    - Abra el flujo de trabajo **Process.xaml** en la carpeta Framework. Haga una llamada a **System1\System1_NavigateTo_WIDetails.xaml**. Importe y enlace el argumento que se debe tomar de la cola utilizando el método **SpecificContent**, como se ha descrito.
    - La actividad **Invoke** y el argumento WIID deberían quedar así:

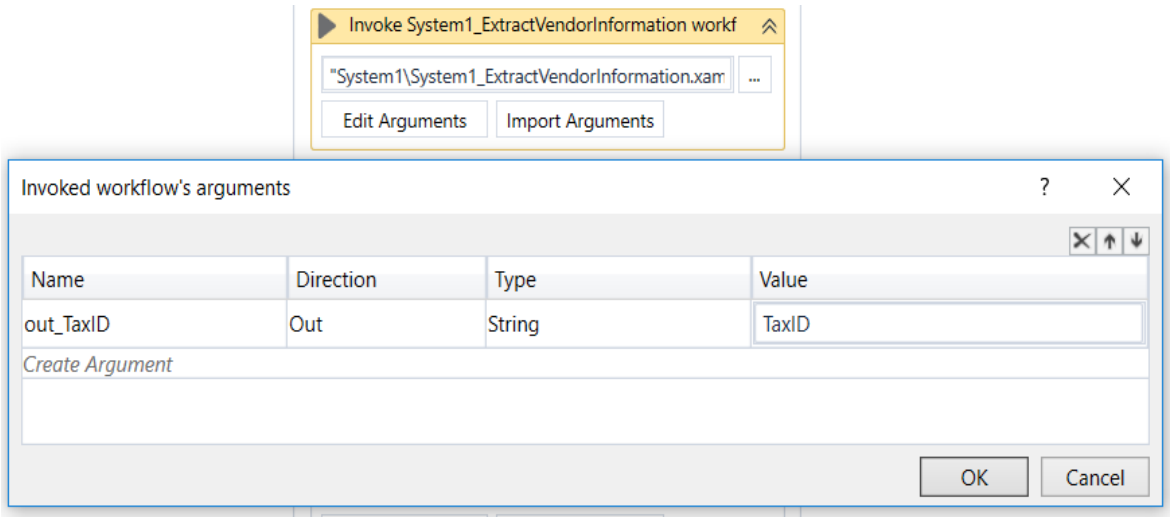


- A continuación, cree un flujo de trabajo con secuencia en blanco en la carpeta System1. Lo utilizaremos para recuperar el valor de TaxID de la página Work Item Details. Podemos dar el nombre System1_ExtractVendorInformation.xaml a este flujo de trabajo.
  - Comience con una nota. La condición previa es esta: «La página Work Item Details está abierta».
  - Cree un argumento de salida de tipo String con el nombre out_TaxID, lo utilizaremos más adelante en el proyecto.
  - Añada una actividad **On Element Appear** e indique el párrafo **Vendor Information** en la página Work Item Details. Asigne a la propiedad **RepeatForever** el valor False.
  - Añada una actividad **Get Text** en la secuencia **Do** que forma parte de **On Element Appear**. Indique el párrafo mencionado anteriormente para recuperar el texto en él.

- En el panel **Properties** de la actividad **Get Text**, cree una variable con el nombre VendorInformation en el campo **Output**.
- A continuación, detrás de la actividad **On Element Appear**, añada una actividad **Assign** para asignar el valor del argumento out_TaxID al valor de TaxID, que se puede recuperar utilizando la variable ya creada VendorInformation.
- La actividad **Assign** debería quedar así:



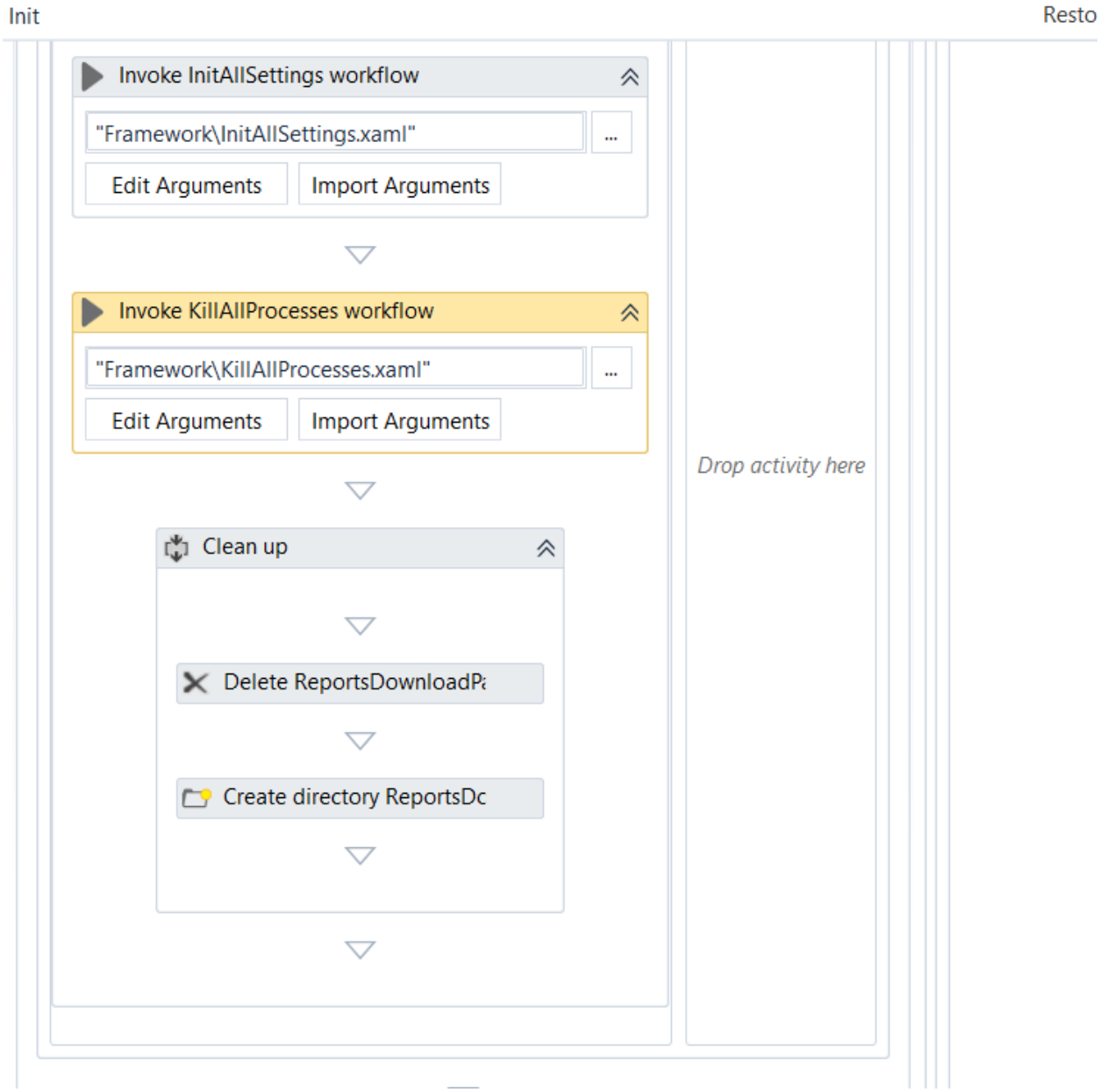
- Abra **Process.xaml** y cree una variable String con el nombre TaxID para almacenar el valor del argumento de salida procedente del archivo de flujo de trabajo creado anteriormente. Haga una llamada a **System1\System1_ExtractVendorInformation.xaml** y enlace el argumento.
- La llamada al flujo de trabajo debería quedar así:



- A continuación, cree un flujo de trabajo para ir hasta la página del panel de control. El flujo de trabajo debe incluir una actividad **Click** para seleccionar la página del panel de control. Podemos dar el nombre **System1_NavigateTo_Dashboard.xaml** a este flujo de trabajo.
- Haga una llamada al flujo de trabajo en el archivo **Process.xaml**.
- Una vez recuperado el valor de TaxID, tenemos que ir a la página **Download Monthly Report**. Para ello, creemos una secuencia en blanco con el nombre **System1_NavigateTo_MonthlyReport**.
    - Añada una nota específica a esta secuencia. La condición previa es que la página del panel de control esté abierta.
    - Añada una actividad **Attach Browser** e indique la página del panel de control.
    - Añada 2 actividades **Click** para ir a **Reports** y después a **Download Monthly Report**.
    - En el panel **Properties** de estas dos actividades, marque la casilla en del campo **Simulate Click**. El menú **Reports** solo se visualiza si pasa el puntero del ratón sobre el botón **Reports**. Para hacer clic en **Download Monthly Report**, utilice UiExplorer, haga una pausa de 3 segundos utilizando la tecla **F2** y verifique que el elemento está visualizado en la pantalla antes de indicar el botón.
- La descarga de facturas existentes puede dar algunos problemas. No obstante, para evitar posibles excepciones, debemos verificar que el entorno esté limpio cada vez que se inicie el robot. Para ello, podemos eliminar la carpeta **Download Reports** mencionada en el archivo **Config** y después volver a crearla de cero. Abra el estado Init en el archivo Main.xaml. Añada

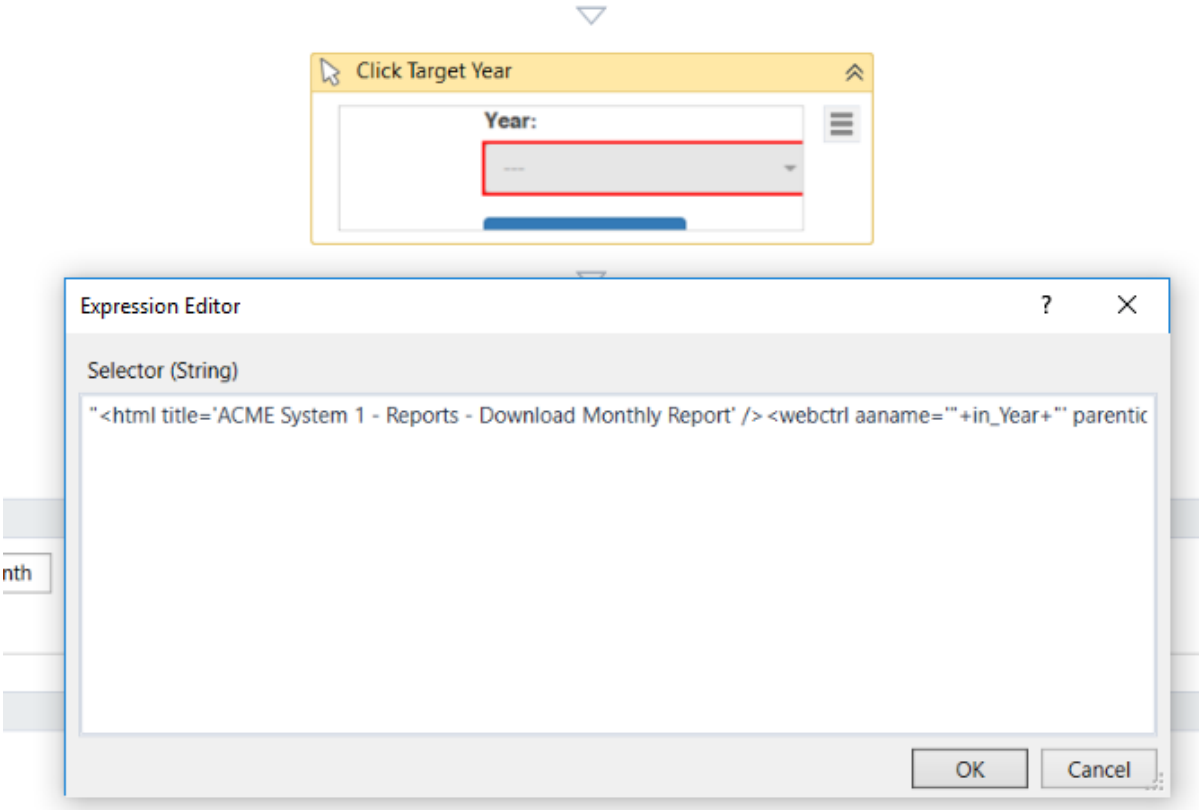
una secuencia con el nombre **Clean Up** detrás de la llamada al archivo KillAllProcesses.xaml. Utilizaremos dos actividades: **Delete** y **Create Directory**.

- En el panel **Properties** de la actividad **Delete**, asigne el campo **Path** al valor en el archivo **Config**. De este modo, siempre podrá tener un nuevo directorio Data\Temp vacío cuando se ejecuta el robot. La secuencia Clean Up debería quedar así:



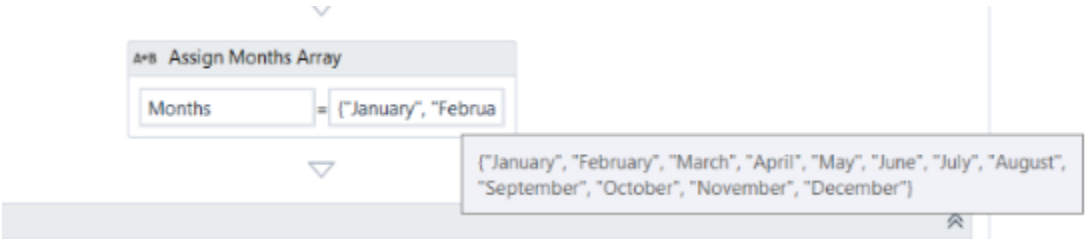
- A continuación, volviendo al archivo **Process.xaml**, haga una llamada al archivo **System1\System1_NavigateTo_MonthlyReport.xaml** creado anteriormente.

- Cree una variable con el nombre **ReportYear.** Utilice una actividad **Assign** para definir su valor como el año anterior.
- El paso siguiente es crear el informe anual. Para ello, tenemos que crear un archivo de secuencia en blanco con el nombre **System1_CreateYearlyReport.xaml**.
    - Comience con una nota adecuada. La condición previa es que la página Monthly Report esté abierta en la aplicación ACME System 1.
    - Cree tres argumentos de entrada del modo siguiente:
        - **in_TaxID**, procedente del archivo principal. Almacena el valor de TaxID.
        - **in_Year**, procedente del archivo principal. Almacena el año para el que se va a crear el informe.
        - **in_ReportsDownloadPath**, la carpeta donde se van a descargar los informes mensuales.
    - Cree un argumento de salida con el nombre **out_YearlyReportPath** para guardar la ruta de acceso al archivo del informe anual creado después de combinar todos los informes mensuales.
    - Añada un nuevo elemento en el archivo Config, en la hoja Settings, para indicar la ruta de acceso de la carpeta donde se descargan los informes. Complete el campo Name escribiendo **ReportsDownloadPath** y el campo Value con **Data\Temp**.
    - Cree una variable de tabla de datos con el nombre dt_YearlyReport. Se utiliza para combinar todos los informes mensuales. Asigne su valor a **new Datatable** utilizando una actividad **Assign**.
    - A continuación, añada una actividad **Type Into** para escribir el valor de TaxID en la aplicación ACME System 1. La página Monthly Report debería estar ya abierta en este paso.
    - Añada una actividad **Click** para seleccionar el año. Active la propiedad **Simulate Click** y después seleccione el año de destino. De este modo, es posible ejecutar la actividad en segundo plano aunque el menú desplegable no esté abierto y el elemento no sea visible. Cambie el atributo aaname del selector del argumento **in_Year**. La actividad **Click** y el selector deberían quedar así:



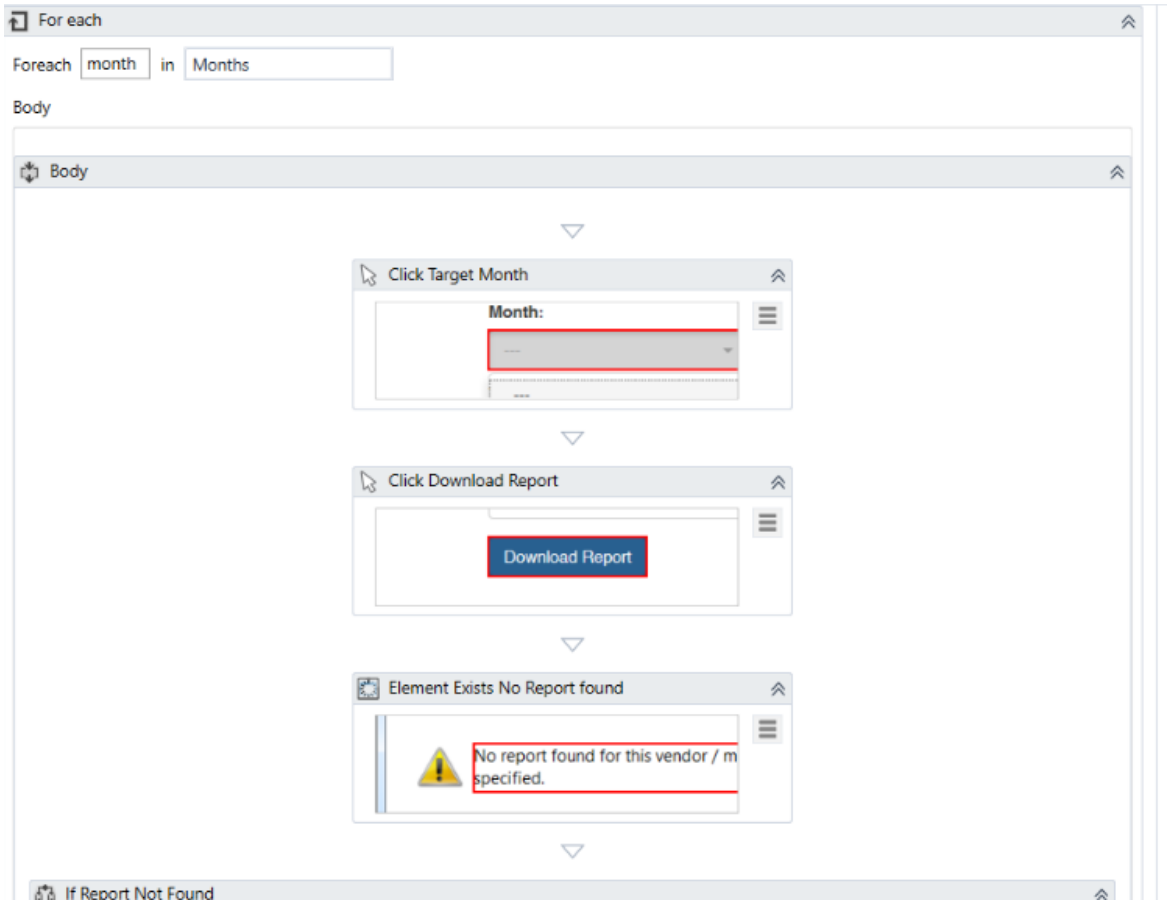
- Otra opción es utilizar una actividad **Select** en lugar de la actividad **Click**. En este caso, verifique que se haya actualizado el selector para indicar el año, utilizando UiExplorer.

- Cree una variable matriz de Strings con el nombre **Months**. Utilice una actividad **Assign** para asignar sus valores según las opciones de la lista desplegable **Month**. La actividad **Assign** debería quedar así:



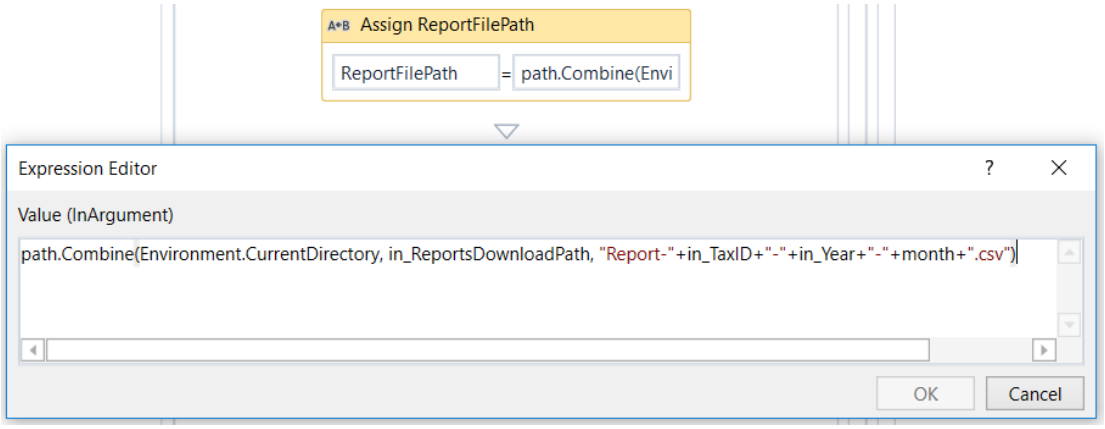
- A continuación, para descargar los informes de cada mes, necesitamos añadir una actividad **For Each** para recorrer toda la matriz Months, seleccionar el mes especificado en el cuadro desplegable y descargar el informe.

- Dentro de la actividad **For Each**, añada una actividad **Click** para seleccionar el mes de destino. Edite el selector de forma similar al selector del año, utilizando el atributo dinámico aaname. Como alternativa, puede utilizar una actividad **Select**.
- A continuación, añada una actividad **Click** e indique el botón Download. Seleccione la propiedad **Simulate Click**.
- Dado que algunos informes no existen, añada una actividad **Element Exists** e indique la etiqueta de la ventana emergente que se muestra en este caso. Cree una variable booleana con el nombre ReportNotFound en la propiedad O**utput**.
- Por ahora, la actividad **For Each** debería quedar así:



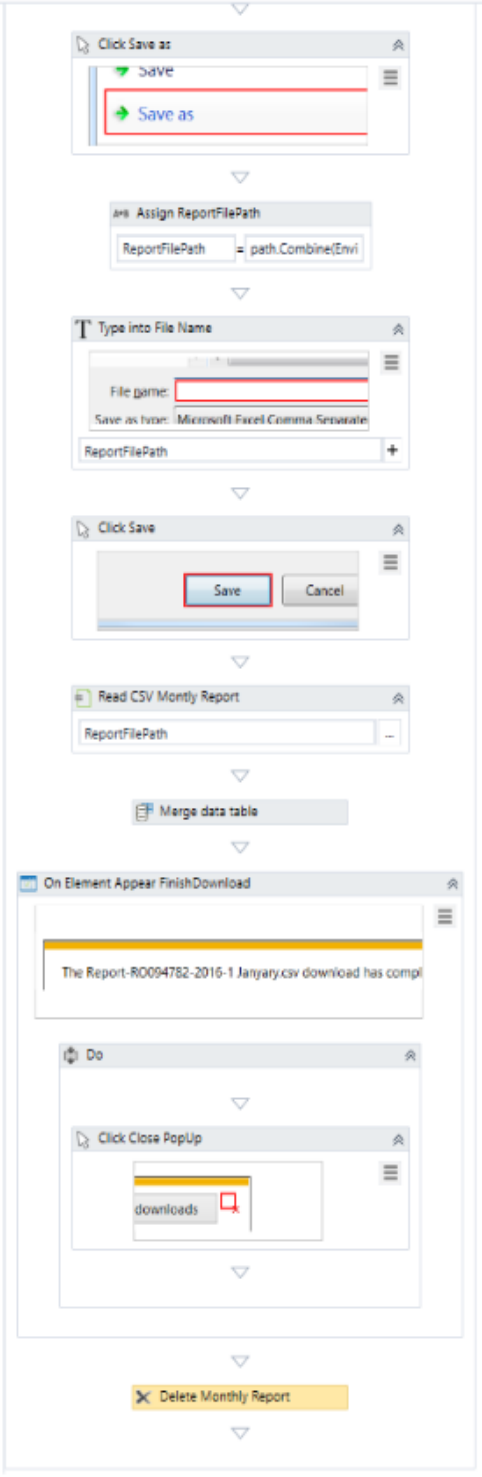
añadir una actividad **If** para comprobar si se ha encontrado el informe. Utilice la variable ReportNotFound en **Condition**. En la sección **Then**, añada una actividad **Click** y defina su objetivo en el botón **OK** de la ventana emergente, para pasar al mes siguiente.

- En la sección Else, descargue el informe utilizando las actividades siguientes:
    - Añada una actividad **Click** y diríjala hacia el botón **Save as**. Active la propiedad **Simulate Click**.
    - Añada una actividad **Assign**. Cree una variable con el nombre ReportFilePath y asigne su valor a la ruta de acceso y el nombre de archivo del informe mensual que se ha descargado en formato .csv.



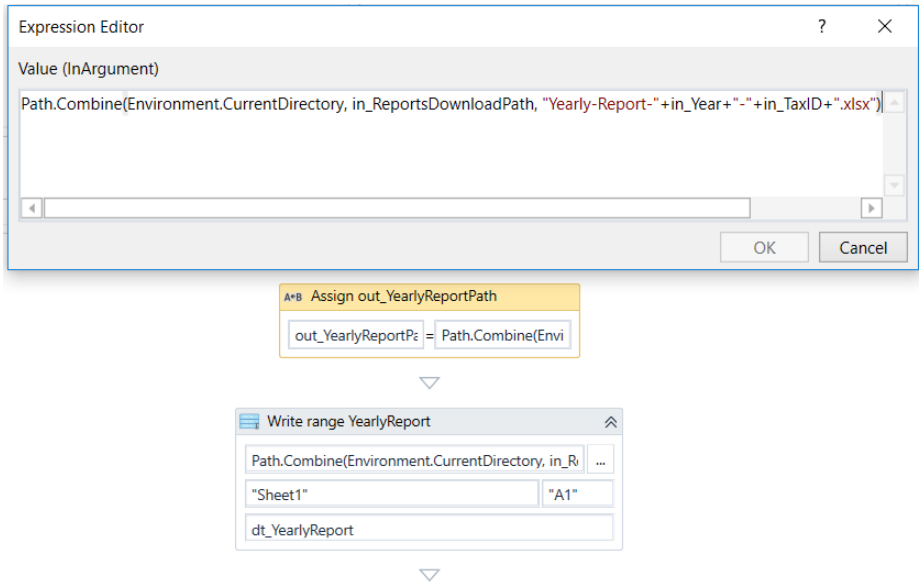
    - Utilizando una actividad **Type Into**, rellene el valor de ReportFilePath en el campo File Name de la ventana Save As. Seleccione la propiedad **Simulate** Type.
    - Añada una actividad **Click** y diríjala hacia el botón **Save**. Active la propiedad **Simulate Click**.
    - Lea el archivo csv que se ha descargado. En la propiedad Output cree una variable de tabla de datos con el nombre dt_MonthlyReport.
    - A continuación, utilizando una actividad **Merge Data Table**, añada los valores de **dt_MonthlyReport** a la tabla de datos **dt_YearlyReport**.
    - La descarga de un informe mensual no siempre tarda el mismo tiempo. Para verificar que el archivo está totalmente descargado antes de descargar el siguiente informe mensual, añada una actividad **On Element Appear** e indique el elemento emergente de descarga. Actualice el selector con un carácter comodín para darle valores dinámicos de atributo. Active la propiedad **Wait Visible**.

- En la sección **Do** de la actividad **On Element Appear**, añada un **Click** para cerrar el elemento emergente. Seleccione la propiedad **Simulate Click**.
- Añada una actividad **Delete File** para eliminar el archivo del informe mensual antes de descargar el siguiente.
- La sección **Else** debería quedar así:

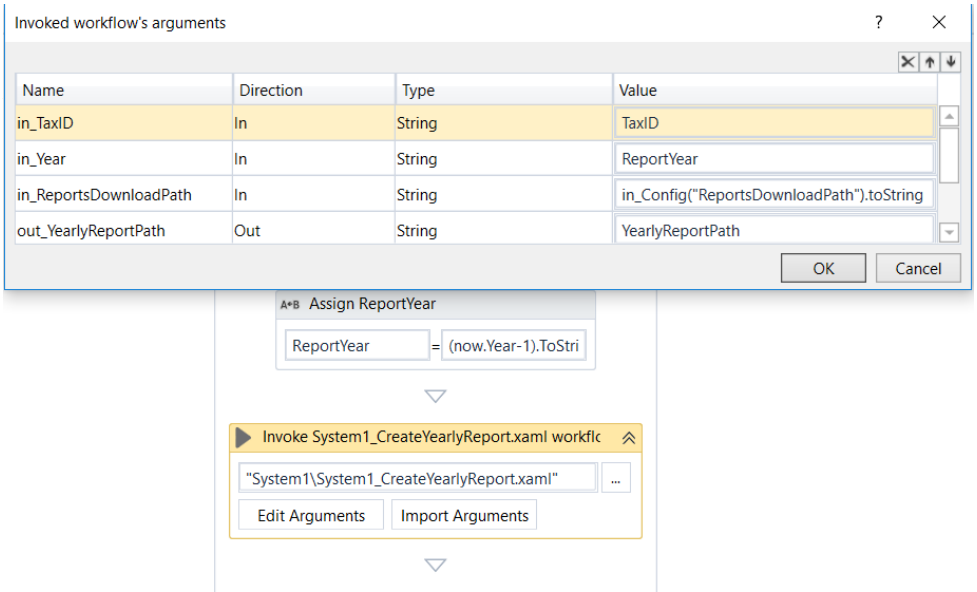


- La siguiente actividad de la secuencia es **Assign**, que se utiliza para definir el valor del argumento **out_YearlyReportPath**. Asegúrese de

que el nombre del archivo Excel del informe anual cumpla el modelo del archivo PDD y que la ruta de acceso sea la del archivo Config.

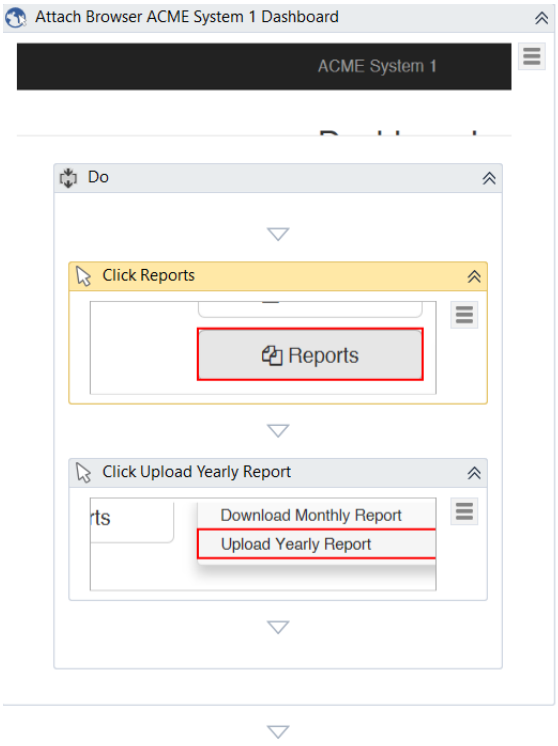


- Defina valores predeterminados para los argumentos de entrada y haga una prueba del flujo de trabajo.

- Vuelva al flujo de trabajo **Process**.
  - Haga una llamada al archivo **System1\System1_CreateYearlyReport.xaml** creado anteriormente. Importe y enlace los argumentos.
  - Cree una variable String con el nombre YearlyReportPath. Se utiliza para obtener el valor del argumento **out_ yearlyReportPath** en el flujo de trabajo anterior.
  - La llamada y los argumentos deberían quedar así:
  - 

ón, haga una llamada a **System1\System1_NavigateTo_Dashboard.xaml** para volver a la página del panel de control.

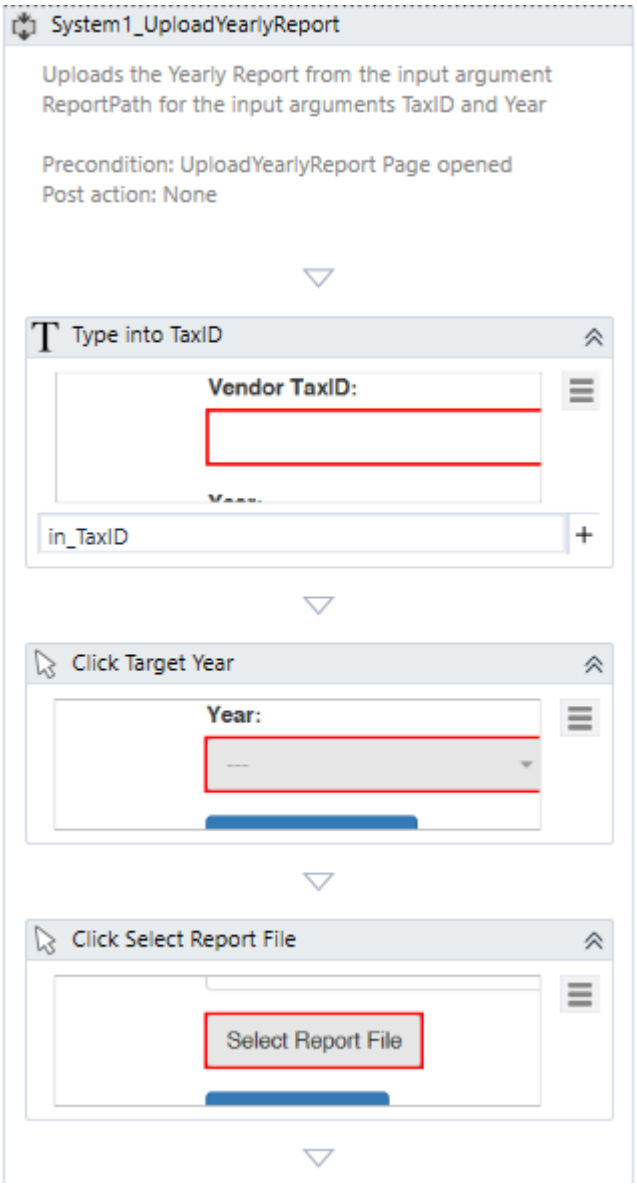
- Una vez creado el archivo del informe anual, tenemos que ir a la página Reports - Upload Yearly Report de la aplicación ACME System 1. Para ello tenemos que crear una secuencia en blanco con el nombre **System1_NavigateTo_UploadYearlyReport**.
  - Inicie el nuevo flujo de trabajo añadiendo una nota. La condición previa es que la página del panel de control esté abierta.
  - Añada una actividad **Attach Browser** e indique la página del panel de control.
  - Añada una actividad **Click** para seleccionar el botón **Reports**. Active la propiedad **SimulateClick**.
  - A continuación, añada otra actividad **Click** para seleccionar el botón **Upload Yearly Report**. Utilice **UiExplorer** para hacer clic en este botón, como en el flujo de trabajo **System1_NavigateTo_MonthlyReport.xaml**. Active la propiedad **SimulateClick**.
  - El flujo de trabajo debería quedar así:



- Vuelva al flujo de trabajo **Process**.

Itinerario – Creación de un informe anual para el proveedor 22

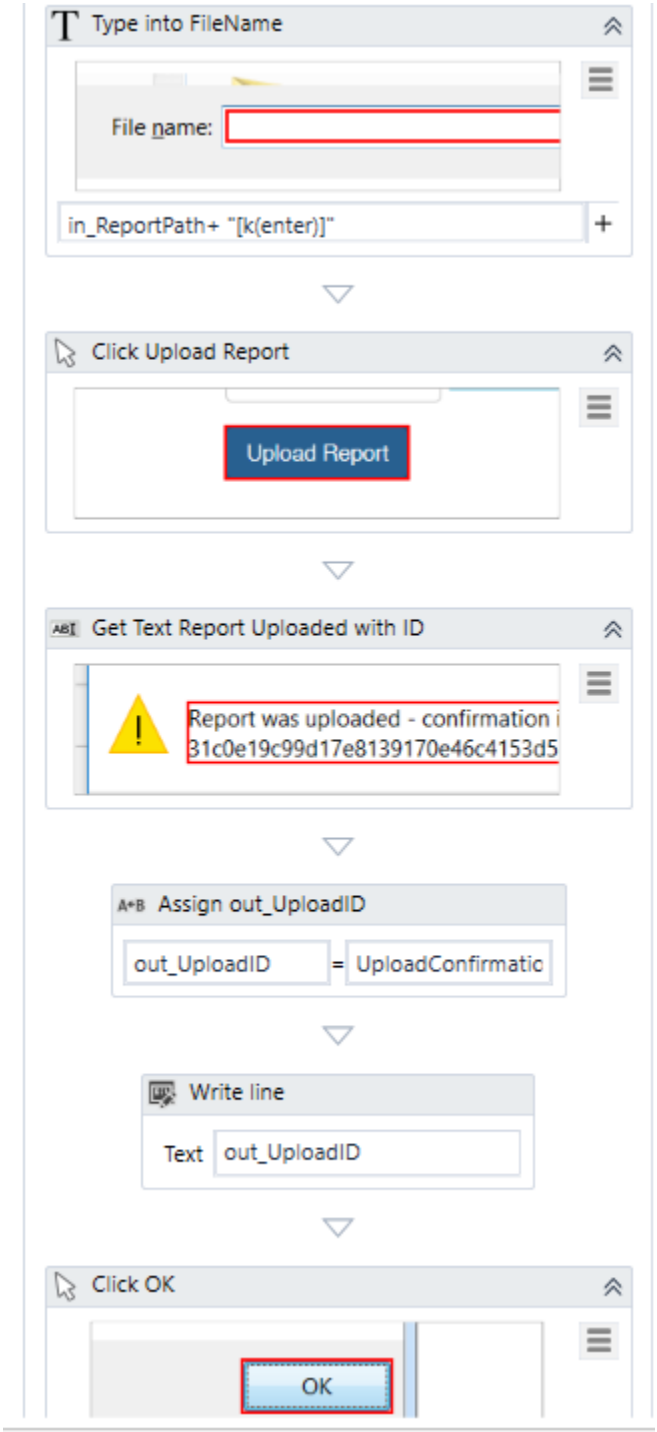
- Haga una llamada al archivo **System1\System1_NavigateTo_UploadYearlyReport.xaml**.
- Después de ir a la página **Reports - Upload Yearly Report**, debemos cargar el informe anual. Para ello, creemos una secuencia en blanco con el nombre **System1_UploadYearlyReport**.
  - Inicie la nueva secuencia añadiendo una nota. La condición previa es que la página Reports - Upload Yearly Report esté abierta.
  - La información necesaria para cargar el archivo del informe anual es taxID, ruta de acceso al archivo y año. Cuando se ejecuta la carga, se genera un ID de confirmación. Por tanto, en este flujo de trabajo debemos utilizar 3 argumentos de tipo String: **in_TaxID**, **in_ReportPath** e **in_Year**, y un argumento de salida String: **out_UploadID**.
  - Con todo esto preparado, añada una actividad **Type Into** e indique el campo **Vendor TaxID**. Utilice el argumento **in_TaxID** como texto.
  - Utilice una actividad **Click** en el menú desplegable **Year**. Actualice el selector utilizando el argumento **in_Year**.
  - Utilice la actividad **Click** en el botón **Select Report File**.
  - Por el momento, la secuencia debería quedar así:



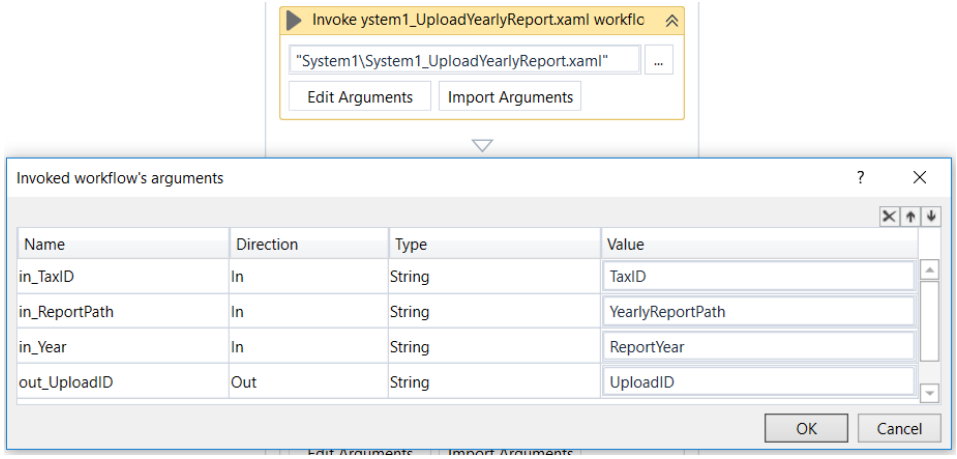
- Ahora, se visualiza la ventana **Choose file to Upload**, así que tenemos que utilizar una actividad Type Into para asignar la ruta de acceso al archivo del informe anual (**in_ReportPath**) y pulsar Intro.
- Seleccione el botón Upload utilizando una actividad **Click**.
- Se visualiza una ventana emergente con el ID de confirmación de carga, así que utilicemos una actividad **Get Text** para recuperar su valor. En la propiedad **Output**, cree una variable con el nombre UploadConfirmation.
- Asigne el valor del argumento **out_UploadID** según el ID de confirmación utilizando una actividad **Assign**. Utilice el método Substring para recuperar el valor del modo siguiente:

UploadConfirmation.Substring("Se ha cargado el informe, el id. de confirmación es ".Length)

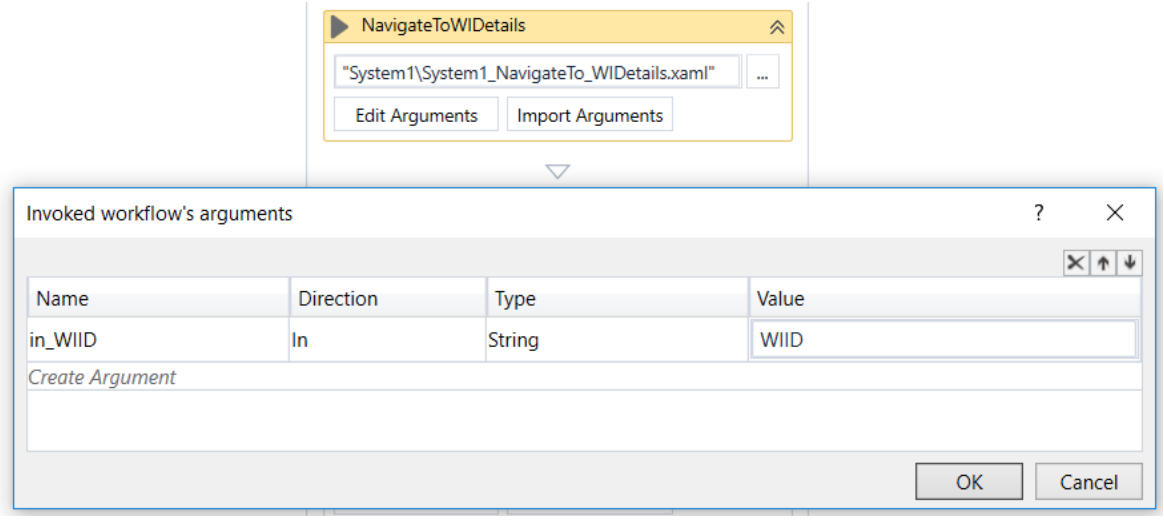
- Utilice una actividad **Click** para seleccionar el botón **OK**. Ya hemos concluido.
- La última parte del flujo de trabajo debería quedar así:



- Vuelva al flujo de trabajo **Process**.
    - Haga una llamada al archivo **System1\System1_UploadYearlyReport.xaml** y enlace los argumentos correspondientes. Cree una variable en el flujo de trabajo **Process** para almacenar el valor del argumento **out_UploadID**. Asigne a la variable el nombre UploadID.
    - La llamada debería quedar así:



    - Al hacerlo, habremos cargado el archivo del informe anual y por tanto debemos actualizar el estado de los elementos de trabajo.
    - Haga una llamada al archivo **System1\System1_NavigateTo_Dashboard.xaml** para ir a la página del panel de control.
    - Haga una llamada al archivo **System1\System1_NavigateTo_WIDetails.xaml** para ir a la página Details de un elemento de trabajo concreto. Como hemos visto, hay un argumento en este flujo de trabajo: WIID.



- Actualice el estado del elemento de trabajo a **Complete** haciendo una llamada al archivo **System1\System1_UpdateWorkItem.xaml**. Compruebe que se hayan asignado los argumentos correctos en las secciones **Comment** y **Status**. Como se ha indicado en el archivo PDD, cambie el estado a **Completed** y el valor de Comment a **Uploaded with ID [uploadID]**.
- Para concluir debemos dejar la aplicación en su estado inicial para poder procesar el siguiente elemento. Para ello, debe hacer una llamada al archivo **System1\System1_NavigateTo_Dashboard.xaml** para volver a la página del panel de control.

- Ya hemos concluido la implementación del proceso. A continuación debemos probar todo el proceso. Ya debería haber probado todos los flujos de trabajo por separado después de haberlos creado utilizando valores predeterminados para los argumentos.
  - Ejecute el flujo de trabajo **Main** varias veces y compruebe que se ejecuta correctamente en todos los casos. Si no es así, corrija los errores y vuelva a ejecutarlo.
  - Use la opción **Reset test data** en el menú User options para generar un nuevo conjunto de datos para realizar la prueba.
  - Utilice el **Dispatcher** para cargar nuevos elementos en la cola si fuera necesario.

Notas de implementación del proceso

Hemos empezado con un proceso **Dispatcher** que se utilizó para cargar elementos de cola en Orchestrator. Después hemos procesado cada elemento de la cola utilizando el **Performer**. Observe que el estado de una transacción en la cola cambia después de ser procesada. Todos los elementos son independientes entre sí y se pueden procesar en paralelo utilizando varios robots Performer.