Hands-On Lab 2015

Introducción

El objetivo de este laboratorio es descubrir las ventajas que ofrece GXtest en la automatización de pruebas funcionales, construyendo de manera sencilla un conjunto breve de casos de prueba automatizados.

Para la realización del taller se utilizará la aplicación Ajax Sample, la cual forma parte de las aplicaciones de ejemplo de GeneXus. (http://samples.genexus.com/ajaxsample/home.aspx)

Todos los archivos sobre los que se hace referencia en el laboratorio, se encuentran en el escritorio, en la carpeta **Material GXtest.**

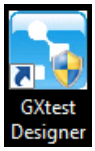
A continuación le acercamos links de utilidad sobre GXtest y Abstracta. En particular, en nuestra wiki encontrará documentación y tutoriales de interés de GXtest.

Crear un Proyecto

¿Cómo podemos trabajar con GXtest si tenemos varias aplicaciones para probar, con diferentes versiones y ambientes?
Cuando trabajamos con GXtest, organizamos nuestro trabajo en proyectos de test, permitiendo crear uno o más proyectos de automatización en GXtest.

De esta manera podremos trabajar ordenadamente, asignando a cada proyecto una aplicación con una versión y su correspondiente ambiente, y se podrá trabajar de manera independiente sobre cada proyecto.
Los proyectos en GXtest se crean en el componente GXtest Designer, y para poder crear el primer caso de prueba, lo primero que debemos hacer es crear el proyecto.

Abrimos **GXtest Designer**, seleccionando el acceso directo en nuestro escritorio que luce como la siguiente imagen:

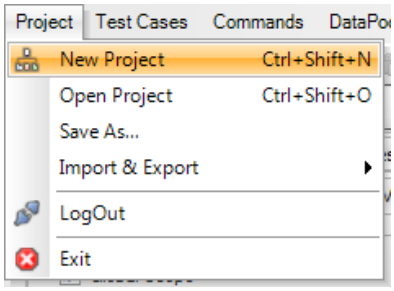


Nos vamos a encontrar con la página de inicio:

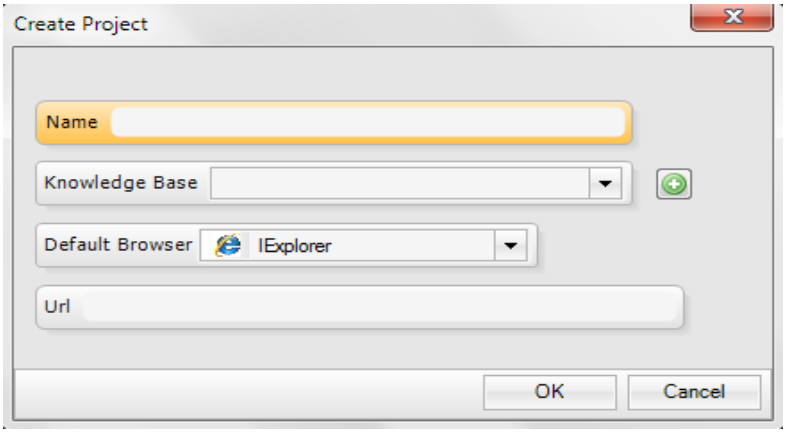
En esta página tenemos acceso a los manuales de usuario, y documentación de soporte de GXtest disponible en la Wiki de GXtest, que nos van a ser de gran utilidad durante y luego del curso cuando trabajen con GXtest en su realidad de negocio.

Debemos indicarle a GXtest que vamos a crear un nuevo proyecto de la siguiente manera:

Debe ir a **Project->New Project**.



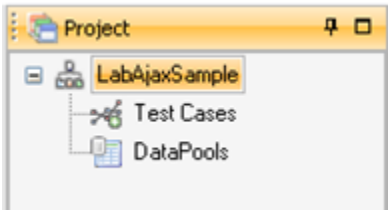
Aparecerá una ventana como la siguiente:



- *Campo "Name":* Ingresar el nombre del proyecto, por ejemplo ***LabAjaxSample***.
- *Campo "Knowledge Base":* ahora se debe asociar la Knowledge Base (KB) de la aplicación a automatizar. Haga click en el botón "+" e indique el archivo llamado "**Curso GXtest.*gxt***" ubicado en la carpeta Material GXtest del escritorio. Luego de unos segundos, GXtest informará que se importó correctamente la KB.
- *Campo "Url":* indica donde está publicada la aplicación que deseamos testear. Ingresar la siguiente url: **http://localhost/LabGXtest/home.aspx**

Para finalizar, presione el botón Ok y se creará el proyecto.

Podrá ver en este video los pasos antes descritos para crear un nuevo proyecto en GXtest.



A la derecha en la pantalla, tenemos el árbol del proyecto, el cual contiene un nodo principal con el nombre del proyecto y dos contenedores: *Test Cases* que agrupará los casos de prueba, y *Datapools* que agrupará los archivos con los datos asociados a los casos de prueba.

Seguramente si Ud. no cuenta con conceptos de programación en GeneXus, se habrá hecho la siguiente pregunta:

¿Qué es una KB?

Como vimos es necesario asociar nuestro proyecto en *GXtest* con al menos una KB para poder comenzar a trabajar.

KB es la abreviación de *Knowledge Base*, es el concepto de proyecto en GXtest pero en *GeneXus*.

Ésta se crea al momento de comenzar a trabajar en *GeneXus* para crear una aplicación.

Es en ella donde se definen todos los objetos, pantallas con sus controles y lógica asociada, que van a formar una aplicación.

> **Tip:** Veremos más adelante que tener nuestro proyecto asociado a la información de la KB nos proporciona grandes ventajas sobre otras herramientas de automatización del mercado, principalmente a la hora de mantener los casos de prueba funcionando.

Explorar la aplicación de ejemplo

Trabajemos ahora con una aplicación de ejemplo de GeneXus llamada “Ajax Sample”, en donde automatizaremos diferentes pruebas durante el Lab. Es un sencillo sistema web de facturación desarrollado en GeneXus Ev2. En ella podemos dar de alta, baja, editar y consultar varias entidades como Clientes, Ciudades, Países, Productos y Facturas.

Para conocerla vayamos a la url de la aplicación y recorramos brevemente sus funcionalidades, creando, editando y eliminando distintas entidades.

http://localhost/LabGXtest/home.aspx

Una vez recorrida e investigada la aplicación sobre la que vamos a trabajar, podemos setear los datos de la aplicación a sus valores iniciales, desde la siguiente url:

http://localhost/LabGXtest/dentry.aspx

Presione los botones en la pantalla en el siguiente orden, de modo de inicializar los datos de la aplicación:

Ingrese a la pantalla de trabajar con clientes, y valide que el cliente "Carlos Perez" existe y aparece solamente una vez en la lista (puede hacerlo rápidamente ingresando el nombre "Carlos" en el filtro *First Name* que se ubica arriba de la grilla). En caso que no aparezca o esté duplicado, vuelva a inicializar los datos.

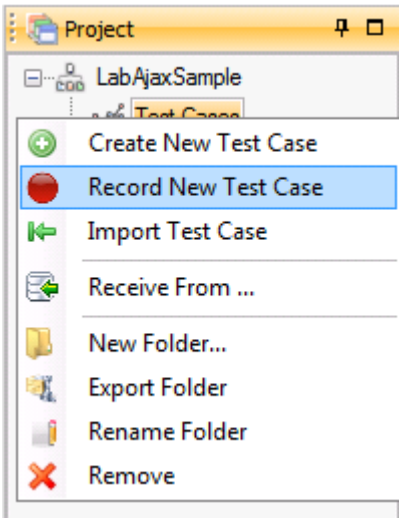
Grabar un nuevo caso de prueba

Nos disponemos entonces a grabar nuestro primer caso de prueba, en el proyecto AjaxSample que acabamos de crear, que consiste en dar de alta una nueva factura en el sistema.

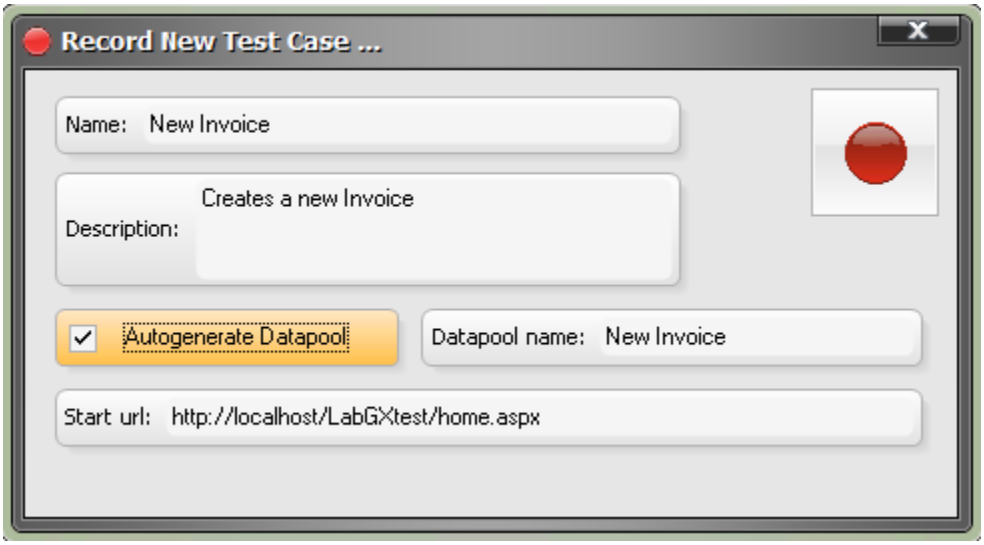
Antes de comenzar, sepa los pasos a ejecutar en el caso de prueba:

1. Ingresar a la aplicación.
2. Acceder a la funcionalidad "Work with Invoices" por el menú.
3. Seleccionar el botón "+" para agregar un nuevo pedido.
4. Ingresar nombre de la empresa.
5. Ingresar una línea de la factura indicando:
    a. Producto
    b. Cantidad
6. Validar que el subtotal, los impuestos y el total sean correctos.
7. Confirmar.

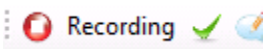
Procedamos entonces a grabar el caso de prueba. Seleccionamos la opción para comenzar a grabar desde GXtest Designer, haciendo clic derecho sobre el nodo Test Cases -> Record New Test Case:



En la ventana que aparece, ingresamos el nombre del caso de prueba "**New Invoice"**, una descripción de la prueba, seleccionamos el checkbox "**Autogenerate Datapool"**, y dejamos la URL que aparece por defecto.
Hacemos clic en el botón rojo y en un instante se abrirá el navegador Internet Explorer.

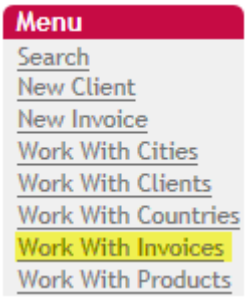


Observar que en la barra de herramientas de Internet Explorer se encuentra un menú de GXtest Recorder , y el estado de la misma estará en "Recording".

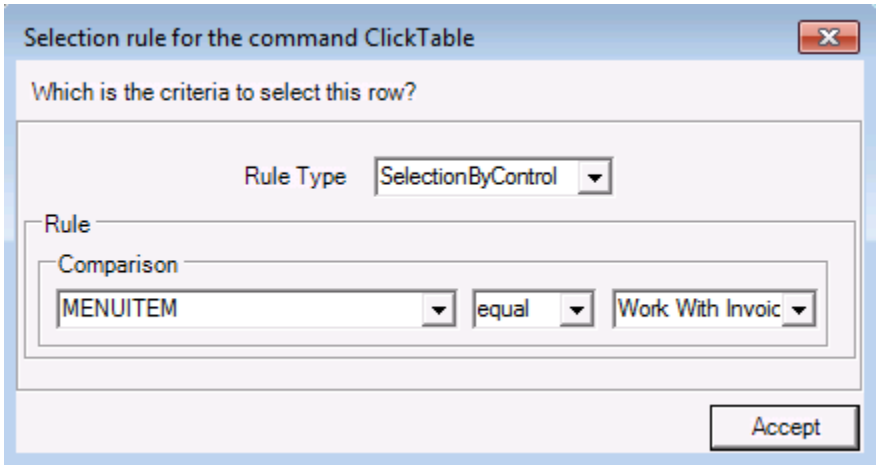


Ahora lo que debemos hacer es seguir **paso a paso** el flujo de prueba definido, mientras GXtest Recorder graba nuestras acciones sobre la aplicación.

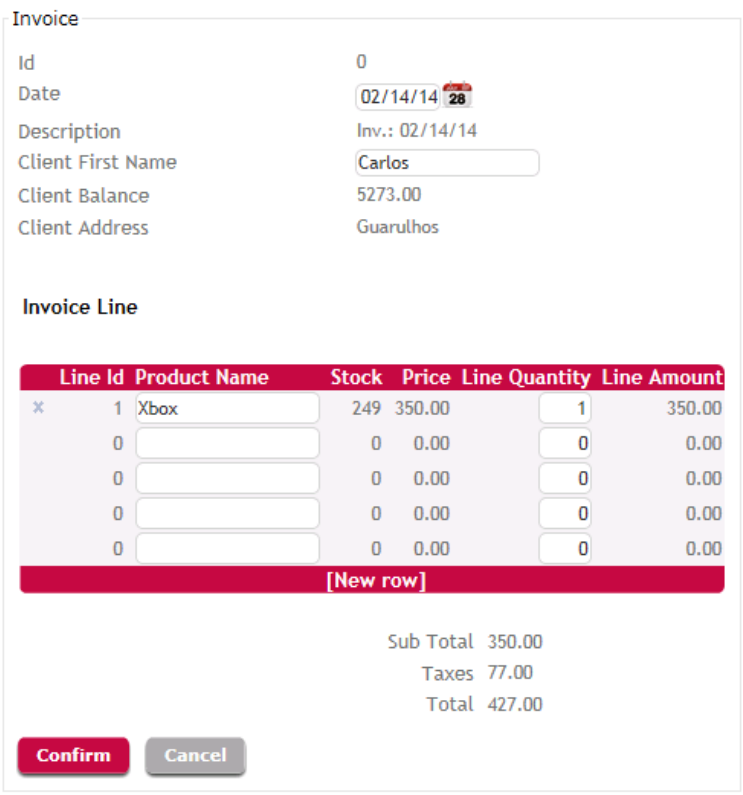
1. Haga click en el link *Work With Invoices*.



En ese momento (debido a que se hizo click en una grilla) GXtest pedirá que se le indique cuál fue el criterio para seleccionar dicha fila de la tabla. Debe dejar la opción que viene por defecto y presionar *Accept* (más adelante explicaremos este concepto):

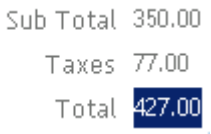


2. Clic en el botón "+" para agregar una nueva factura. 
3. Ingrese el nombre del cliente *Carlos*. Luego ingrese una línea de la factura, indicando el producto *Xbox* y como cantidad *1*. Presione ***Tabulador***.
4. La aplicación calculará los valores de la nueva factura ingresada. Ahora se desea validar que el importe total sea el esperado (427).



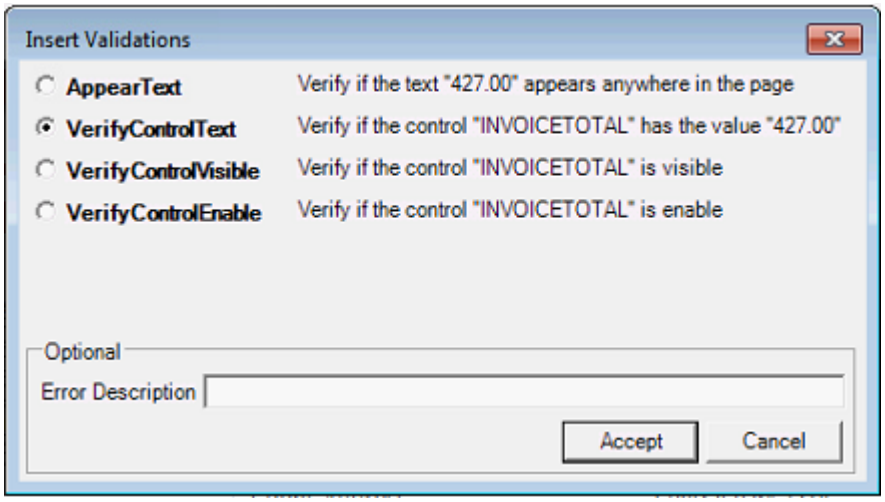
Para ingresar esta validación debe seguir los pasos que se detallan a continuación para el campo total:

i. seleccione el número que muestra el valor del total en la pantalla (427.00)

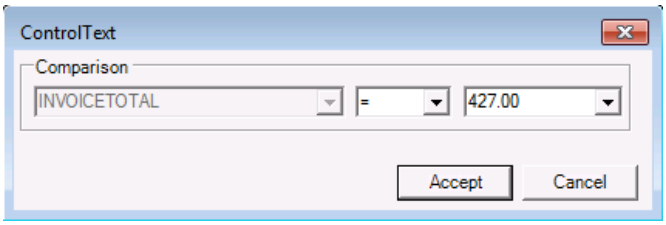


ii. luego seleccione el botón "*Insert validation*" de la barra de GXtest Recorder  (o haciendo clic con el botón derecho sobre el texto seleccionado -> Insert GXtest Validation).

iii. En la ventana que aparece, nos permitirá elegir el tipo de validación a agregar. En este caso seleccione **VerifyControlText**, la cual permite comparar un valor de la pantalla con otro valor y presione *Accept*.

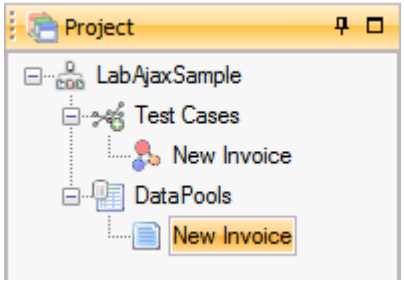


iv. Se abrirá una nueva ventana que indica el criterio de comparación. Dejar el valor por defecto y simplemente seleccionar *Accept*.



5. Para confirmar la factura, presione el botón ***Confirm***.
6. Por último, cierre el navegador para terminar de grabar el caso de prueba.

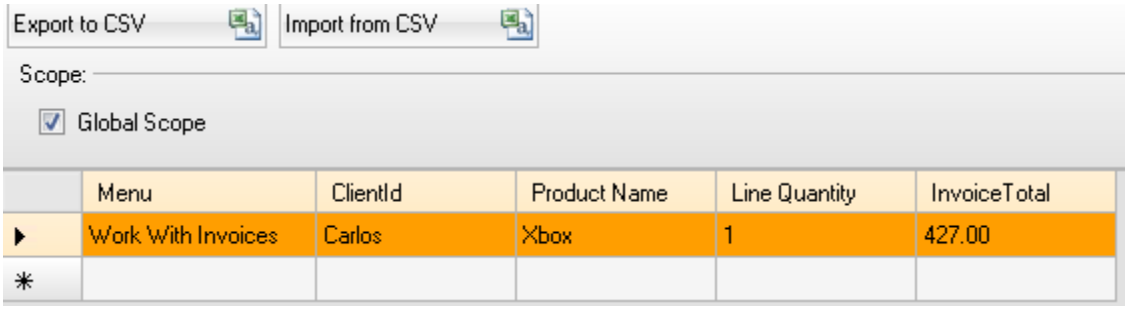
De esta forma quedó grabado el caso de prueba en GXtest, y su datapool asociado.



GXtest ofrece Datapools como fuente de datos para poder parametrizar nuestros casos de prueba. Un Datapool es una estructura con formato tabular, compuesto por filas y columnas donde podemos almacenar nuestros datos de prueba.

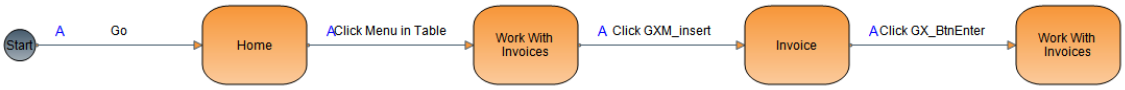
Podemos abrir el Datapool creado automáticamente por GXtest llamado New Invoice, haciendo doble clic sobre el mismo, en el árbol de proyecto.

Dentro del datapool, nos encontraremos con los datos que utilizamos durante la grabación del caso de prueba:

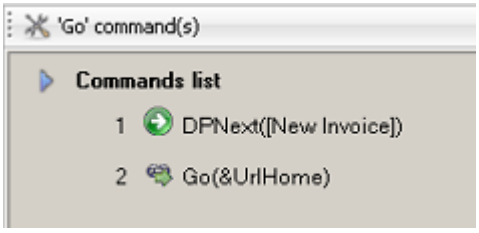


Si prestamos atención, podemos ver que estos son todos los datos que utilizamos al grabar.

Ahora que sabemos dónde están nuestros datos de prueba, repasemos los elementos del caso de prueba *New Invoice*, y prestemos especial atención a los comandos que manipulan los datos de entrada. Veremos cómo están vinculados a un Datapool. Abrir el caso de prueba *New Invoice*:



GXtest expresa las acciones que el usuario realiza en el navegador a través de comandos. Cada comando se asocia a una página (representadas por los nodos anaranjados) o a una arista (que representa las transiciones de una página a otra). Por ejemplo, en la primera arista se encuentra el comando “Go”, el cual se puede observar en la parte inferior en la lista de comandos al seleccionar la arista inicial:

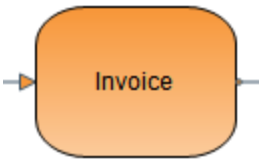


Estos dos comandos representan que al comenzar el caso de prueba, vamos a ir a la pantalla de inicio de la aplicación (indicada por el comando Go y la variable &URLHome), y además utilizaremos el comando DPNext para inicializar el datapool que tiene nuestros datos (New Invoice).

Los otros comandos que se pueden ver en el caso de prueba son:

- Click: representa un clic del ratón en un elemento.
- ClickTable: representa un clic en un elemento que se encuentra dentro de una grilla.
- FillInput: representa la acción de ingresar un valor en un campo editable.
- FillInputTable: ídem al anterior, pero para campos que están dentro de una grilla.
- VerifyControlText: representa la validación de que un control tenga cierto valor esperado.
- DPNext: Obtiene la próxima fila de datos del Datapool.

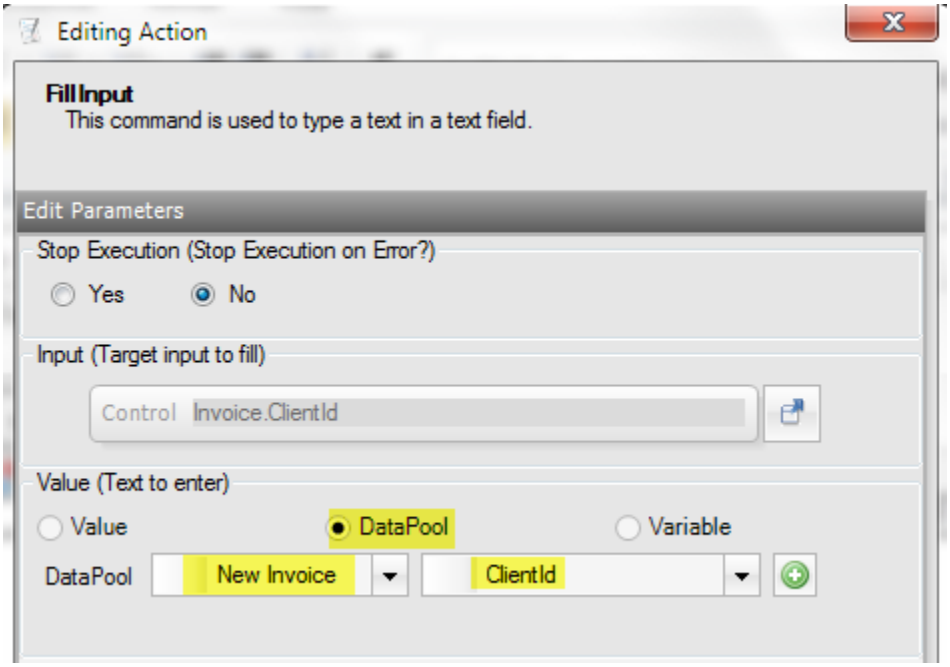
Si seleccionamos el nodo *Invoice* en nuestro modelo, podemos ver los comandos que allí se ejecutan.



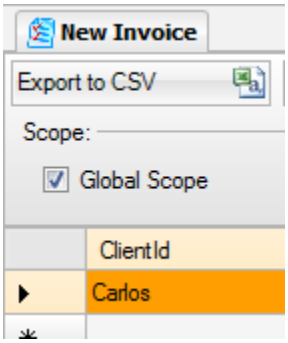
Observemos la lista de comandos:

Si abrimos cualquiera de estas acciones podremos ver cómo el valor del dato de entrada lo toma del datapool asociado: [New Invoice]

Por ejemplo si abrimos el primer comando (FillInput) haciendo doble clic sobre el comando:



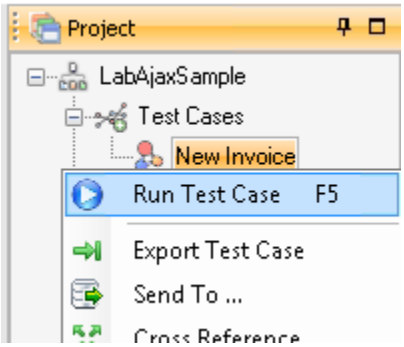
El valor de entrada corresponde al guardado en el Datapool New Invoice, columna ClientId:



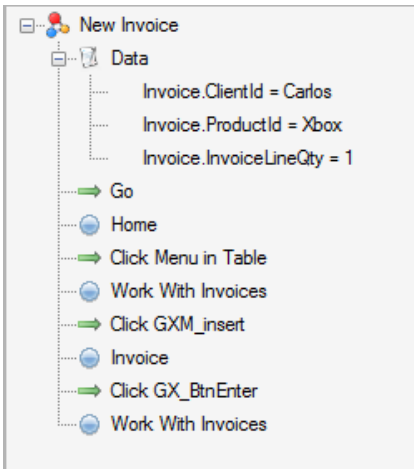
De esta manera si queremos modificar el juego de datos con el que ejecuta el caso de prueba, lo podremos hacer fácilmente, y lo más importante es que podemos agregar nuevos juegos de datos. De igual manera, podríamos modificar el comando y en lugar de utilizar el datapool, configurar un valor fijo, o un valor almacenado en una variable, que haya sido cargada antes por otra sección del caso de prueba.

Ejecutar el caso de prueba

Una vez que se haya podido entender el modelo creado por GXtest, ejecute el caso de prueba. Para esto, puede abrir el caso de prueba y luego presionar el botón de ejecutar en la barra de herramientas de GXtest Designer, o la tecla F5. También es posible ejecutar un caso de prueba utilizando el menú contextual del botón derecho del ratón en el árbol de proyecto.



Finalizada la ejecución, el resultado detallado se mostrará en un panel lateral ubicado a la izquierda dentro de GXtest Designer. Si navega en el menú de resultados, podrá observar los datos que utilizó GXtest en la ejecución y recorrer la secuencia de todos los comandos ejecutados con sus respectivos parámetros y tiempos de ejecución.

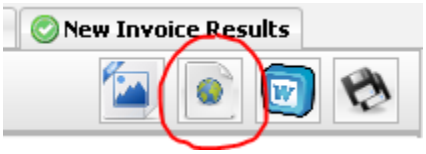


Si selecciona el nodo Invoice dentro de los resultados, podrá ver los pasos ejecutados, comando por comando, y el detalle de cómo estaba la aplicación en ese momento.

Documentación automática

Si deseamos guardar los resultados de esta ejecución como prueba para poder comparar con otras versiones de la aplicación, o simplemente como un documento que especifique la prueba realizada, podemos utilizar las funcionalidades de documentación automática de GXtest, las cuales se encuentran arriba del árbol de resultados. Utilizando estas funcionalidades, es posible generar un documento Word con el paso a paso de la prueba y sus resultados, o también un documento HTML.

Genere la documentación del caso de prueba en formato HTML presionando el siguiente botón, selección el destino para el archivo, y examine el archivo generado.

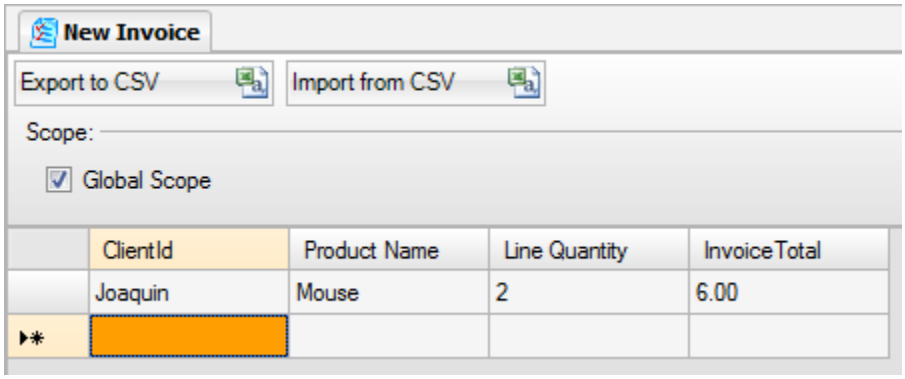


> **Tip:** Estas funcionalidades de documentación le ahorran mucho tiempo al tester, el cual muchas veces debe documentar paso a paso las pruebas, incluyendo los datos y las capturas de pantalla de sus pruebas manuales. GXtest genera documentos que ya incluyen todo esto con un solo clic, los cuales pueden ser adaptados a los formatos de cada empresa, incluso utilizando plantillas predefinidas.

Cambio de datos de prueba

Es posible editar los valores incluidos en el Datapool como se muestra en la imagen, o también podemos

importar datos desde un archivo CSV (archivo de texto con valores separados por coma).



En la carpeta Material GXtest en su escritorio, hemos dejado un archivo "NewInvoice.csv", el cual incluye más datos para incluir en el datapool y completar nuestra prueba.

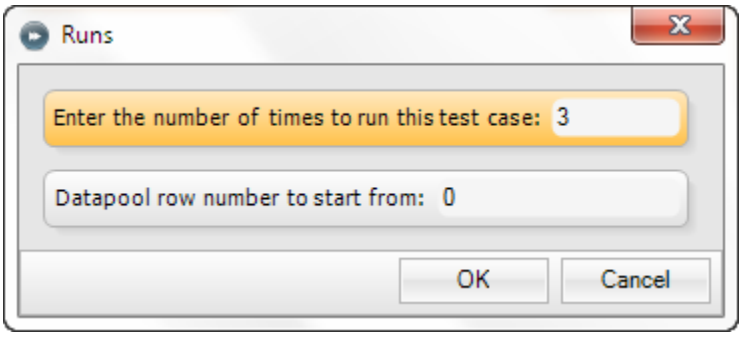
Para agregar estos datos, debemos importar dicho archivo a nuestro datapool, haciendo clic en el botón "*Import from CSV*" (sobre el datapool abierto). Le especificamos el archivo, y luego de importar deberá quedar como se muestra a continuación:

ClientId	Product Name	Line Quantity	InvoiceTotal
Joaquin	Mouse	1	3.66
John	Keyboard	2	12.20
Julio	Notebook	9000	109789020

Si le aparecen cuatro filas en lugar de tres, puede borrar la fila que sobra seleccionando toda la fila (haciendo clic en la primera columna sin título, y en la fila correspondiente) y presionar la tecla "Suprimir" del teclado.

> **Tip:** También es posible copiar los datos desde una planilla Excel directamente al datapool (utilizando Control+C, Ccontrol+V). Ambas opciones (importar el CSV, o copiar desde Excel) le permiten al tester importar a GXtest sus datos de prueba que ya tenía definidos para sus pruebas manuales, disminuyendo el tiempo que le llevaría automatizarlas.

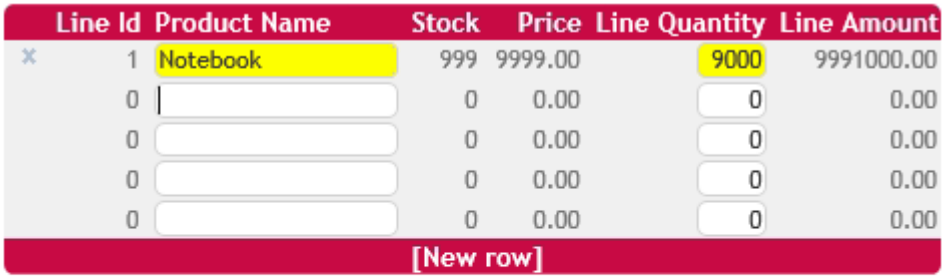
Abrir el caso de prueba y ejecutar el caso de prueba 3 veces presionando el botón en la barra de herramientas de GXtest Designer, o presionando Shift + F7.



Esto hará que se abra el navegador y se ejecute el caso de prueba con 3 iteraciones, tomando en cada ejecución una fila de datos distinta del Datapool.

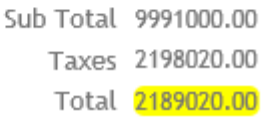
Analice los resultados de las tres ejecuciones y estudie por qué se da el error en la tercera ejecución.

En los resultados, en particular en el caso que falló, podemos ver que el nodo “Invoice” tiene una marca roja, la cual significa que tuvo algún error. Seleccione dentro de los comandos de la página Invoice, el comando VerifyControlText para ver cómo estaba en ese momento la aplicación:



Line Id	Product Name	Stock	Price	Line Quantity	Line Amount
1	Notebook	999	9999.00	9000	9991000.00
0		0	0.00	0	0.00
0		0	0.00	0	0.00
0		0	0.00	0	0.00
0		0	0.00	0	0.00

[New row]



Si no puede ver los valores ingresados en la página (si le aparecen amarillos y vacíos), puede ser debido a propiedades de seguridad del navegador en el que ejecuta la prueba. Puede solucionar esto cambiando la visualización del resultado de HTML a bitmap. Esto se logra haciendo clic en el botón  , el cual intercambia la vista entre HTML o bitmap.

Analice los valores de la página. ¿Los valores del total le parecen correctos? ¿Qué le sucedió a la aplicación?

Tip: Notar la importancia de poder trabajar por separado los datos de prueba, del flujo de prueba en sí. Esto nos permite poder modificar o agregar nuevos datos para que el caso de prueba varíe, sin necesidad de tocar el diseño de la prueba. A partir de un único caso de prueba, podemos generar varias pruebas interesantes simplemente agregando nuevas filas al datapool, jugando con datos borde, o utilizando técnicas de testing más avanzadas como combinaciones por pares o clases de equivalencia. Esto no siempre es posible lograrlo con varias de las herramientas de automatización del mercado, donde cambiar o agregar nuevos datos es un proceso lento, muy tedioso, y propenso a introducir falsos positivos en las pruebas.

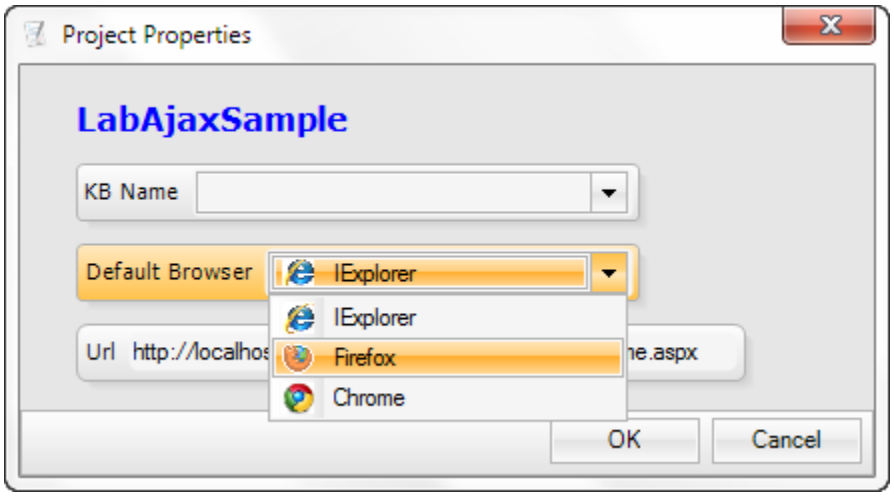
Ejecución en otros navegadores

Si nuestra aplicación está diseñada para ejecutarse con varios navegadores, ¿Únicamente podremos ejecutar las pruebas automatizadas en Internet Explorer?

GXtest nos provee la posibilidad de ejecutar en varios navegadores, si bien únicamente podemos grabar los casos desde Internet Explorer, sí podemos ejecutarlos en Mozilla Firefox y Google Chrome.

Para ello debemos configurar nuestro proyecto de GXtest para que ejecute los casos de prueba en el navegador que optemos.

Esta opción la configuramos haciendo clic con el botón derecho en nodo inicial del árbol de proyecto (LabAjaxSample), y seleccionamos *Properties*. Uno de los atributos de nuestro proyecto es el navegador. Cuando creamos este proyecto al principio dejamos el valor por defecto, Internet Explorer. Cambiemos ese valor a Firefox, y presionemos OK.



Ejecutar nuevamente el caso de prueba, y validar se ejecuta correctamente en Firefox.

Tip: Notar que el mismo caso de prueba, es posible ejecutarlo sobre 3 diferentes navegadores, e

incluso en diferentes ambientes de la aplicación (.net, java, ruby). Si además nos interesa ejecutar en diferentes sistemas operativos, podemos obtener múltiples ambientes combinando esas tres variables. GXtest se encarga de que el mismo caso de prueba funcione en todas las combinaciones, para que el tester solamente se deba encargar de diseñar el flujo. Esto no ocurre así en otras herramientas de automatización, donde muchas veces es necesario mantener un caso de prueba específico para cada ambiente y navegador. GXtest le da una gran ventaja al tester, dado que con un solo caso de prueba, logra ejecutar múltiples pruebas.

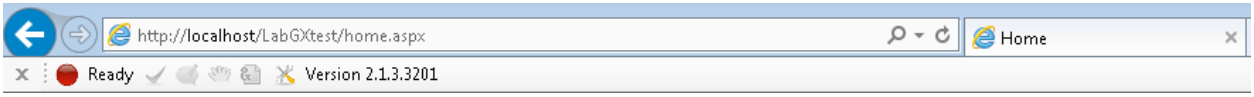
Grabar un caso de prueba de manera Offline

El componente GXtest Recorder tiene la capacidad de funcionar en modo "*offline*", esto es, puede grabar un caso de prueba desde Internet Explorer, de manera independiente de GXtest Designer. De hecho, el Recorder puede ser instalado por separado y grabar los casos de prueba sin tener GXtest Designer.

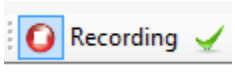
Los casos de prueba son similares al que ya grabamos, la única diferencia es que se inicia el grabado directamente desde el navegador (no desde el Designer). Esto implica, que una vez grabado el caso de prueba *offline*, se guarda como un archivo comprimido (zip) y puede ser importado al proyecto en GXtest Designer de manera manual.

> **Tip:** Esta funcionalidad resulta de gran utilidad cuando un usuario quiere reportar un error, ya que teniendo GXtest Recorder en su navegador, puede grabar el flujo de acciones que reproducen el error y enviar el zip resultante como parte del reporte de error. De la misma manera se puede utilizar este mecanismo al brindar soporte, o incluso testers que no tengan GXtest instalado pueden instalar solamente el Recorder, pues funciona independiente y no necesita licencias.

Para grabar un caso de prueba de manera *offline*, abrimos el Internet Explorer y abrimos la URL de inicio de nuestra aplicación (http://localhost/LabGXtest/home.aspx). Una vez abierta la página de inicio, presionamos en la barra de GXtest Recorder el botón rojo para comenzar a grabar.



El texto a la izquierda del botón dirá "Recording" para indicar que se están grabando las acciones sobre el navegador.

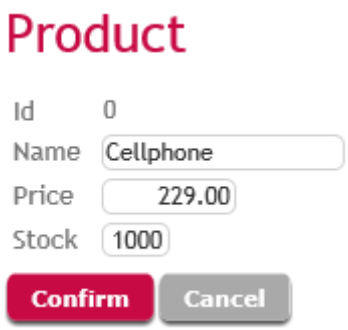


Grabemos entonces el alta de un nuevo producto, siguiendo los siguientes pasos:

1 - Hacemos clic en "*Work With Products*". En la ventana de selección del criterio, presionamos *Accept* para dejar el criterio por defecto.

2- Hacemos clic en el botón *Insert* para agregar un nuevo producto: 

3- Agregamos los datos del nuevo producto:



4- Presionamos el botón *Confirm*, y luego presionamos nuevamente el botón rojo en la barra del Recorder para finalizar la grabación del caso de prueba. Al detener la grabación *offline*, GXtest nos preguntará donde deseamos guardar el caso de prueba grabado. Ingresar el nombre "*New Product*" y guardar el archivo por ejemplo en el escritorio.

Podrá ver en este video los pasos antes descritos para grabar un caso de prueba offline en GXtest.

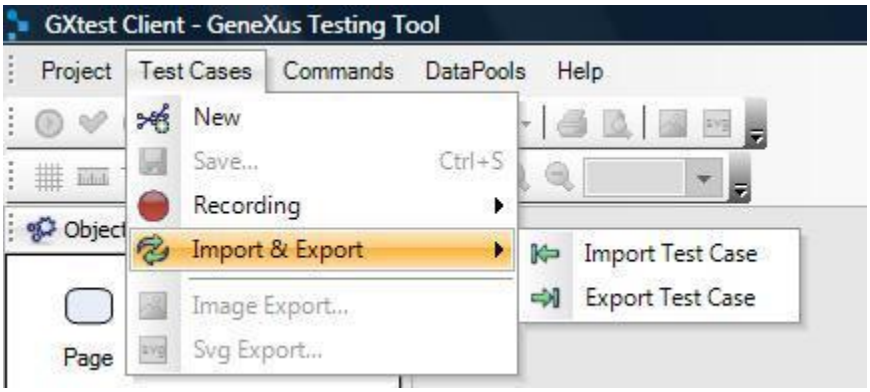
Ahora nos resta importar el caso de prueba dentro de nuestro proyecto GXtest.

Importar un caso de prueba

Tanto la importación y exportación de un caso de prueba permite mover un caso de prueba desde una instalación de GXtest a otra.

Además, es posible importar los casos de prueba grabados de manera offline, como lo hicimos recién.

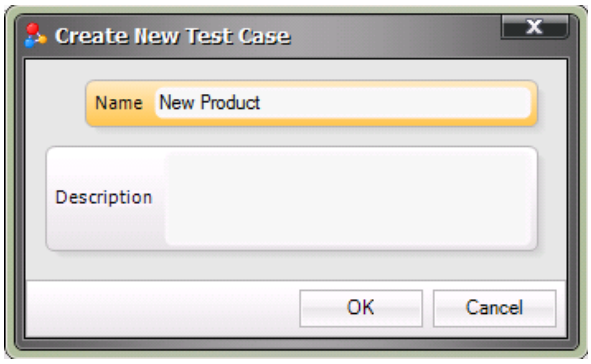
Para acceder a esta funcionalidad, debemos seleccionar de la barra del menú principal de GXtest Designer: ***Test Cases → Import & Export*** *como se muestra en la siguiente imagen:*



Importemos el caso de prueba generado en la sección anterior. Vamos al menú **Test cases -> Import &**

Export -> Import test case.

Seleccionamos el archivo "**New Product.zip**" que habíamos guardado en el escritorio, y luego nos preguntará el nombre del caso de prueba. Dejamos "*New Product*" y presionamos OK.

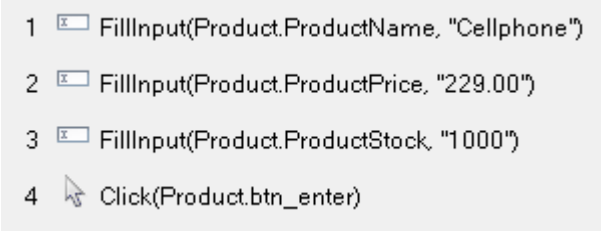


Veremos que el caso de prueba se importa correctamente, y no tiene ninguna diferencia con los casos de prueba que grabamos "*online*" desde GXtest Designer.

Podrá ver en este video los pasos antes descritos para importar un caso de prueba a GXtest Designer.

Parametrización con Datapool

Si explora el caso de prueba importado, verá que esta vez los datos de prueba sí quedaron incrustados en el flujo de prueba. Esto es porque los datapools no se generan de manera automática si grabamos de manera *offline*. Podemos ver esto en la arista "Click GX_BtnEnter" (la última transición del caso de prueba que grabamos).



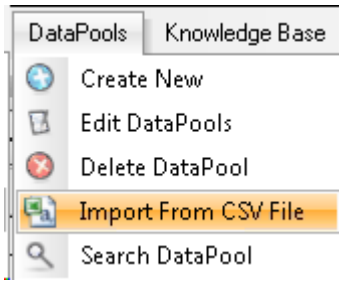
Esto ya comentamos que es una gran desventaja para el mantenimiento y mejora de las pruebas, por lo que primero que nada, cambiaremos los datos fijos por valores almacenados en un Datapool.

Lo bueno es que no es necesario que grabemos el caso de prueba nuevamente, porque siempre podemos editar el caso de prueba, crear un nuevo datapool por separado, y asociarlo a los comandos que ya grabamos.

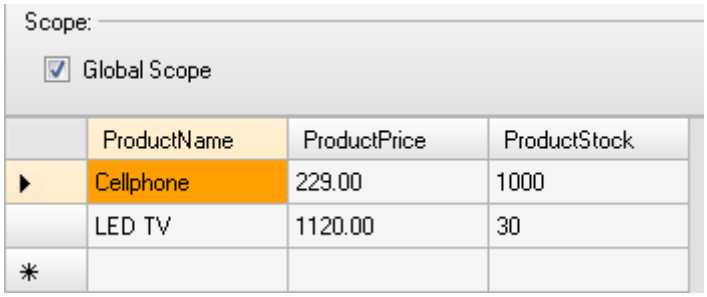
Si bien podemos crear un nuevo datapool de forma manual, ya tenemos un archivo CSV con datos y la estructura que necesitamos para el datapool. Entonces podemos crearlo a partir del CSV (que es diferente

a lo que ya hicimos antes, donde importamos los datos a un datapool que ya existía).

Para esto, ir al menú Datapools, y seleccionar la opción "*Import From CSV File*". Seleccionar el archivo "New Product.csv" ubicado en la carpeta Materiales GXtest.

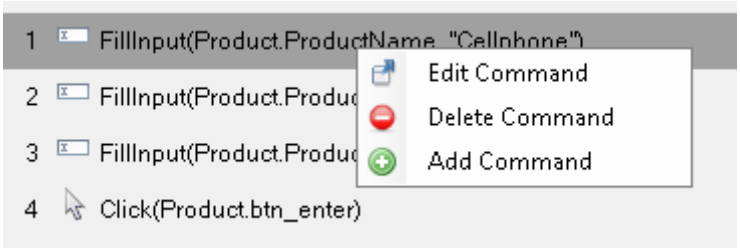


El datapool "*New Product*" contendrá las siguientes columnas y valores:

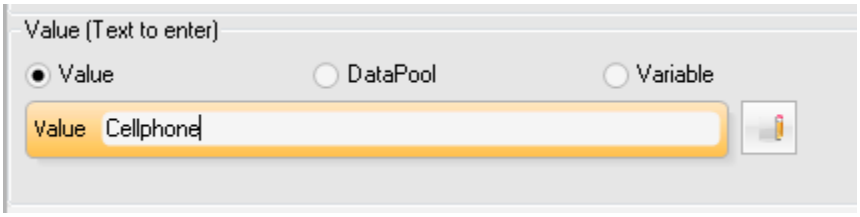


Ahora, necesitamos modificar el caso de prueba para que utilice los valores del datapool, en lugar de los datos fijos, así que volvamos al caso de prueba "*New Product*".

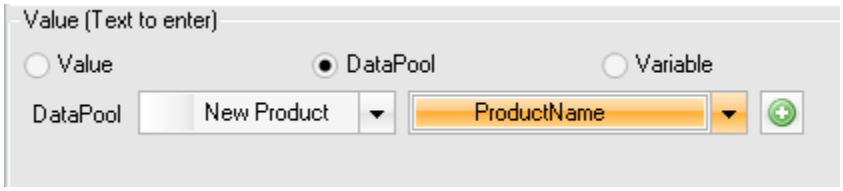
Cada uno de los comandos FillInput que ya vimos puede ser editado, haciendo doble clic en el comando, o haciendo clic con el botón derecho y seleccionando "*Edit Command*".



De esta manera, se abre la ventana de edición, y tenemos el valor fijo "Cellphone".

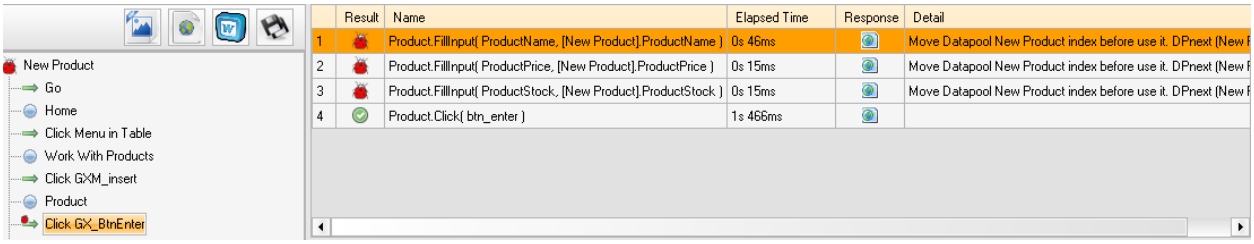


Debemos cambiar el valor utilizado, por la columna “ProductName” del datapool que recién importamos.



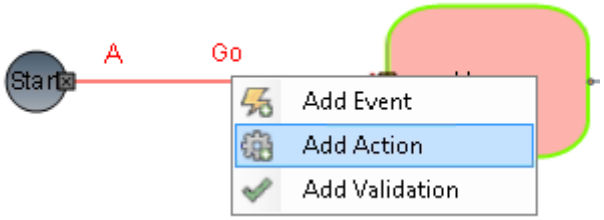
Reemplazar de la misma manera el resto de los comandos FillInput sobre el producto. Luego de las modificaciones, los comandos deberían verse así:

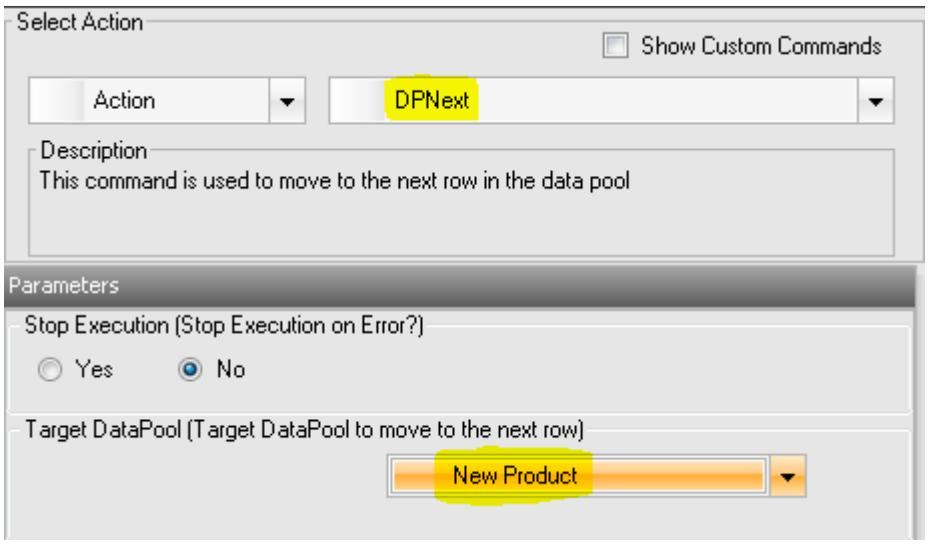
Ejecute el caso de prueba y analice los resultados. ¿Qué sucedió?



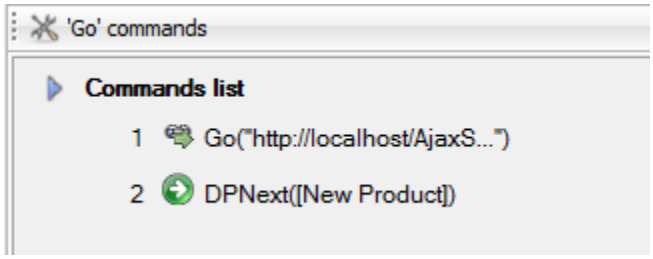
¡GXtest nos está indicando que nos olvidamos de mover el datapool antes de utilizarlo!

Para solucionarlo, debemos agregar un comando DPNext al comienzo del caso de prueba, para que GXtest inicialice el datapool y lo deje listo para ser utilizado. Para esto, seleccionar la arista Go (la primera transición del caso de prueba) y agregar el comando DPNext, haciendo clic con el botón derecho y seleccionando “Add Action”.





Luego de agregarlo, debería verse así:



Ahora, ¿Qué es el comando DPNext, y qué hace?

¿Recuerda que en el primer caso de prueba que grabamos, al ejecutar 3 veces se utilizaban las diferentes filas de datos que teníamos en el datapool New Invoice?

Este comando hace posible que GXtest tome la siguiente fila o el siguiente juego de datos del Datapool durante la ejecución. Si desea, puede validar que al inicio del caso de prueba New Invoice, GXtest ya agregó automáticamente este comando para poder utilizar el datapool.

El único parámetro que necesita el comando DPNext es el Datapool que queremos asociar al caso de prueba.

Siempre que vamos a trabajar con un Datapool en un caso de prueba, debemos agregar el comando DPNext asociado al Datapool antes del primer comando de nuestro caso de prueba que lo invoca.

Ejecute el caso de prueba y valide que ahora ejecuta correctamente.

Manejo de la KB

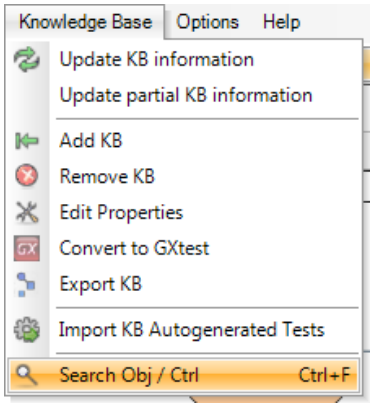
En esta sección, veremos cómo manejar los cambios en la aplicación bajo testing cuando el equipo de desarrollo libera nuevas versiones, y cómo mantener los casos de prueba de manera que se adapten a la nueva versión del software a testear.

La KB en GXtest

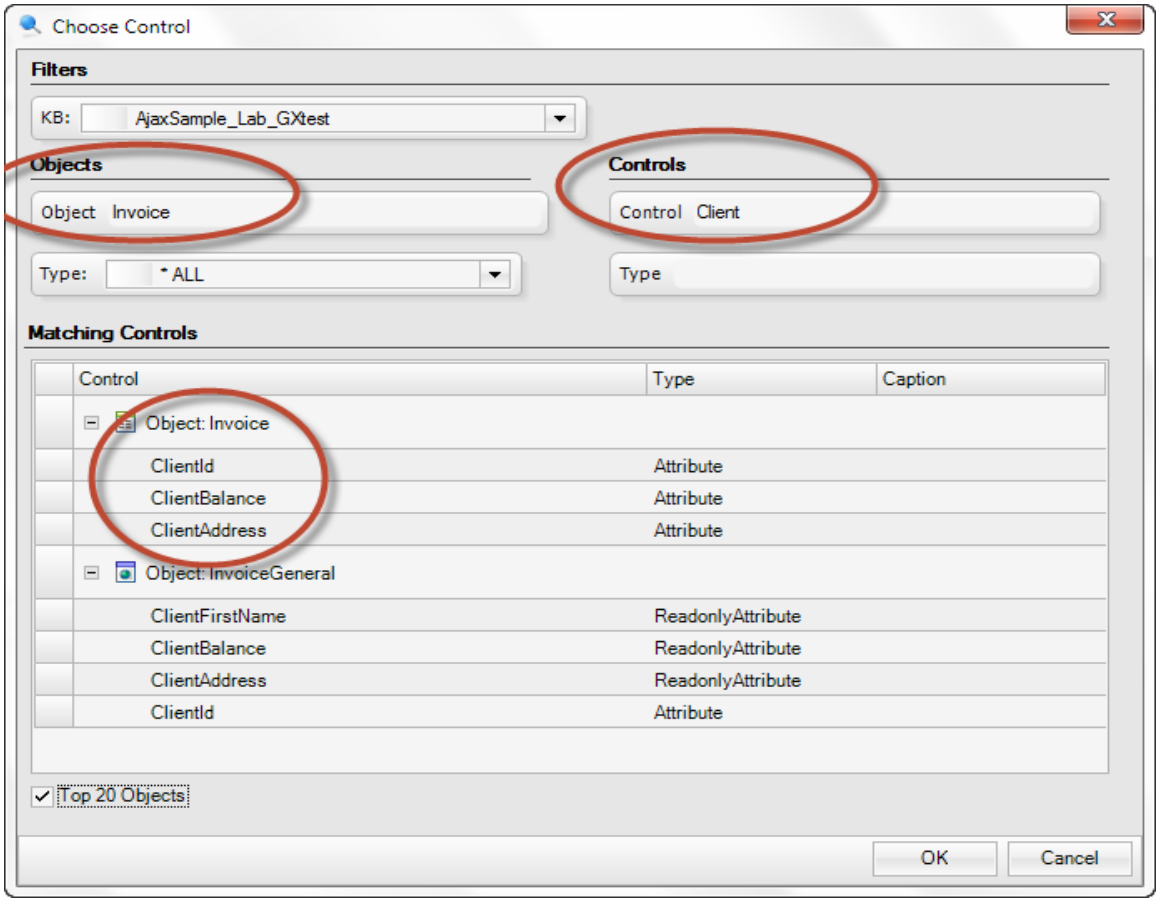
Al comienzo del taller, hablamos brevemente sobre la base de conocimiento o KB de la aplicación. Desde el punto de vista de GXtest, es la información de las pantallas de la aplicación bajo test y sus controles, sobre los que automatizaremos las pruebas. Algunos ejemplos de objetos de la KB que ya vimos: las páginas Invoice, WWProduct, entre otras. En cuanto a los controles, ejemplos son el botón Confirm dentro de la página Invoice, el campo ProductName en la pantalla Product, etc.

Entonces, la KB en GXtest es simplemente un conjunto de objetos o páginas, donde cada una tiene un grupo de controles que pueden ser botones, links, grillas, etiquetas, cajas de texto, columnas de una grilla, etc.

Estos controles y objetos se pueden ver fácilmente en GXtest Designer en el menú *Knowledge Base* -> *Search Obj / Control*, donde podemos filtrar por nombre de objeto, y el tipo o nombre del control.



Por ejemplo, podemos ver los controles de la página Invoice, que comienzan con el nombre “Client”



De esta forma, podemos validar los controles disponibles en la KB y sus objetos.

Es por esto que cuando creamos un proyecto, debemos asociar al menos una KB al proyecto. Esto se debe a que necesitamos saber el conjunto de páginas y controles sobre los que automatizaremos las pruebas. Toda esta información, es mantenida de forma automática por GXtest, aunque algunas veces necesita la intervención del usuario para resolver algunos conflictos (a continuación veremos cuáles).

A medida que pasa el tiempo y testeamos nuestra aplicación, seguramente tengamos nuevas versiones de la aplicación que sean liberadas por el equipo de desarrollo. Dado que ya tenemos muchas de las pruebas automatizadas con GXtest para la versión anterior, seguramente debamos adaptar algunos casos de prueba para testear la nueva versión y sus cambios.

Por ejemplo, imagine que en la nueva versión de la aplicación no aparece más un botón que se cliqueaba en un caso de prueba, dado que se quitó de la aplicación, o se movió la funcionalidad hacia otra pantalla. Ese caso de prueba entonces fallará en su ejecución, dado que no va a tener ese botón en la aplicación para hacer clic. Es por esto, que debemos mantener nuestros casos de prueba, a medida que la aplicación cambia.

Para esto, GXtest tiene una funcionalidad que lo diferencia de las herramientas tradicionales de automatización: nos permite hacer un análisis de impacto que tendrá sobre nuestras pruebas la nueva versión, nos alertará de los casos de prueba que se “rompan”, e incluso corregirá de manera automática algunos casos de prueba.

Cuando el equipo de desarrollo libera una nueva versión de la aplicación, como testers debemos pedirle la información acerca de la nueva versión (el archivo gxt que importamos al comienzo, pero esta vez para la nueva versión).

Para actualizar la información de la KB en GXtest, tenemos la funcionalidad llamada “Update KB information”, la cual analiza la totalidad de la KB existente en GXtest y la compara con la nueva versión, validando que todas las páginas y controles utilizados en los casos de prueba sigan existiendo, o en caso contrario, ayudar al tester a resolver estos problemas (algunos de manera automática, y otros con intervención del usuario).

Ahora procederemos a hacer dos pequeños cambios en la aplicación (como la haría un desarrollador), de manera de evaluar los cambios. No se preocupe si nunca trabajó con GeneXus, ya que los cambios son lo suficientemente simples como para que los pueda realizarlos sin conocimientos previos de la herramienta.

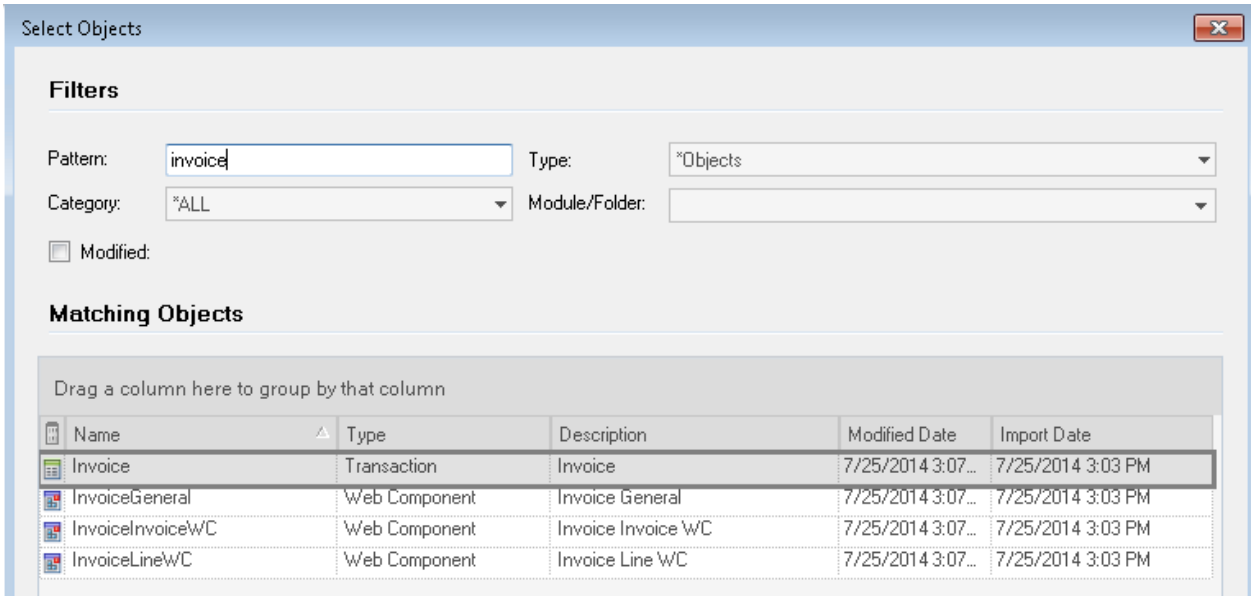
Modificar la KB

Con la finalidad de ver cómo se comporta GXtest frente a los cambios, introduciremos dos modificaciones a la aplicación bajo pruebas:

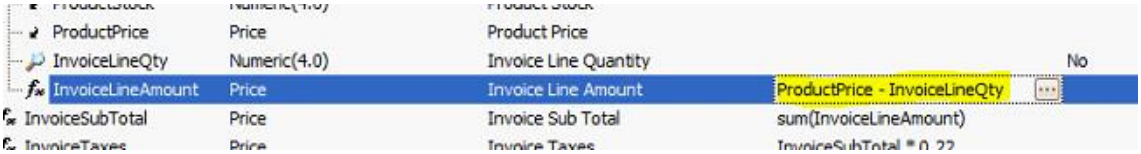
1) Introducir un bug a propósito

2) Cambiar de nombre un control de la aplicación en la KB GeneXus

Para esto, abra GeneXus X Evolution 3 desde el ícono en el escritorio, y dentro de la AjaxSample (que se abrirá por defecto), abra la transacción Invoice, presionando Ctrl + O y luego elija el objeto Invoice en la ventana que aparece.

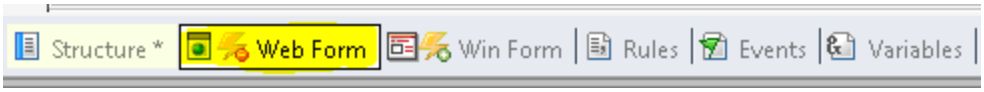


Cambie la fórmula de Subtotal de la línea (InvoiceLineAmount).
Donde dice:
*ProductPrice * InvoiceLineQty*
Cámbielo por:
ProductPrice – InvoiceLineQty

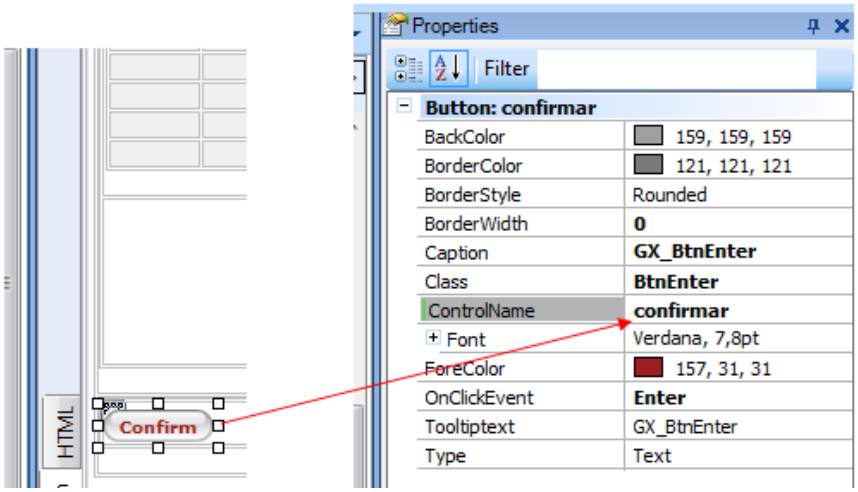


Eso definitivamente hará que el subtotal se calcule incorrectamente, y lo haremos con el propósito de ver que GXtest nos avise de este bug de la aplicación (dado que ya tenemos un caso de prueba que valida esto).

Ahora, cambiaremos un control de la pantalla, para validar como GXtest nos ayuda a adaptar los casos de prueba a los cambios de la aplicación.
Para cambiar algún control de la KB, haga clic en la pestaña de WebForm en la barra inferior del objeto.

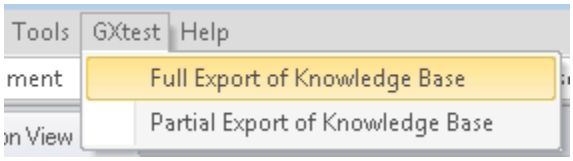


Cambiemos el nombre del botón Confirm por otro. Para esto, seleccione el botón haciendo clic sobre éste, y en el panel de propiedades de la derecha, cambie el ControlName, del valor “btn_enter” por “confirmar”.



Salve los cambios y presione F5 para que los cambios se propaguen a la aplicación. Cuando GeneXus termine de regenerar la aplicación, se abrirá el navegador en la página inicial.

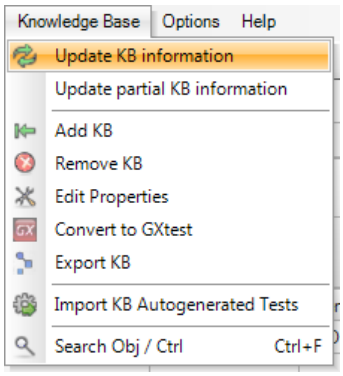
Ahora, debemos actualizar los datos que tiene GXtest sobre la aplicación, para que nos ayude a adaptar los casos de prueba.
Desde GeneXus, exportaremos la nueva versión desde el menú GXtest -> *Full export of Knowledge Base.*



Guarde el archivo en la carpeta "Material GXtest" del escritorio, y sobreescriba el archivo existente **Curso GXtest.*gxt*** .

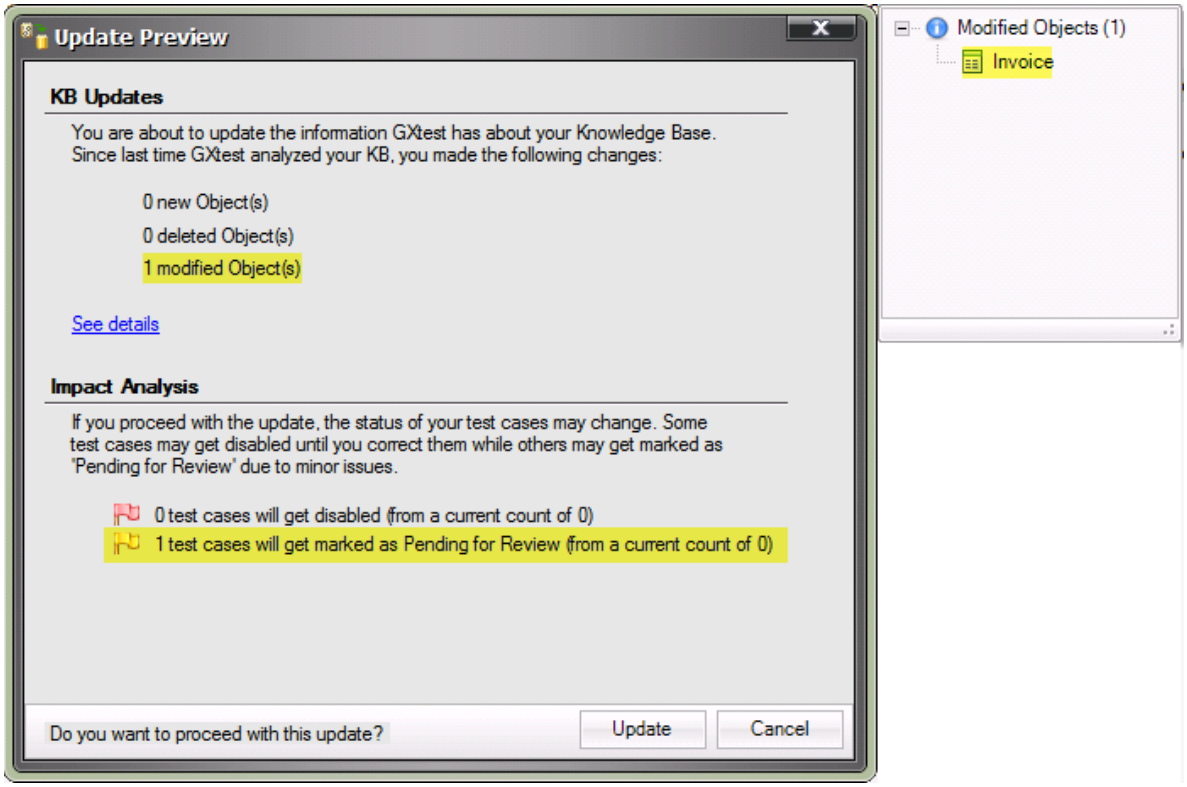
Impactar los cambios en GXtest

Volvamos a GXtest, e impactemos los cambios de la nueva versión. Dado que "pisamos" el archivo gxt que importamos al principio, no es necesario volver a indicar un nuevo archivo, por lo tanto simplemente debemos ir al menú **Knowledge Base -> Update KB Information**.



GXtest leerá la información de la KB y analizará cuántos objetos han sido agregados, borrados y modificados, en relación con la KB original que importamos al comienzo del laboratorio.
Por otro lado brindará un detalle de que casos de prueba quedarán en estado deshabilitados (por tener cambios que inhabilitan su ejecución) o pendientes de revisión (aquellos que GXtest aplicó una determinada heurística para su adaptación y requiere que el usuario acepte el cambio aplicado).

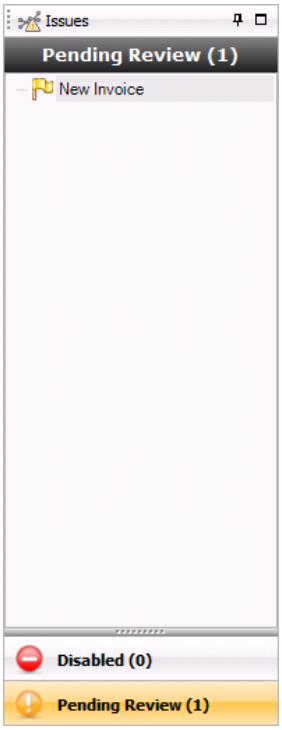
La pantalla de análisis de impacto mostrará lo siguiente:



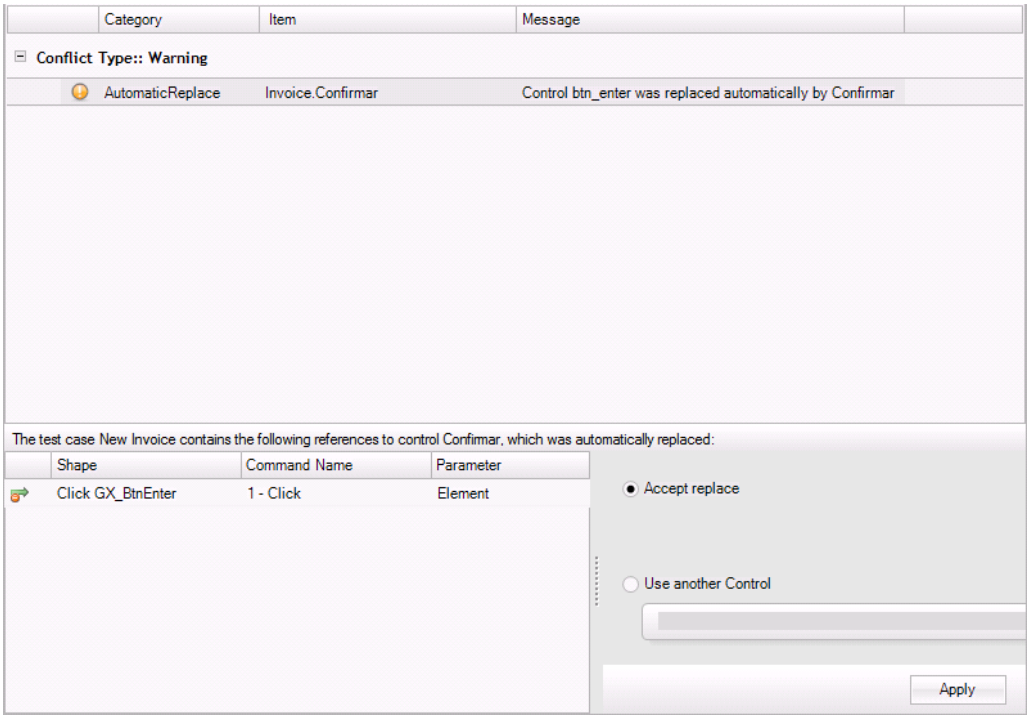
Analizando la pantalla, y a su criterio, ¿cuáles son los casos de prueba que quedarán pendientes de revisión?
Podríamos pensar que serían los casos de prueba que utilizan la transacción Invoice, pero en realidad, los casos de prueba que se ven afectados por el cambio son solamente los que utilizan el botón “Confirmar” que modificamos. Si hubiéramos modificado un control que el caso de prueba no utiliza (aunque sí utiliza la transacción Invoice), no hubiéramos tenido ningún impacto en los casos de prueba.

> **Tip:** En otras herramientas de automatización, esto no es posible, dado que no tienen conocimiento de la aplicación. GXtest, por ser diseñada específicamente para aplicaciones hechas con GeneXus, toma ventaja de esto y nos permite analizar los cambios, e incluso corregir automáticamente las pruebas.

Haga clic en el botón Update para proceder con la aplicación de los cambios, y una vez finalizado este proceso, verá que aparecen los casos de prueba pendientes de revisión en el panel “Issues” de la derecha.



Haciendo doble clic en el caso de prueba “New Invoice” marcado en el panel, veremos los conflictos que debemos revisar.



GXtest nos está avisando que aplicará un reemplazo en el caso de prueba, donde se utilizaba el control btn_enter, lo reemplazará por el control Confirmar, dado que sospecha que es el mismo control que cambió de nombre. En caso de que esta suposición sea incorrecta, es posible seleccionar manualmente otro control para utilizar en lugar del control que no existe más.

Haga clic en el botón *Apply* en la parte inferior derecha de la pantalla y para dar por resueltos los conflictos.

Ejecute el caso de prueba “New Invoice”, para así verificar:

1) Que se detecta el bug introducido en el cálculo del total de la factura.

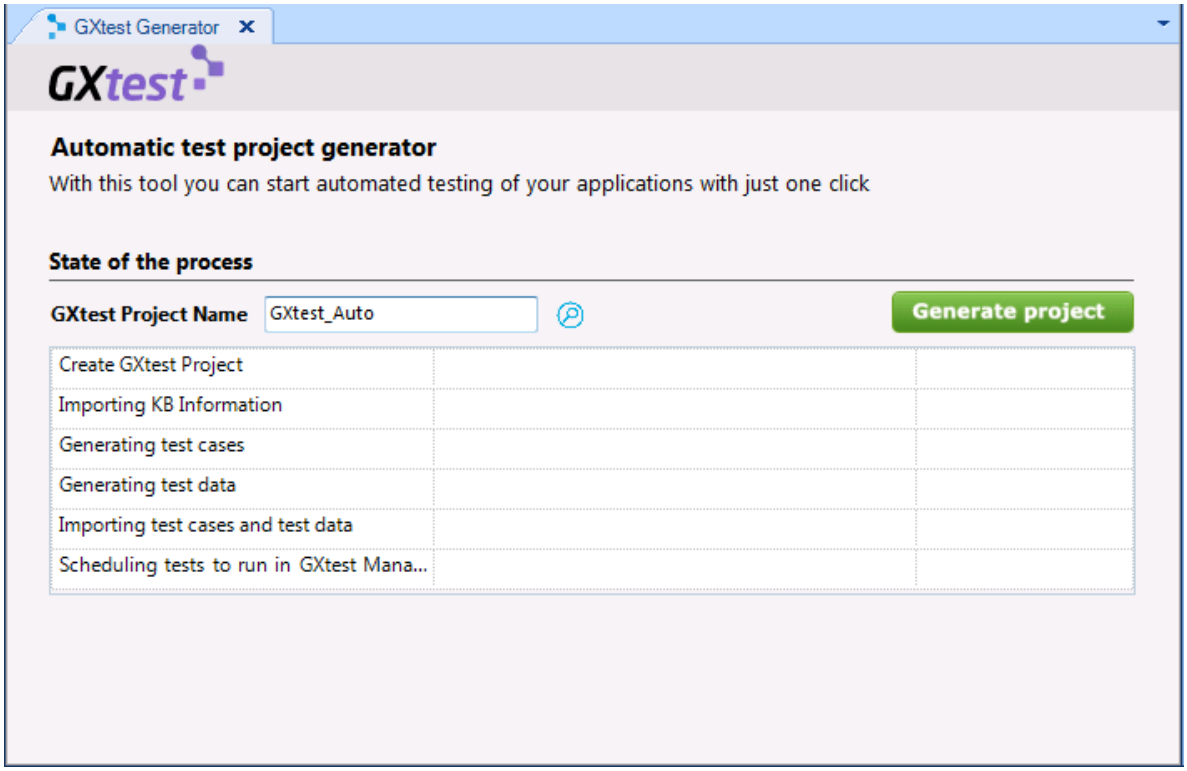
2) Que GXtest reconoce el nuevo control por más que tenga otro nombre.

One Click Startup: ¡Empiece rápido y fácil con 1 clic!

Ahora veremos cómo GXtest nos ayuda a comenzar con automatización muy rápidamente, creando y generando un proyecto con un juego de pruebas en forma automática a partir de la KB de nuestra aplicación. Esta funcionalidad la ejecutamos desde GeneXus a través de la extensión de GXtest. Para ello debemos abrir GeneXus desde el ícono ubicado en el escritorio.

Para empezar a generar pruebas automáticamente, debemos ir al menú GXtest -> "*Autogenerate test cases*".

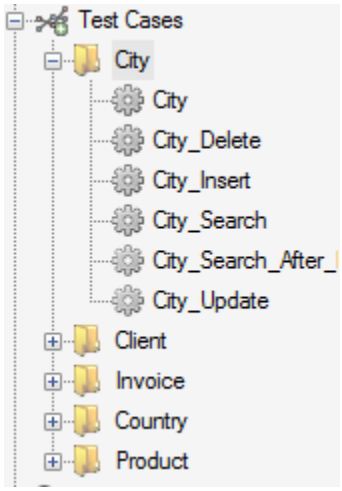
Esto nos abrirá una ventana dentro de GeneXus, donde podemos iniciar el proceso desde el botón "*Generate Project*".



Observe cómo en la ventana se notifica paso a paso desde la creación del proyecto a la creación de los casos de prueba.
Una vez que notifica que ya fue creado e importado nuestros casos de prueba y el proceso se ha completado, verifique en GXtest Designer que el proyecto fue creado e importado a nuestra herramienta

de pruebas. Para ello, puede abrir GXtest Designer desde el botón “Add or Edit Test cases” que aparece en la esquina inferior izquierda de la pantalla.

Observe que se han creado a partir de nuestra aplicación (por ejemplo para AjaxSample) los siguientes Casos de prueba y Datapools:



Compruebe, ejecutando un caso de prueba (por ejemplo City, asociado a la transacción con el mismo nombre) cómo GXtest generó automáticamente a partir de la KB en GeneXus, las pruebas de Alta, Modificación, Baja y Consulta para dicha transacción, así como para cada uno de los objetos transaccionales de nuestra aplicación.
Simplemente seleccione el caso de prueba que desea ejecutar y presione F5.

Es importante comentar que la utilidad de la generación automática no es probar la aplicación completamente, sino darle al tester una suite de pruebas con las cuales empezar a automatizar, para que pueda a partir de aquí mejorarlas, adaptarlas o incluirlas en otras pruebas.

Conclusiones

Con esta práctica pudimos demostrar lo fácil que es automatizar pruebas con GXtest (usando GXtest Recorder) y ejecutarlas con distintos juegos de datos.

Pudimos ver cómo variar los datos de manera simple, reutilizando los casos de prueba.

Pudimos analizar los cambios en la aplicación realizados por los desarrolladores en unos pocos minutos, adaptando los casos de prueba para que continúen ejecutándose en la nueva versión del sistema bajo pruebas.

Por otro lado, con 1 sólo clic desde GeneXus podemos generar pruebas automáticas que testean distintas transacciones y "Work With's" permitiendo aplicar técnicas de testing en las pruebas generadas:

- Encontrar bugs rápidamente.
- Reutilizar dichos casos de prueba para crear casos de prueba más complejos.
- Ejecutar pruebas de regresión luego de aplicar un nuevo Build de GeneXus o un cambio en la aplicación.

Lo invitamos a seguir descubriendo toda la suite de productos de GXtest en http://genexus.com/gxtest y solicitar una versión de prueba por 30 días.

Este taller fue tan solo una primera aproximación a la herramienta. Para profundizar sus conocimientos y aprender a dominar todas las funcionalidades de GXtest, lo invitamos a realizar el curso presencial de GXtest pudiendo obtener la certificación *Analista GXtest*.

La información del curso así como fechas agendadas para su dictado en Montevideo las encontrará en el sitio GXtraining, http://training.genexus.com/

En el curso podrá conocer otras funcionalidades de GXtest, como:

- GXtest Manager, un repositorio de pruebas, donde gestionar, agendar y ejecutar sus pruebas.
- GXtest Executor, para ejecutar en forma distribuida y desatendida sus tests.
- Generación automática de scripts para pruebas de performance, para OpenSTA o JMeter.

Así mismo si desea probar GXtest en la aplicación de su negocio, podrá coordinar con Abstracta y comprobar cómo GXtest facilitará notablemente la automatización de pruebas funcionales de su aplicación GeneXus, reduciendo los costos generados por bugs y largos procesos de testing manual.