

Práctico Cobertura de Código

Testing en GeneXus

Copyright ♥ Abstracta

All rights reserved. This document may not be reproduced by any means without the express permission of Abstracta. The information contained herein is intended for personal use only

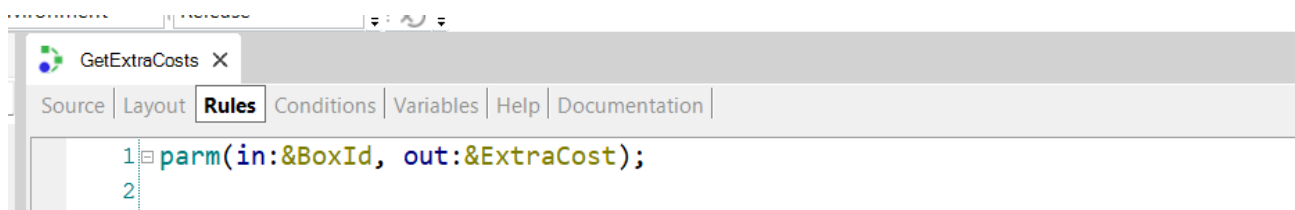
Introducción

En esta capacitación práctica se introducirá la funcionalidad de *Test Coverage*, realizándose un análisis de cobertura de código de una prueba unitaria.

Test Coverage

Es posible conocer el porcentaje de líneas de código ejecutadas para cada objeto llamado por un Unit Test. En esta sección le guiaremos para que pueda crear y ejecutar una prueba unitaria que realice llamadas a otros objetos de la KB y analizar el cubrimiento que realiza ¡Veamos el procedimiento!

- 1) Abrir el procedimiento **'GetExtraCost'** y hacer click en la pestaña 'Rules'. Podemos ver que recibe como parámetro de entrada **&BoxId** (Id de la caja a la cual le va a determinar el costo extra) y devuelve como parámetro de salida **&ExtraCost** (costo extra para cada caja según la ciudad asignada para el envío).



- 2) Hacer clic en la pestaña Source. Podrá observar en la lógica del procedimiento, que se asigna el costo extra para cada caja que coincide con el parámetro de entrada **&BoxId** siempre y cuando la caja esté lista para ser entregada. De lo contrario, devuelve un **ExtraCost = '0'**.



Para este procedimiento tenemos 12 líneas ejecutables de código. Si el *BoxId* que se ingresa, tiene asignado a la variable *BoxIsSetToBeDelivered* el valor *'true'*, se ejecutarán 11 líneas de este procedimiento. De lo contrario, sólo se ejecutará la sentencia declarada dentro del *'else'*.

Buscaremos generar una cobertura total, creando casos de prueba que cubran todo el código del procedimiento. ¡Empecemos!

- 1) Antes de comenzar, abrir la aplicación y seleccionar el Box con Id=1. Actualizar el campo *'Be Delivered'* que se encuentra en *false* por defecto, escribir *'true'* y seleccionar *'Confirm'*.

Arrival Date: 09/13/22

Delivery Date: / / 12:00 AM

Is Retained: ☐

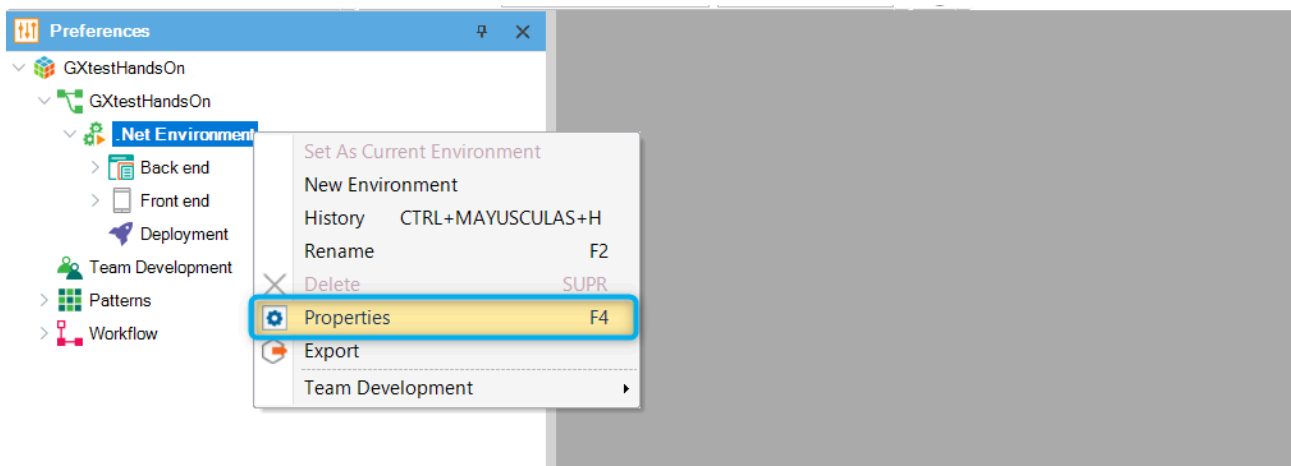
Be Delivered: true

Customer Id: 1

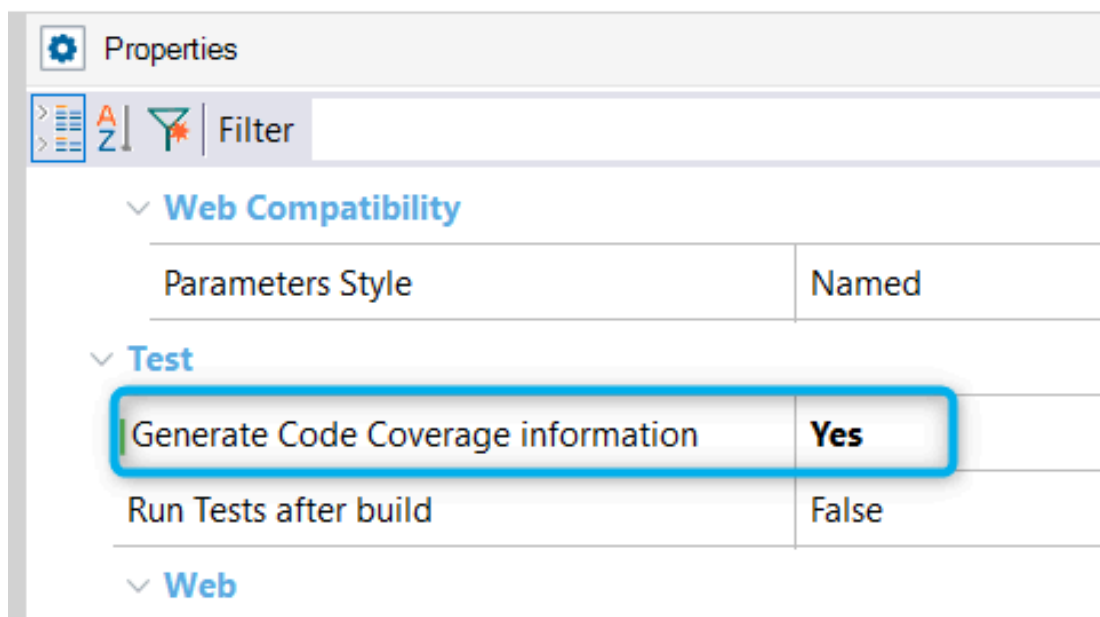
Container Id: 1

CONFIRM CANCEL

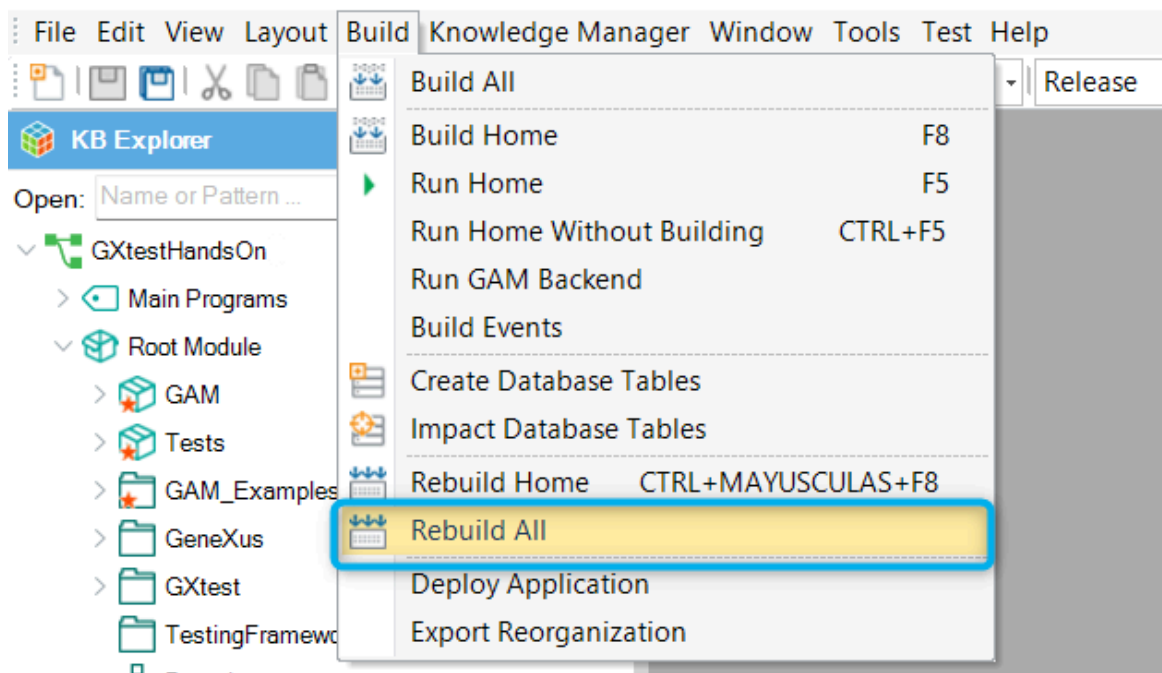
- 2) Volver a la KB y crear un *Unit Test* sobre el procedimiento **'GetExtraCost'** haciendo clic derecho sobre el mismo y seleccionando *'Create Unit Test'*.
- 3) Abrir las propiedades del ambiente haciendo click derecho y seleccionando la opción *"Properties"*.



- 4) Identificar la propiedad 'Generate Code Coverage Information' y cambiar a 'Yes'.



- 5) Una vez habilitado, desde el menú 'Build' seleccionar 'Rebuild All'. Este paso es necesario para que se vuelvan a generar los objetos con rastreadores de cobertura de código.



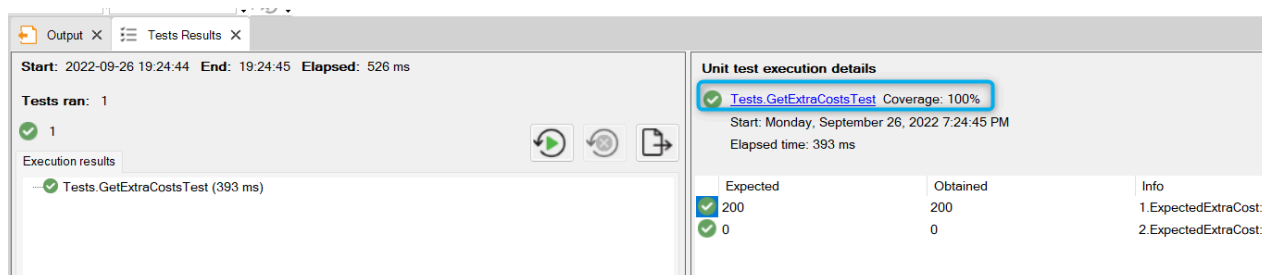
- 6) Abrir 'GetExtraCostsTestData' y modificar en el *TestCaseId=1* los valores **BoxId = 1** y **ExpectedExtraCost = 200**. En el *TestCaseId=2* ingresar los valores **BoxId = 2** y **ExpectedExtraCost = 0**.

El código del Data Provider es:

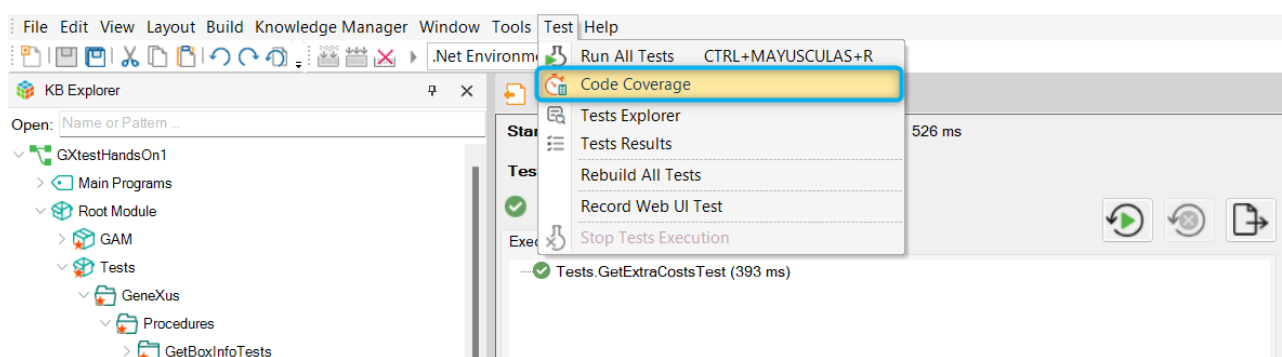
```
GetExtraCostsTestSDT
{
    TestCaseId = '1'
    BoxId = 1
    ExpectedExtraCost = 200
    MsgExtraCost = '200'
}
GetExtraCostsTestSDT
{
    TestCaseId = '2'
    BoxId = 2
    ExpectedExtraCost = 0
    MsgExtraCost = '0'
}
```

- 7) Guardar los cambios y ejecutar 'GetExtraCostsTest'.

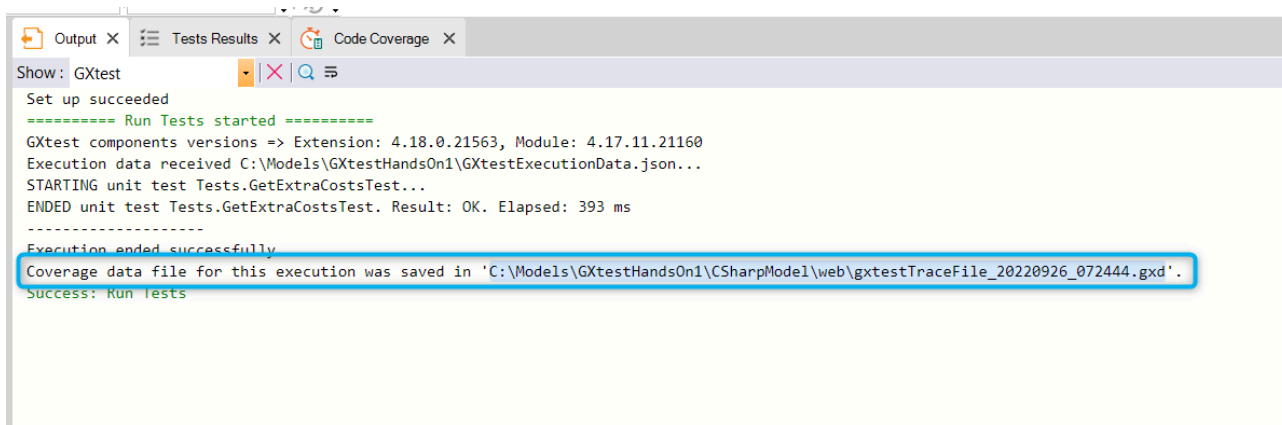
Al terminar la ejecución del *test*, se mostrarán los resultados en el panel *Tests Results*. Vemos que se realizó una cobertura del 100% del código de los objetos llamados por el test.



- 8) Seleccionar la pestaña 'Test' y hacer clic sobre la opción 'Code Coverage'.



- 9) Copiar la ruta indicada en el output de GXtest donde se guardó la información de la ejecución.

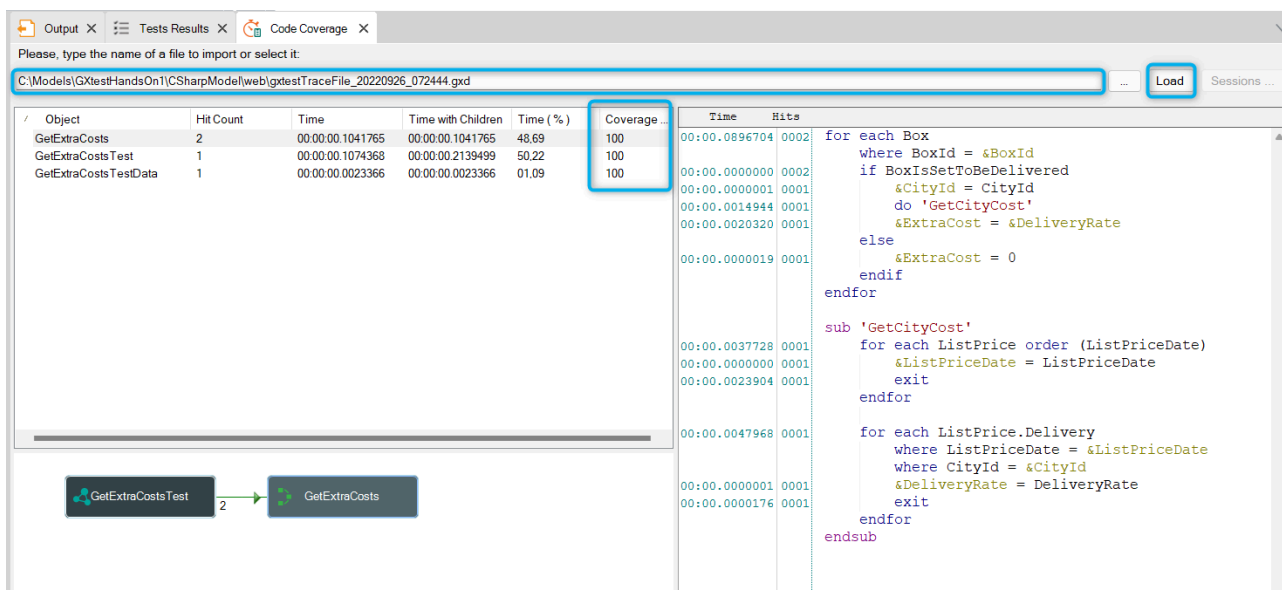


The screenshot shows the 'Output' tab of the GXtest application. The text in the window is as follows:

```
Set up succeeded
----- Run Tests started -----
GXtest components versions => Extension: 4.18.0.21563, Module: 4.17.11.21160
Execution data received C:\Models\GXtestHandsOn1\GXtestExecutionData.json...
STARTING unit test Tests.GetExtraCostsTest...
ENDED unit test Tests.GetExtraCostsTest. Result: OK. Elapsed: 393 ms
-----
Execution ended successfully
Coverage data file for this execution was saved in 'C:\Models\GXtestHandsOn1\CSharpModel\web\gxtestTraceFile_20220926_072444.gxd'.
Success: Run Tests
```

The path 'C:\Models\GXtestHandsOn1\CSharpModel\web\gxtestTraceFile_20220926_072444.gxd' is highlighted with a blue box.

- 10) Pegar el path en el campo indicado dentro del Code Coverage y hacer clic en 'Load'



The screenshot shows the 'Code Coverage' window. At the top, there is a text box with the path 'C:\Models\GXtestHandsOn1\CSharpModel\web\gxtestTraceFile_20220926_072444.gxd' and a 'Load' button. Below this, there is a table with columns: Object, Hit Count, Time, Time with Children, Time (%), Coverage, Time, and Hits. The table contains three rows of data for the 'GetExtraCosts' test.

Object	Hit Count	Time	Time with Children	Time (%)	Coverage	Time	Hits
GetExtraCosts	2	00:00:00.1041765	00:00:00.1041765	48.69	100	00:00.0896704	0002
GetExtraCostsTest	1	00:00:00.1074368	00:00:00.2139499	50.22	100	00:00.0000000	0001
GetExtraCostsTestData	1	00:00:00.0023366	00:00:00.0023366	01.09	100	00:00.0014944	0001

Below the table, there is a diagram showing the flow of the test execution. It starts with 'GetExtraCostsTest' (2 hits) and leads to 'GetExtraCosts' (2 hits).

Observar que se muestra el porcentaje de cubrimiento en la columna 'Coverage'.

Se logró un cubrimiento del 100% del código, ya que en los casos de prueba se ingresó un *BoxId=1* que cumple con la primera condición, y por tanto recorre las 12 líneas para ese escenario, mientras que *BoxId=2* se corresponde con la ejecución de la sentencia 'else', completando una cobertura total.



MONTEVIDEO - URUGUAY
SAN FRANCISCO - USA
LONDRES - UK

Sarmiento 2465, 11300
100 Pine St., Ste. 1250, CA 94111
71-75 Shelton Street, WC2H 9JQ

+598 2711 0561
+1 415 745 3678
+44 203 696 6682