Guía de desarrollo, Anexo 03.06

Guía de Programación en Unified Functional Testing v14.00

Oficina de Pruebas

GERENCIA INFORMÁTICA
JOSEFA VALCÁRCEL, 44
28027-MADRID

Índice General

1 INTRODUCCIÓN 4

2 TERMINOLOGÍA Y ACRÓNIMOS 5

3 BENEFICIOS DE LA AUTOMATIZACIÓN 6

4 ACCESO A LA APLICACIÓN 7

5 OPCIONES DE CONFIGURACIÓN DE UFT 8

5.1 INSTALACIÓN ADD-IN DE APPLICATION LIFECYCLE MANAGEMENT 8
5.2 PAGE/FRAME OPTIONS 11
5.3 ASOCIAR LA LIBRERÍA DE FUNCIONES A TODOS LOS NUEVOS TESTS 12
5.4 CAPTURAR IMÁGENES CUANDO SE PRODUCEN ERRORES O WARNING EN LA EJECUCIÓN. 14

6 INTEGRACIÓN ALM Y UFT 15

6.1 CREAR SCRIPTS UFT EN ALM 15
6.2 ABRIR SCRIPTS UFT DESDE ALM 18
6.3 CONECTAR A ALM DESDE UFT 18

7 REPOSITORIO DE OBJETOS 23

7.1 ACCEDER AL REPOSITORIO DE OBJETOS 23
7.2 DEFINIR NUEVOS OBJETOS EN EL RO LOCAL 25
7.3 ACTUALIZACIÓN DE OBJETOS 28
7.4 DEFINICIÓN DEL CICLO DE VIDA DEL RO 29
7.4.1 Asociación de RO a Actions 31
7.4.2 Gestión del RO Local 32
7.4.3 Gestión del RO Compartido 33

8 REGLAS DE NOMENCLATURA 36

8.1 SCRIPTS FUNCIONALES AUTOMATIZADOS 36
8.1.1 Ejemplo 37
8.2 COMPONENTES COMUNES AUTOMATIZADOS 37
8.2.1 Ejemplo 38

9 RECURSOS DE LOS TESTS 39

9.1 TABLA DE DATOS 40
9.1.1 Sintaxis 44
9.2 REPOSITORIOS DE OBJETOS COMPARTIDOS 45
9.2.1 Sintaxis 45
9.3 ESCENARIOS DE RECUPERACIÓN 45
9.3.1 Sintaxis 46
9.4 LIBRERÍAS FUNCIONALES 46
9.4.1 Sintaxis 46

10 NORMATIVAS DE DESARROLLO A SEGUIR 48

10.1 CASO DE PRUEBA GUI 48
10.1.1 Creación nuevo test o Componentes Comunes Automatizados 48
10.1.2 Grabación 50
10.1.3 Scripting 51
10.1.4 Sincronización 51
10.1.4.1 Definición 51
10.1.4.2 Sincronización General 52

10.1.5 Checkpoints *53*
10.1.5.1 Definición 53
10.1.5.2 Insertar Checkpoints durante la grabación 54
10.1.5.3 Insertar Checkpoints después de la grabación 56
10.1.5.4 Nomenclatura de los Checkpoints 57
10.1.6 Parametrización *58*
10.1.6.1 Definición 58
10.1.6.2 Insertar Parámetros de Entrada 59
10.1.6.3 Insertar Parámetros de Salida durante la grabación 62
10.1.6.4 Insertar Parámetros de Salida después de la grabación 64
10.1.6.5 Número de Iteraciones 69
10.1.7 Reusabilidad de Código *71*
10.1.7.1 Crear un Action 72
10.1.7.2 Llamar a un Action reutilizable 73
10.1.8 Recovery Scenarios *74*
10.1.9 Sentencia Exist *76*
10.1.10 Ejecución de Scripts desde UFT e Informe de Resultados *77*
10.2 CONSTRUCCIÓN DEL CASO DE PRUEBA API 80
10.2.1 Importación del proyecto SoapUI *80*
10.2.1.1 Configuración del proyecto SoapUI 81
10.2.2 Creación del test desde UFT *83*
10.2.3 Datos de entrada en Tabla de Datos *91*
10.2.4 Almacenar el caso en ALM *102*

11 EJECUCIÓN DESDE ALM 106

11.1 BATERÍAS DE PRUEBAS 106
11.1.1 Diseño de las Baterías de Pruebas *106*
11.1.2 Ejecución de las Baterías de Pruebas Automáticas *109*
11.1.3 Ejecución de Baterías de Pruebas de Forma Desatendida *110*
11.1.4 Análisis de resultados de ejecución *112*
11.1.4.1 Pruebas GUI 113
11.1.4.2 Pruebas API 114
11.2 PROBLEMÁTICAS EN LA EJECUCIÓN DE BATERÍAS DE PRUEBAS (TEST SETS) 116
11.2.1 Construcción de Scripts Independientes *117*
11.2.2 Errores no controlados *117*

1 Introducción

El presente Documento es la Guía de Programación de Scripts de Pruebas Funcionales con la Herramienta de **HPE** (ahora **MicroFocus**) **Unified Functional Testing v. 14.00**.

Esta Guía **no es un Manual de Programación** de **Unified Functional Testing (UFT)**. Los Manuales Oficiales de UFT los podemos encontrar en la ayuda on-line de la herramienta.

Las reglas de nomenclatura, recomendaciones y procedimientos que aquí se exponen son **de obligado cumplimiento** para todas aquellas personas que vayan a desarrollar Scripts en **UFT** para la DGT: La Oficina de Pruebas, Empresas externas de Desarrollo, etc…

Si no se siguen estas normas, el Grupo de Calidad de la DGT podría rechazar la entrega de los Scripts y/o de sus componentes asociados.

2 Terminología y Acrónimos

En este apartado se ofrece la lista de los términos y acrónimos específicos que aparecen en el presente documento, y sus definiciones:

Término /Acrónimo	Definición
ALM	*Application Lifecycle Managent*. Herramienta de Micro Focus para la gestión del ciclo de vida completo de un sistema informático: gestión de releases, requisitos, casos de prueba, recursos de pruebas, ejecuciones de pruebas, incidencias, informes de pruebas, etc.
API	*Application Programming Interface*. Interfaz de Programación de Aplicaciones.
CCA	Componente Común Automatizado.
DENT	Datos de Entrada.
GUI	*Graphic User Interface*. Interfaz Gráfica de Usuario.
LIBF	Librería Funcional.
PF	Prueba Funcional.
PFA	Prueba Funcional Automatizada.
RECU	Escenario de Recuperación.
RO	Repositorio de Objetos.
ROBJ	Repositorio de Objetos.
Script	En el contexto de la herramienta UFT, llamamos script de forma genérica a una prueba funcional automatizada (PFA), o a un componente común automatizado (CCA). Un Script de UFT consta de código fuente VBasic que se ejecuta, y otros recursos de pruebas como son Tablas de datos, Repositorios de Objetos, Librerías de funciones, etc.
Transacción de negocio	En el contexto de este documento de pruebas, una transacción de negocio equivale a una navegación por la aplicación en pruebas, mediante la cual se realiza una operación de negocio en dicha aplicación. Otros sinónimos son: proceso de negocio, navegación de negocio, flujo de negocio, caso de uso de negocio, etc.
UFT	*Unified Funtional Testing*. Herramienta de Micro Focus para la ejecución de scripts de pruebas funcionales automatizadas, tanto de GUI como de API.

3 Beneficios de la Automatización

La automatización de las pruebas permite optimizar la calidad del software, acelerar el proceso de pruebas e incrementar la eficiencia de las pruebas.

La automatización de las pruebas ofrece las siguientes ventajas frente a las pruebas manuales:

- Reutilización: Las pruebas automatizadas pueden ser reutilizadas.
- Repetición: La ejecución de las pruebas automáticas puede ser repetida cuantas veces se quiera.
- Fiabilidad: Las pruebas automáticas proporcionan siempre el mismo resultado.
- La ejecución de las pruebas automáticas se pueden planificar y puede realizarse de forma desatendida.

Se recomienda automatizar las pruebas tanto de GUI como APIs:

✓ De regresión.
✓ Que se van a ejecutar un número elevado de veces.
✓ De procesos críticos de la aplicación.
✓ Cuando se quieran minimizar los tiempos de ejecución de las baterías de pruebas y/o realizar dichas ejecuciones de forma desatendida.
✓ Que tengan un volumen de datos de entrada grande o cuyo manejo sea costoso de realizar.

No se automatizan las pruebas:

X Que requieren de una sola ejecución
X Que no se pueden predecir sus resultados
X De aplicaciones inestables

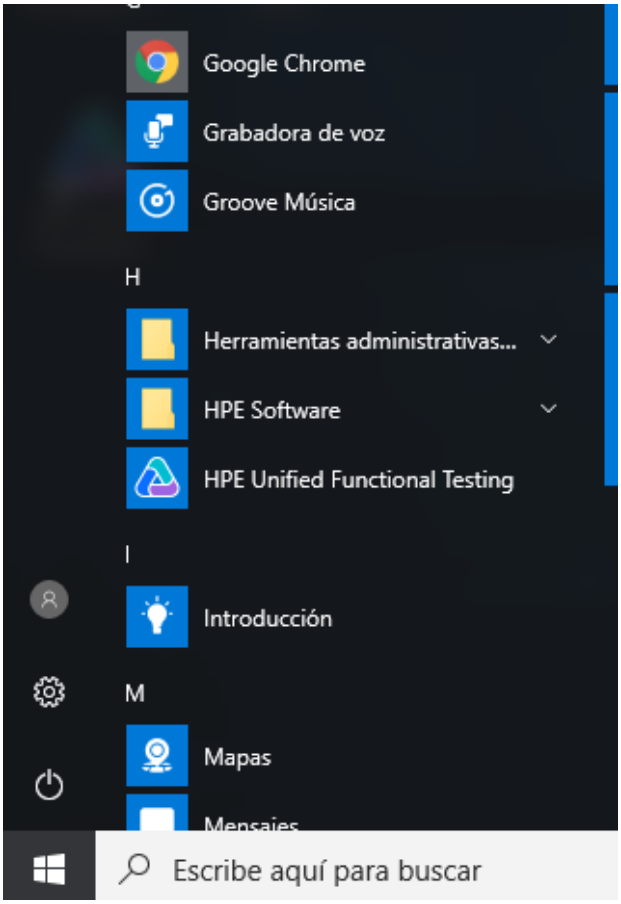
En definitiva, el objetivo de la automatización funcional es el de disponer de una serie de casos de prueba que puedan ser ejecutados repetidas veces de una forma rápida y fiable por un menor coste que si se realizase de forma manual.

Se utilizará la herramienta UFT para automatizar test funcionales tanto de Interfaz Gráfico de Usuario (GUI Tests) como de Interfaz de Programación de Aplicaciones (API Tests).

4 Acceso a la aplicación

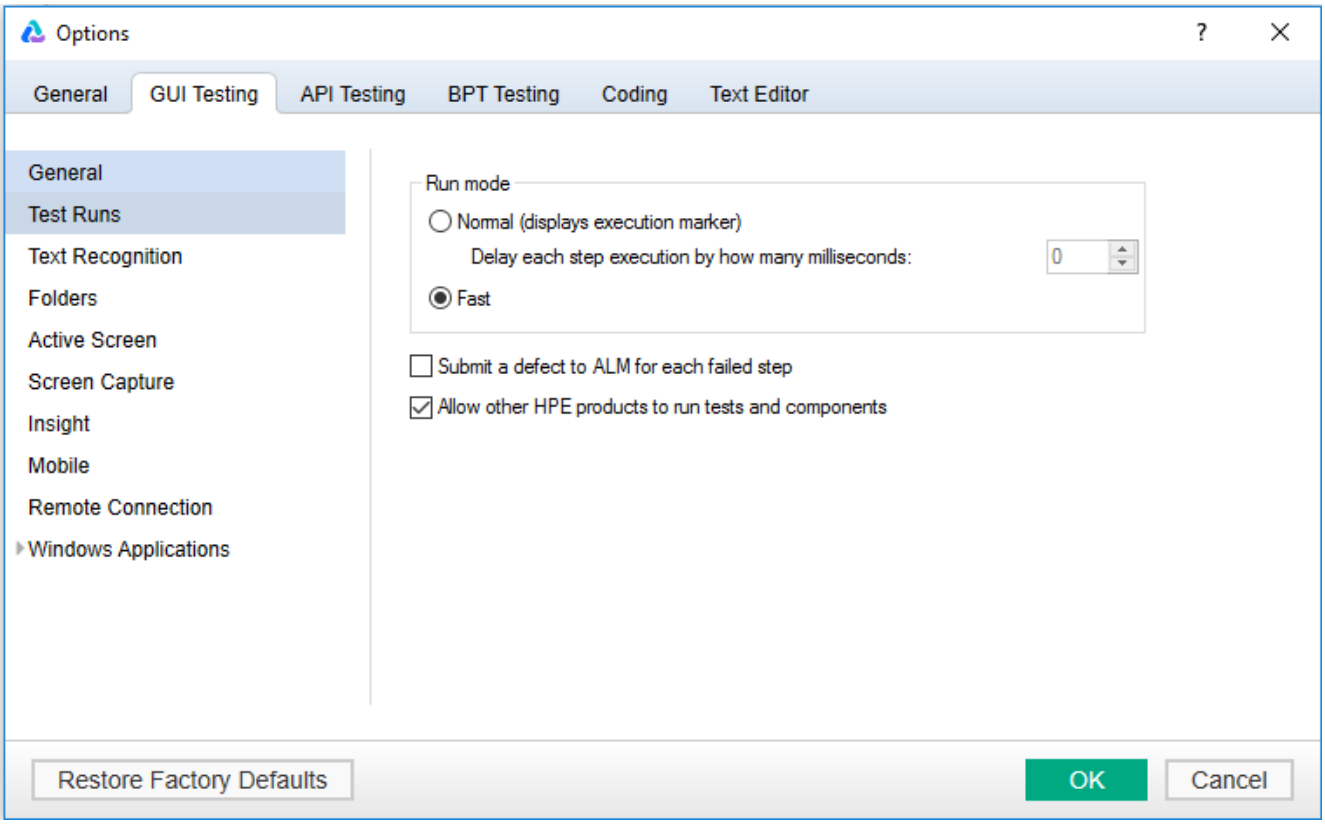
El acceso a la aplicación se realiza haciendo doble clic sobre el icono del Escritorio

o mediante "**Inicio->Programas-> HPE Unified Functional Testing**"



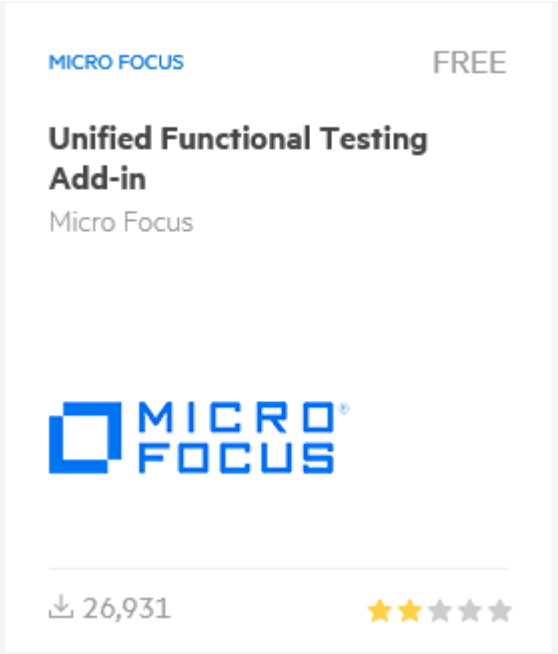
5 Opciones de Configuración de UFT

Desde el menú ***Tools->Options*** se accede a la ventana de opciones de configuración de UFT.

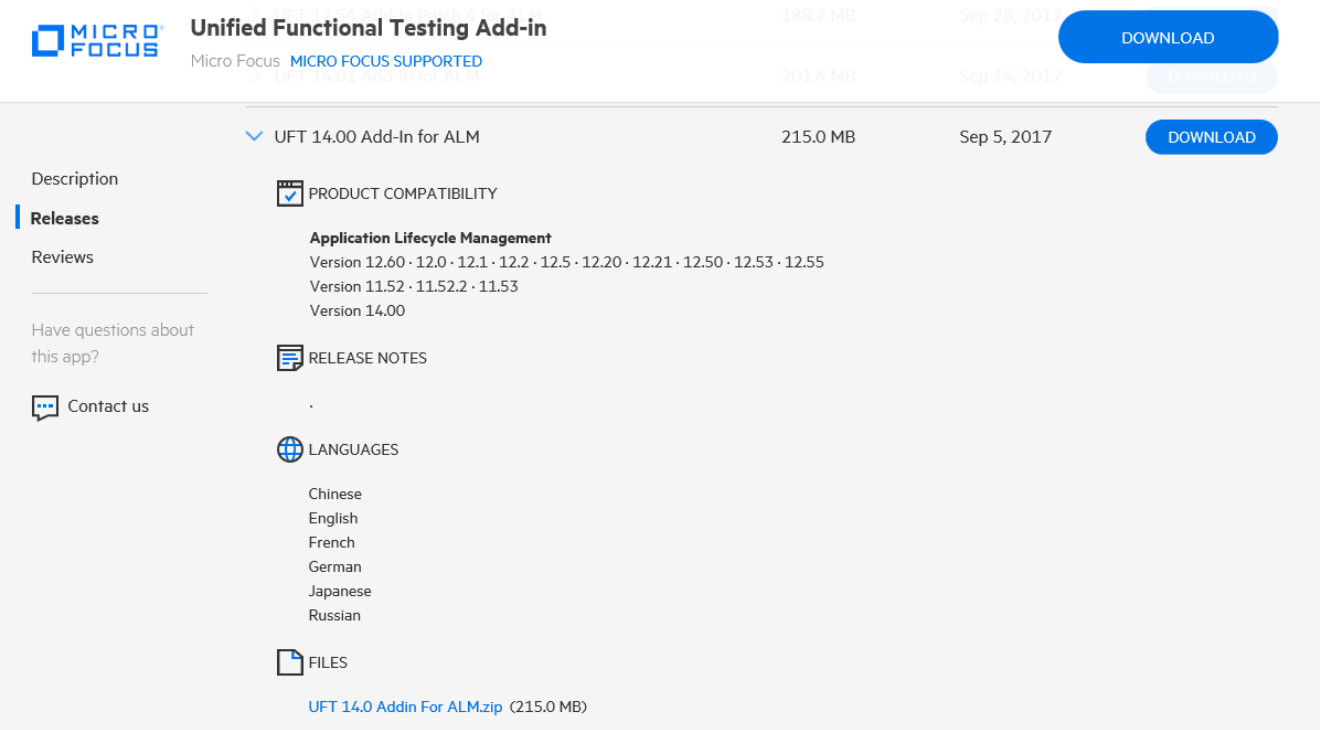


5.1 Instalación Add-In de Application Lifecycle Management

Para poder visualizar el código fuente de los scripts de UFT y los Test Report de las ejecuciones de tests automáticos de UFT, se deberá instalar un add-in. Para ello desde **ALM → Tools → More HP ALM Add-ins** se llega a la página de Micro Focus, y en esta página buscaremos *Unified Functional Testing Add-in*.



Al acceder deberemos descargarnos el que corresponda a nuestra versión de UFT y que soporte la versión de ALM que estamos utilizando.

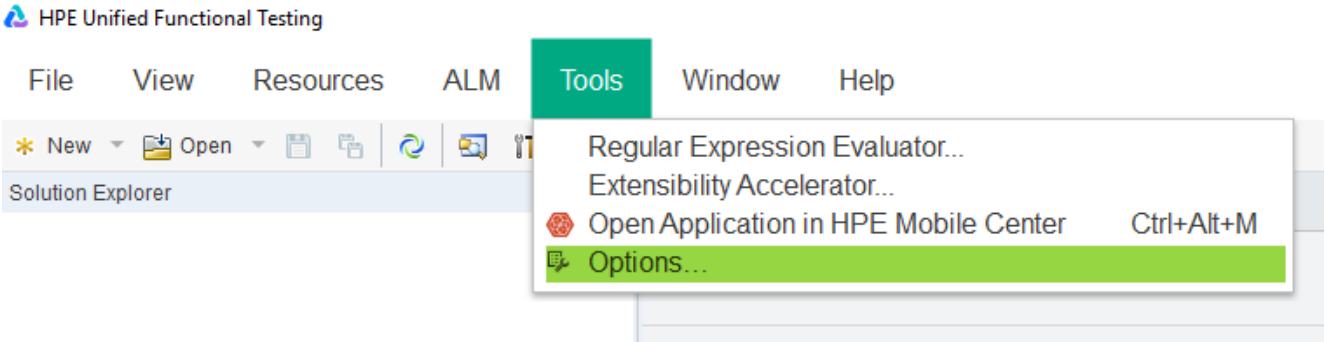


Para finalizar se instala el add-in.

Para poder lanzar desde ALM la ejecución de tests automatizados contra una máquina con UFT, desde dicho UFT se activará esta opción para la integración con ALM.

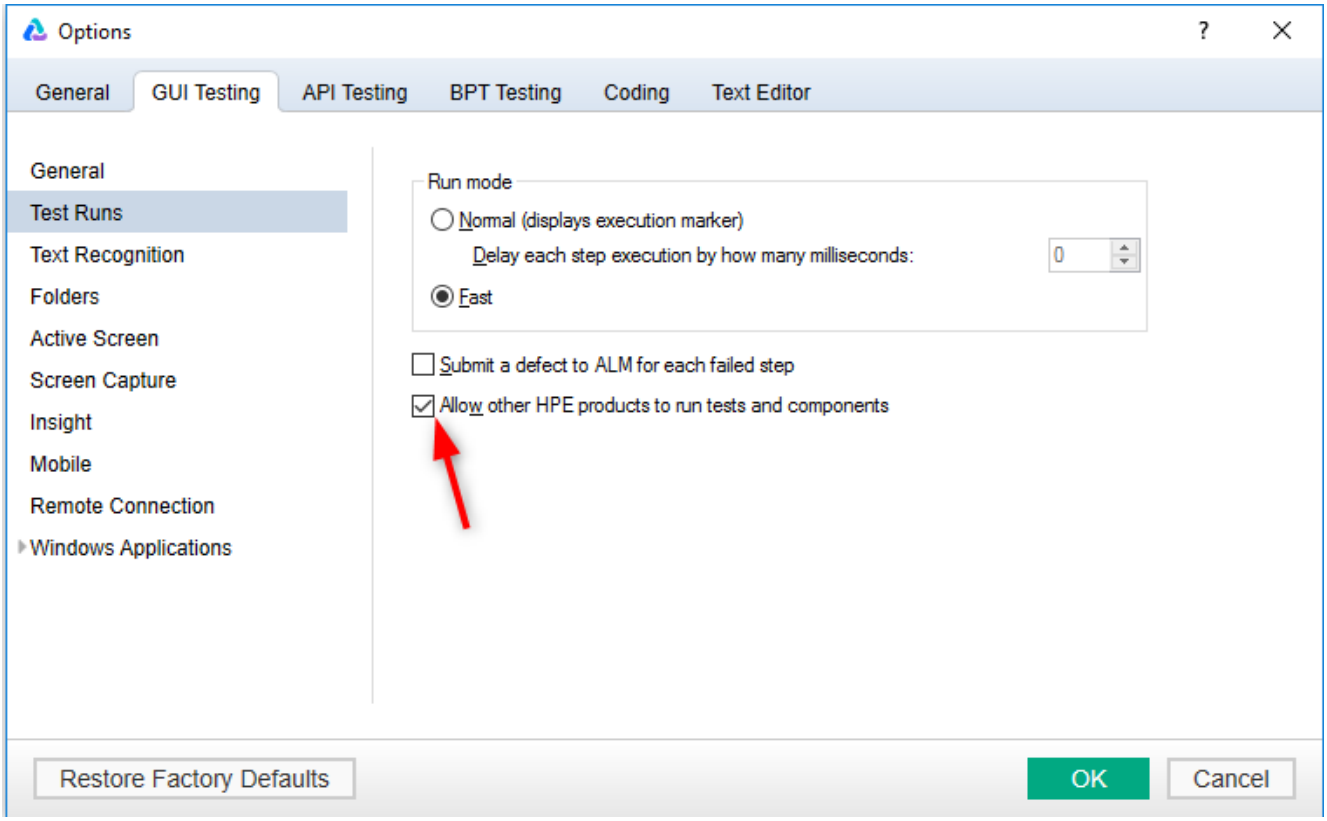
Los pasos a realizar son:

1. Ir al menú y ejecutar "***Tools->Options***"



2. Pulsar en la pestaña GUI Testing y en el grupo "***Run***"

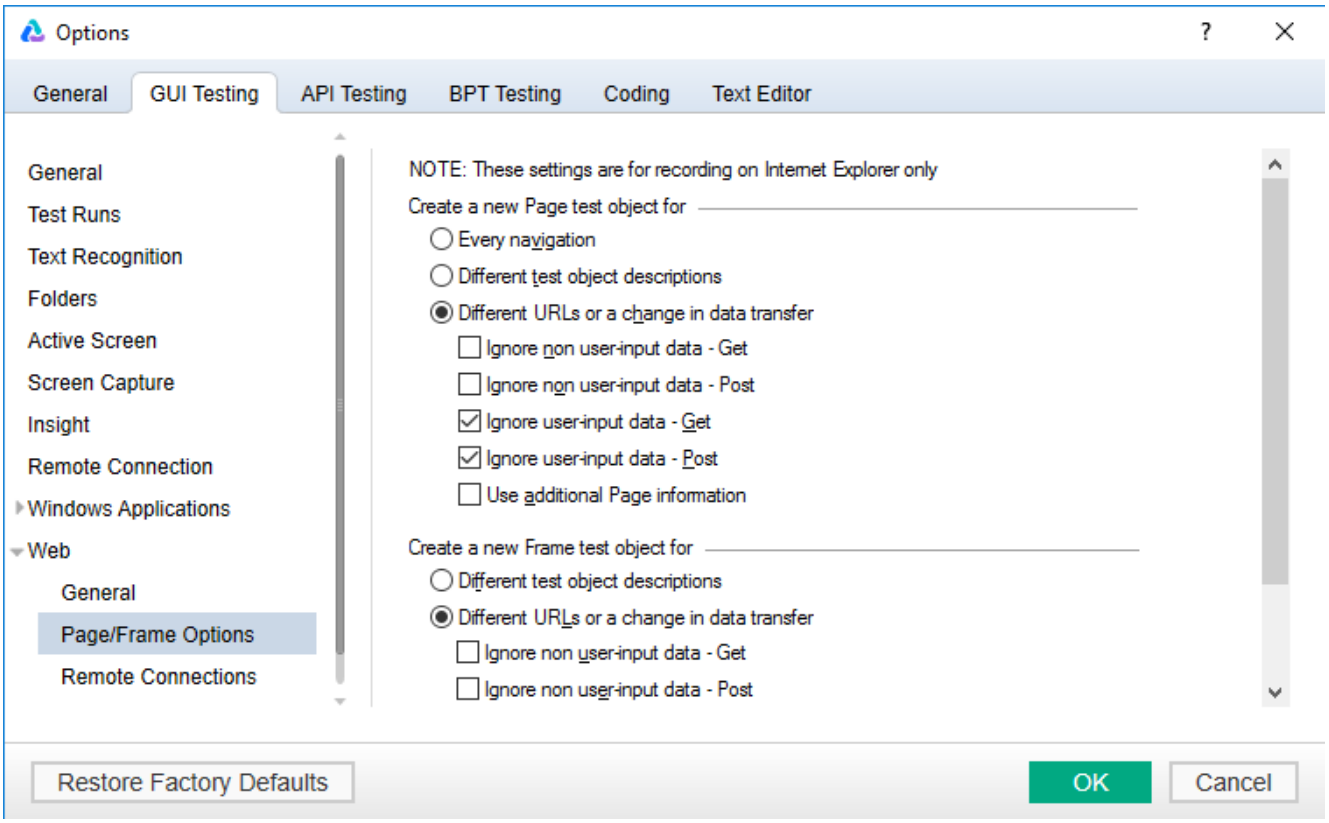
3. Activar la casilla "Allow other HP products to run tests and components"



4. Pulsar en el botón "***OK***"

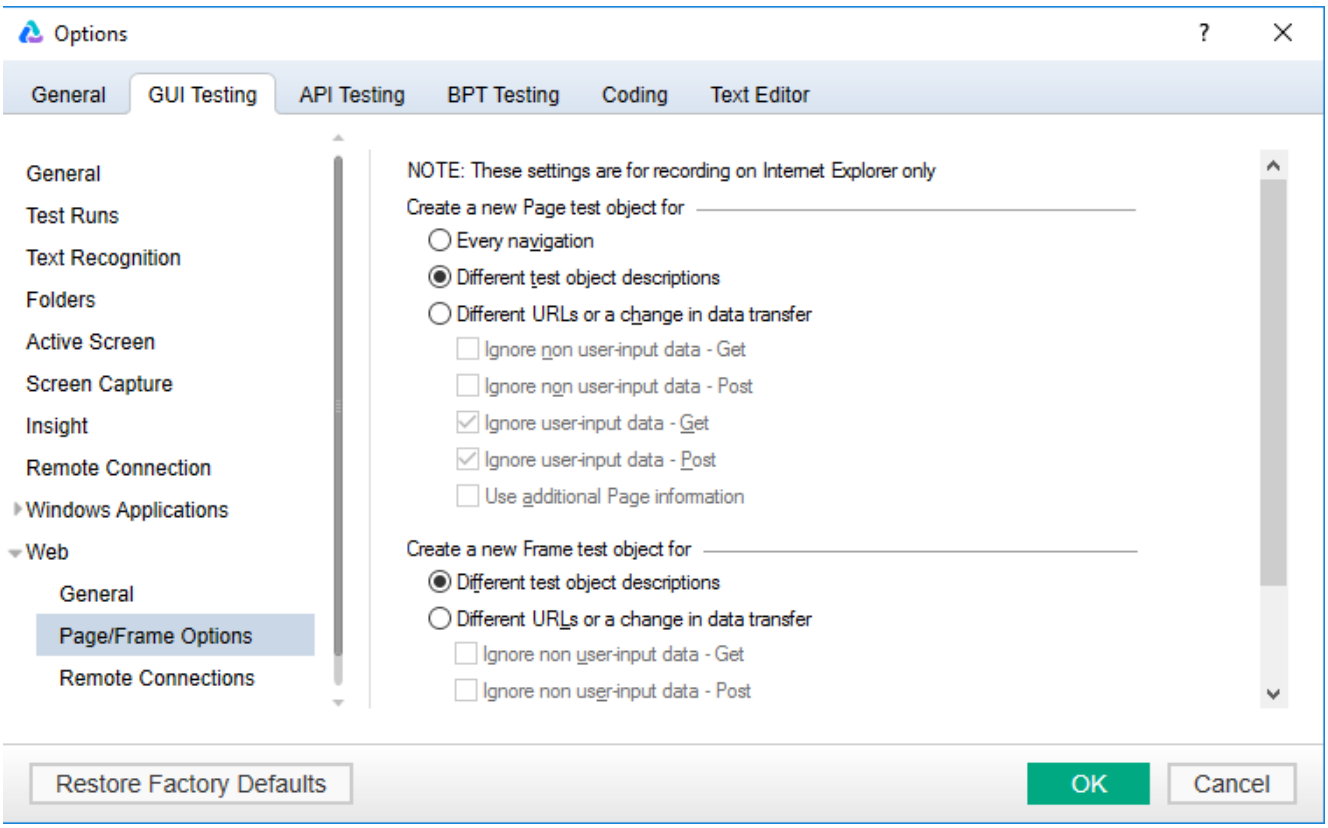
5.2 Page/Frame Options

Para aplicaciones Web la configuración por defecto que tiene la herramienta UFT para añadir objetos de tipo *Page* y *Frame* es la que aparece en la imagen siguiente:



Con esta configuración puede ocurrir que se añadan al Repositorio de Objetos numerosos objetos de tipo *Page* y *Frame* que realmente son el mismo objeto. Esto hace que la localización de los objetos dentro de la jerarquía de objetos sea complicada y el script sea menos mantenible.

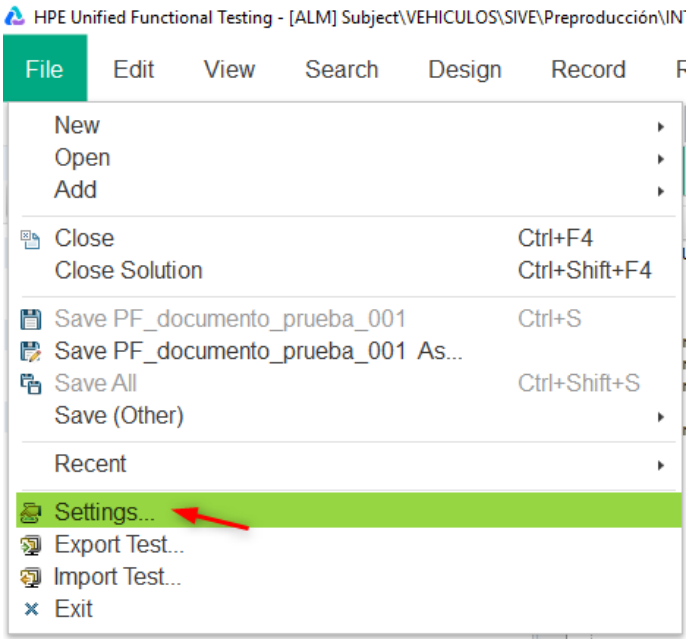
Para evitar esta situación se puede configurar la herramienta para que solo añada al Repositorio de Objetos los objetos *Page* y *Frame* cuando tenga diferente descripción, para ello hay que configurar la herramienta con las opciones que se ven en la siguiente pantalla:



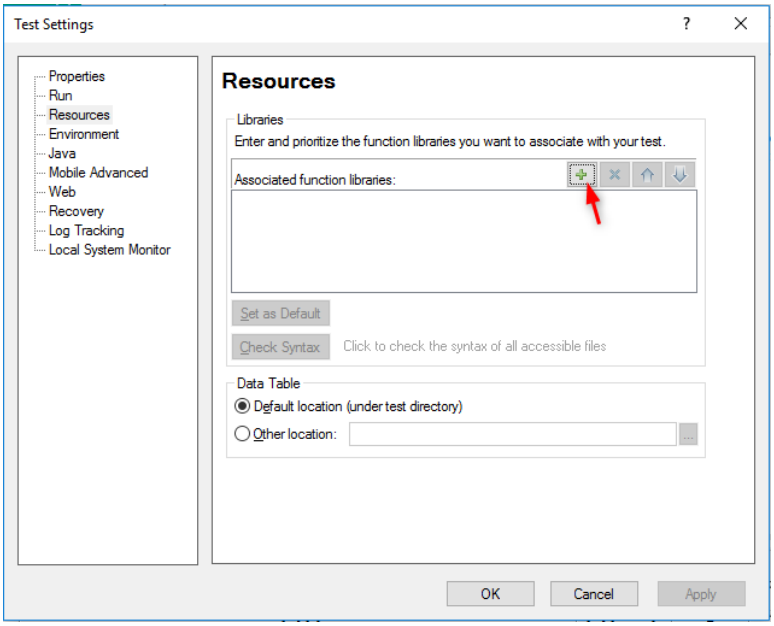
5.3 Asociar la librería de funciones a todos los nuevos tests

Las librerías de funciones creadas para el aplicativo se deben asociar a todos los nuevos tests que se creen en la herramienta, de manera que se deben de asociar una primera vez antes del comienzo de la grabación de los tests. Los pasos necesarios son los siguientes:

- Ir a File > Settings.



- Seleccionar la pestaña **Resources**.
- Pulsar el botón Adds a new file to the file list (icono +).

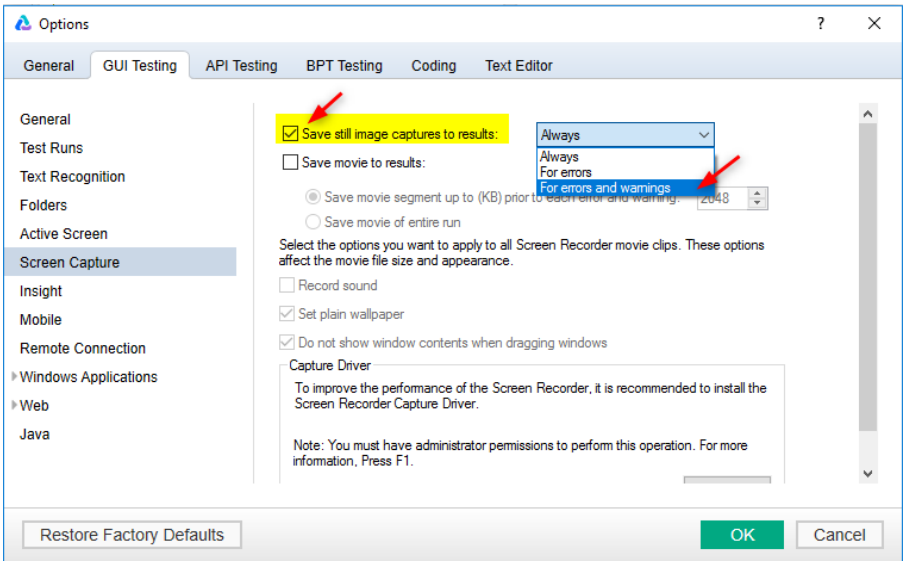


- Seleccionar la ruta de la librería de funciones.
- Pulsar el botón **Set as Default**.
- Pulsar el botón **Aceptar**.

5.4 Capturar imágenes cuando se producen errores o warning en la ejecución.

Cuando la ejecución de un script automático falla o se ejecuta con warning es necesario guardar la imagen de la aplicación para facilitar la resolución de la incidencia. Los pasos son los siguientes:

- En Tools>Options seleccionar la pestaña GUI Testing >Screen Capture.
- Habilitar el check Save still image captures to results.
- En la lista asociada a la derecha seleccionar la opción **For errors and warnings**.



- Pulsar el botón **Aceptar**.

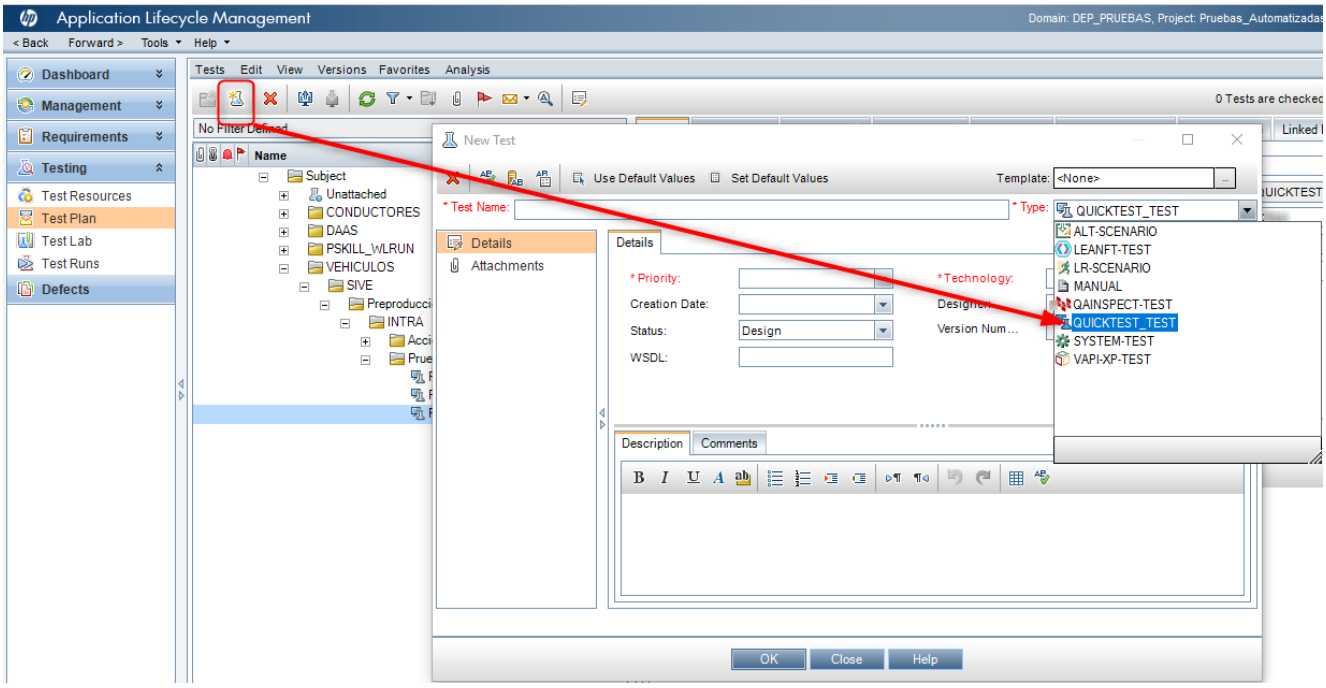
Importante: Estas configuraciones son por cliente de UFT de manera que es necesario que todos los miembros del equipo tengan implementada esta configuración en su máquina antes del comienzo de la automatización.

6 Integración ALM y UFT

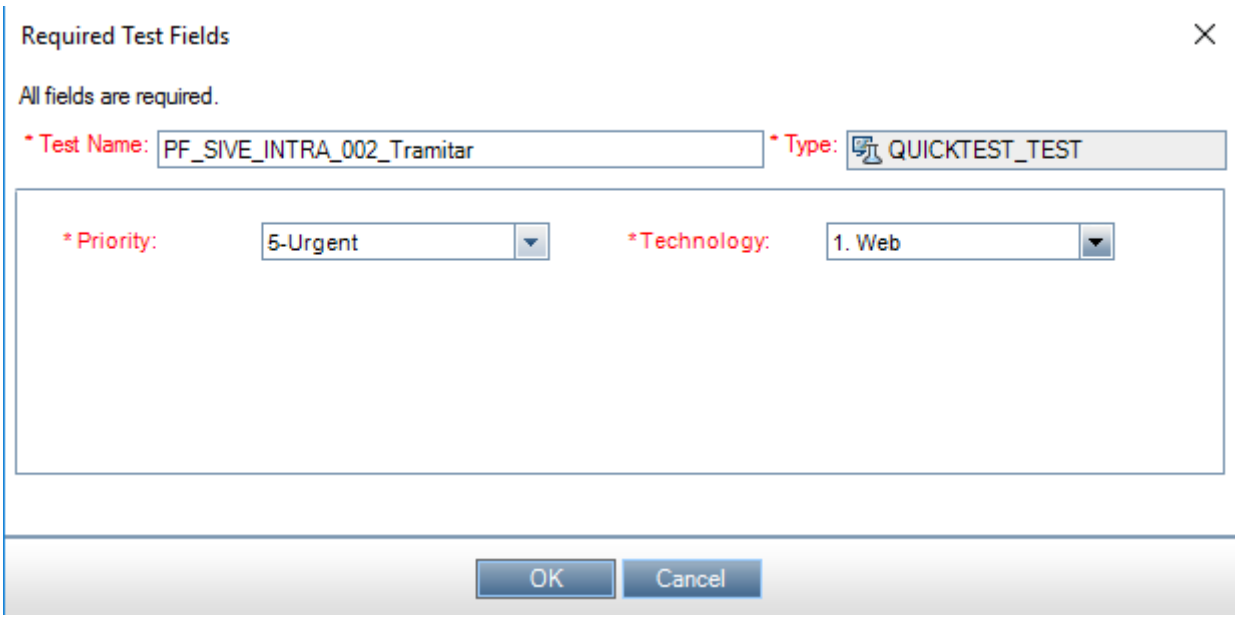
6.1 Crear Scripts UFT en ALM

Para crear un nuevo script UFT en ALM se deben realizar los siguientes pasos:

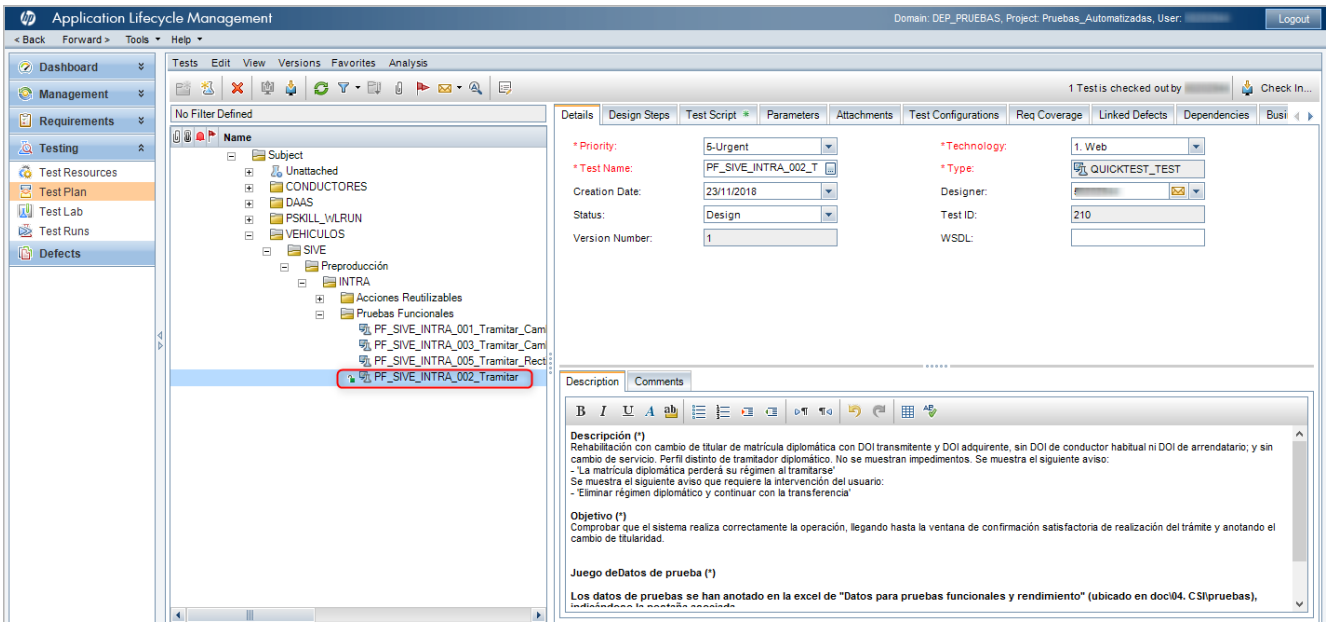
Una vez conectado a ALM y posicionado en el módulo Test Plan sobre la carpeta donde se quiere añadir un caso de prueba automático con UFT, añadir un nuevo caso de prueba pulsando sobre **"*New Test*"**. Seleccionar un nuevo caso de tipo **QUICKTEST_TEST,** introducir el nombre que se quiere dar al nuevo caso de prueba y pulsar **"*OK*"**.



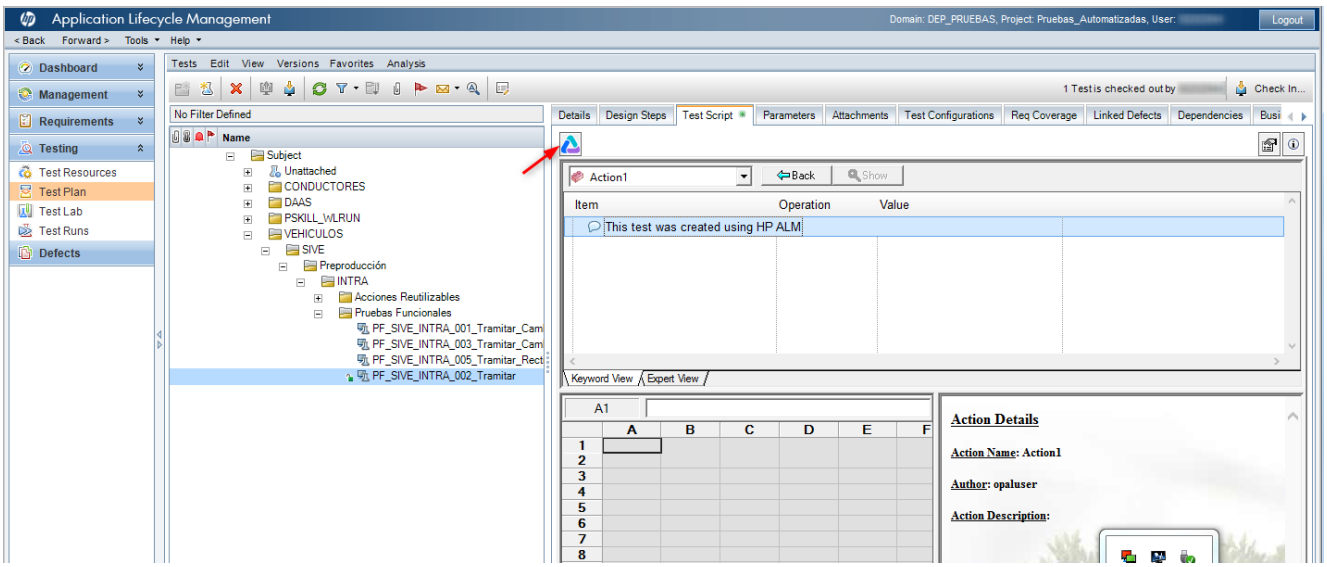
Aparece la ventana que solicita campos obligatorios del caso de prueba. Seleccionar lo que corresponda y pulsar **"*OK*"**.



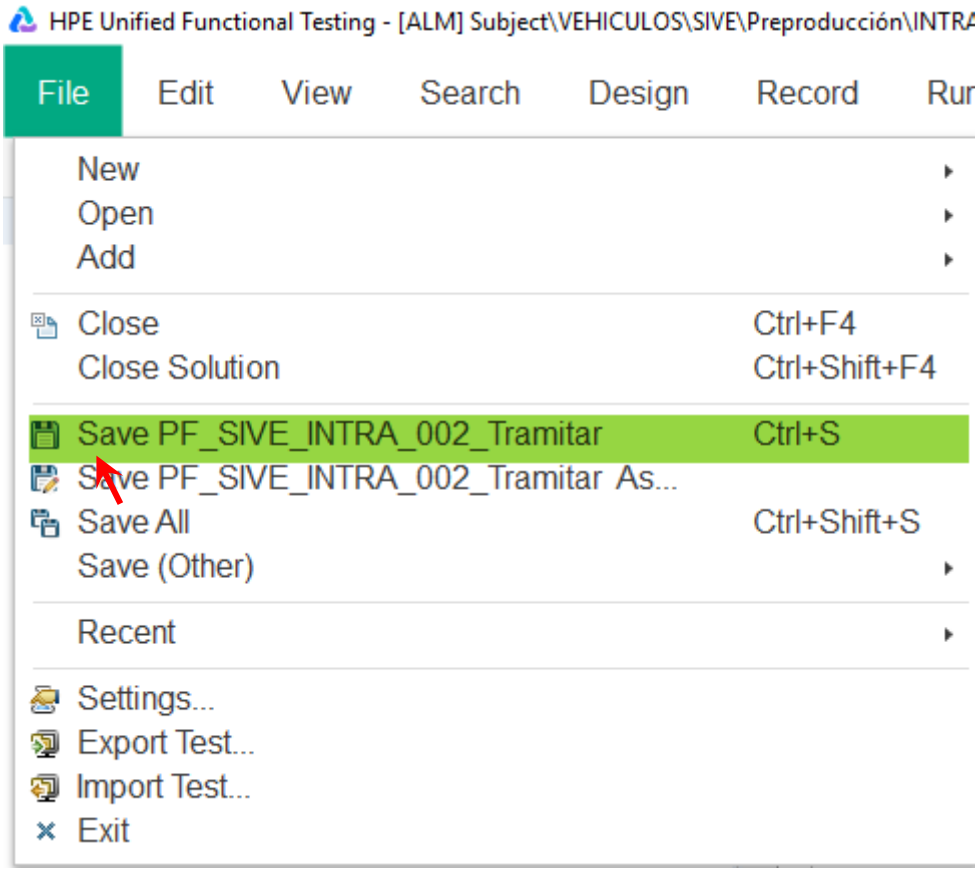
Se crea el nuevo caso de prueba. Completar el resto de campos del caso de prueba en las pestañas Details, Design Steps, Attachments y Req Coverage de ALM.



Desde la pestaña Test Script se puede visualizar el script y abrirlo en UFT pulsando el botón ***"Launch UFT".***

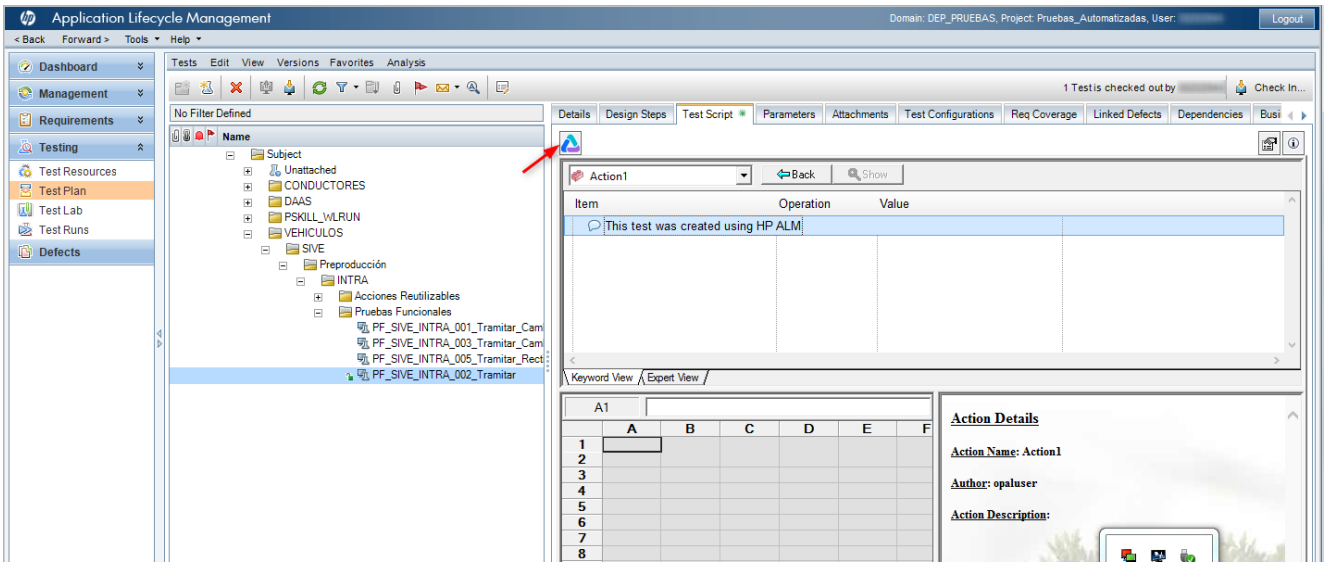


Realizar la grabación/codificación de la prueba y guardar el script pulsando "***File->Save***".



6.2 Abrir Scripts UFT desde ALM

Seleccionar el script y desde la pestaña Test Script pulsar el botón "***Launch UFT***".

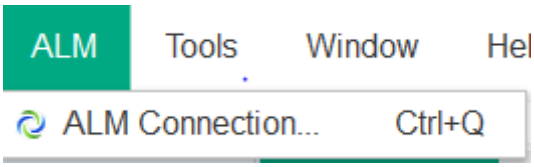


6.3 Conectar a ALM desde UFT

Si no se lanza la herramienta UFT desde ALM, como se ha visto en los apartados anteriores, se debe realizar la conexión al proyecto de ALM correspondiente para poder guardar o abrir scripts en UFT.

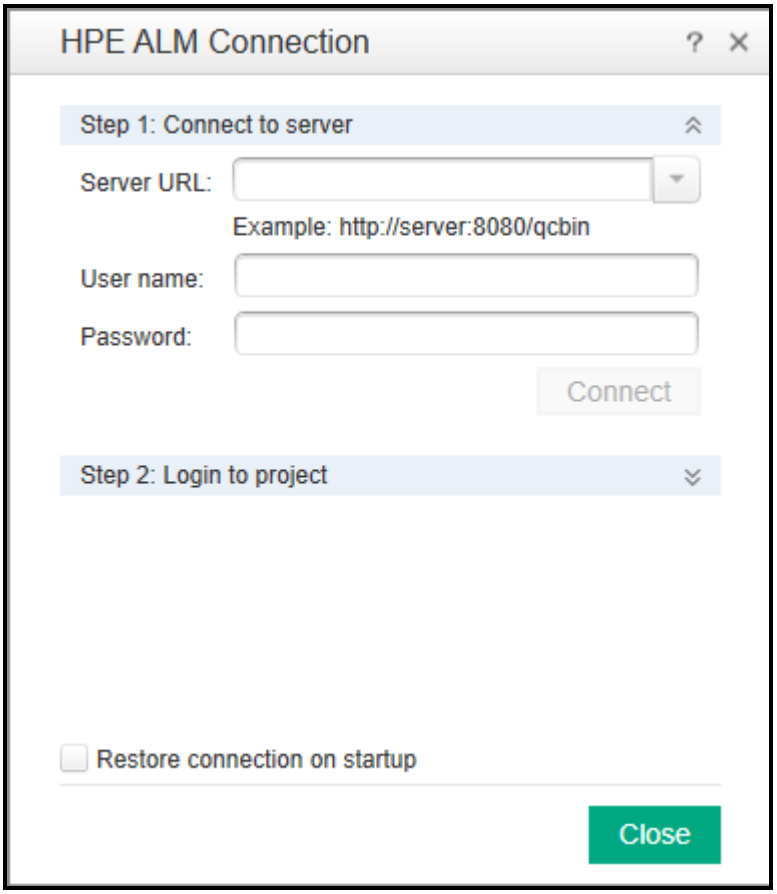
Los pasos a seguir para conectar desde UFT al proyecto de ALM son:

1. Conectarse a UFT pulsando en el menú "ALM -> ALM Connection".



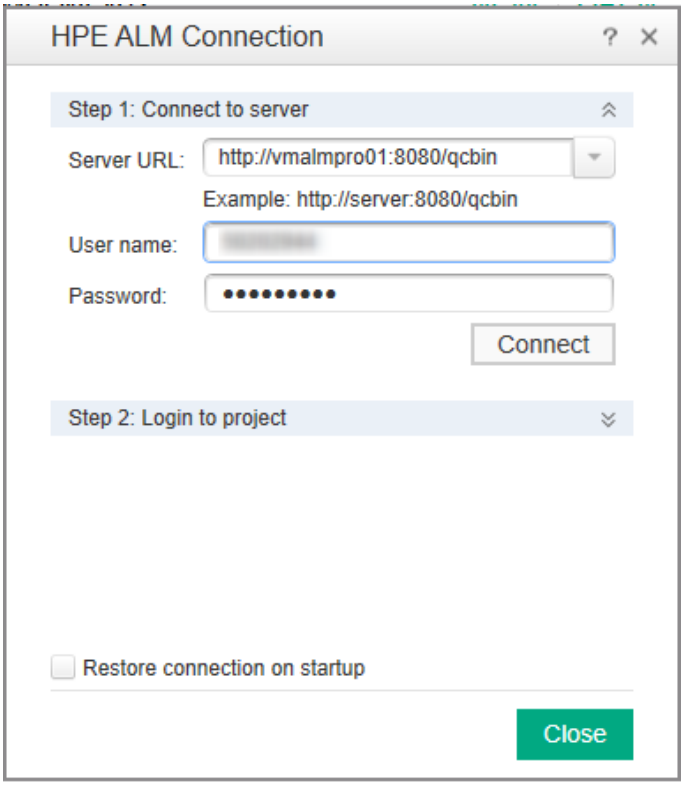
2. Introducir la URL del servidor de Application Lifecycle Management en la casilla "**Server URL**"

En nuestro caso: http://<Máquina Servidor ALM>:8080/qcbin/

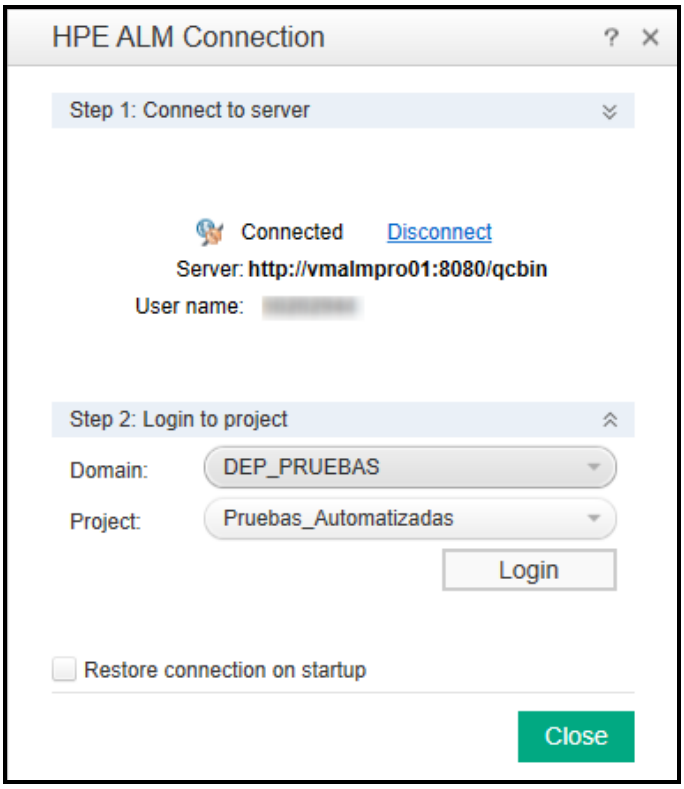


3. Introducir el nombre de usuario (DNI) y clave en el campo "***User Name***" y "***Password***" respectivamente.

4. Pulsar en el botón "***Connect***".

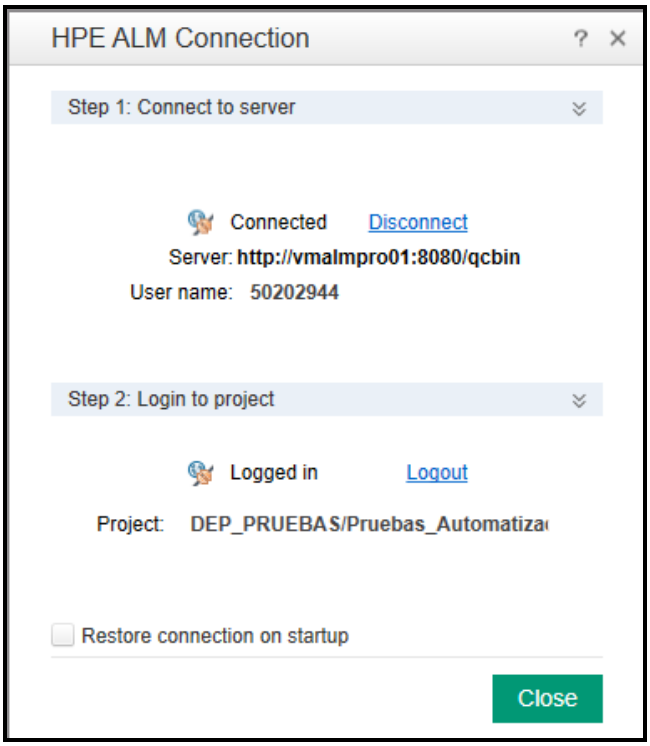


5. Seleccionar un proyecto en el campo "***Project***".

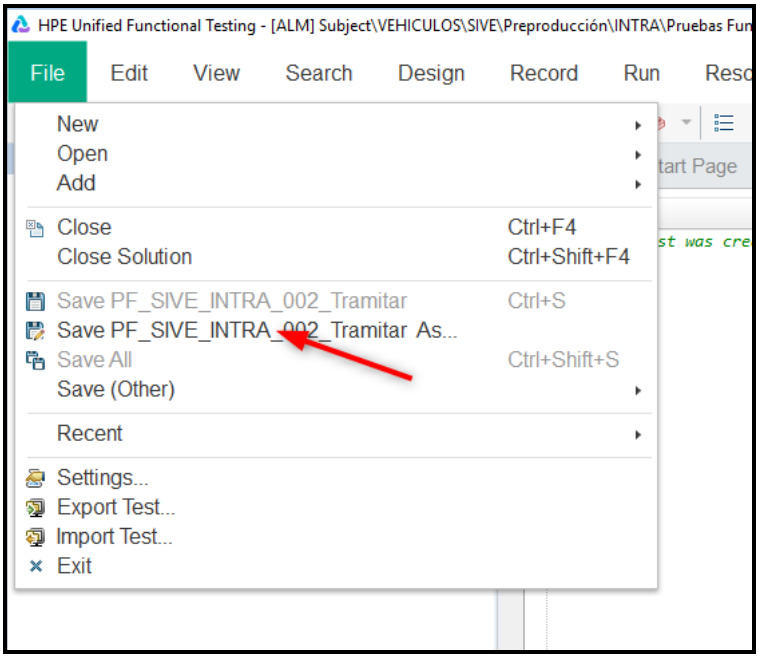


6. Pulsar en el botón "***Login***"

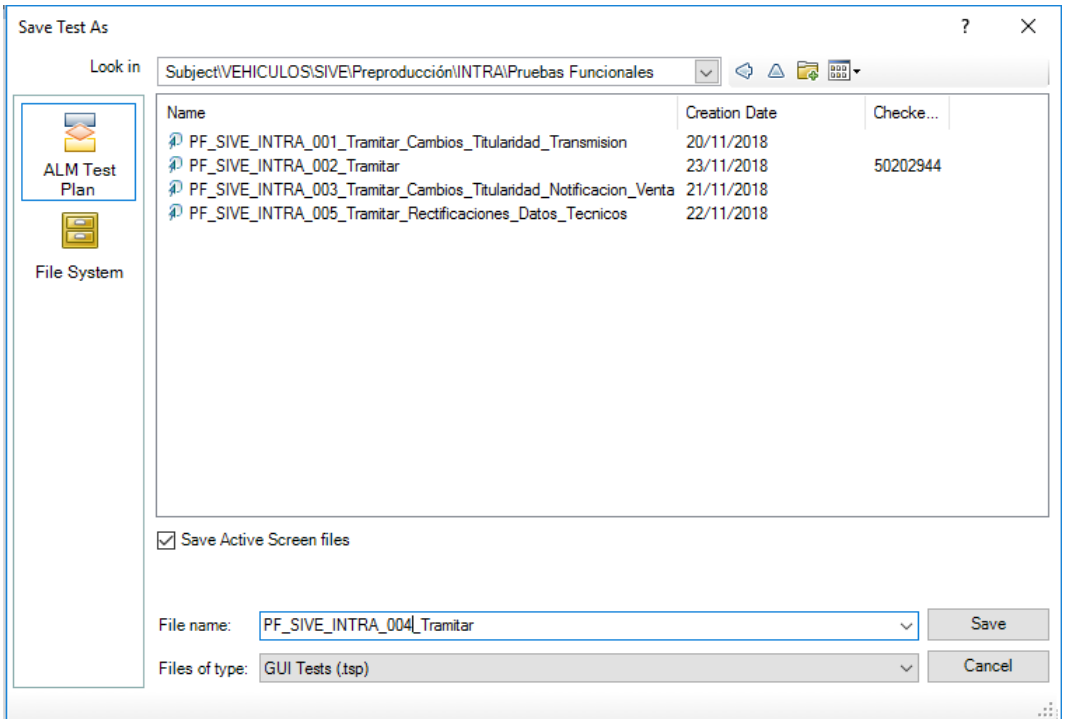
7. Cuando aparezca "Logged in" pulsamos en el botón "***Close***".



8. Desde el menú File se puede seleccionar "***Open***", "***Save***" o "***Save As…***" para abrir o guardar un script respectivamente.



9. Desde esta ventana ya se tiene acceso a las carpetas de ALM y se pueden abrir o guardar los scripts en el proyecto.



7 Repositorio de objetos

Cuando se realiza la grabación de un script funcional, UFT automáticamente guarda todos los objetos que reconoce en un repositorio local. Por defecto, crea un repositorio de objetos por cada "Action" del script.

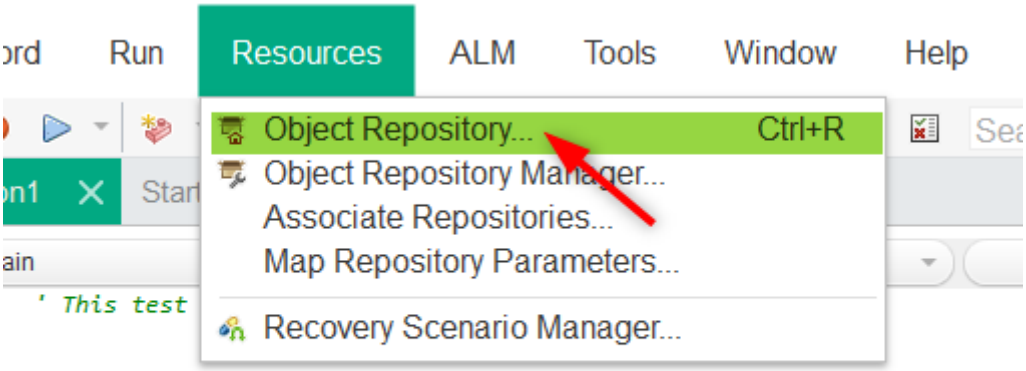
Cada objeto reconocido del repositorio de objetos estará definido por unas propiedades que lo identifican y lo hacen único.

Se pueden utilizar repositorios de objetos compartidos, es decir, almacenar todos los objetos de la aplicación en un repositorio común y usar dicho repositorio en diferentes scripts.

7.1 Acceder al repositorio de objetos

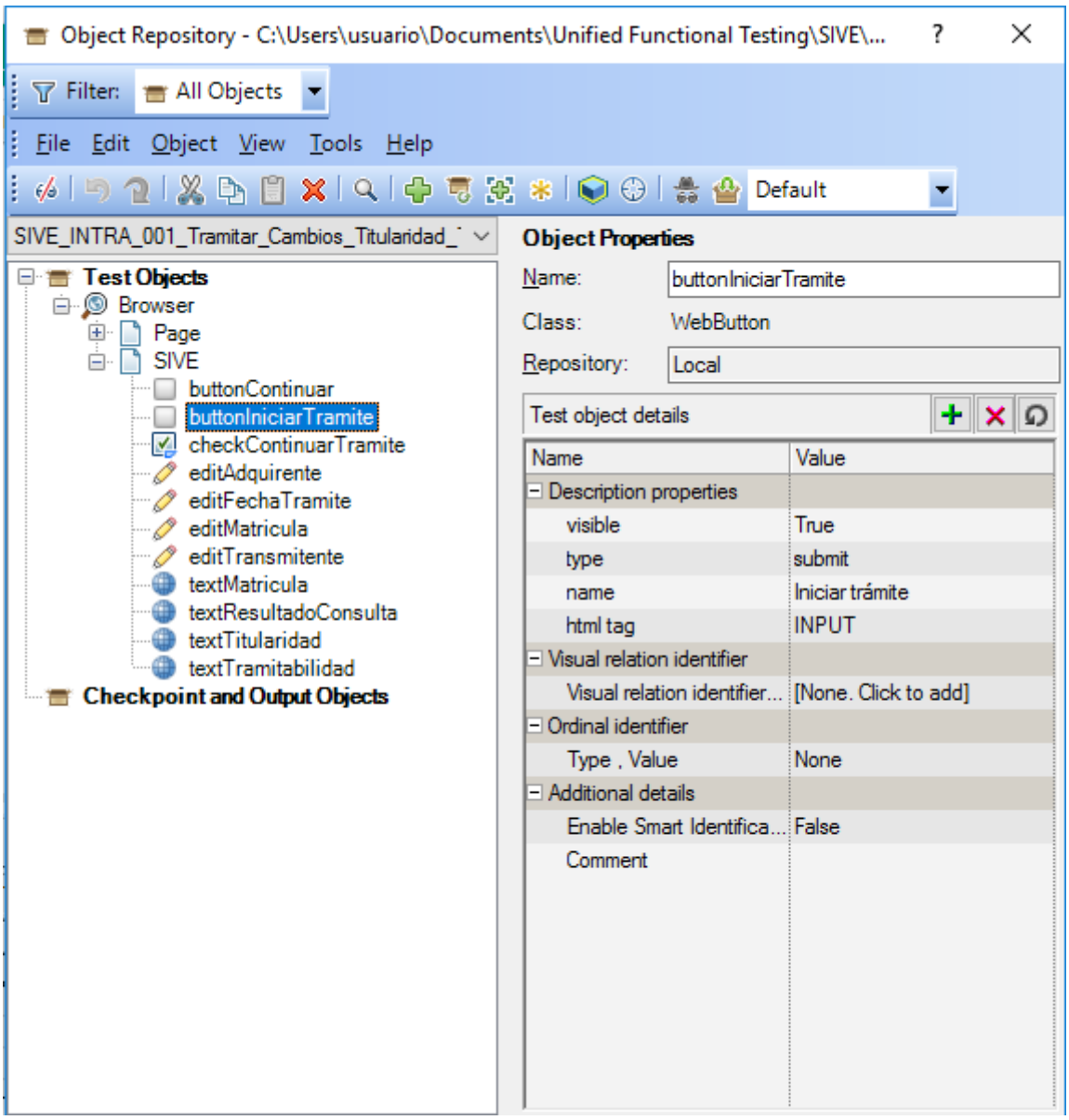
Para acceder al repositorio de objetos se deben realizar los siguientes pasos:

1. Ir al menú y ejecutar "***Resources->Object Repository***" o pulsar sobre el botón de la barra de herramientas.



Aparecerá la pantalla del Repositorio de Objetos.

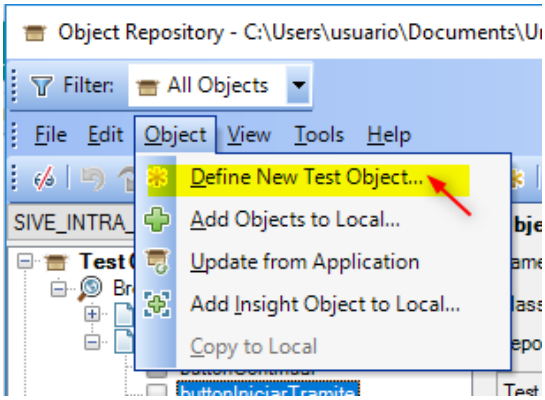
En la parte izquierda están todos los objetos reconocidos, y en la parte derecha las propiedades del objeto seleccionado.

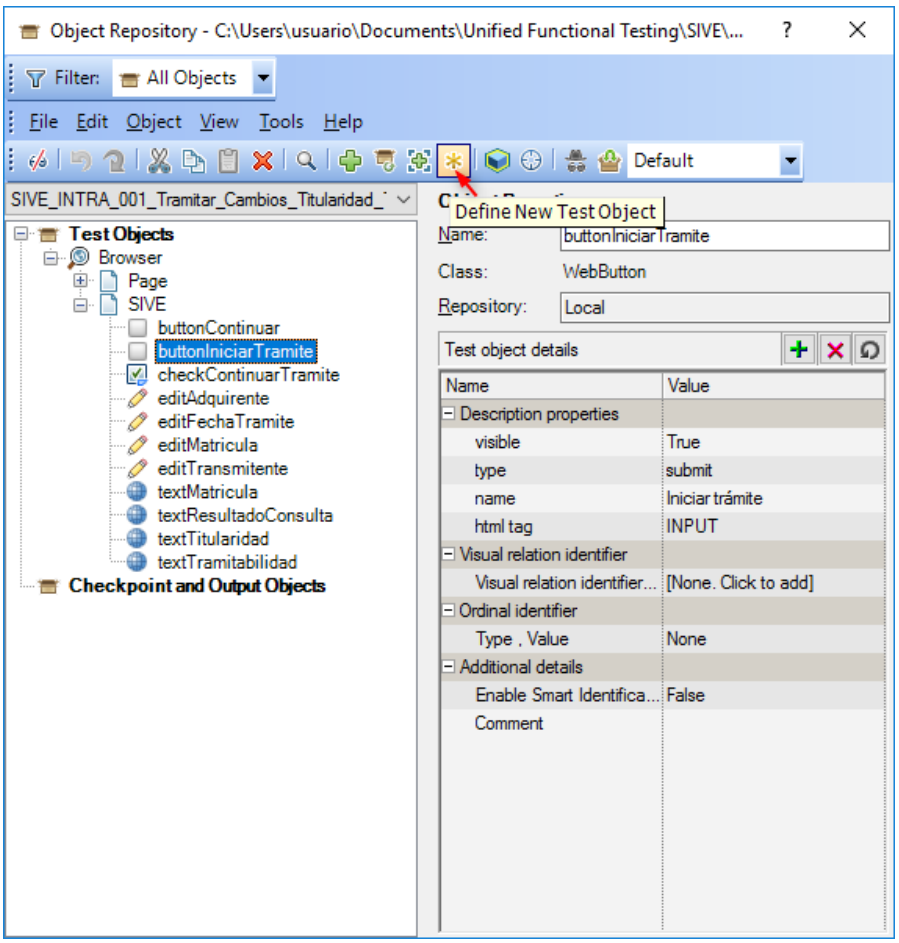

Object Repository - C:\Users\usuario\Documents\Unified Functional Testing\SIVE\...
Filter: All Objects
File Edit Object View Tools Help
Default
SIVE_INTRA_001_Tramitar_Cambios_Titularidad_
Test Objects
Browser
Page
SIVE
buttonContinuar
buttonIniciarTramite
checkContinuarTramite
editAdquirente
editFechaTramite
editMatricula
editTransmitente
textMatricula
textResultadoConsulta
textTitularidad
textTramitabilidad
Checkpoint and Output Objects
Object Properties
Name: buttonIniciarTramite
Class: WebButton
Repository: Local
Test object details
Name
Value
Description properties
visible True
type submit
name Iniciar trámite
html tag INPUT
Visual relation identifier
Visual relation identifier... [None. Click to add]
Ordinal identifier
Type , Value None
Additional details
Enable Smart Identifica... False
Comment

7.2 Definir nuevos objetos en el RO local

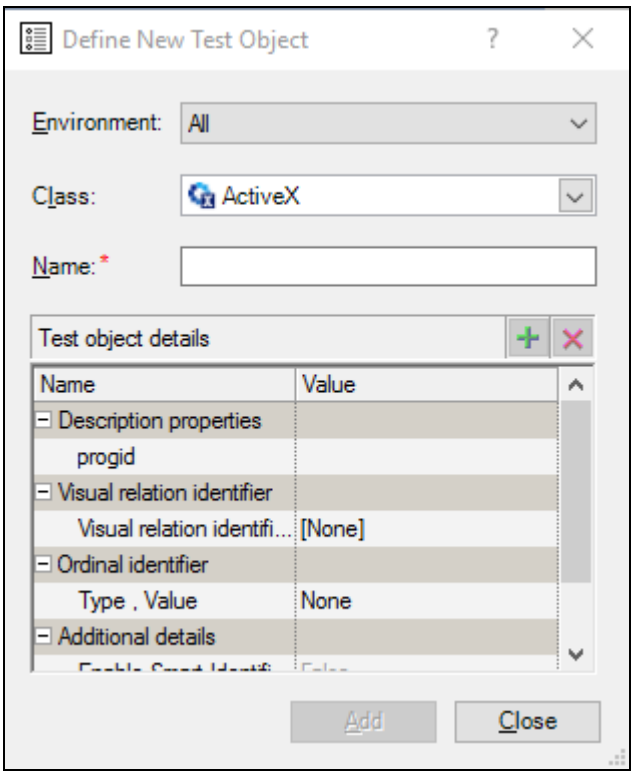
Desde la ventana del Object Repository se pueden definir nuevos objetos en el repositorio local:

1. Ir al menú "***Object-> Define New Test Object***" o pulsar directamente en el botón "✻"





2. Seleccionar en el campo "***Class***" la clase del nuevo objeto, introducir en el campo "***Name***" el nombre del nuevo objeto y rellenar los valores de los campos necesarios en el área "*Test object details*".



3. Pulsar en el botón "***Add***"
4. Pulsar en el botón "***Close***"

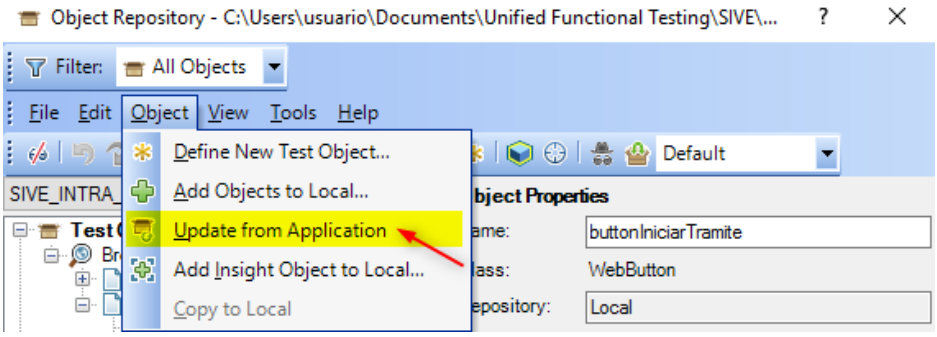
7.3 Actualización de objetos

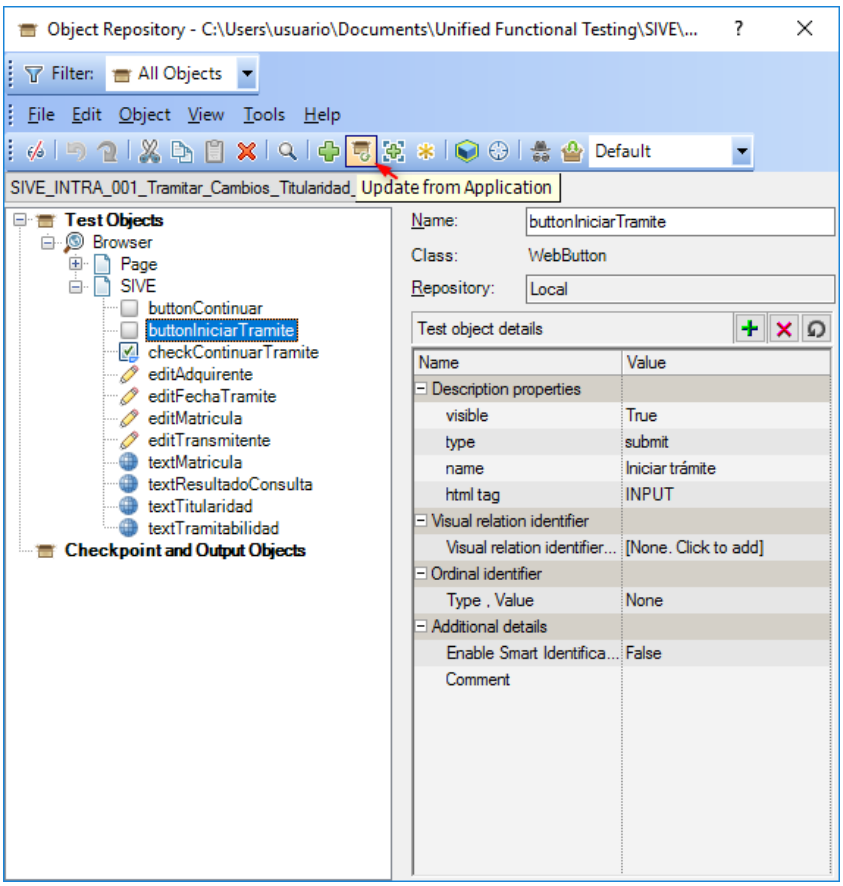
Cuando un objeto dentro del RO no es encontrado por UFT en la aplicación bajo pruebas durante la ejecución de un script porque ha sido modificado en la aplicación por Desarrollo, se pueden actualizar las propiedades de dicho objeto en el RO, bien, a mano, o bien con "***Update from application***".

Los pasos a realizar son:

1. Ejecutar la misma transacción de negocio en la aplicación bajo pruebas hasta llegar al objeto que no se encuentra.

2. Seleccionar en el RO el objeto a actualizar, e ir al menú "***Object-> Update form Application***" o pulsar directamente en el botón " "





3. Con el puntero del ratón, seleccionar el objeto correspondiente en la aplicación bajo pruebas.

7.4 Definición del ciclo de vida del RO

UFT maneja dos tipos de repositorios de objetos:

1. **Repositorio local**: propio de cada acción de UFT y accesible únicamente por la acción propietaria del caso de prueba.
2. **Repositorio global o compartido:** al que acceden todos los casos de pruebas del proyecto de automatización.

Se recomienda el uso de repositorios de objetos compartidos en las siguientes situaciones:

- Aplicaciones con muchos elementos comunes.
- Tests que se componen de múltiples acciones y se utiliza frecuentemente la llamada a otras acciones.
- Las propiedades de los objetos en la aplicación cambian frecuentemente.

Se recomienda el uso de repositorios locales en las siguientes situaciones:

- Aplicaciones con pocos elementos comunes.
- Tests que se componen de acciones simples.
- No se esperan cambios en las propiedades de los objetos.

Se hace necesario valorar el uso de uno u otro tipo de repositorio dependiendo de la situación del aplicativo que se desea automatizar.

Cada uno de los Repositorios de Objetos debe contener la siguiente estructura inicial de 2 niveles:

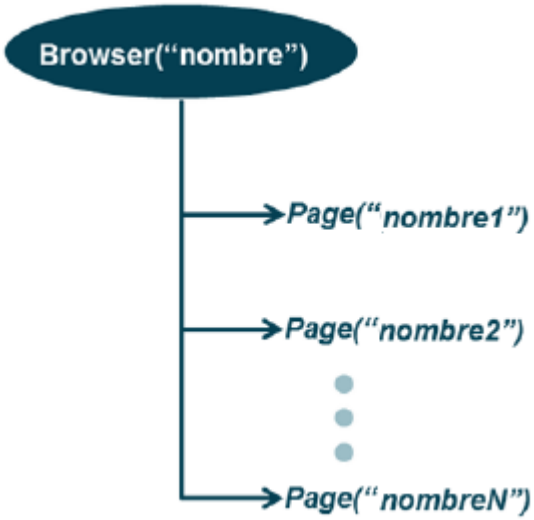


Ilustración 4 Estructura Repositorios de Objetos

Siendo:

- Objeto *Browser*("nombre"): Objeto de la clase Browser correspondiente con el nivel 1 y que identifica el navegador desde el que se ejecuta la prueba. Por cada Repositorio de Objetos existirá un único objeto de la clase Browser. Ejemplo: Browser("Browser")
- Objeto Page("nombre"): Objeto de la clase Page correspondiente con el nivel 2 que identifica una página web y que incluye los objetos de interacción y demás objetos necesarios para la Automatización de funcionalidades. En la medida de lo posible se utilizará un único objeto de clase Page, pero si se requiriera más de uno el nombre deberá ser descriptivo de la página donde nos encontramos, ejemplo: Page("AltaUsuario").

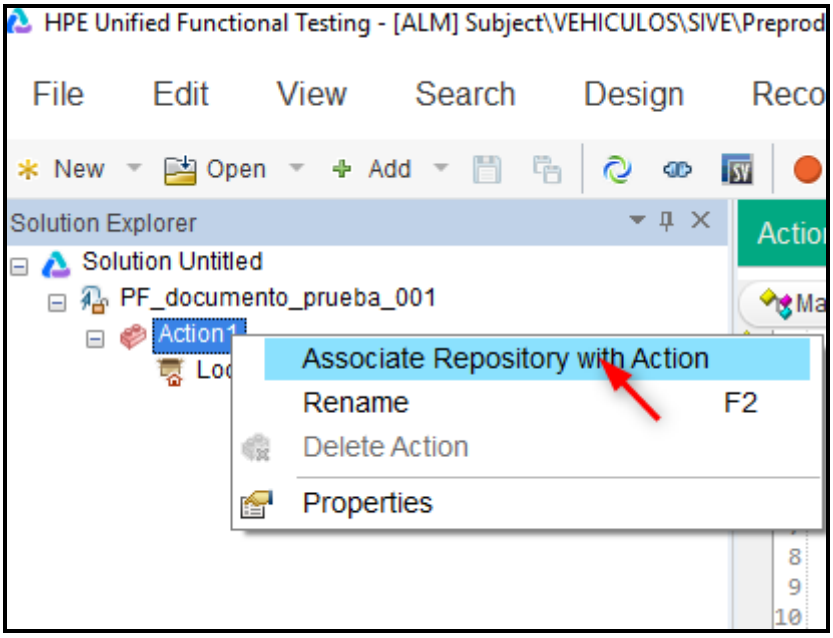
En el nivel 3 se almacenarán los objetos atómicos (botones, cajas de texto, listas desplegables), o agrupaciones de objetos atómicos, cuyo nombre debe cumplir las siguientes reglas:

- Comenzar por carácter del alfabeto en minúscula.
- En caso de tratarse de un objeto atómico (botón, caja de texto, tabla, etc.), el nombre debe corresponderse con el nombre de la etiqueta mostrada en la GUI, y en caso de que no se incluya deberá tener un nombre descriptivo que permita identificar de forma clara y unívoca al objeto en la pantalla.
- Si el nombre es compuesto, cada una de las palabras que forman el nombre compuesto, a partir de la segunda palabra, comenzará por letra mayúscula.

7.4.1 Asociación de RO a Actions

Para asociar el RO a cada action de un nuevo script es necesario:

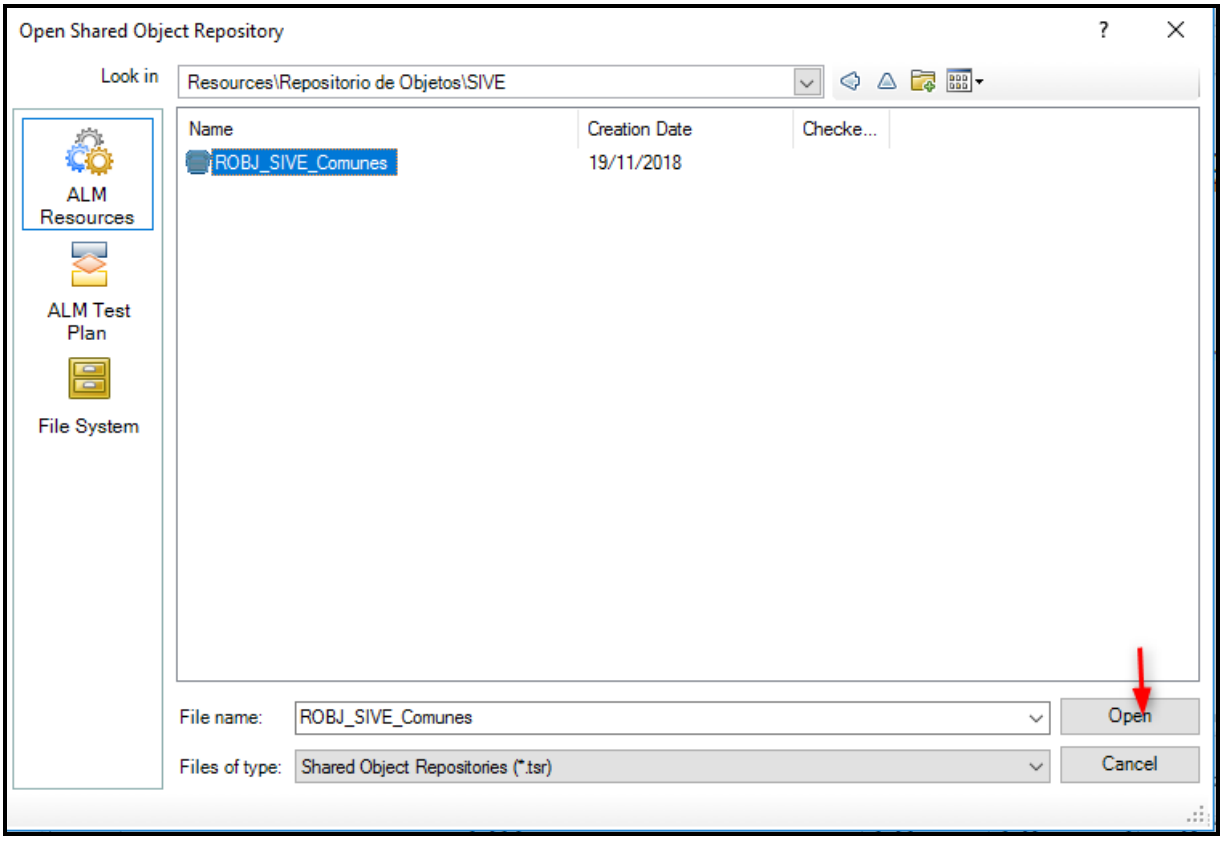
- Abrir script
- Seleccionar una acción y pulsar con el botón derecho.
- Seleccionar la opción “Asociate Repository with Action”



- Seleccionar el repositorio compartido que se encuentra guardado en ALM en la siguiente localización:

  [ALM] Resources\Repositorios de Objetos

- Pulsar el botón Open.



7.4.2 Gestión del RO Local

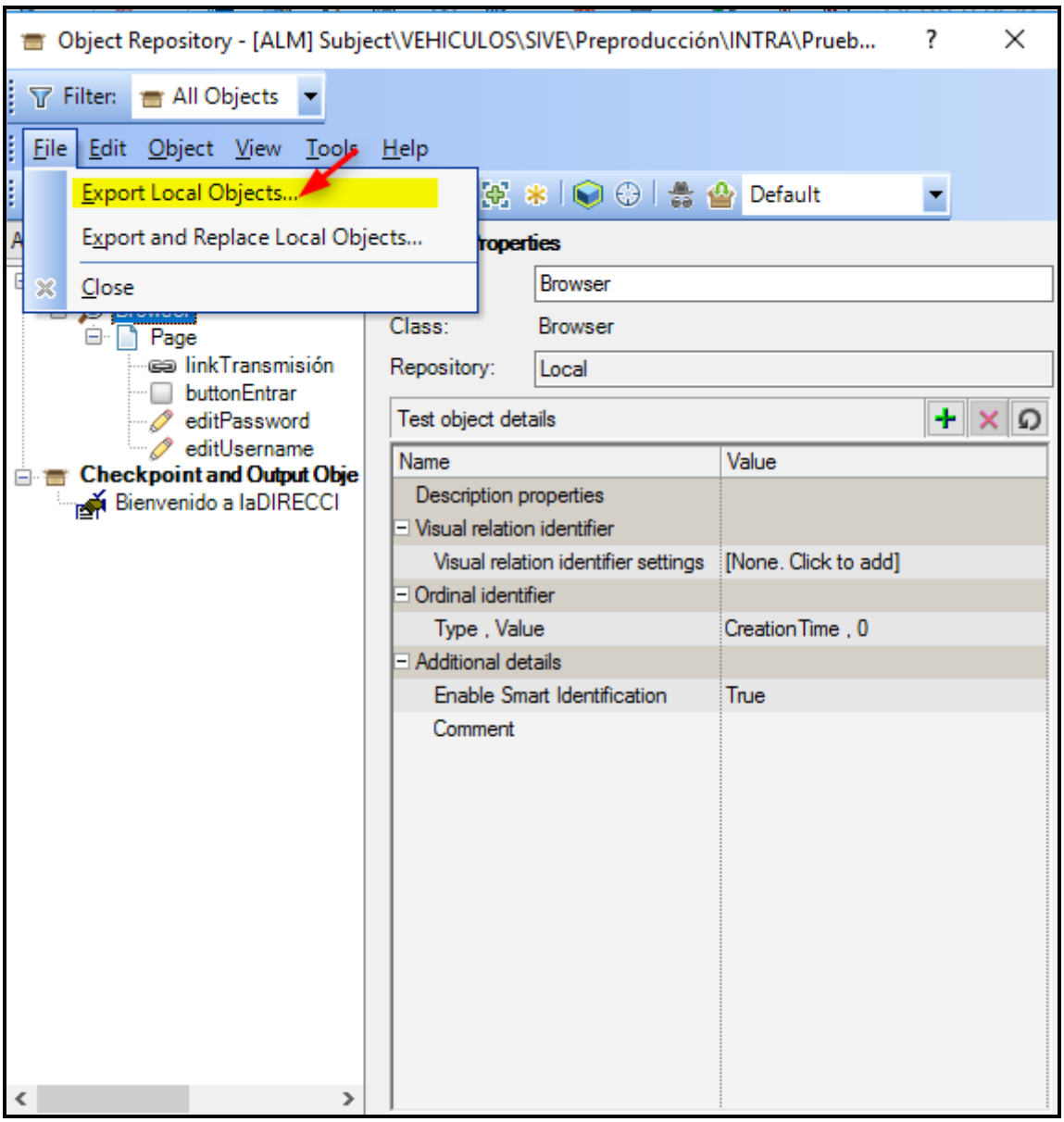
Una vez terminada y depurada una nueva Action, pasaremos los objetos nuevos del RO local de la Action al repositorio compartido, de forma que solo se deberán mantener los elementos contenidos en el apartado **Checkpoints and Output Objects** ya que serán exclusivos para cada Action. Esto es porque no es factible su reutilización ya que se han creado a medida de las necesidades específicas de esa pantalla en concreto.

Cualquier otro objeto de pruebas (**Test Object**) deberá gestionarse como se indica en la "Gestión del RO Compartido".

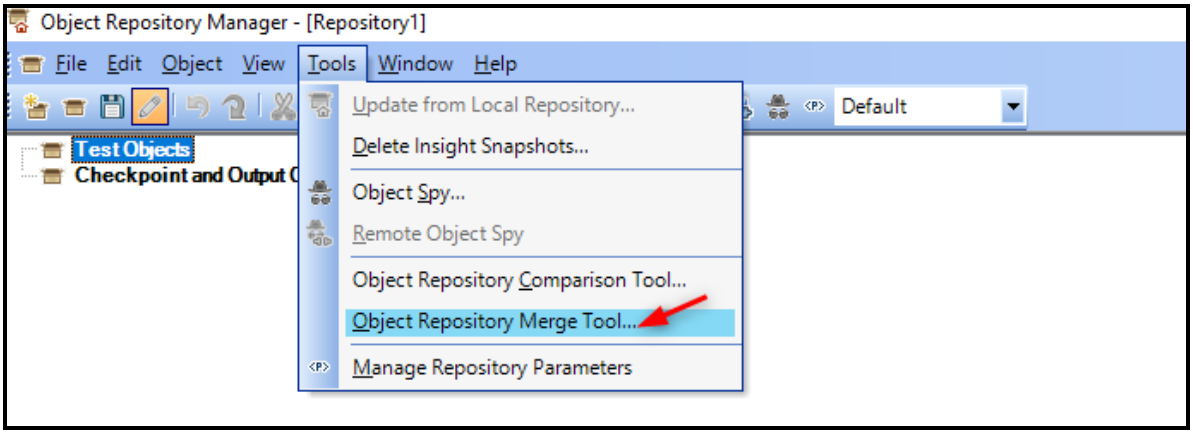
7.4.3 Gestión del RO Compartido

Una vez terminada y depurado una nueva action, para incorporar los objetos de RO local de la action al RO Compartido ubicado en ALM, deberemos realizar los siguientes pasos:

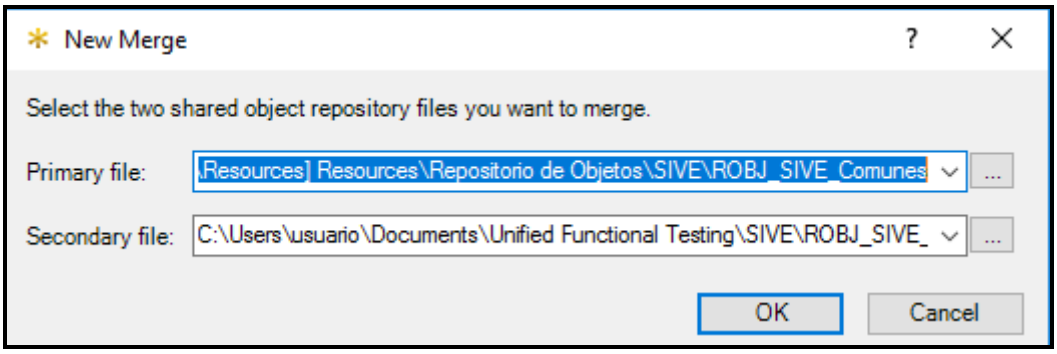
- Abrir RO Local con Object Repository
- Exportar el RO Local a una carpeta local (A MODO DE BACKUP). Opción: “Export Local Objects”. En este repositorio compartido, borrar los Checkpoints y los output Checkpoints.



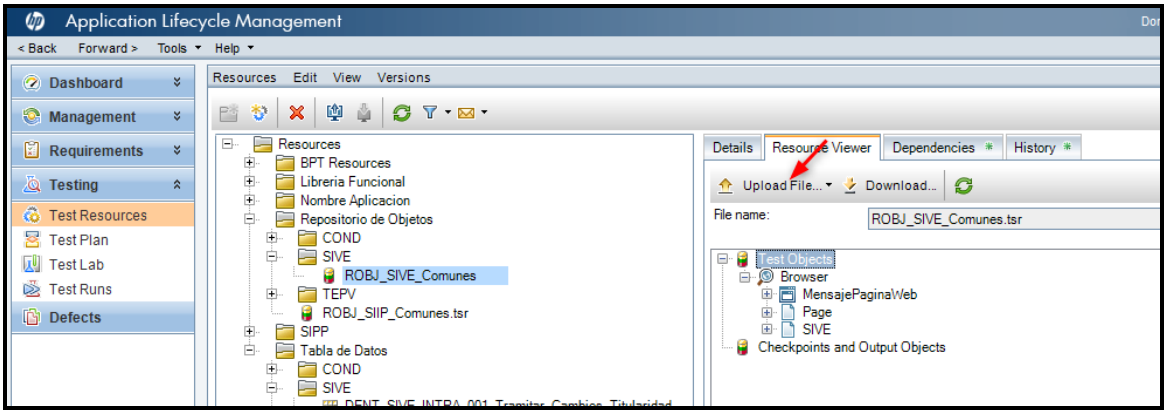
- Descargar el RO Compartido de ALM al file system del PC, crear una copia de seguridad y renombrarla añadiendo la fecha del día.
- Desde la ruta “Resources>Object Repository Manager Tools” seleccionar “Tools>Object Repository Merge Tool”.



- Seleccionar como Primary el RO Compartido de ALM que está en el PC. Seleccionar como Secondary el RO compartido exportado del RO local del script y pulsar "OK"



- Validar que no hay conflictos, hasta que no los haya NO SE GUARDA EL RO. En caso de conflictos deberá abortarse el merge y solucionarse en el script y su RO asociado
- Una vez solucionados los posibles conflictos, “Save” y reemplazar el RO compartido de ALM original (sin _copia) en el File System del PC
- Asociar el RO compartido obtenido al script y quitar referencias a otros (es posible que sea necesario refrescar la conexión con ALM para coger los cambios)
- En ALM > Test Resources, reemplazar el RO compartido existente por el nuevo del PC local (opción "Upload file")



- Asociar el RO Compartido de ALM al script

8 Reglas de Nomenclatura

8.1 Scripts Funcionales Automatizados

Son los Scripts de los casos de pruebas funcionales automatizados. Se graban a través de la Herramienta: **UFT**

La sintaxis es casi igual que la que se describe en el Plan de Pruebas Funcionales, añadiremos la "A" para indicar que es un caso automatizado:

PFA_<APLI>_<Nº CasoPrueba>_<BreveDescripcion>_<NºScript>

PFA_<Aplicación>_<999>_<Descripción>_<000>

Donde:

Aplicación	**Es el acrónimo que asigna la DGT a cada Aplicación**
999	**Nº del Caso de Prueba, es un índice secuencial. Comienza por 001**
Descripción	**Es la descripción de la Transacción de negocio**
000	**Es el nº de script del caso de prueba. Es un índice secuencial. Comienza por 001**

NOTA: Cuando se trabaje con subsistemas, el nombre Aplicación debe entenderse como aplicacion_subsistema y el contador debe reiniciarse por subsistema.

Para el caso de WebServices (API Tests) como norma general por cada operación de cada servicio deberá tener, al menos, dos Test Case, uno para las pruebas del caso positivo y otro para las pruebas del caso negativo.

8.1.1 Ejemplo

PFA_CCAL_001_Consulta_Vehiculos_001

PFA_SIVE_INTRA_001_Cambio_Titularidad_CTI_TD_001

8.2 Componentes Comunes Automatizados

Son los Scripts que corresponden a fragmentos de pruebas funcionales automatizados, porque contienen Actions reutilizables que se pueden llamar dentro de los casos de pruebas funcionales automatizados propiamente dichos. Se graban a través de la Herramienta: **UFT**

La sintaxis empieza por "CCA" de Componente Común Automatizado:

CCA_<APLI>_<BreveDescripcion>

CCA_<Aplicación>_<BreveDescripción>

Donde:

Aplicación **Es el acrónimo que asigna la DGT a cada Aplicación**

BreveDescripción **Es la descripción del componente común automatizado**

NOTA: Cuando se trabaje con subsistemas, el nombre Aplicación debe entenderse como aplicación_subsistema.

8.2.1 Ejemplo

CCA_ATEX_CON_Login

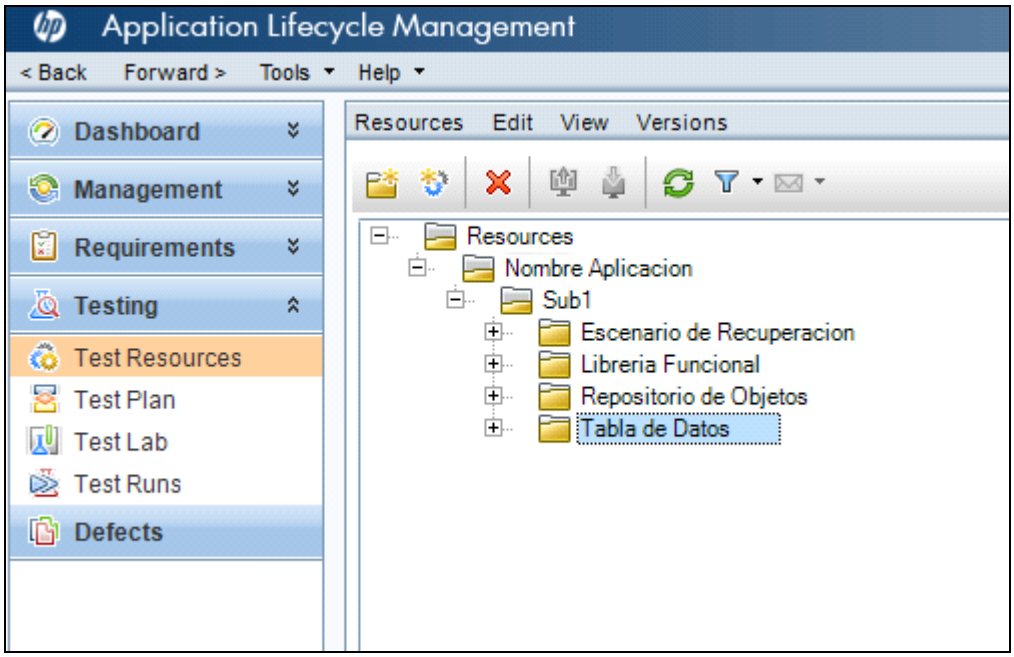
CCA_ATEX_CON_Logout

9 Recursos de los Tests

En el módulo Test Resources de ALM se deben adjuntar todos los recursos adicionales del script automático, como son:

- Tabla de Datos
- Librerías Funcionales
- Escenarios de Recuperación
- Repositorios de Objetos Compartidos

En este módulo se debe tener una estructura como la que se muestra en la imagen



Se debe crear la siguiente jerarquía de carpetas:

🗁 Carpeta con el nombre de la aplicación

🗁 Carpeta del Subsistema (en caso de tener susbsistemas la aplicación)

Y dentro de esta jerarquía incluir una carpeta para cada tipo de recurso:

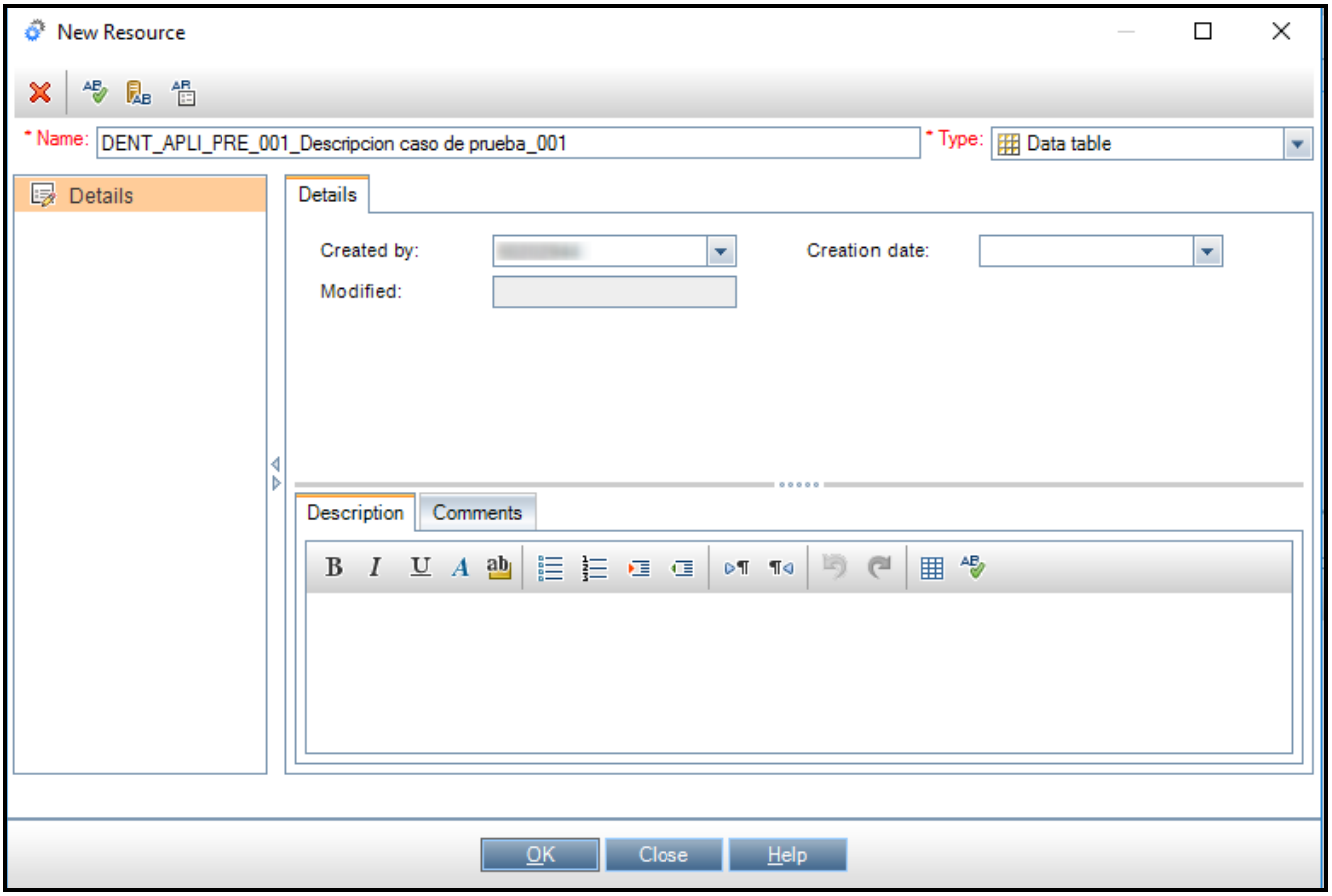
🗀 Escenario de Recuperación

🗀 Librería Funcional

🗀 Repositorio de Objetos

🗀 Tabla de Datos.

Para añadir un nuevo recurso se debe pulsar sobre el botón, que abre la siguiente ventana donde se debe especificar el nombre y el tipo de recurso. Adicionalmente se puede incluir una descripción y comentarios.



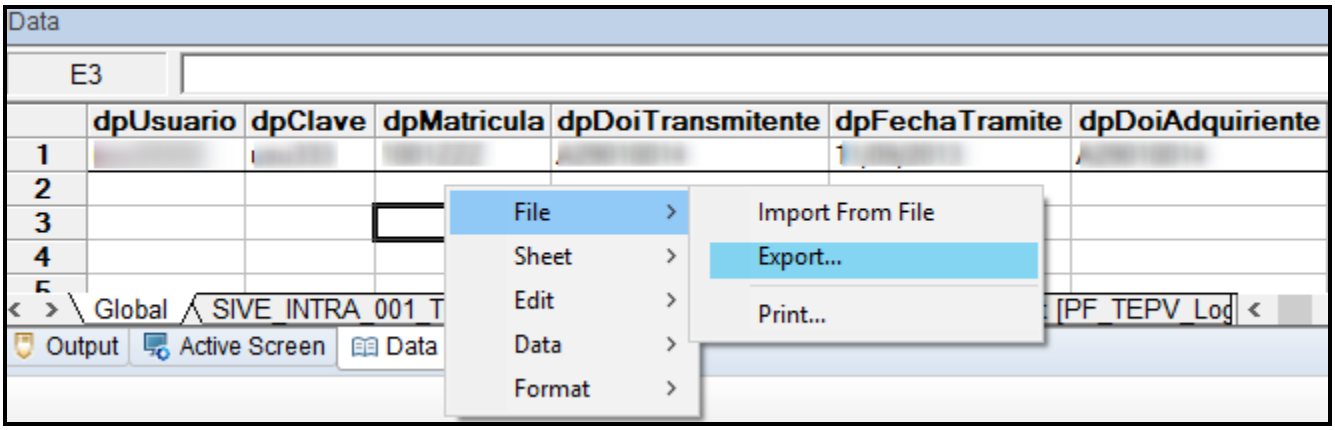
9.1 Tabla de Datos

En los casos de prueba automatizados el fichero de datos de entrada debe tener el formato adecuado para que pueda ser utilizado por el scritp UFT. De esta forma se evita tener duplicada información de los datos de entrada (en ALM y en el script UFT) y los datos de entrada pueden ser actualizados directamente desde ALM (en el módulo 'Test Resources') sin necesidad de tener que abrir el script UFT.

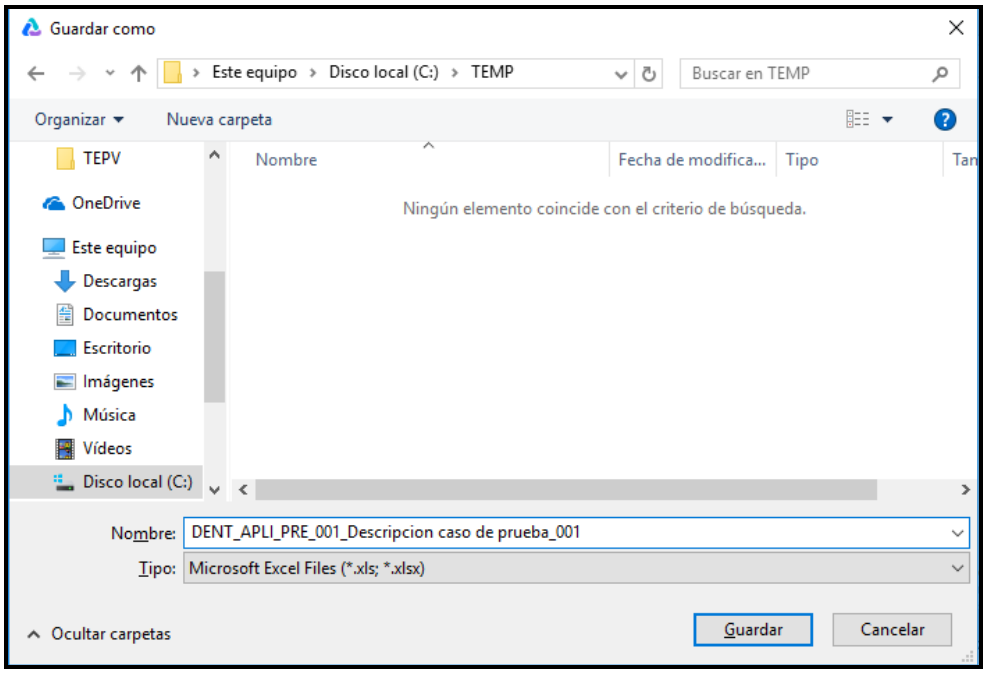
Los casos de prueba automatizados con UFT utilizan lo que se conoce como Data Table como mecanismo para parametrizar los datos de entrada. Estas datas tables tienen estructura de libro Excel que puede configurarse para cada caso de prueba.

Los pasos a seguir para establecer la data table a utilizar por un script son los siguientes:

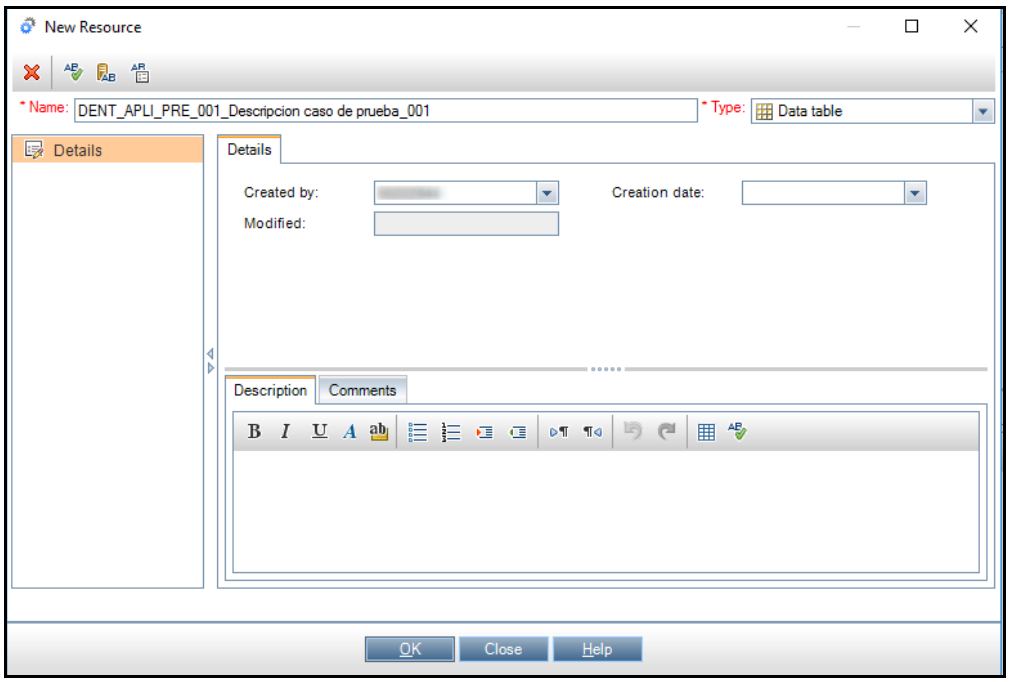
1. Exportar tabla del script a fichero Excel: Una vez que se ha finalizado el script con su propia data table por defecto (*default location*) se puede exportar esta data table al file system.
    a. Para exportar el fichero pulsar sobre el botón derecho y seleccionar File->Export



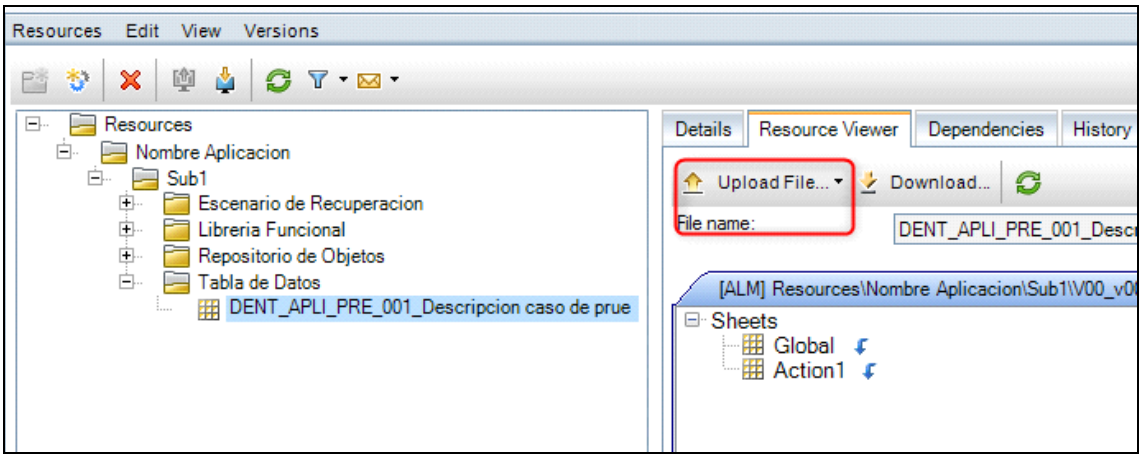
    b. Guardar en una capeta del file system:



2. Crear el recurso en ALM, para este fichero Excel:
   a. En el módulo "Test Resources", seleccionar la carpeta "Tabla de Datos" correspondiente, y seleccionar la opción de menú "New resource" o bien, pulsar el botón . En la ventana "New Resource", introducir el nombre del recurso, y seleccionar Type = "DataTable":

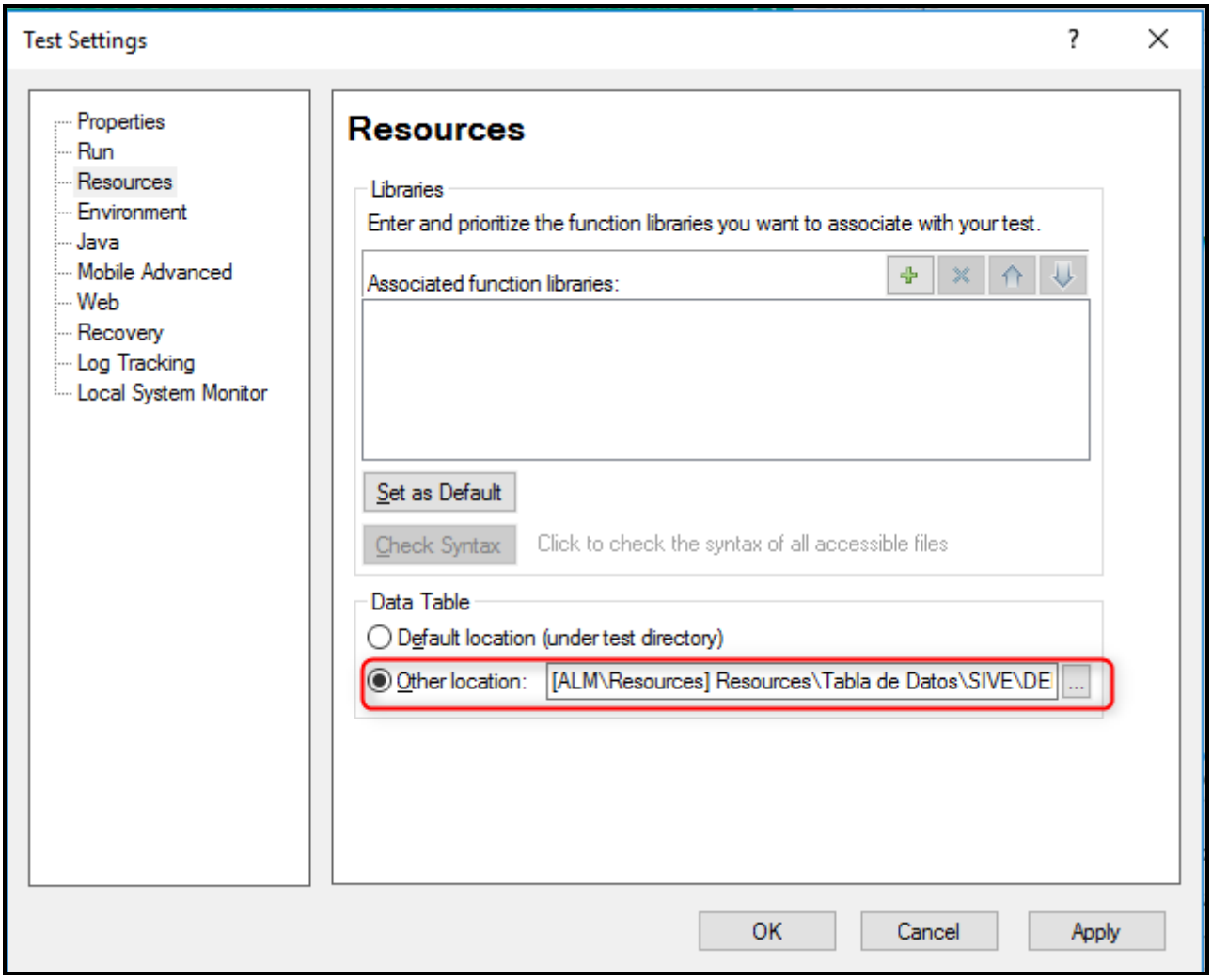


b. Actualizar el recurso correspondiente con la tabla exportada. Seleccionar el recurso a actualizar y pulsar sobre el botón "*Upload File*". Seleccionar la tabla exportada en fichero Excel previamente.



3. Asociar el Recurso de ALM al script de UFT:

a. En UFT, con el script abierto, acceder a través del menú *File->Settings* a la pantalla de *Test Settings*. Seleccionar la pestaña *Resources*. Seleccionar la opción "*Other location*" y pichar en el botón para seleccionar la Data Table creada en el módulo test resources de ALM.



A partir de ahora, los cambios que se hagan sobre los datos en la tabla del script en UFT se reflejarán en el fichero del recurso en ALM, y los cambios en el fichero Excel que se suban al recurso en ALM, se reflejarán en la tabla del script

9.1.1 Sintaxis

DENT_<APLI>_<ENTORNO>_<Nº CasoPrueba>_<BreveDescripcion>_<NºScript>

DENT_<Aplicación>_<ENTORNO>_<999>_<Descripción>_<000>

Donde:

Aplicación	**Es el acrónimo que asigna la DGT a cada Aplicación**
Entorno	**Es el entorno donde tiene validez los datos (PRE;DESA)**
999	**Nº del Caso de Prueba del script asociado**
Descripción	**Es la descripción del script asociado**
000	**Es el nº de script del caso de prueba.**

Ejemplo: DENT_ CCAL_PRE_001_ Consulta_Vehiculos_001

NOTA: Cuando se trabaje con subsistemas, el nombre Aplicación debe entenderse como Aplicación_subsistema.

9.2 Repositorios de Objetos Compartidos

Se deben incluir los Repositorios de Objetos Compartidos que se utilicen en los scripts dentro del módulo Test Resources de ALM.

9.2.1 Sintaxis

ROBJ_<APLI>_<Descripcion>

ROBJ_<Aplicación>_<Descripción>	
Donde:	
Aplicación	**Es el acrónimo que asigna la DGT a cada Aplicación**
Descripción	**Es la descripción del repositorio de objetos**

Ejemplo: ROBJ_CCAL_Comunes

NOTA: Cuando se trabaje con subsistemas, el nombre Aplicación debe entenderse como Aplicación_subsistema.

9.3 Escenarios de Recuperación

Se deben incluir los Escenarios de Recuperación que se utilicen en los scripts dentro del módulo Test Resources de ALM.

9.3.1 Sintaxis

RECU_<APLI>_<Descripcion>

RECU_<Aplicación>_<Descripción>	
Donde:	
Aplicación	**Es el acrónimo que asigna la DGT a cada Aplicación**
Descripción	**Es la descripción del escenario de recuperación**

Ejemplo: RECU_CCAL_Sistemas

NOTA: Cuando se trabaje con subsistemas, el nombre Aplicación debe entenderse como Aplicación_subsistema.

9.4 Librerías Funcionales

Se deben incluir las Librerías Funcionales que se utilicen en los scripts dentro del módulo Test Resources de ALM.

9.4.1 Sintaxis

LIBF_<APLI>_<Descripcion>

LIBF_<Aplicación>_<Descripción>	
Donde:	
Aplicación	**Es el acrónimo que asigna la DGT a cada Aplicación**
Descripción	**Es la descripción de la librería**

Ejemplo: LIBF_CCAL_Comunes

NOTA: Cuando se trabaje con subsistemas, el nombre Aplicación debe entenderse como aplicación_subsistema.

10 Normativas de Desarrollo a seguir

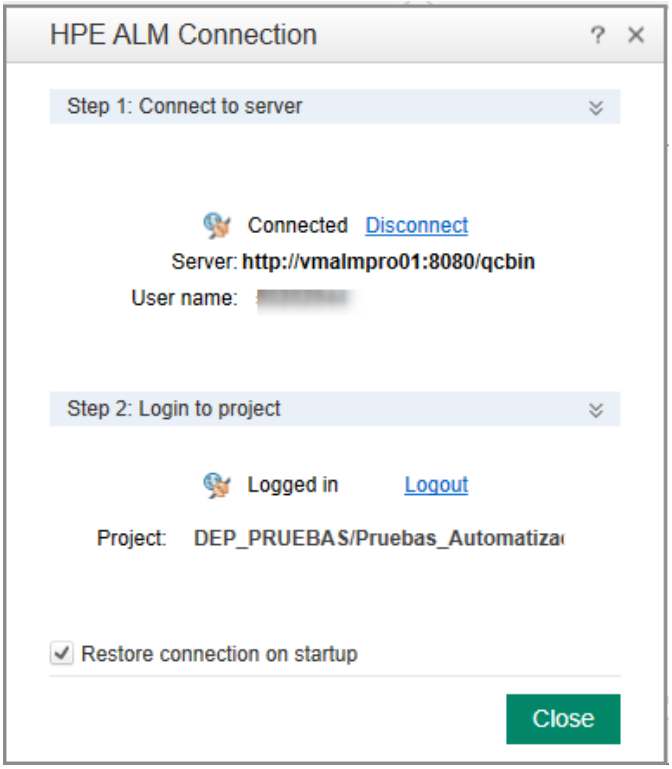
10.1 Caso de prueba GUI

10.1.1 Creación nuevo test o Componentes Comunes Automatizados

Como se ha explicado en el punto "6.1 Crear Scripts UFT en ALM" los tests se pueden crear directamente desde ALM o bien se pueden crear desde UFT y almacenarlos en ALM.

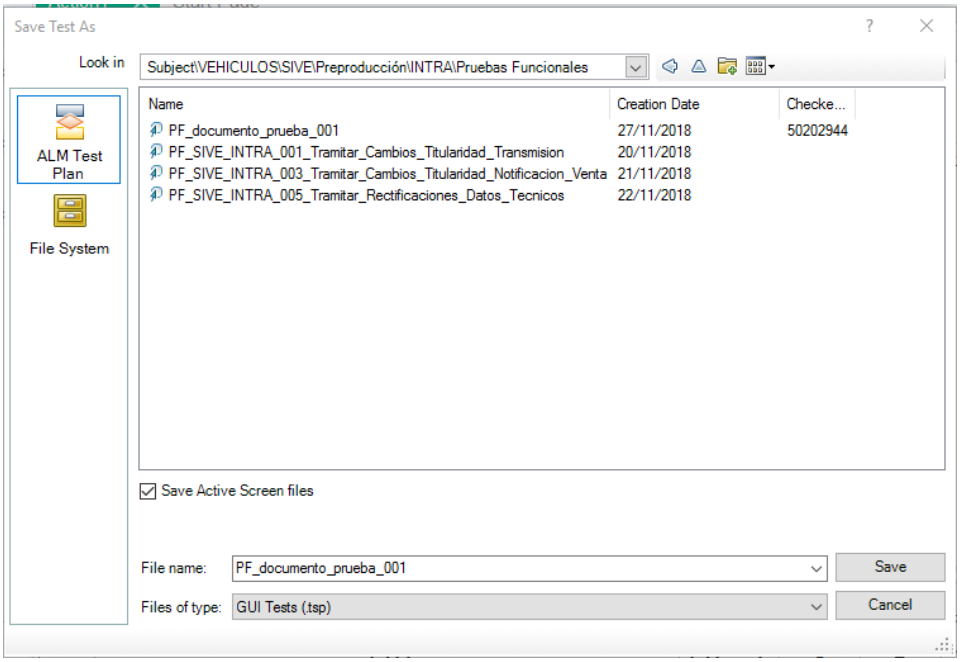
Para crearlos desde UFT se realizan los siguientes pasos:

- Abrir UFT.
- Conexión de UFT con ALM:
  - Seleccionar la opción de menú **ALM>ALM Connection**. Se abre la ventana **HPE ALM Connection**.
  - En el panel **Step 1: Connect to server** rellenar el campo **Server URL** con la dirección del servidor de ALM (http://vmalmpro01:8080/qcbin) y pulsar el botón **Connect**.
  - En el panel Step **2: Authenticate user information** rellenar los campos **User name** y **Password** con un usuario de acceso a ALM, y pulsar el botón **Authenticate**.
  - En el panel **Step 3: Login to project** seleccionar en los campos **Domain** y **Project** el dominio y proyecto en el cual se va a trabajar, y pulsar el botón **Login**.



- Pulsar el botón **Close**.

- Seleccionar la opción de menú **File>New>Test** o bien pulsar el botón **New Test**.
- Seleccionar la opción de menú **File>Save As**. Aparece la ventana **Save Test to ALM**.
- En el árbol del TestPlan se selecciona la carpeta sobre la cual se quiere guardar el caso de prueba.
- En el campo **Test Name** se introduce el nombre del caso de prueba y se pulsa el botón **Save**.

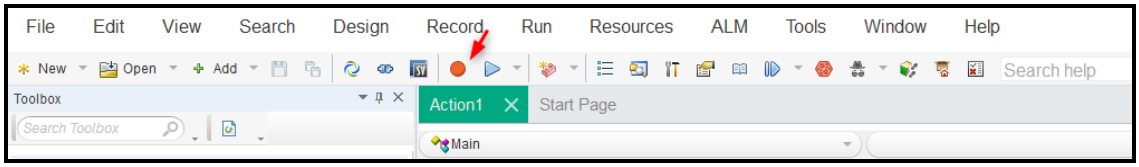


10.1.2 Grabación

Uno de los pasos principales en la creación del caso de prueba automático es la grabación de las acciones de usuario que el tester debe realizar contra la aplicación para completar el proceso que se desea grabar. Esto genera lo que se denomina un test básico.

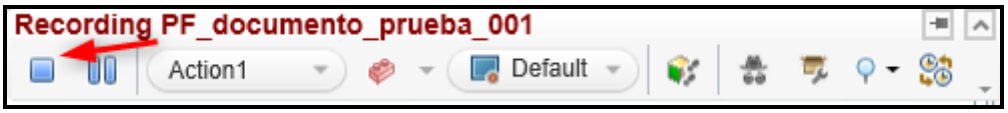
Los pasos necesarios para grabar son los siguientes:

- Posicionar la aplicación en el punto en la cual comienza el proceso que se quiere grabar.
- Posicionar el cursor en UFT en la posición del script en la cual se quiere insertar la grabación del proceso.
- En UFT pulsar el botón **Record**.



- Realizar sobre la aplicación el proceso que se quiere grabar. En este momento UFT estará grabando las acciones de usuario que se realizan sobre el interfaz del aplicativo.

- Una vez terminado el proceso, en UFT se pulsa el botón **Stop** para parar la grabación.



Es necesario ejecutar el script grabado para comprobar que la herramienta automática consigue reproducir las acciones de usuario sin problemas.

10.1.3 Scripting

Existen situaciones en las cuales la grabación del proceso de negocio no es suficiente para la creación del script automático y es necesario combinarla con la introducción de código en la pestaña **Expert View**. Por ejemplo, puede ser necesario introducir estamentos if, for, while, case etc. que permita el control en la ramificación del proceso de negocio.
El lenguaje utilizado por UFT es VBScript.

10.1.4 Sincronización

10.1.4.1 Definición

La **sincronización** permite que la aplicación espere antes de pasar al siguiente paso y así conseguir que el proceso se ejecute completamente.

Es necesaria para sincronizar la ejecución de un Script de UFT con la Aplicación, ya que puede que se intente acceder a un objeto que todavía no haya sido cargado porque tarde más tiempo.

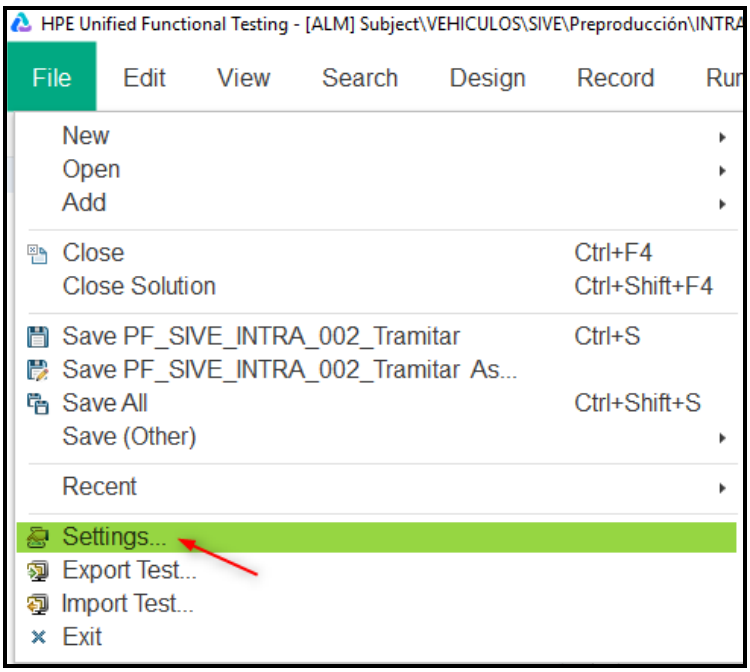
Existe una sincronización general para todos los objetos del test. Adicionalmente se pueden añadir puntos de sincronización para aumentar el tiempo máximo de espera en aquellos pasos de la aplicación que puedan ser necesarios.

10.1.4.2 Sincronización General

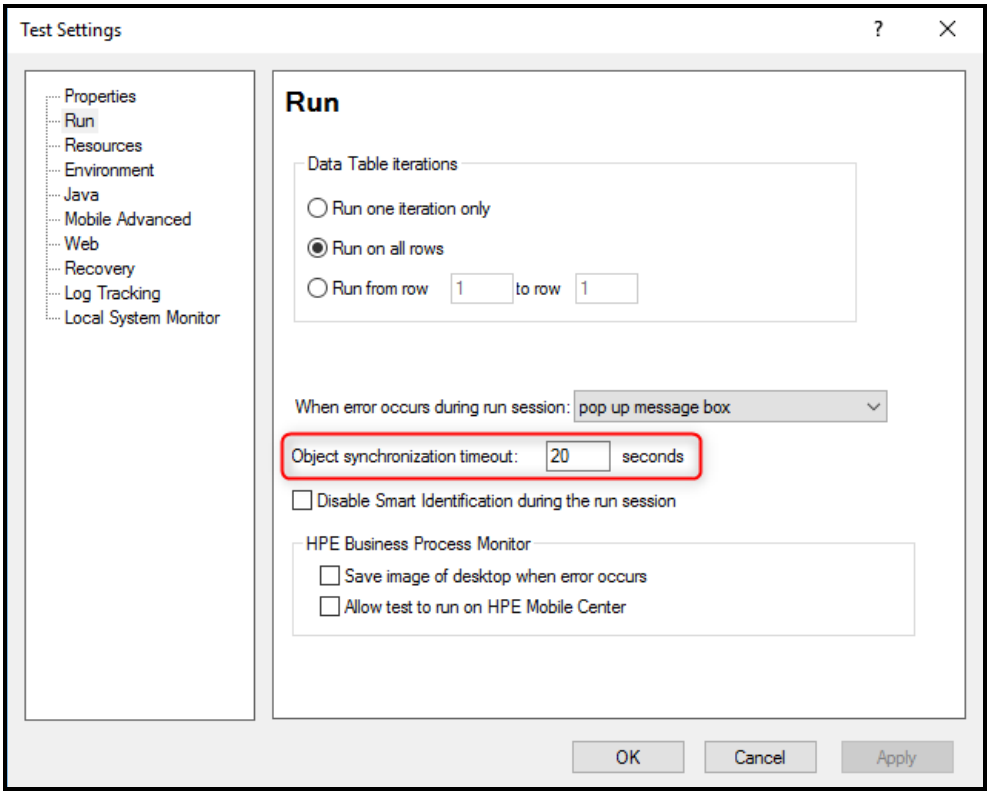
Existe una configuración general de sincronización para todos los objetos del test, por lo que con esta configuración puede ser suficiente para el buen funcionamiento del script.

La forma de acceder y modificar estos valores son:

1. Ir a menú y ejecutar "***File->Settings…***"



2. En la opción "***Run***" se dispone del parámetro "***Object synchronization timeout:***" con el número máximo de segundos de sincronización para todos los objetos. Se recomienda revisar el valor por defecto de 20 segundos a otro inferior alrededor de 5 segundos.



10.1.5 Checkpoints

10.1.5.1 Definición

Para comprobar que una aplicación se comporta como se espera, se deben utilizar puntos de chequeo en cada navegación que se realiza.

En la definición de los pasos del caso de prueba en ALM se deben haber especificado los resultados esperados de cada paso. Todos esos resultados esperados de cada paso se deben corresponder con un checkpoint u otro método de verificación (If, GetROProperty, etc.) dentro del script UFT.

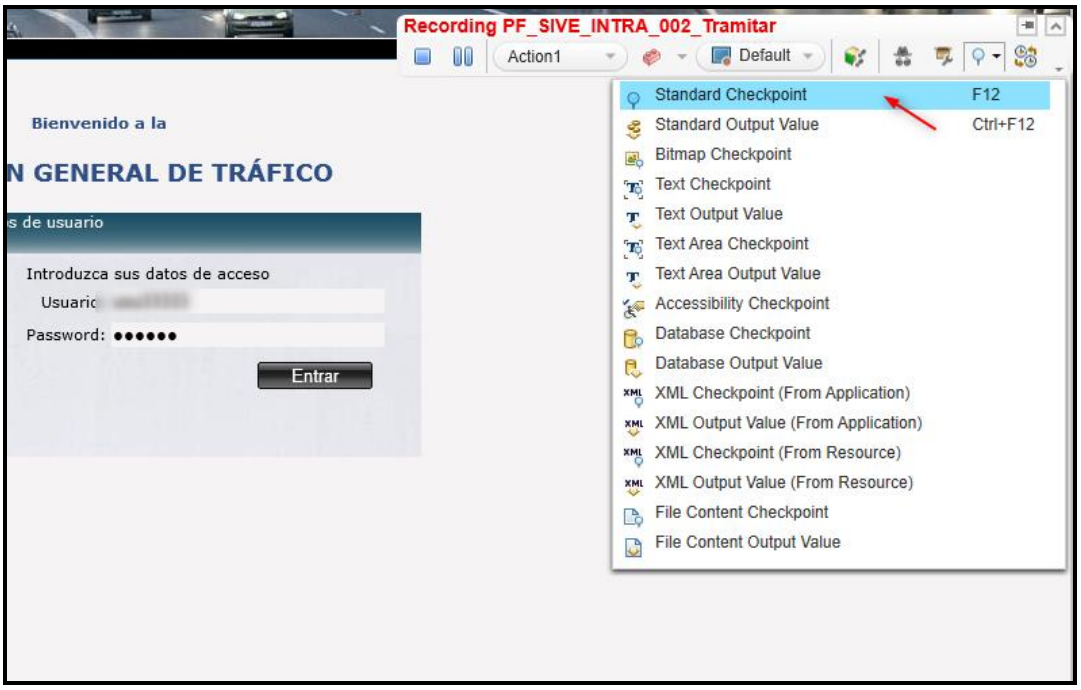
Los checkpoints se pueden insertar durante la grabación o después y pueden ser de diferentes tipos.

10.1.5.2 Insertar Checkpoints durante la grabación

Los pasos a realizar para introducir un punto de chequeo son diferentes dependiendo del tipo de chequeo: Standard Checkpoint, Text Checkpoint, Bitmap Checkpoint, Database Checkpoint,…

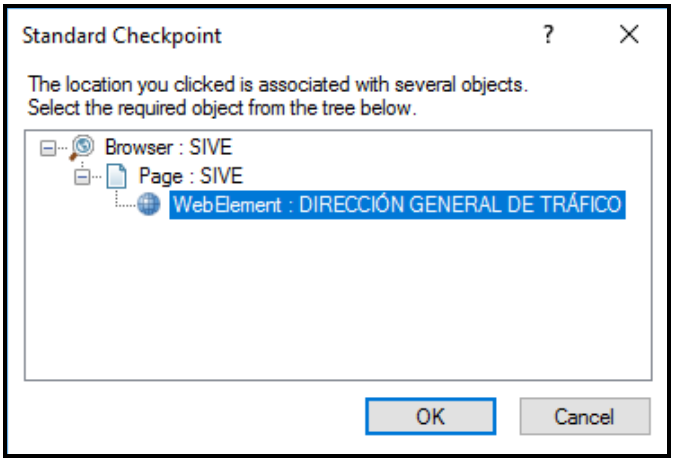
A continuación se definen los pasos a realizan para un chequeo de tipo "Standard Checkpoint":

1. En el momento de establecer el punto de chequeo ir a menú de grabación y ejecutar "***Insert->Checkpoint->Standard Checkpoint***"

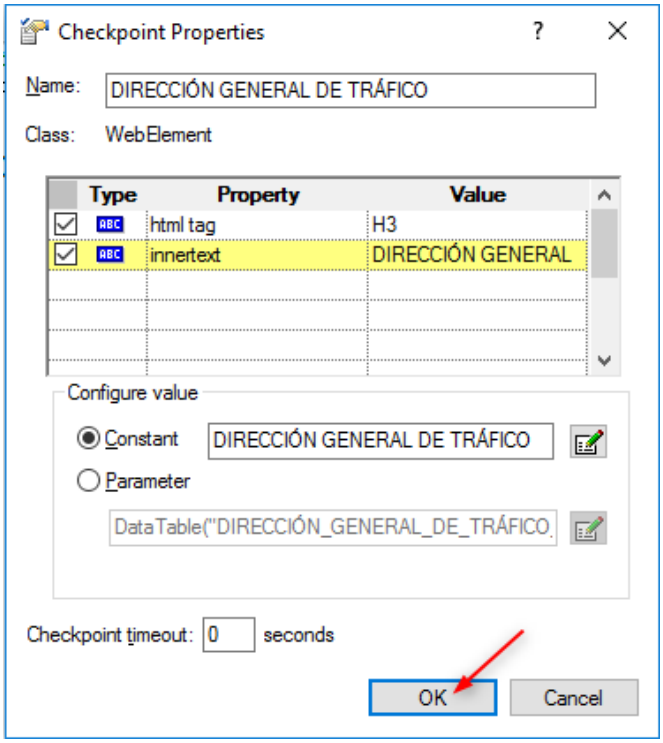


2. En caso de un chequeo de tipo "Standard Checkpoint" se debe seleccionar el objeto a chequear en la pantalla de la aplicación bajo pruebas.

3. Pulsar en el botón "***OK***"



4. Seleccionar las propiedades que se quieren chequear.



5. Pulsar en el botón "***OK***"

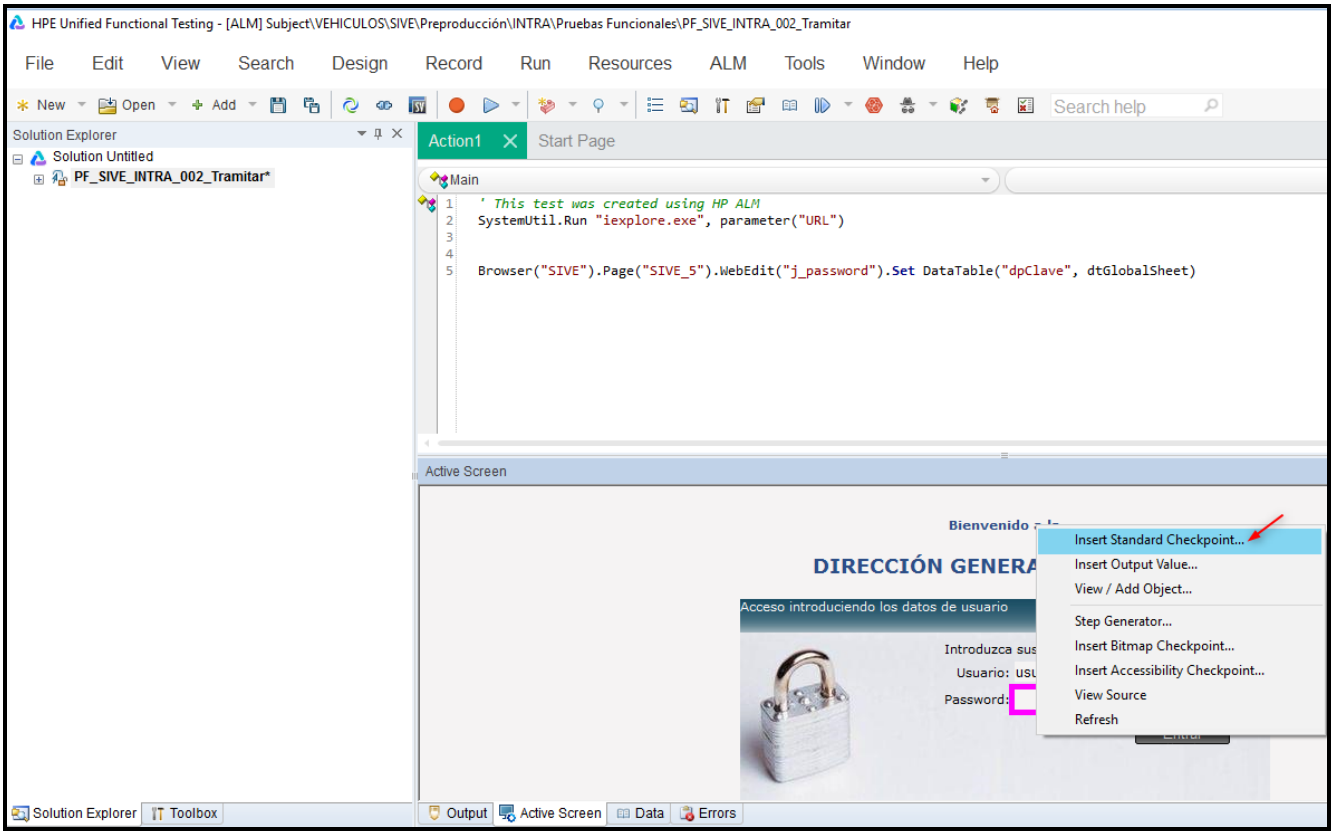
NOTA: En la ejecución, si el punto de chequeo falla el **Status** de ejecución de la prueba será "**Failed**"

10.1.5.3 Insertar Checkpoints después de la grabación

Los pasos a realizar para introducir un punto de chequeo son diferentes dependiendo del tipo de chequeos: Standard Checkpoint, Text Checkpoint, Bitmap Checkpoint, Database Checkpoint,…

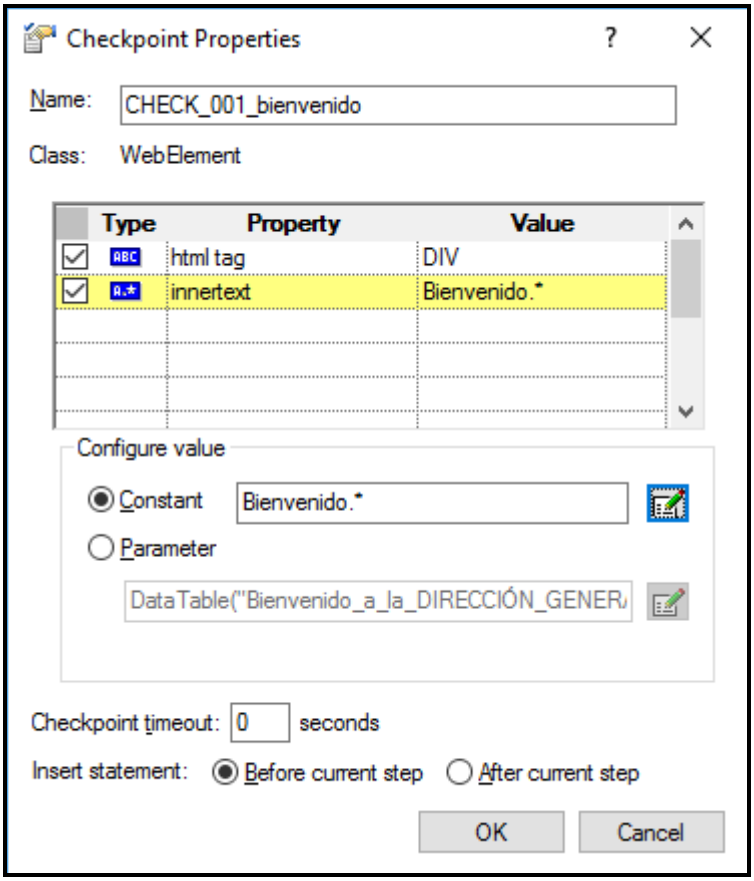
A continuación se definen los pasos a realizan para introducir un chequeo de tipo "*Text Checkpoint*" después de la grabación:

1. Ir a la ventana "***Active Screen***"
2. Seleccionar el objeto a chequear y pulsar el botón derecho.



3. Seleccionar el tipo de chequeo "***Insert Text Checkpoint***"

4. En la ventana de "***Text Checkpoint Properties***" Establecer las propiedades del punto de chequeo y pulsar en el botón "***OK***"



NOTA: En la ejecución, si el punto de chequeo falla el **Status** de ejecución de la prueba será "**Failed**"

10.1.5.4 Nomenclatura de los Checkpoints

Se debe seguir la siguiente sintaxis para la nomenclatura de los checkpoints:

CHECK_<Nº Chequeo Secuencial>_<BreveDescripcion>

En la siguiente imagen se muestra un ejemplo de chequeos dentro de un script UFT.

10.1.6 Parametrización

10.1.6.1 Definición

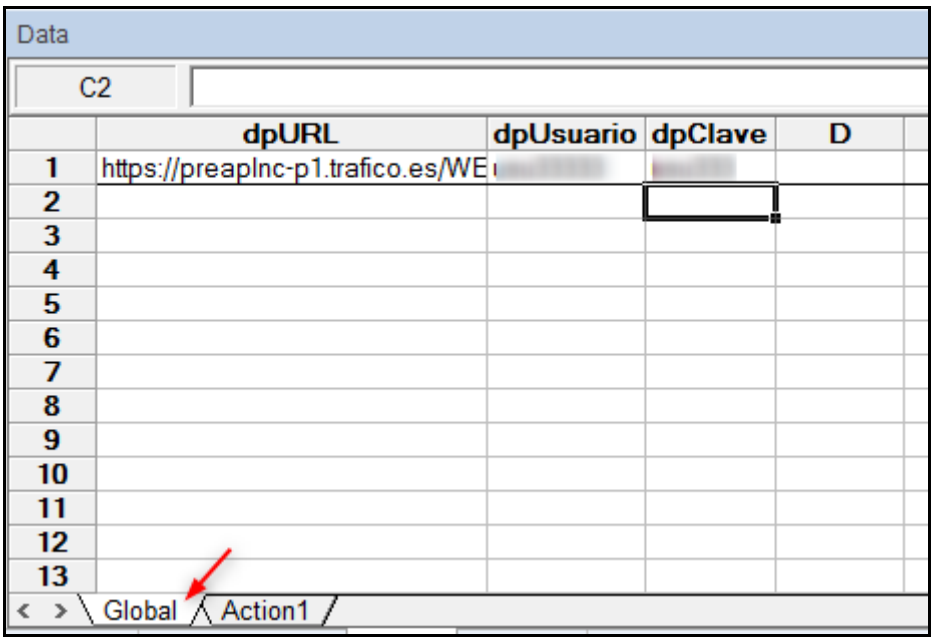
En la grabación de una transacción de negocio con la herramienta **UFT**, se capturan valores constantes. Por ejemplo, el login a una aplicación, en el que se necesita especificar *el usuario y el password* para poder acceder. Además, puede que se necesite que la transacción de negocio se ejecute con diferentes usuarios. Para lo cual es necesario parametrizar el script reemplazando dichas constantes por parámetros de **UFT.**
En general, se debe parametrizar todo contenido susceptible de cambio.

Los tipos de parámetros a utilizar son:

- Entrada
- Salida
- Random
- Environment Parameter

Por defecto, existe un fichero de datos de entrada del tipo XLS que se ubica en la estructura de directorios del script. Dicho fichero contiene 2 hojas.

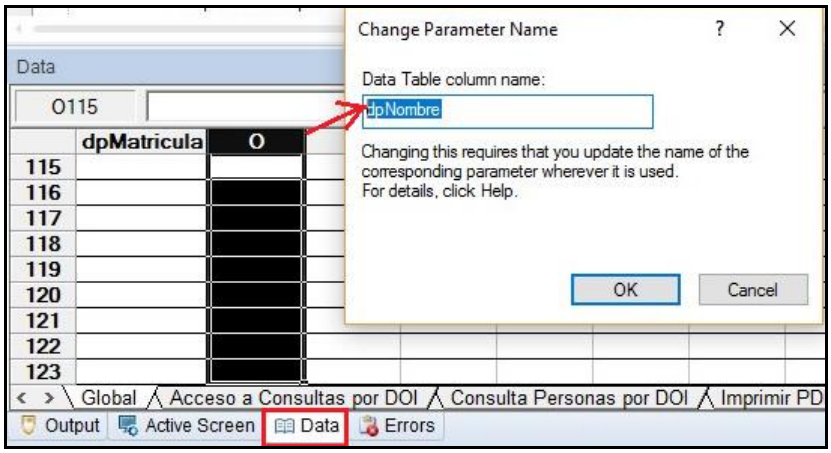
- "**Global**" que es accesible desde todos los Actions del Script.
- "**Action**" que es una hoja por cada Action definido.



10.1.6.2 Insertar Parámetros de Entrada

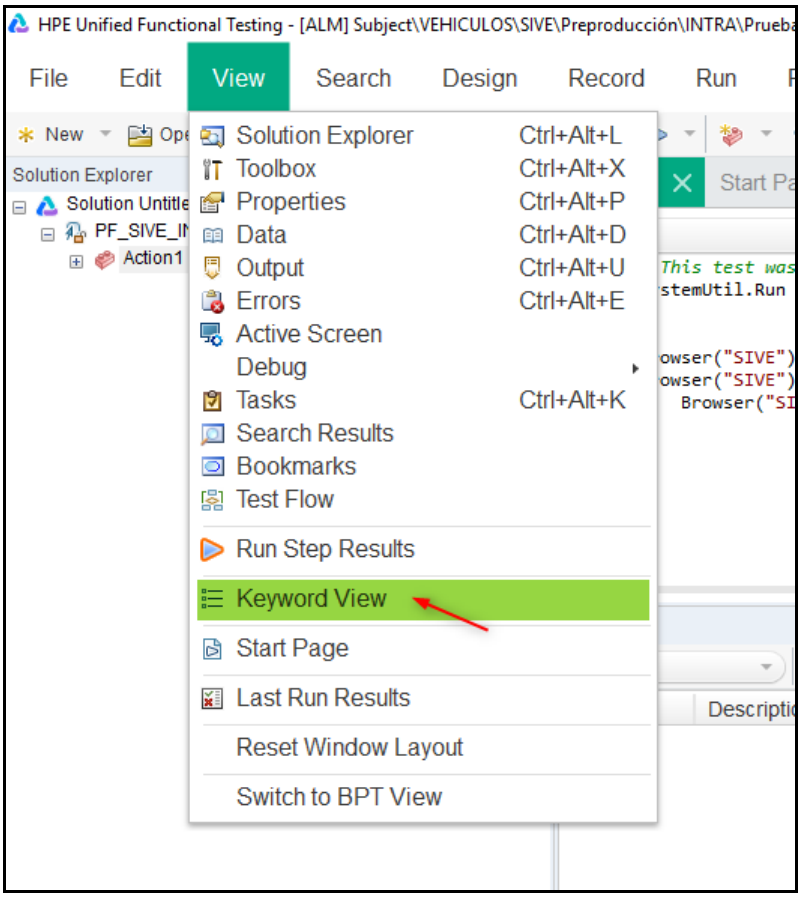
Dos de las formas de insertar parámetros de entrada en la Data Table del script son:

A. Directamente en la Data Table del script: Hacer doble clic sobre la cabecera de una columna de la tabla e introducir el nombre del nuevo parámetro

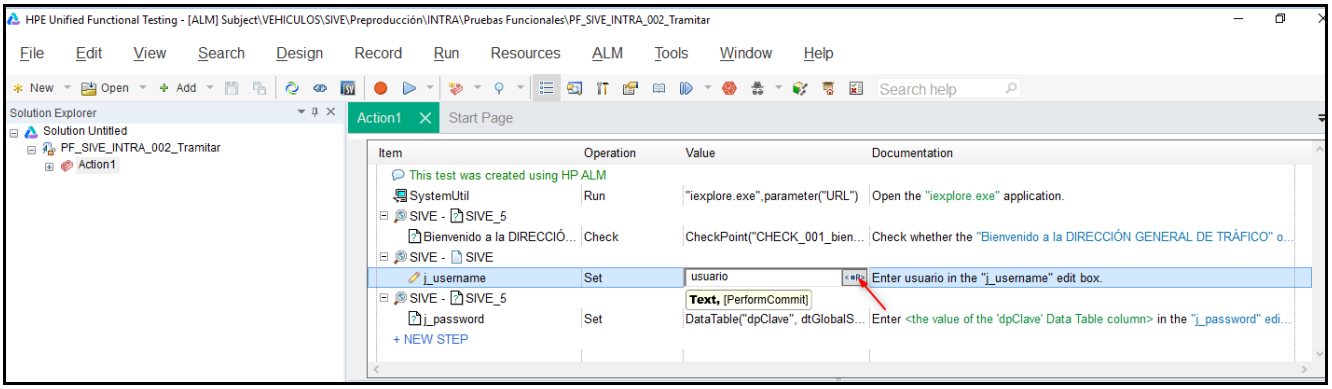


B. A través de la vista "*Keyword view*": Los pasos a realizar para crear un parámetro de entrada a través de "*Keyword view*" son:

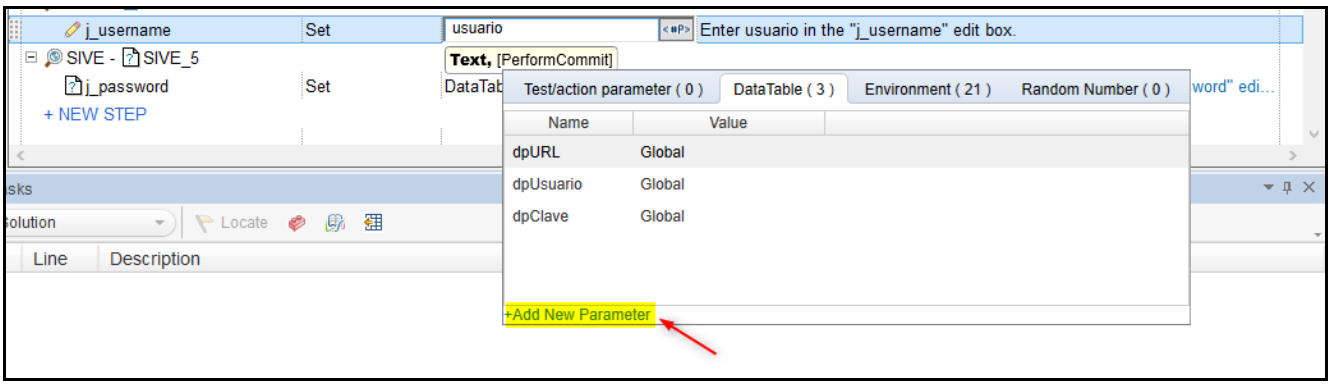
1. Ir a la pestaña "*Keyword view*" (menú View→Keyword View)

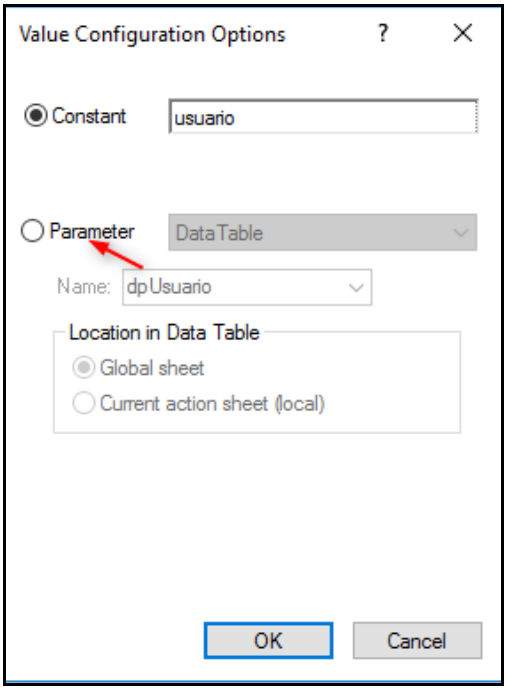


2. Seleccionar el paso que contiene la constante a parametrizar y pulsar en el botón "***Configure the Value***".



3. Seleccionar la opción "***+ Add New Parameter***" → "***Parameter***"



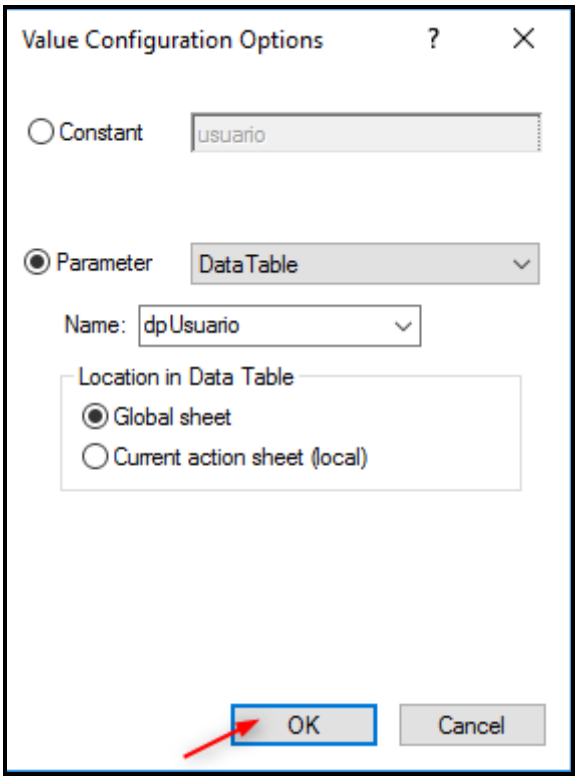


4. Introducir en el campo "***Name***" el nombre del parámetro según la nomenclatura dada en el documento "**Anexo 03.01 Guía Datos de Entrada Pruebas**"

5. Seleccionar su localización en "Location in Data Table":

- "**Global**" Hace referencia a todas las Action del Script.
- "**Action1**" Es una hoja por cada Action definido.

6. Pulsar en el botón "***OK***"



10.1.6.3 Insertar Parámetros de Salida durante la grabación

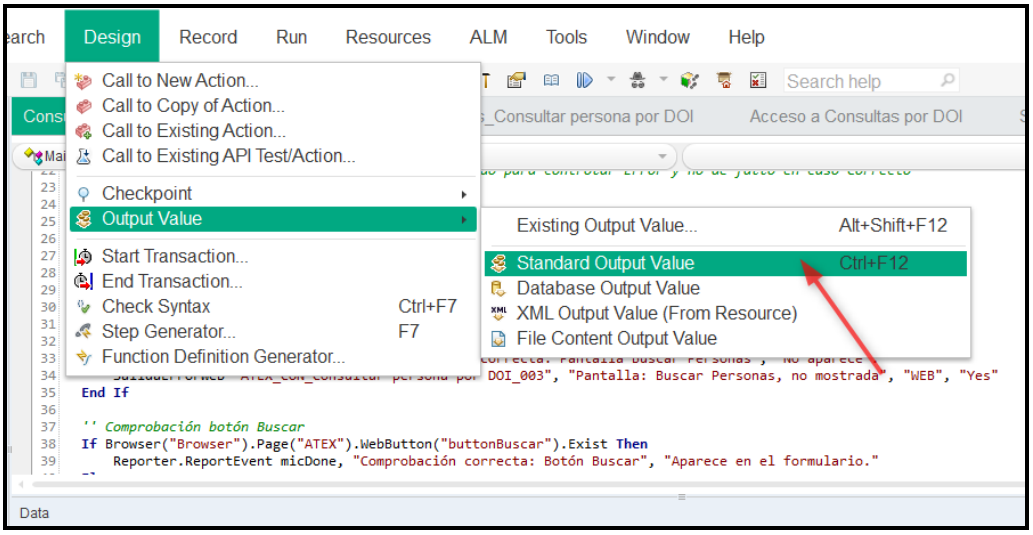
Los parámetros de Salida ("***Output values***") se usan cuando se quieren recoger los valores de salida de un script.

Los valores de los parámetros de salida se pueden utilizar como valores de entrada de otros parámetros.

Existen diferentes tipos de "***Output values***" y dependiendo del tipo los pasos a seguir son diferentes.

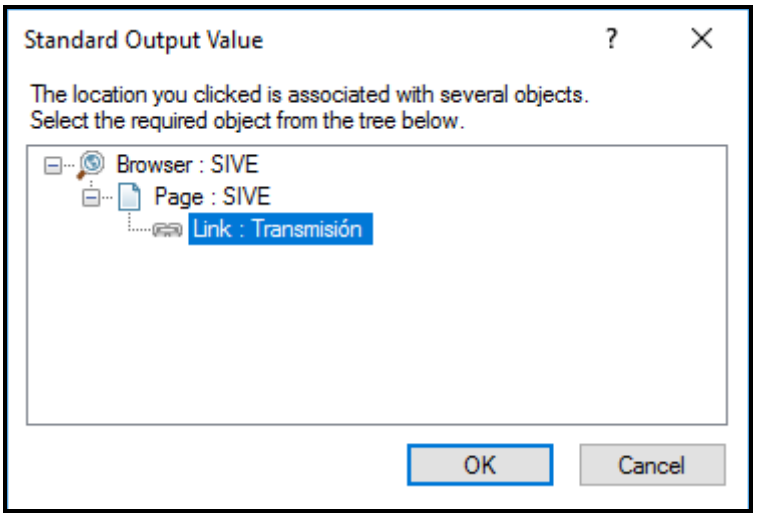
A continuación se explican los pasos a realizar para introducir un *output value* de tipo *Standard* durante la grabación:

1. Seleccionar del menu "Insert->Output Value->Standard Output Value …"

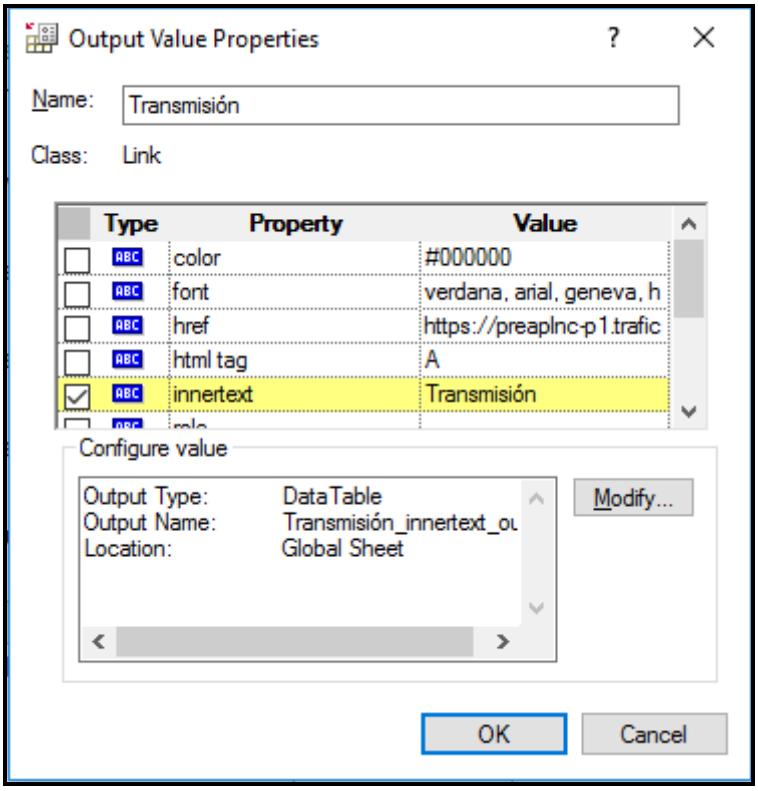


2. Pinchar sobre el objeto de la aplicación bajo pruebas del que se quiere obtener el valor de la propiedad a parametrizar como parámetro de salida.

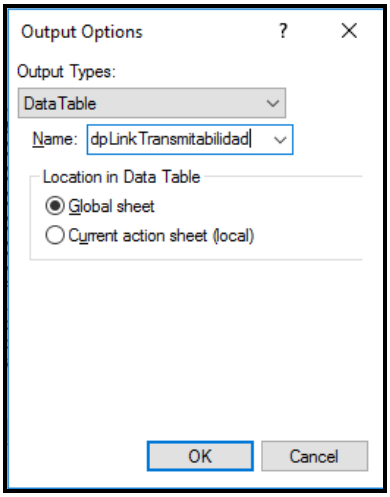
3. Aparece la ventana "***Object Selection***" con el objeto seleccionado. Pulsar "***OK***".



4. Seleccionar la propiedad del objeto cuyo valor se quiere obtener como parámetro de salida. Pulsar el botón "***Modify***"



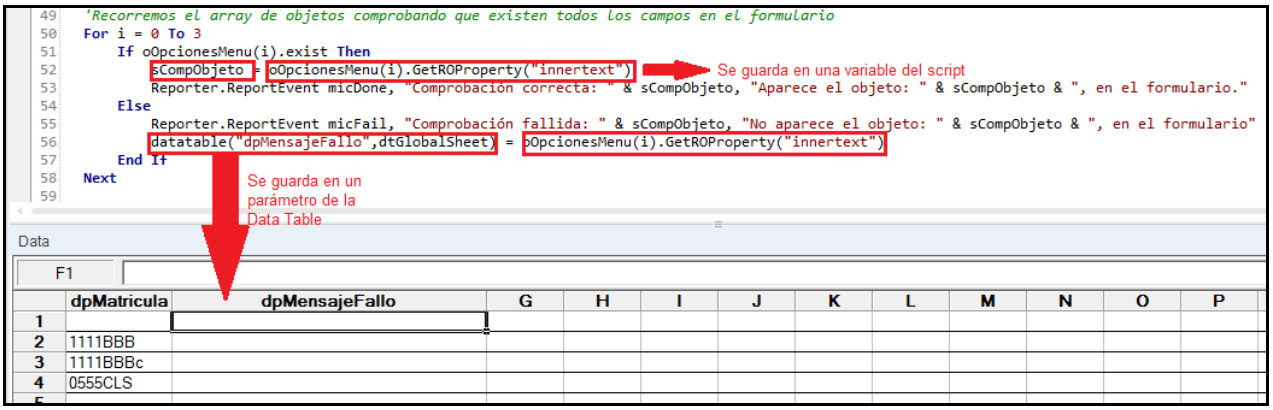
5. En la ventana "***Output Options***" dar un nombre al parámetro y seleccionar la hoja de la tabla de datos donde se quiere localizar. Pulsar "***OK***".



10.1.6.4 Insertar Parámetros de Salida después de la grabación

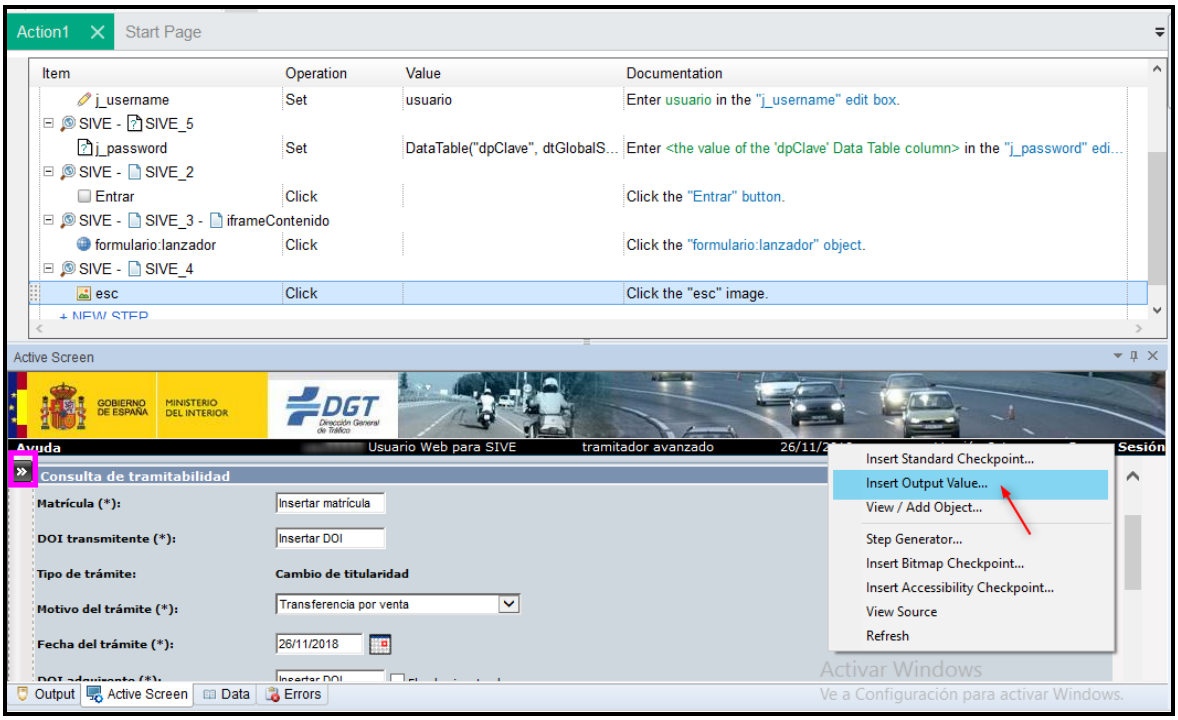
Dos de las formas de capturar valores en parámetros de salida en la Data Table del script son:

A. Directamente en el código del script con el método *GetROProperty*, e indicando la propiedad del objeto cuyo valor queremos capturar, y el nombre de la variable del script o del parámetro de la Data Table donde lo queremos guardar:

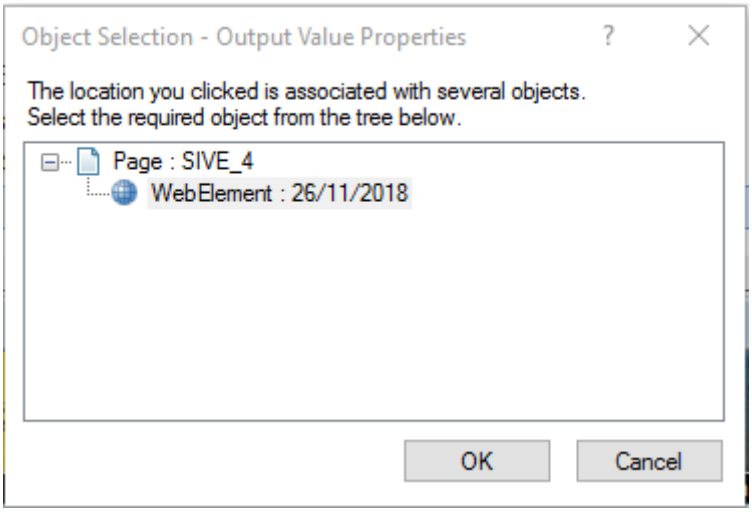


B. A través del panel "*Active Screen*", los pasos son:

1. Ir a la ventana "***Active Screen***"
2. Seleccionar el objeto que contiene el dato que queremos parametrizar.
3. Pulsar en el botón derecho y seleccionar "***Insert Output Value***"

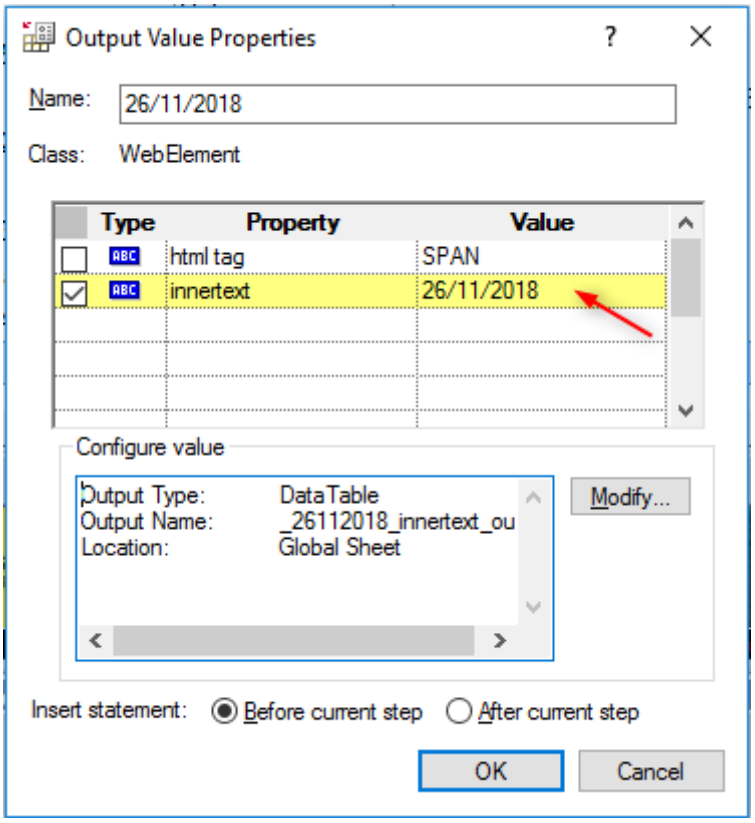


4. Seleccionar el objeto en la ventana "***Object Selection – Output Value Properties***".



5. Pulsar en el botón "***OK***"

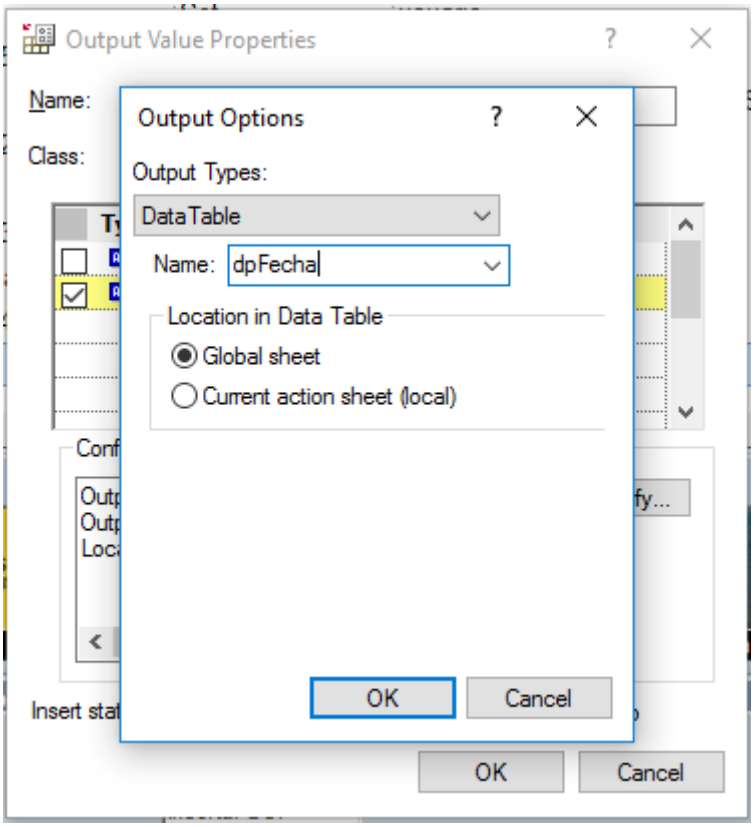
6. Seleccionar la propiedad del objeto que contiene el dato a parametrizar.



7. Pulsar en el botón "***Modify***" para editar las opciones.

8. Introducir en el campo "Name" el nombre del parámetro según la nomenclatura dada en el documento "**Guía Datos de Entrada Pruebas.doc**"

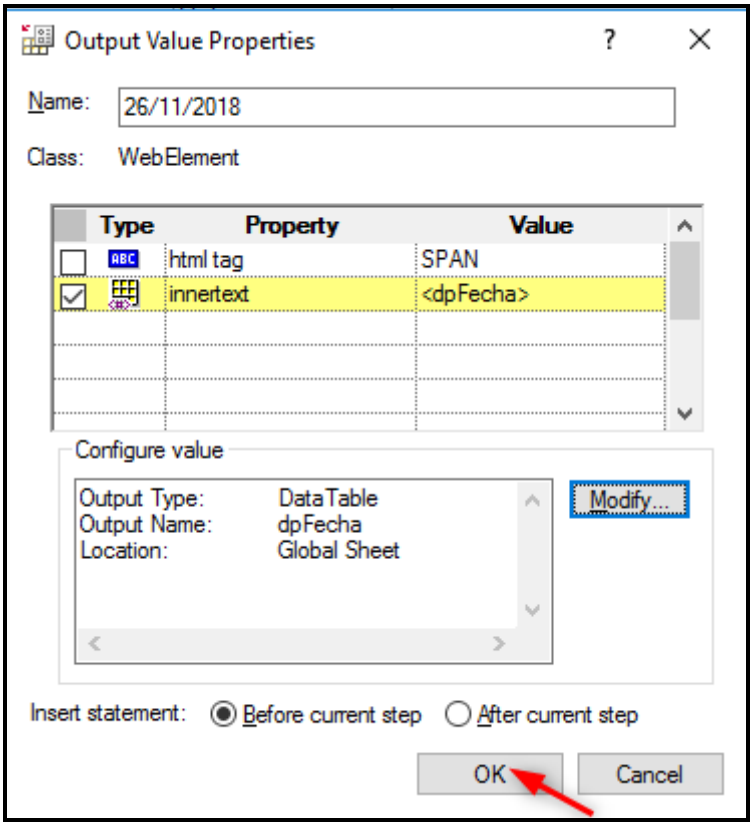
Por ejemplo: dp_FechaMatriculacion



9. Seleccionar la localización:

- "**Global**" Hace referencia a todas las Action del Script.
- "**Action1**" Es una hoja por cada Action definido.

10. Pulsar en el botón "***OK***"

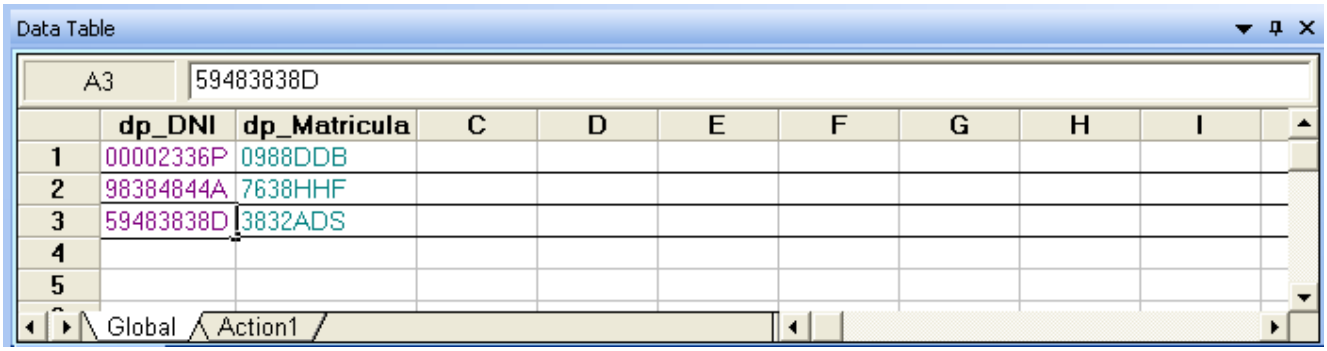


11. Pulsar en el botón "***OK***"

10.1.6.5 Número de Iteraciones

Cada una de las filas de la hoja de datos Global del script se corresponde con una iteración del script.

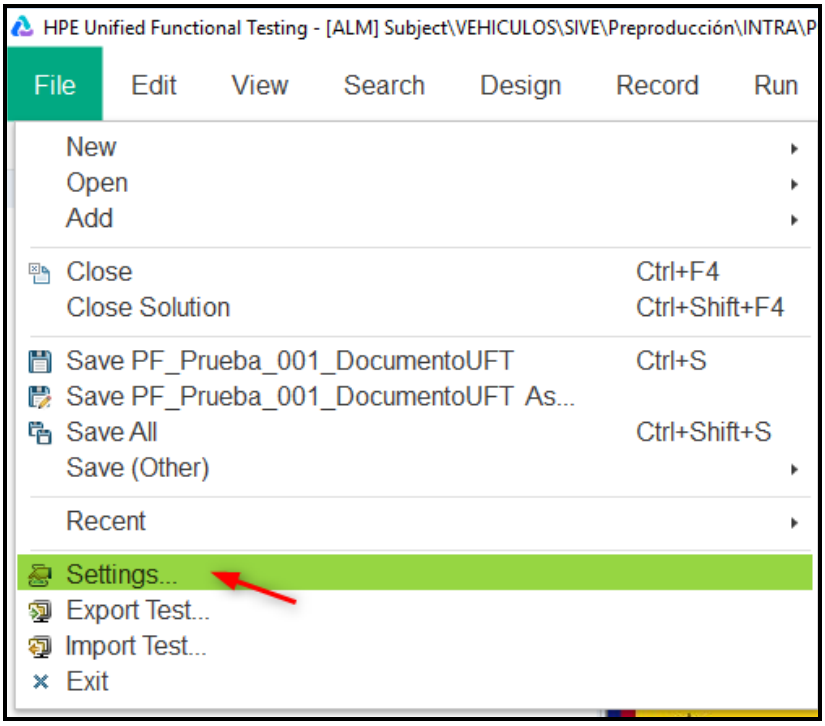
Así por ejemplo, la siguiente hoja de datos global contiene dos parámetros dpDNI y dpMatricula con 3 iteraciones.



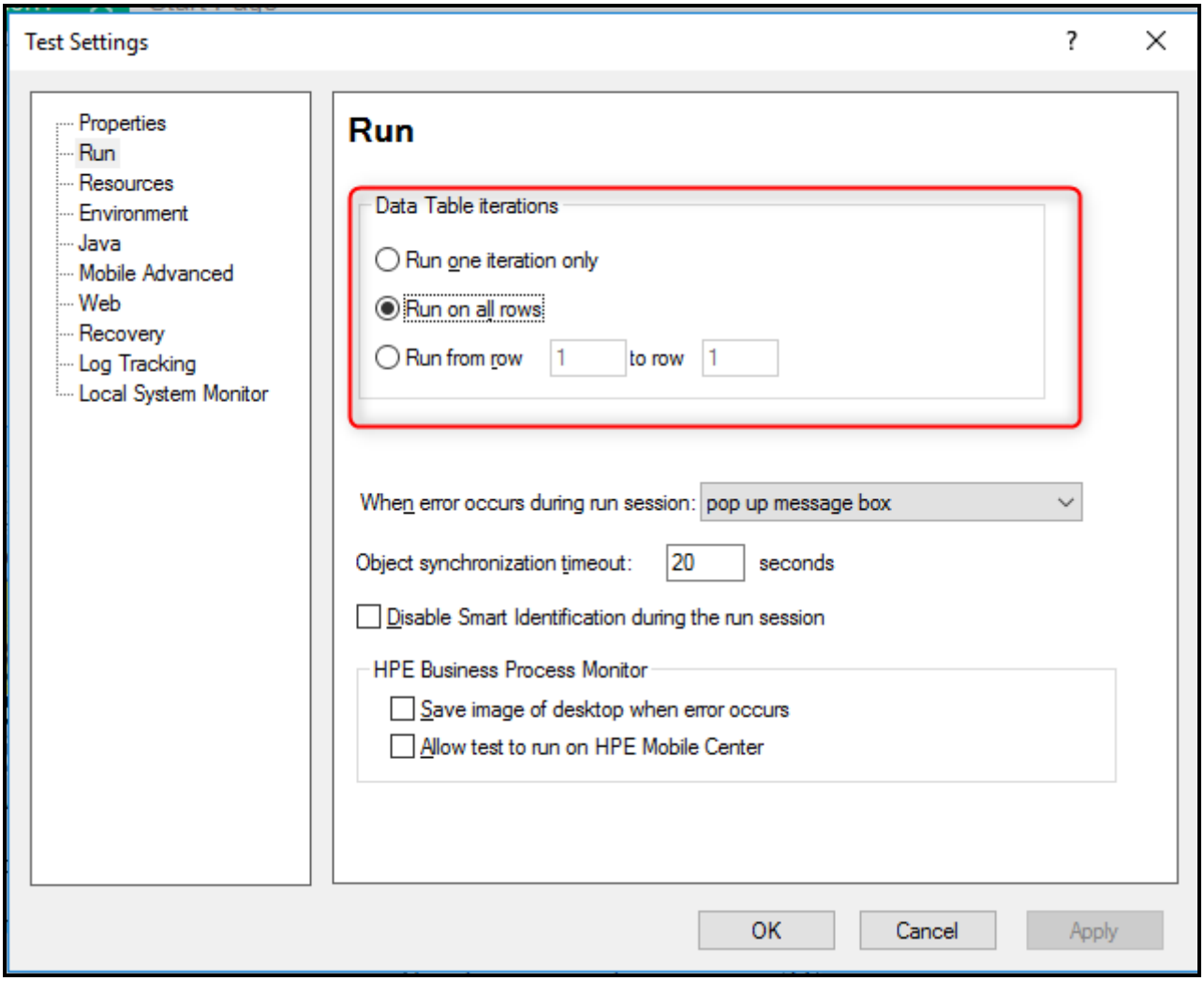
Se puede establecer el número de filas de la hoja de datos a utilizar durante la ejecución del script. La configuración por defecto de la herramienta ejecuta todas las iteraciones.

Para especificar el número de iteraciones a ejecutar de un script se deben realizar las siguientes acciones:

1. Ir a Menú y ejecutar "***File->Settings***…"



2. Pulsar sobre "***Run***"



3. Seleccionar una de las opciones posibles entre:

- Run one iteration only
- Run on all rows
- Run from row to row

4. Pulsar en el botón "***OK***"

Es fundamental para la correcta ejecución de las iteraciones que el **punto de inicio y fin del caso de prueba sea el mismo**. Es decir, cuando termina la ejecución de una iteración y para poder continuar con la siguiente la aplicación debe encontrarse en la situación inicial.
Normalmente, se establece este punto de inicio y fin común en la ventana principal de la aplicación. En el caso de aplicaciones que requieren login en la ventana inmediatamente después del login.

10.1.7 Reusabilidad de Código

La **Reusabilidad** de código consiste en procedimientos generales que se pueden guardar y reutilizar a posteriori.

Para ello se deberán estructurar las transacciones de negocio a grabar en acciones.

Por ejemplo, una transacción de negocio podríamos estructurarla en

Transacción de negocio	
Action1:	**Logado a la aplicación**
Action 2:	**Transacción de Negocio**
Action 3:	**Deslogado de la aplicación**

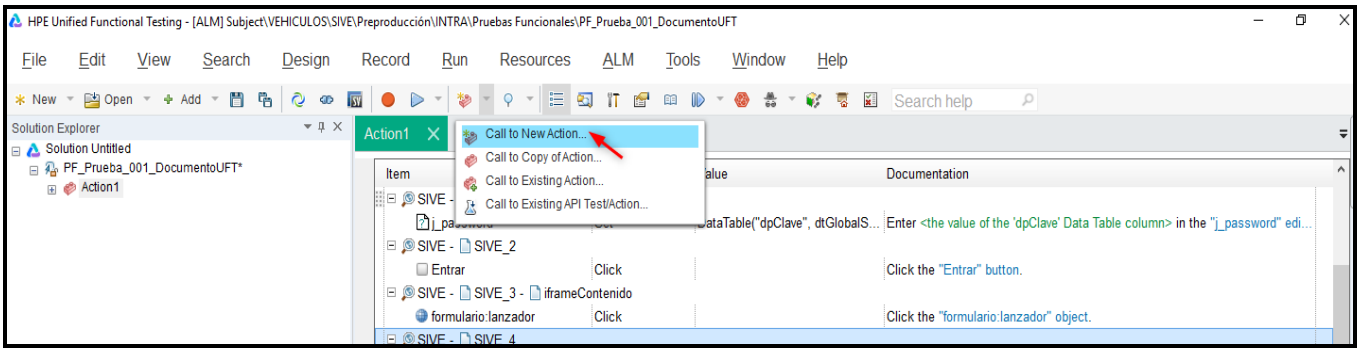
Se deben dividir en tantas Action como subsecuencias lógicas.

10.1.7.1 Crear un Action

Para la creación de un nuevo Action los pasos a realizar son:

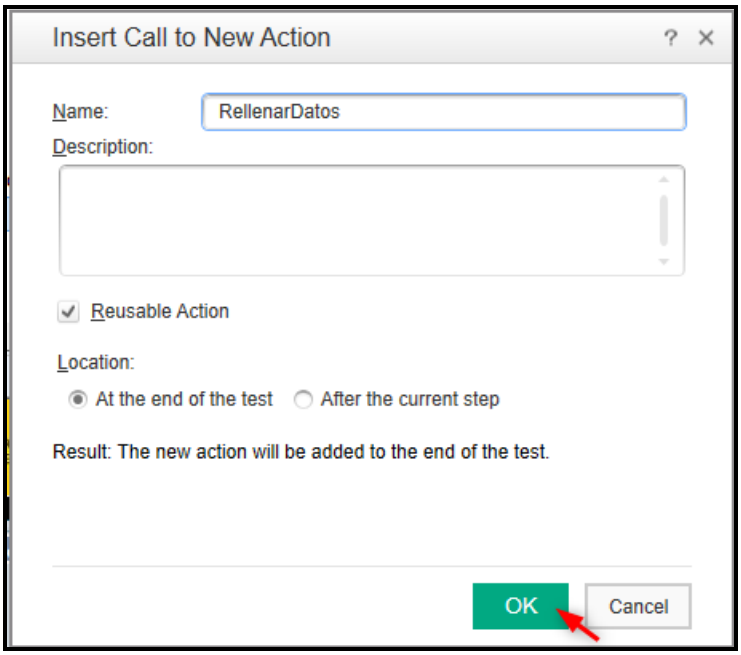
1. Ir a la pestaña "***Keyword view***"

2. Seleccionar el punto donde se quiere definir un nuevo Action, pulsar en la barra de herramientas en el botón y seleccionar la opción "***Call to New Action…***"



3. Introducir en el campo "***Name***" el nombre del Action.

4. Pulsar el botón "**OK**"



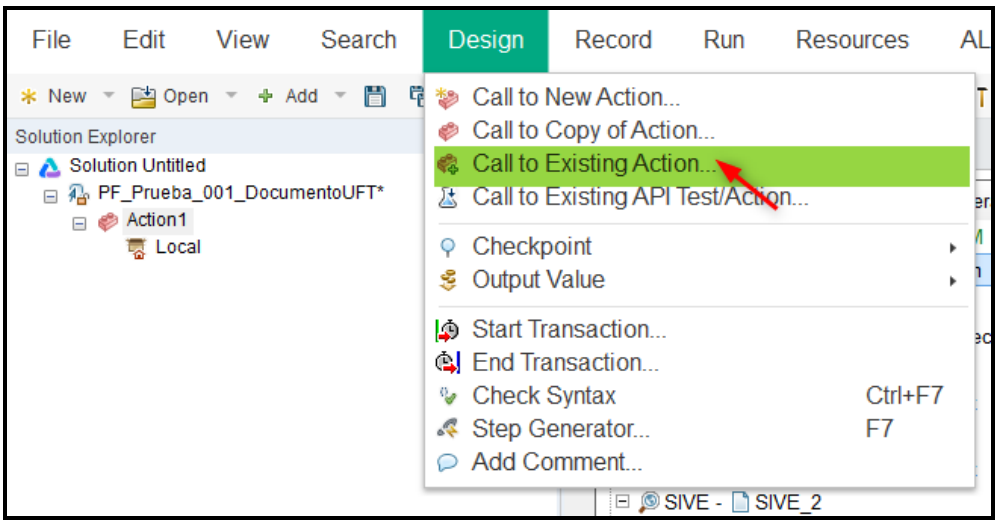
Nota: todas las action se crean como "Reusable Action" por defecto. En caso de no querer que sea reutilizable hay que desmarcar ese check.

10.1.7.2 Llamar a un Action reutilizable

La llamada a un Action reutilizable se puede hacer de diferentes formas aunque en este apartado sólo se explica el método más recomendado ya que solo requiere el mantenimiento del Action en un sólo lugar, es decir, en el Action original.

Los pasos a realizar son:

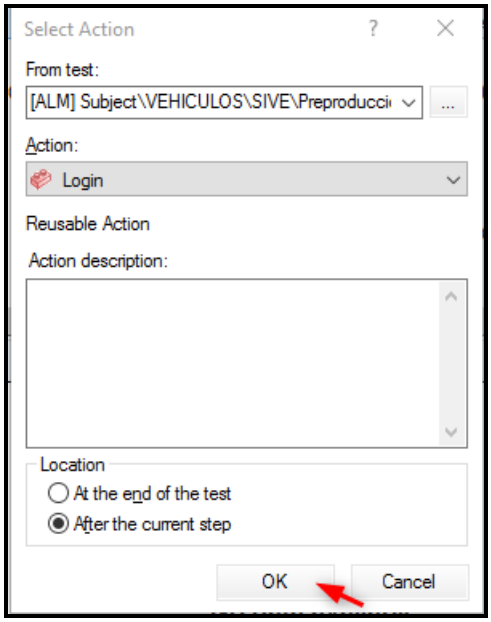
1. Ir a menu y ejecutar "Design->Call to Existing Action …"

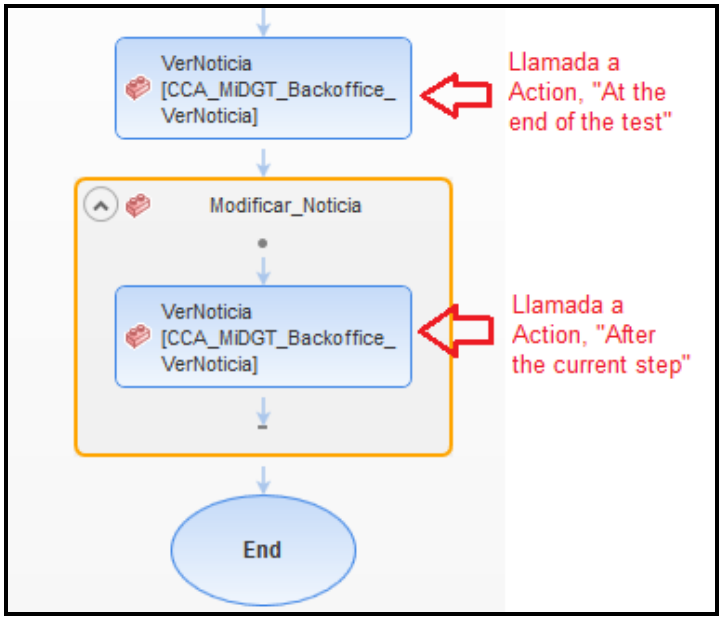


2. Seleccionar en el campo "***From test:***" el test o el componente común automatizado que contiene el Action reutilizable que se quiere usar. Seleccionar en el campo "***Action***" la acción a utilizar.

3. En "Location" seleccionar:

- "At the end of the test", para colocar la llamada a la Action en primer nivel del script (top level). Después se podrá mover desde el final del test a su posición correcta dentro de la secuencia del script.
- O bien, "After the current step", para colocar la llamada a la Action dentro de la Action actualmente seleccionada, en la posición en la que se encuentre el cursor (Action anidada dentro de otra Action)





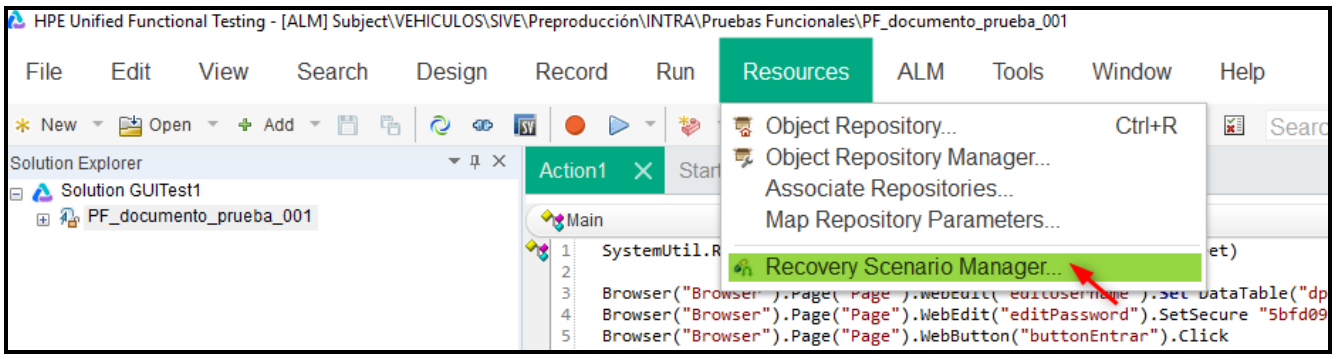
4. Pulsar en el botón "***OK***".

10.1.8 Recovery Scenarios

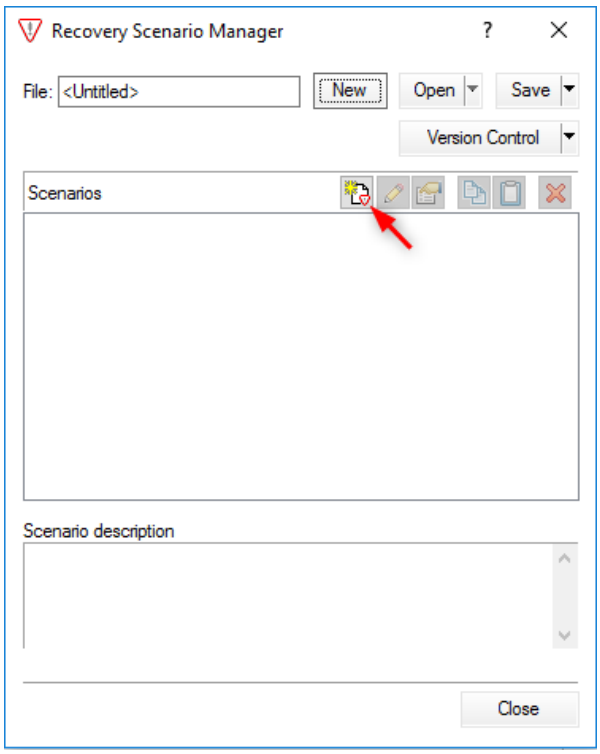
Los *Recovery Scenarios* o Escenarios de Recuperación son el mecanismo de que dispone UFT para el manejo de excepciones.

Una excepción es aquel **evento inesperado** que ocurre ajeno a la ejecución normal de la aplicación y que afecta a la ejecución del test.

Los escenarios de recuperación se gestionan desde el asistente «*Recovery Scenario Manager*» en el menú «*Resources*».



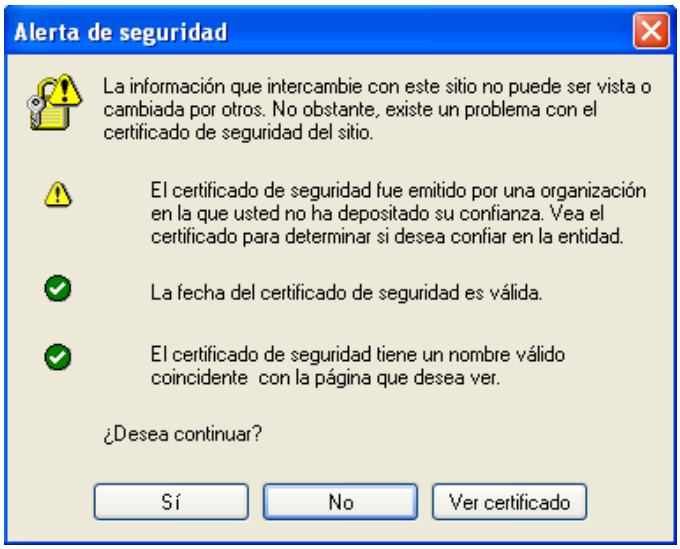
Para la creación de un escenario de recuperación pulsar el botón *New Scenario* de la ventana *Recovery Scenario Manager*. Un asistente guía la creación del *recovery scenario*.



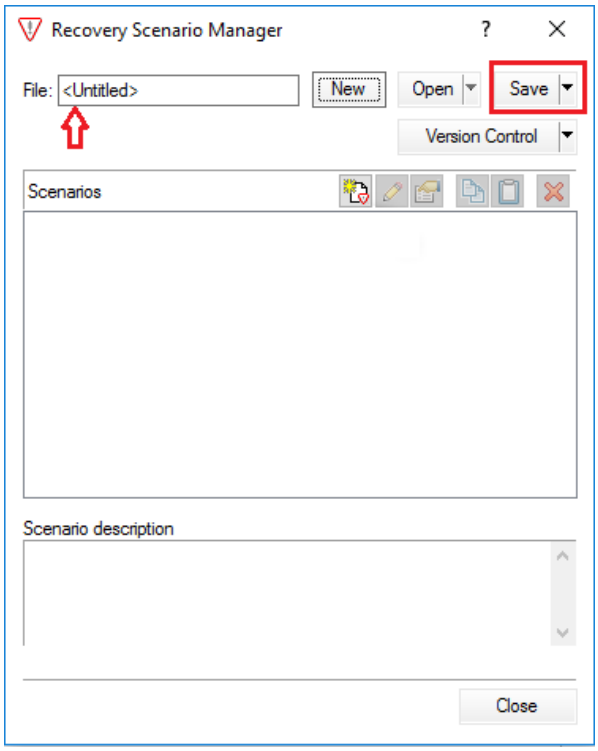
Un escenario de recuperación consta de:

- ***Trigger***: Evento que dispara el escenario de recuperación.
- ***Recovery***: Acciones a realizar para la recuperación.
- ***Post-Recovery***: Acción a realizar después de las operaciones de recuperación.

Un ejemplo *Recovery Scenario* es para capturar la ventana de alerta de seguridad de IE:



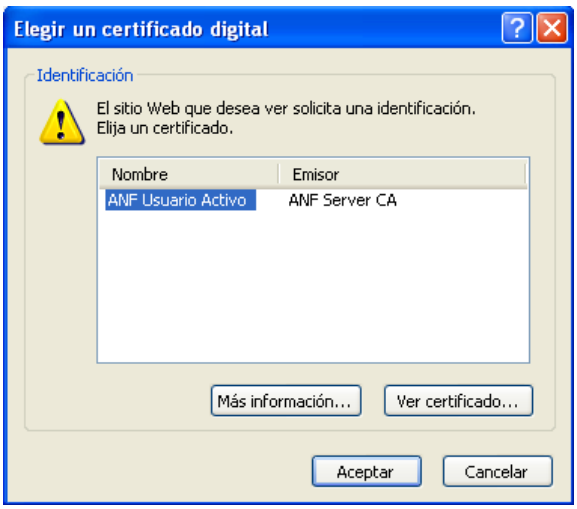
Se pueden crear varios *Recovery Scenarios* y guardarlos juntos en un mismo fichero (*File*). Los ficheros de *Recovey Scenarios* se guardan en ALM, en el módulo de *Test Resources*, como se indica en el apartado "9 Recursos de los Tests".



10.1.9 Sentencia Exist

Utilizando la sentencia Exist se puede comprobar la existencia o no de un objeto.

Un ejemplo de donde se debe utilizar esta función es para preguntar por la existencia o no de la ventana que solicita el certificado de seguridad. Esta ventana aparece en caso de que no se disponga de certificados válidos en sitios de confianza de IE.



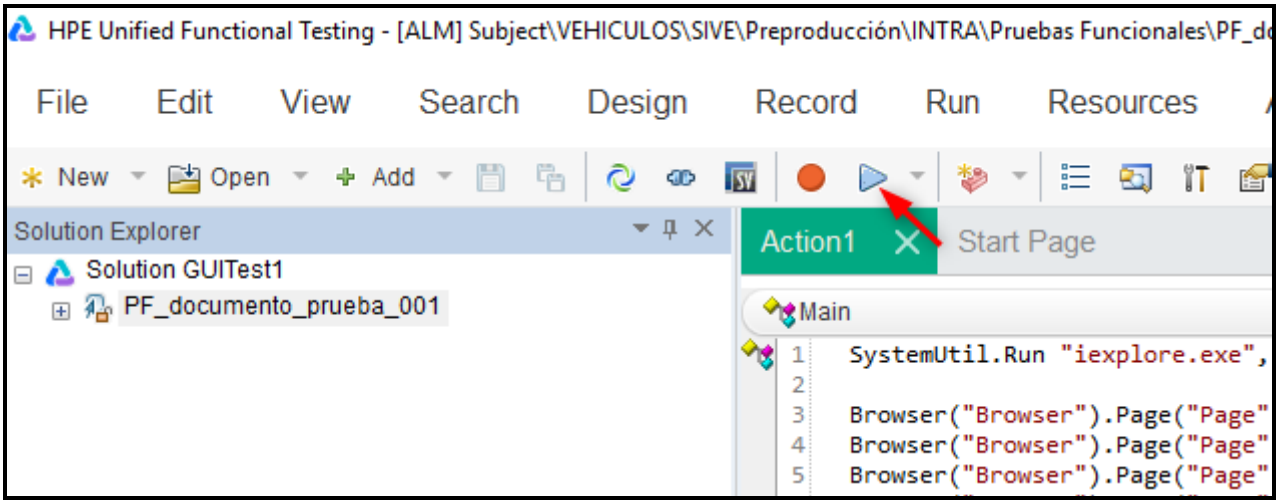
Para no depender la ejecución del test de la configuración de los certificados de los sitios de confianza del IE se utiliza esta solución. Si la ventana existe se realizan los pasos necesarios para seleccionar el certificado correspondiente

```
If Browser("Browser").Dialog("Elegir un certificado").Exist Then
   Browser("Browser").Dialog("Elegir un certificado").Activate
   Browser("Browser").Dialog("Elegir un certificado").Check CheckPoint("Elegir un certificado digital_2")
   Browser("Browser").Dialog("Elegir un certificado").WinListView("El sitio Web que desea").Select "ANF Usuario Activo"
   Browser("Browser").Dialog("Elegir un certificado").WinButton("Aceptar").Click
end If
```

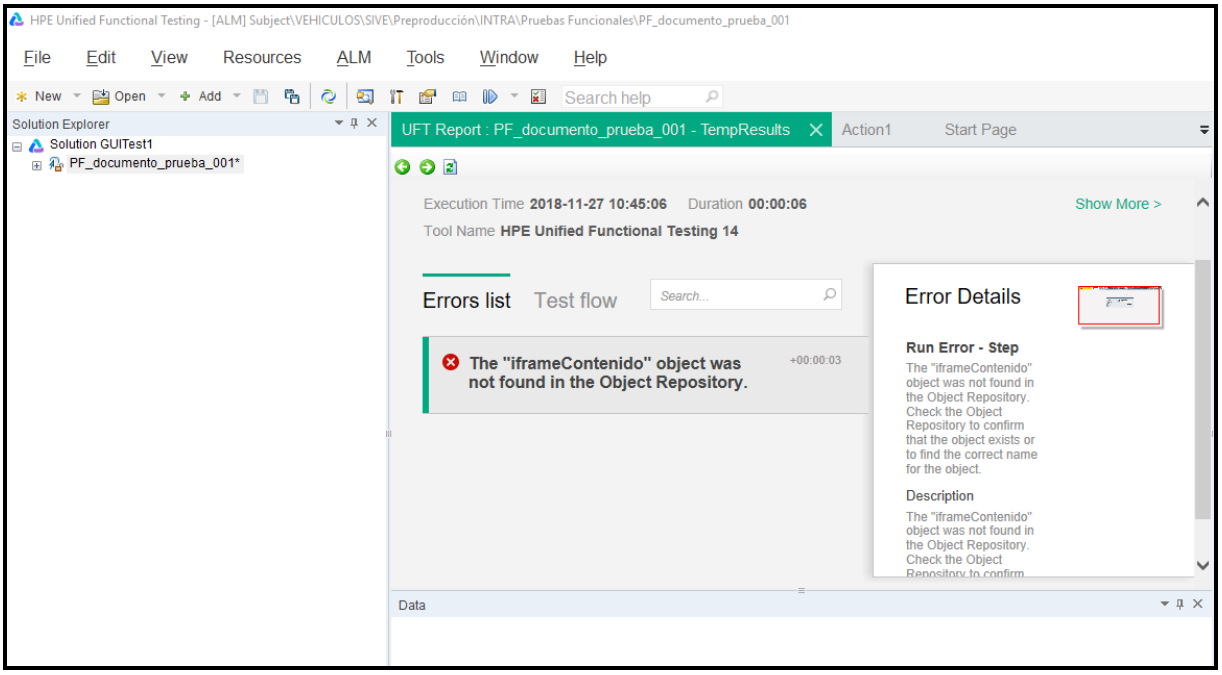
10.1.10 Ejecución de Scripts desde UFT e Informe de Resultados

A continuación se explican los pasos para realizar una ejecución de un scripts desde UFT. No confundir con la ejecución de las baterías de pruebas (Test Sets) que siempre deben ser ejecutadas desde ALM.

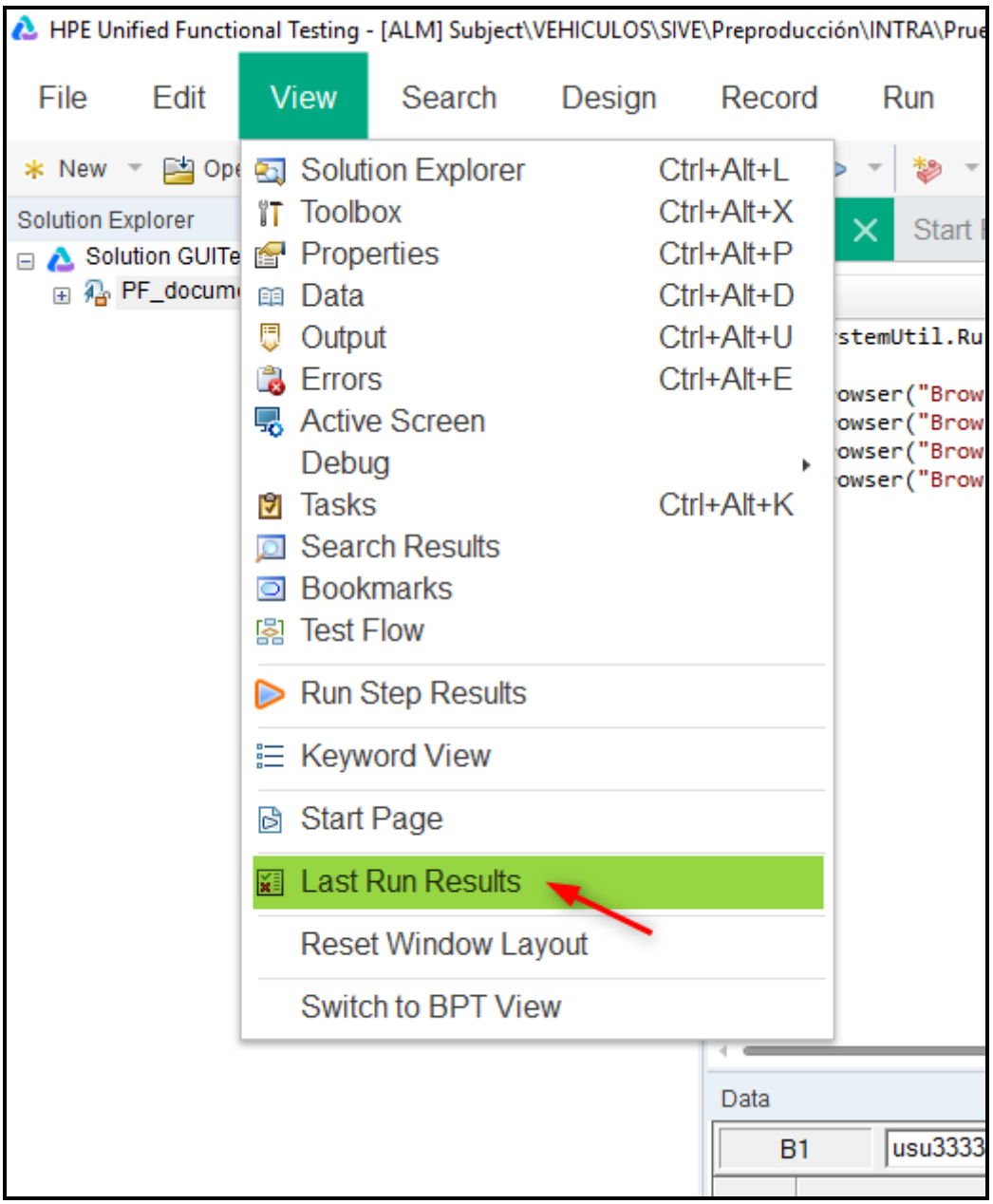
Para ejecutar un script desde UFT pulsar el botón "***Run***".



Una vez finalizada la ejecución se abre una ventana con el resultado de la ejecución.



También se puede llegar a ella mediante la ruta "***View → Last Run Results***"



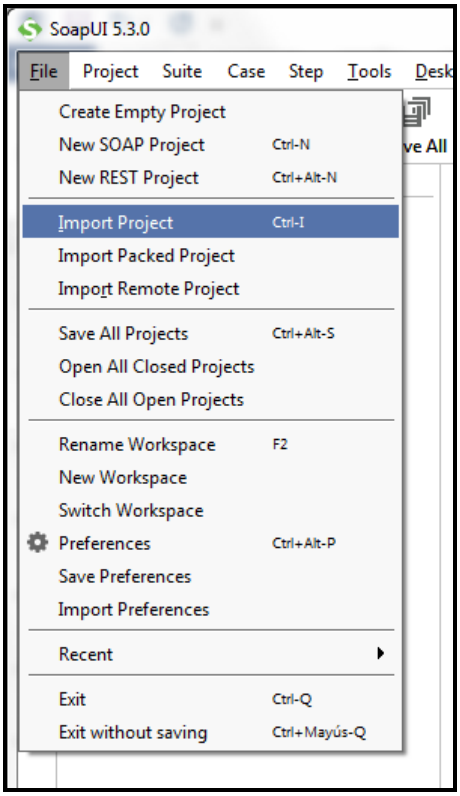
Una vez que se haya verificado que el script se ejecuta correctamente se debe lanzar una ejecución del script desde el módulo Test Lab de ALM con objeto de dejar registrado el informe de resultados en ALM. Ver "*Anexo 03.03 Guia de gestión de proyectos en ALM 11.52*", apartado *"2.4.2 Creación de una batería"*.

10.2 Construcción del Caso de Prueba API

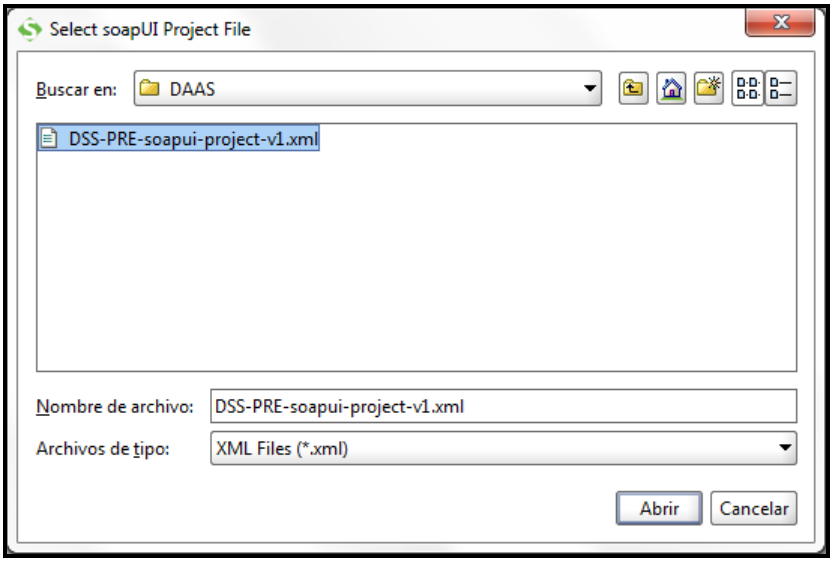
Los pasos necesarios para desarrollar un script de prueba de WebServices automático de UFT a partir de proyecto SoapUI son los siguientes:

10.2.1 Importación del proyecto SoapUI

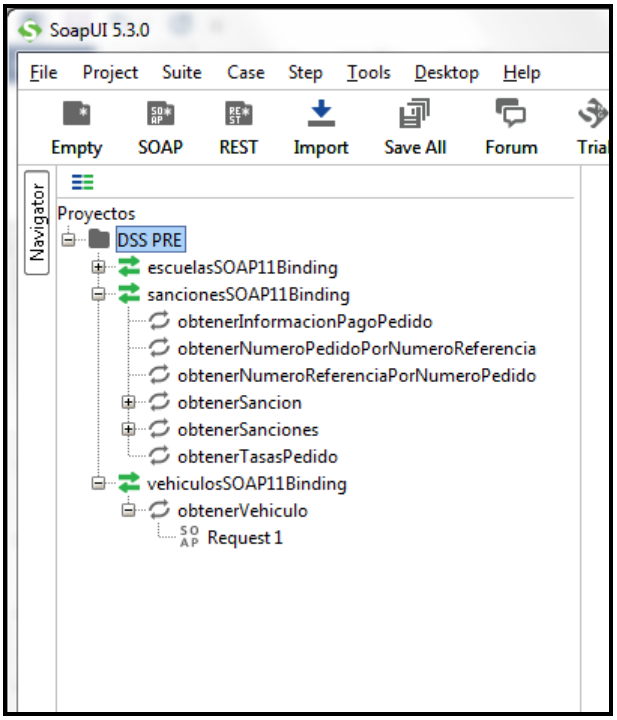
Lo primero que debemos hacer es importar el proyecto en el workspace de nuestro SoapUI:



Pulsamos en “Import Project”, y en la siguiente ventana navegamos hasta la ubicación del fichero .xml:

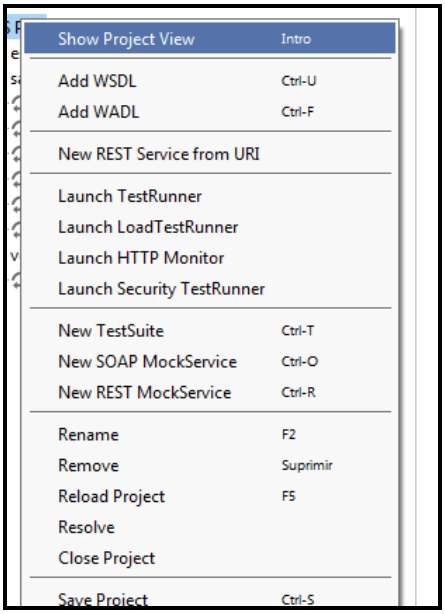


Seleccionamos el fichero, y pulsamos "Abrir". Con esto SoapUI importa el proyecto, que se mostrará en el workspace:

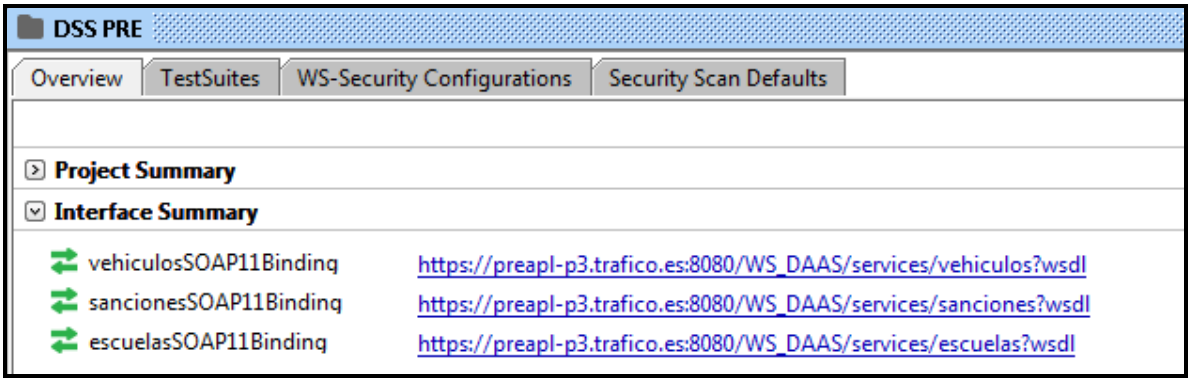


10.2.1.1 Configuración del proyecto SoapUI

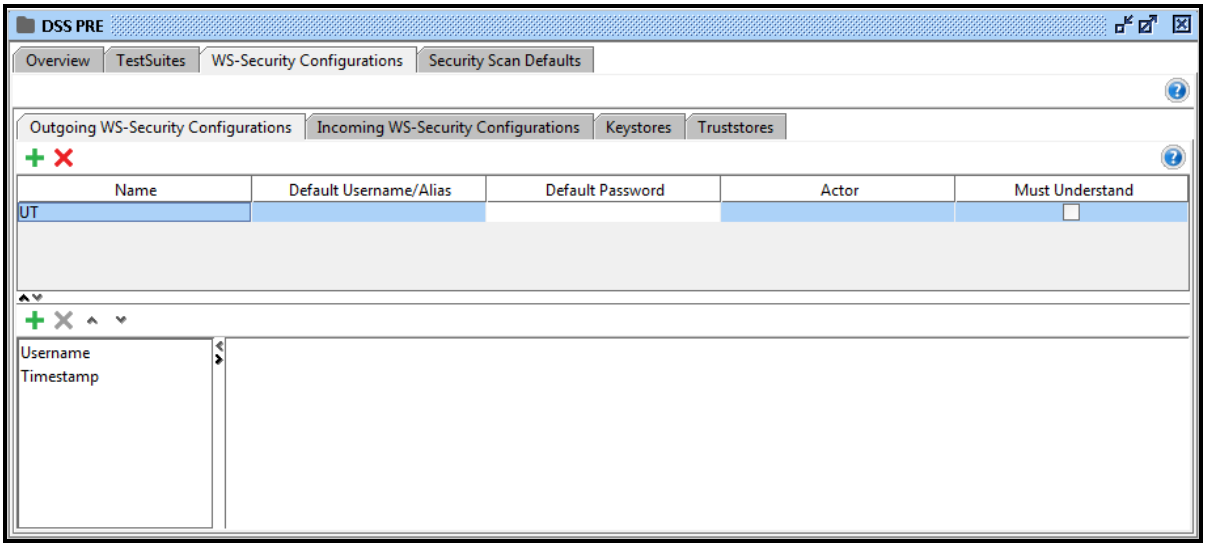
Una vez importado el proyecto, abrimos la ventana "Project View":



En este proyecto en concreto nos fijamos que hay 3 WSDLs, así que anotamos las URLs correspondientes ya que las necesitaremos para crear los casos asociados.



En la ventana de propiedades del proyecto debemos verificar que existe la configuración que se debe utilizar para autentificar las peticiones, en caso de que sea necesario. En el proyecto que se muestra, se autentifica por Usuario/Contraseña y genera un Timestamp.

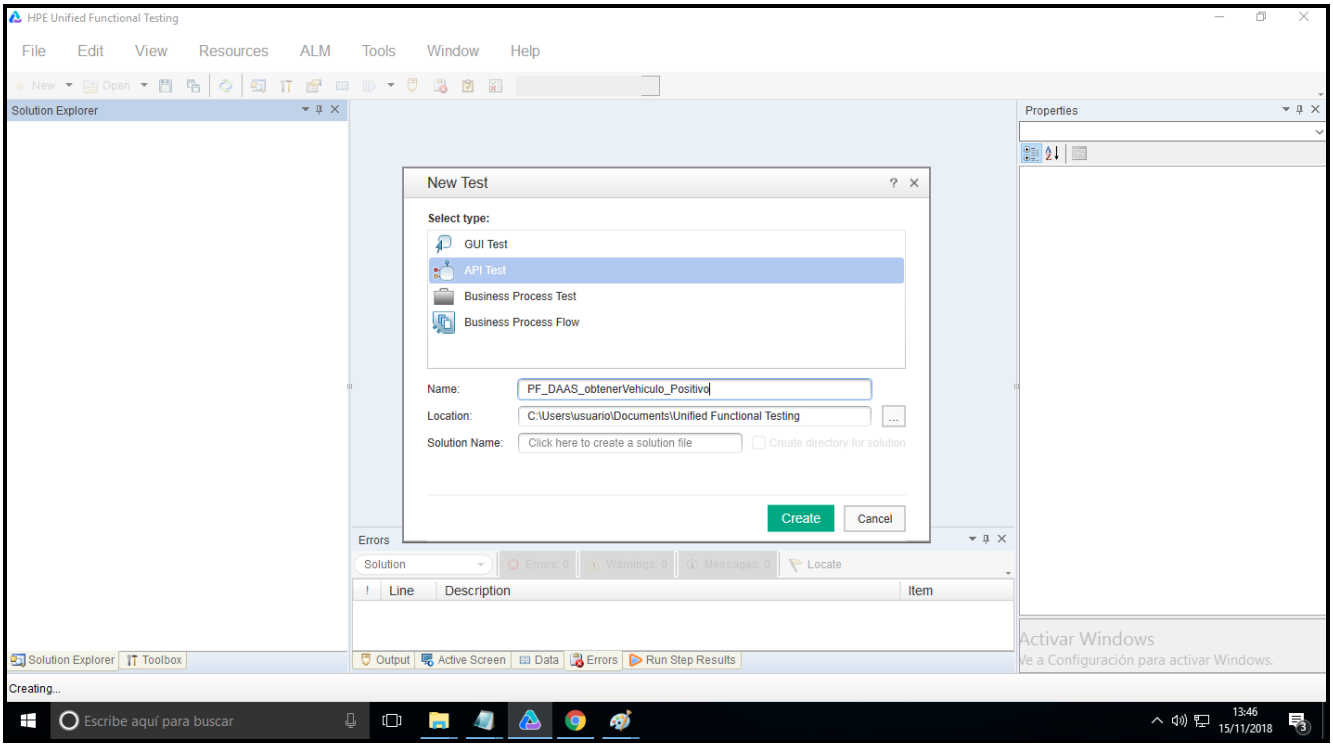


Nota: Esta es la configuración que deberemos tener en cuenta para luego replicarla en UFT

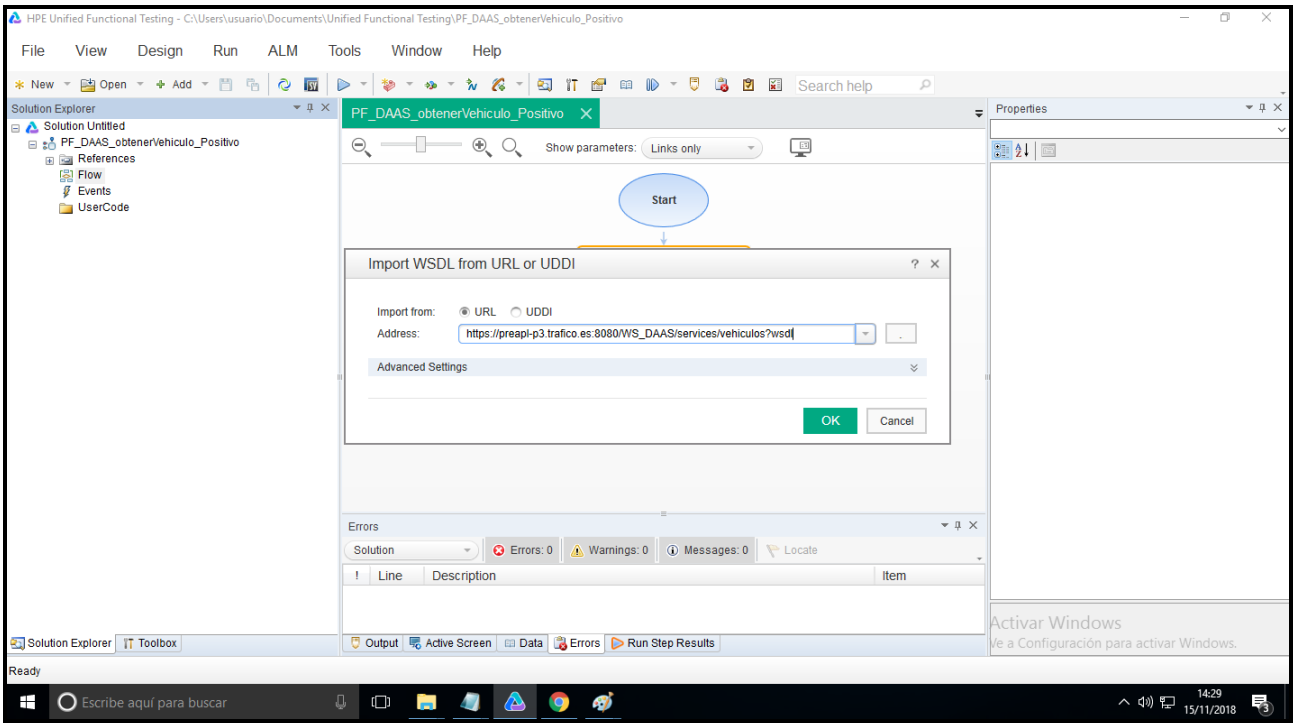
10.2.2 Creación del test desde UFT

A partir de los datos que contiene el proyecto de SoapUI vamos a crear nuestros casos de prueba en UFT. Para ello abriremos UFT y crearemos un nuevo caso de prueba del tipo "API Test" y lo guardaremos en la máquina local. Una vez finalizado la creación de los casos ya sí se almacenarán en ALM.

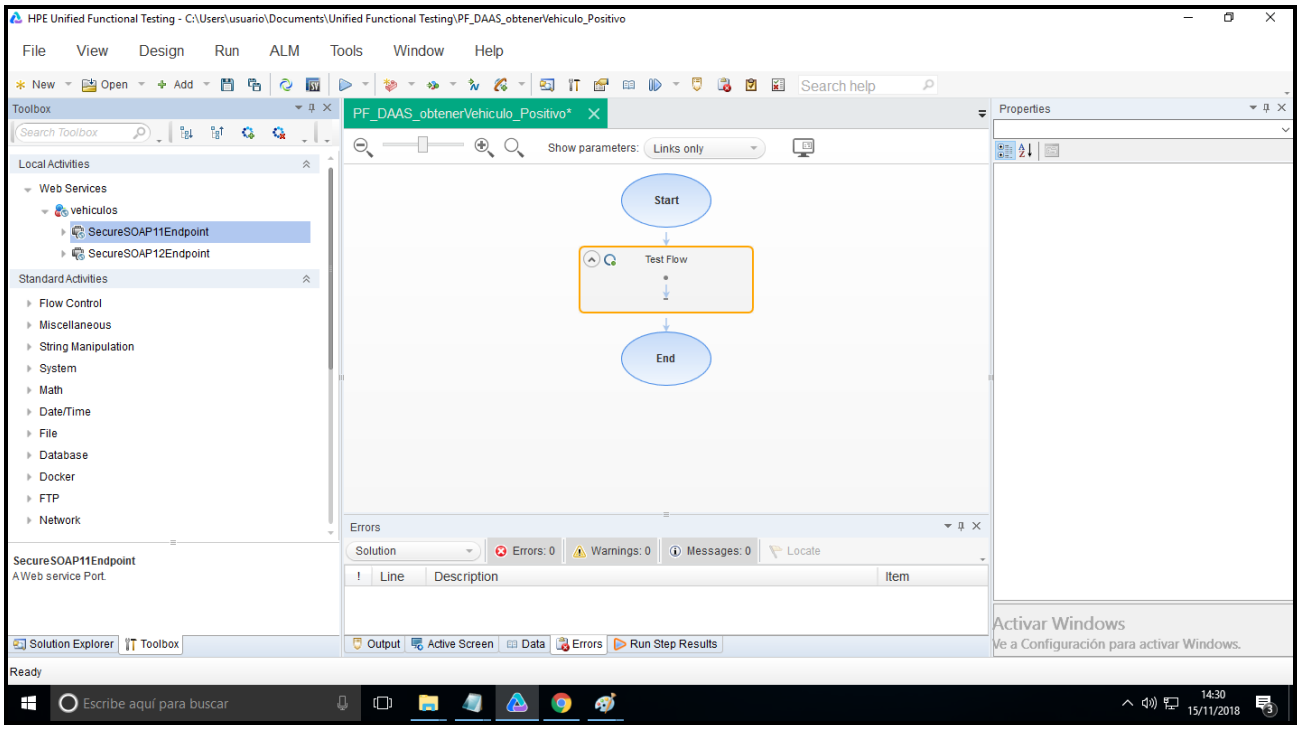
Como norma general por cada operación de cada servicio deberá tener, al menos, dos Test Case, uno para las pruebas del caso positivo y otro para las pruebas del caso negativo.



Ahora ponemos la URL del WSDL que queramos importar:

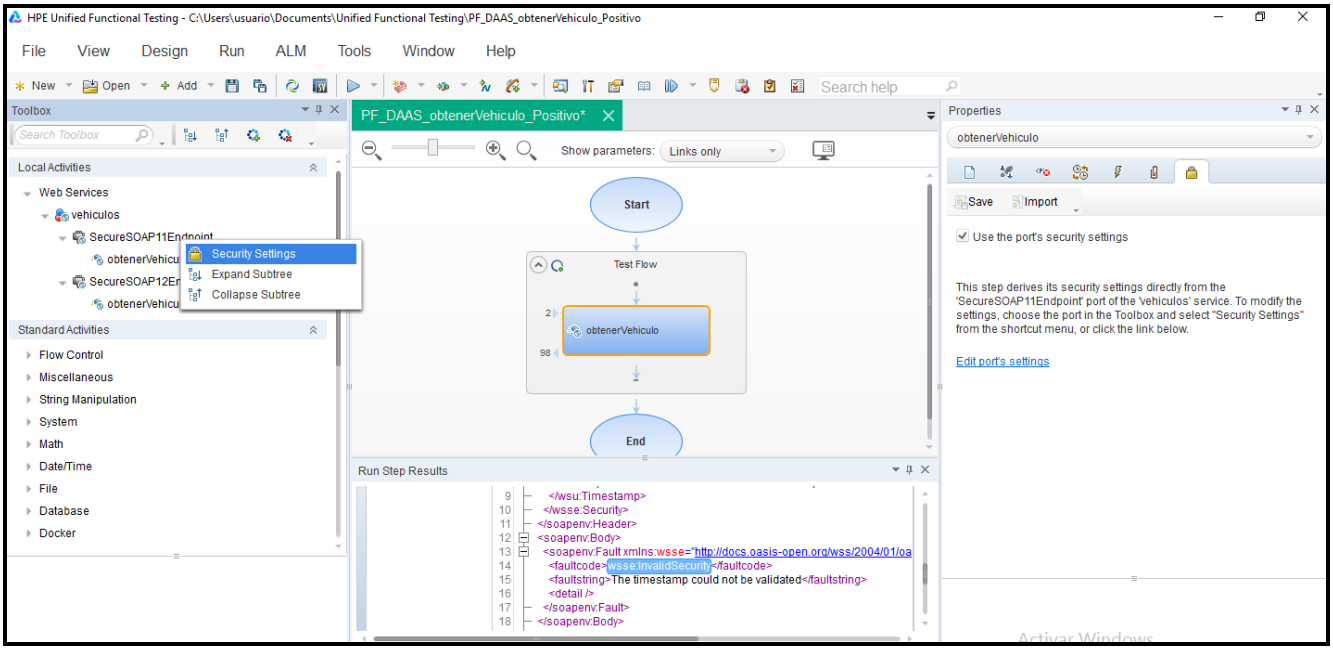


Damos a OK:

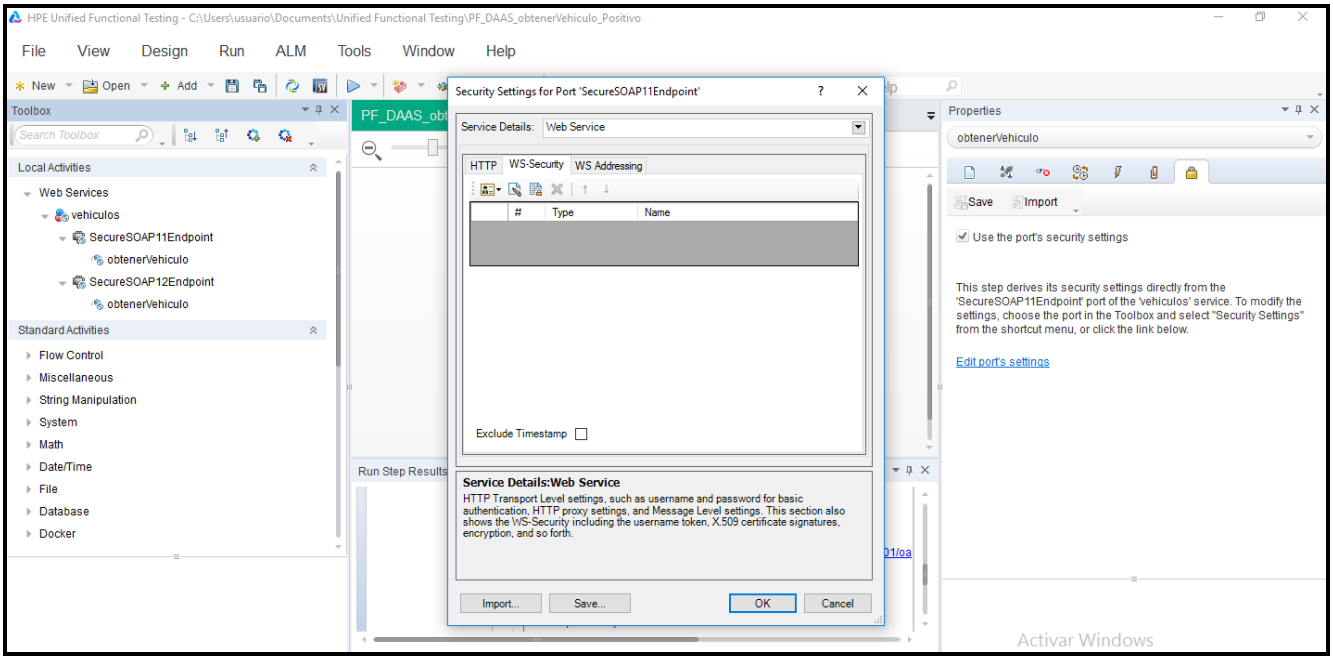


Ahora configuraremos los datos de seguridad necesarios si es que el WebService lo requiere. En este caso sí.

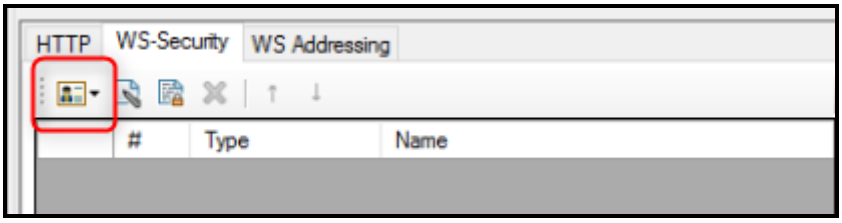
Situados sobre el servicio que vayamos a probar, en este caso “ServiceSOAP11Endpoint” y pulsamos el botón derecho del ratón:

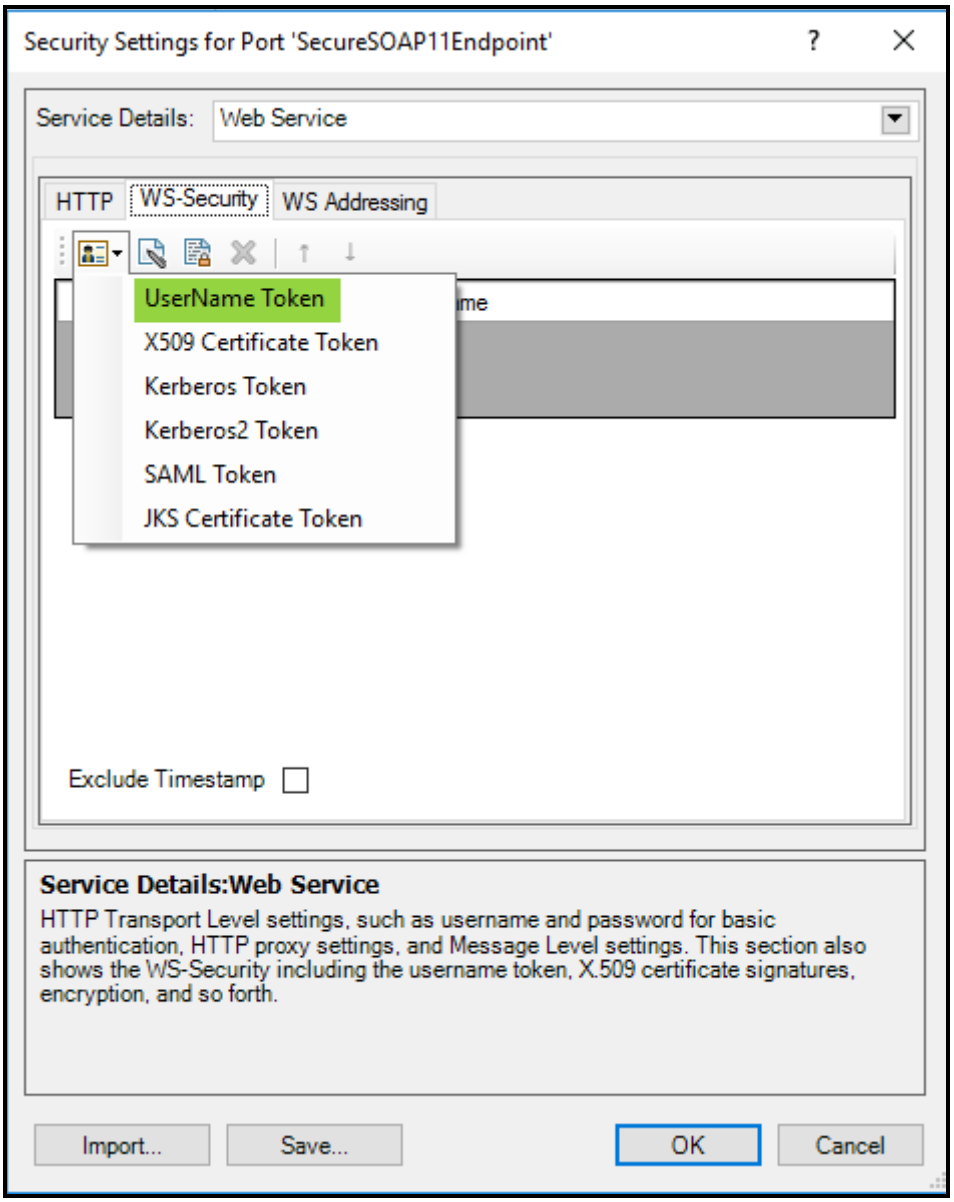


Ahora seleccionamos la opción “Security Settings”

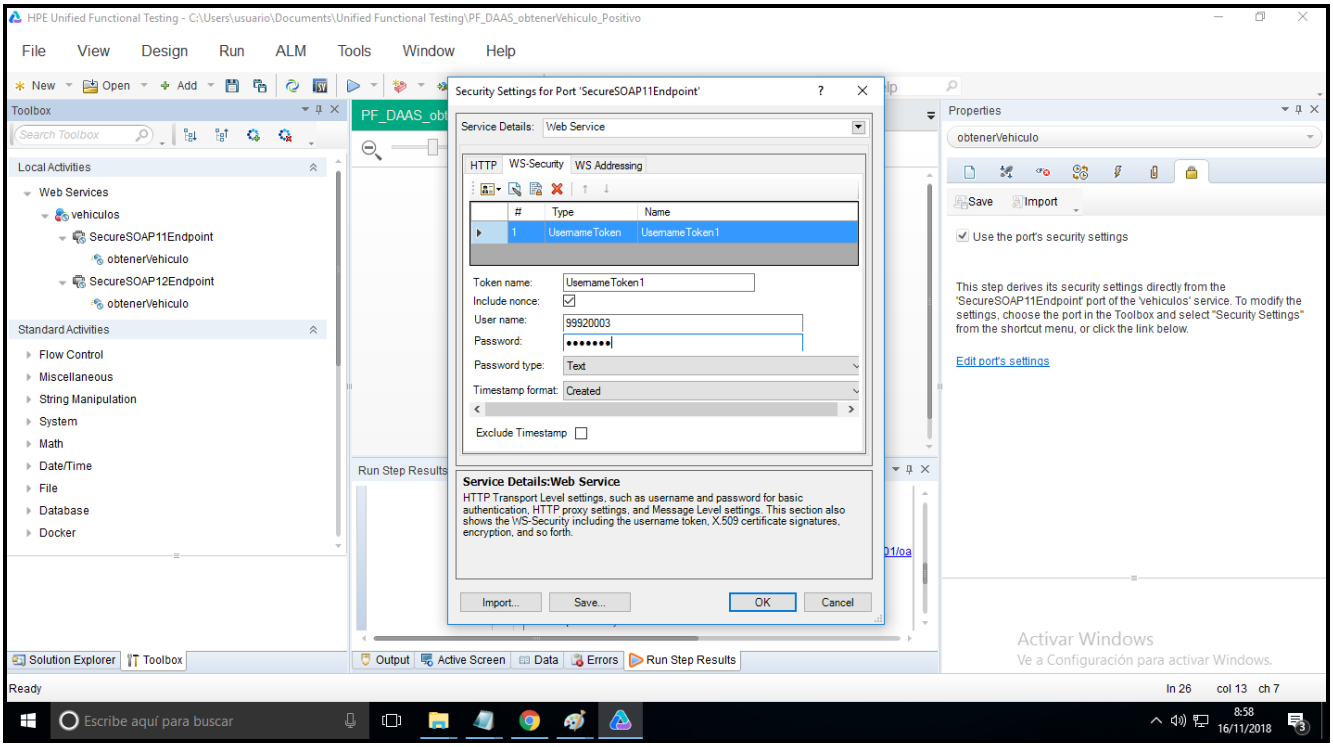


Click en el icono “Security Tokens” y seleccionamos la opción “Username” token:

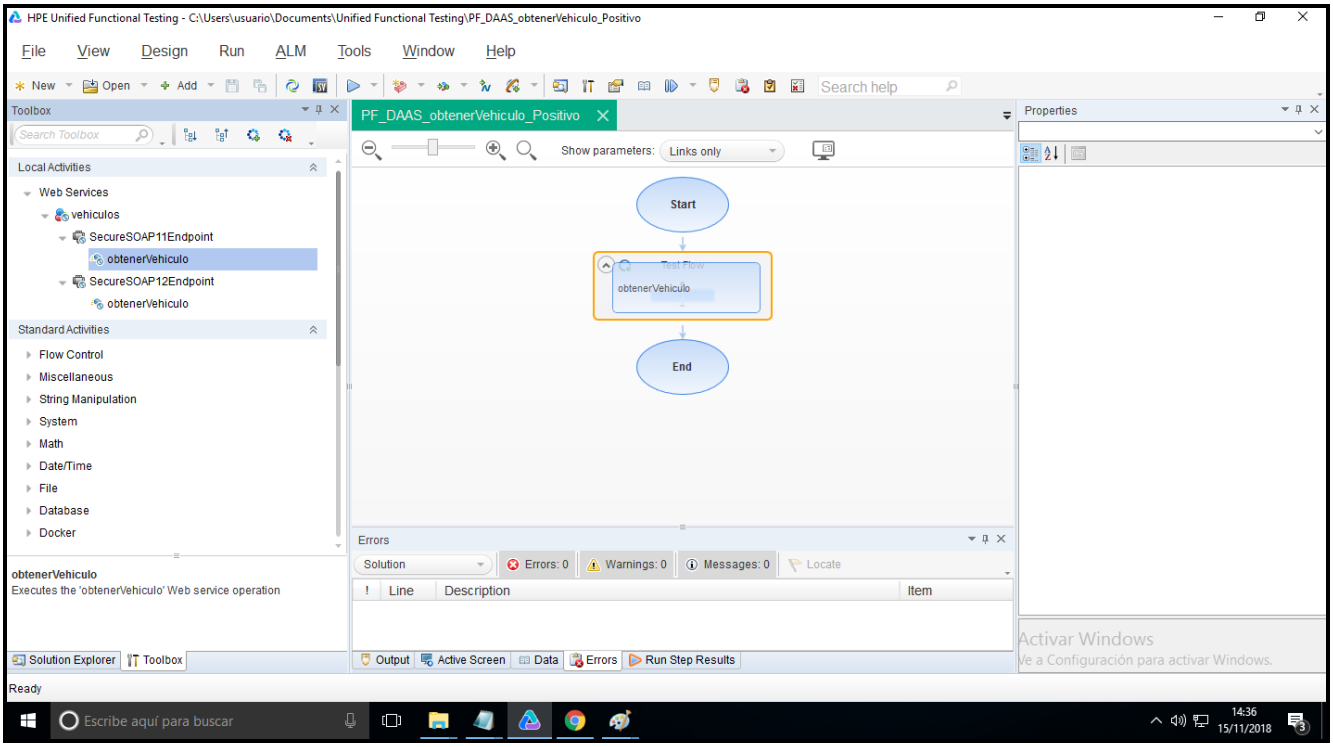




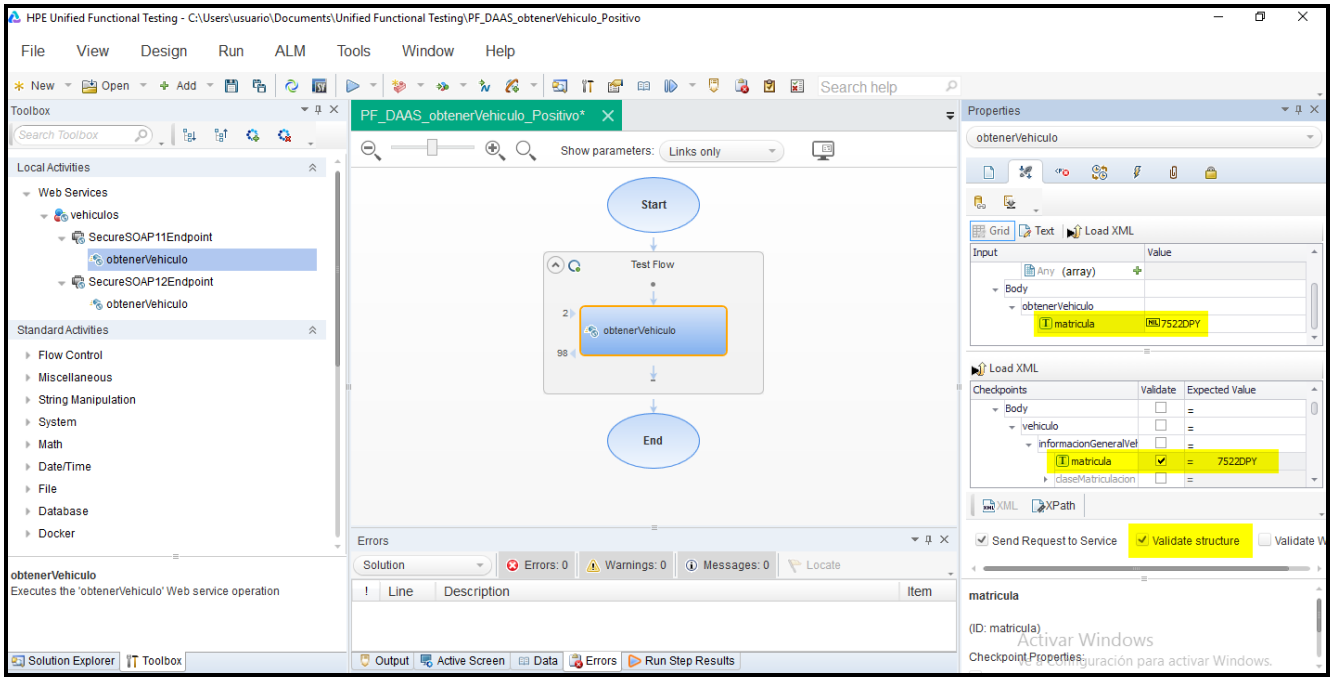
Ahora rellenamos los datos adecuados:



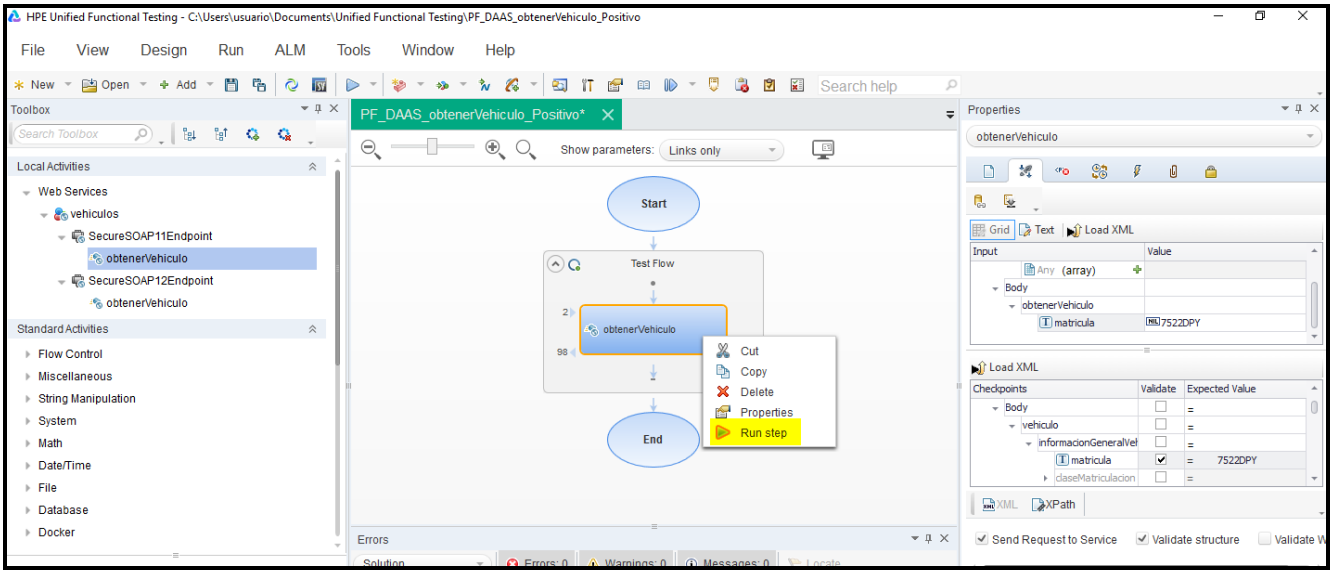
Ahora desplegamos el servicio a probar, SecureSOAP11Endpoint en este caso, y arrastramos la petición (request) bajo prueba, en este caso “obtenerVehiculo”, a la caja TestFlow:

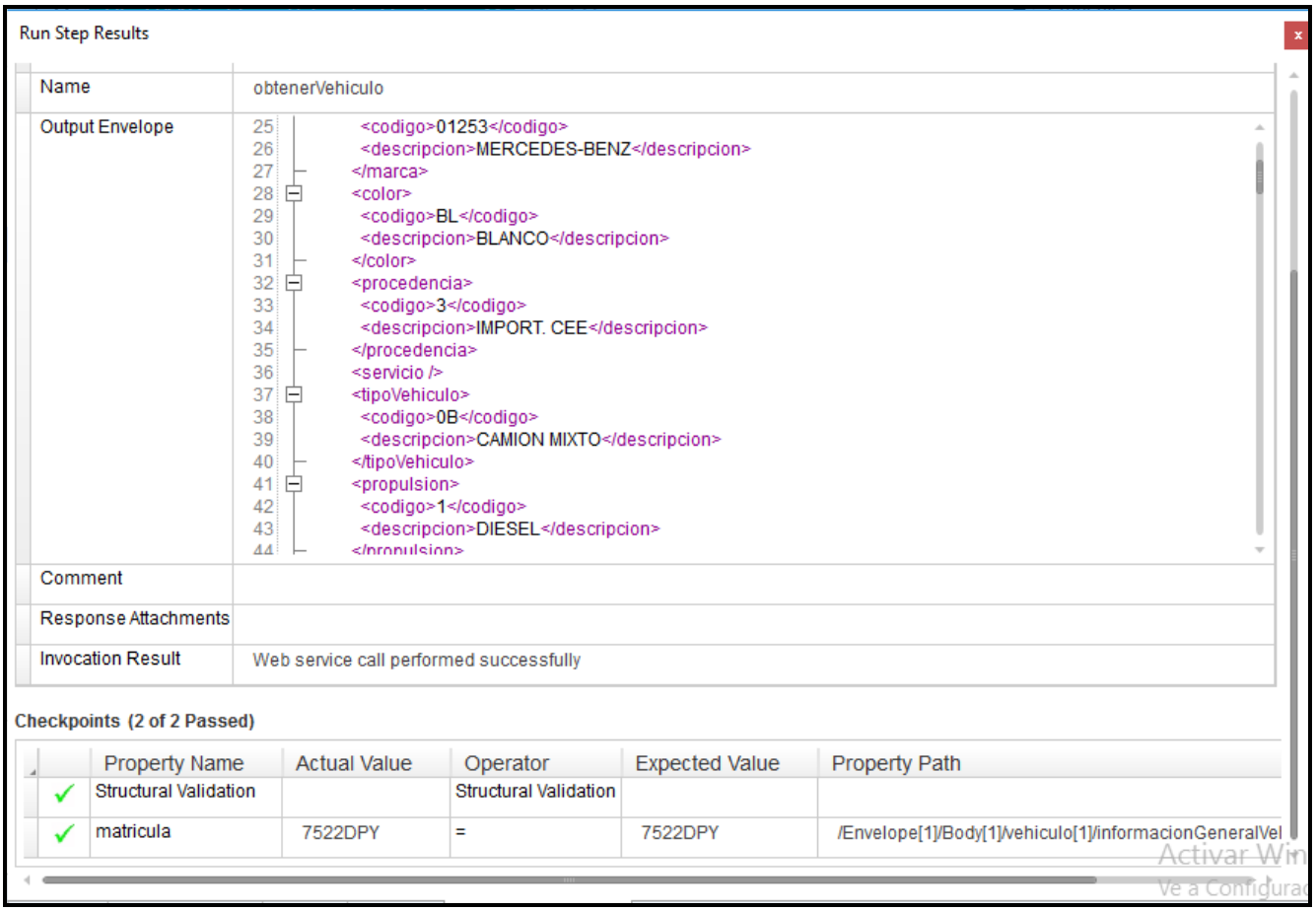


Una vez arrastrado tenemos que modificar la Request de acuerdo a como estaba en proyecto de SoapUI e introducirle los checkpoints que consideremos oportunos. En este caso introducimos la matrícula a comprobar y la validación de la estructura de la respuesta.

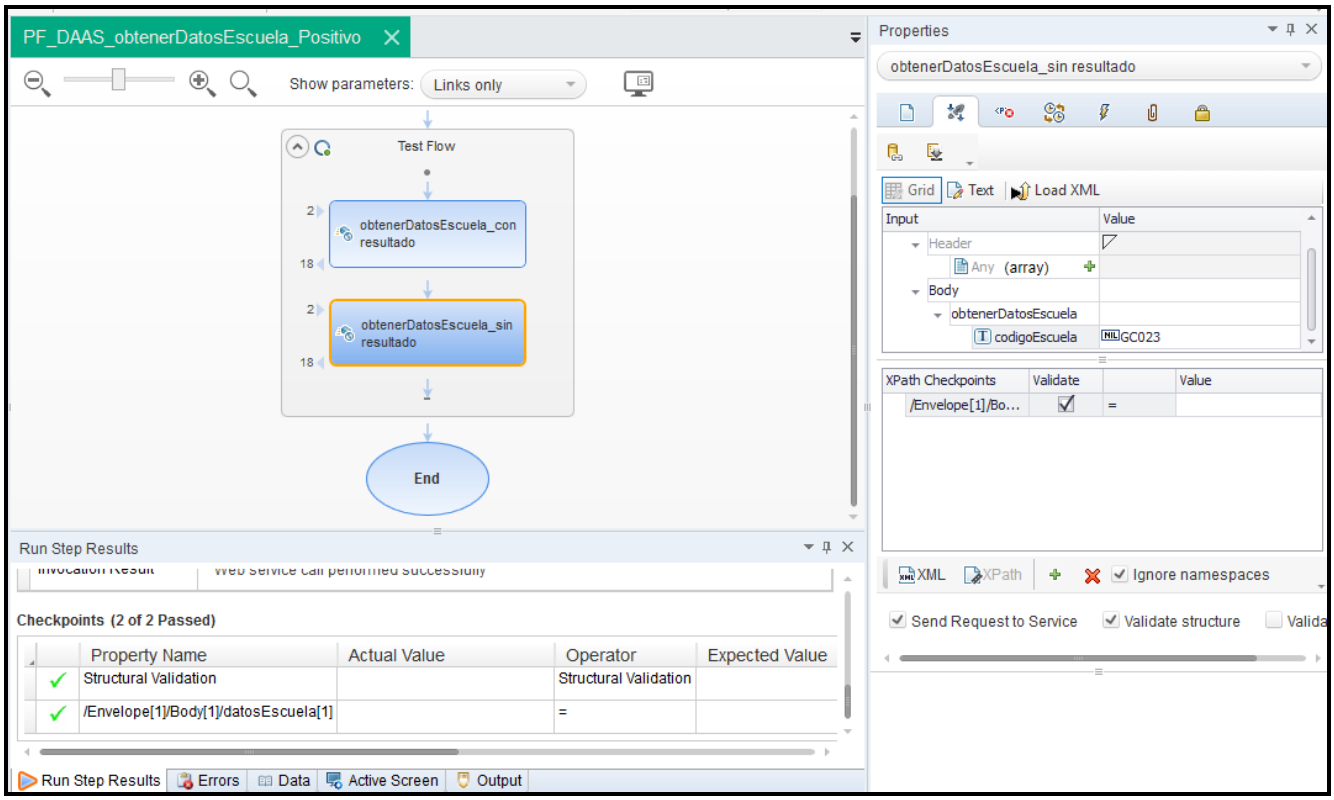


Ahora podemos hacer una comprobación de si funciona el WebService. Situándonos sobre la caja de “obtenerVehiculo” pulsamos botón derecho del ratón y seleccionamos la opción “RunStep”.


HPE Unified Functional Testing - C:\Users\usuario\Documents\Unified Functional Testing\PF_DAAS_obtenerVehiculo_Positivo
File View Design Run ALM Tools Window Help
Toolbox
Local Activities
Web Services
vehiculos
SecureSOAP11Endpoint
obtenerVehiculo
SecureSOAP12Endpoint
obtenerVehiculo
Standard Activities
Flow Control
Miscellaneous
String Manipulation
System
Math
Date/Time
File
Database
Docker
PF_DAAS_obtenerVehiculo_Positivo*
Show parameters: Links only
Start
Test Flow
obtenerVehiculo
End
Cut
Copy
Delete
Properties
Run step
Errors
Properties
obtenerVehiculo
Grid
Text
Load XML
Input
Value
Any (array)
Body
obtenerVehiculo
matricula
7522DPY
Checkpoints
Validate
Expected Value
vehiculo
informacionGeneralVel
claseMatriculacion
XML
XPath
Send Request to Service
Validate structure


Run Step Results
Name
obtenerVehiculo
Output Envelope
25 <codigo>01253</codigo>
26 <descripcion>MERCEDES-BENZ</descripcion>
27 </marca>
28 <color>
29 <codigo>BL</codigo>
30 <descripcion>BLANCO</descripcion>
31 </color>
32 <procedencia>
33 <codigo>3</codigo>
34 <descripcion>IMPORT. CEE</descripcion>
35 </procedencia>
36 <servicio />
37 <tipoVehiculo>
38 <codigo>0B</codigo>
39 <descripcion>CAMION MIXTO</descripcion>
40 </tipoVehiculo>
41 <propulsion>
42 <codigo>1</codigo>
43 <descripcion>DIESEL</descripcion>
Comment
Response Attachments
Invocation Result
Web service call performed successfully
Checkpoints (2 of 2 Passed)
Property Name
Actual Value
Operator
Expected Value
Property Path
Structural Validation
Structural Validation
matricula
7522DPY
=
7522DPY
/Envelope[1]/Body[1]/vehiculo[1]/informacionGeneralVel

Comprobamos si la ejecución ha ido Ok. Si en el mismo caso queremos añadir un segundo step lo pondríamos a continuación de manera similar al primero, es decir, arrastrando la request a la zona del TestFlow.



10.2.3 Datos de entrada en Tabla de Datos

Al igual que para los casos de GUI, para los scripts de API (WebServices) también podemos tener los datos de entrada y en este caso los resultados esperados en una tabla de datos.

Para hacer uso de ello se siguen los siguientes pasos:

Creamos un fichero Excel con los datos de prueba a usar y los resultados esperados siguiendo la nomenclatura establecida para las tablas de Datos.

DENT_<APLI>_<ENTORNO>_<Nº CasoPrueba>_<BreveDescripcion>_<NºScript>

Incluiremos tantas columnas como datos de entrada tenga el servicio que vamos a probar, usando la nomenclatura establecida, es decir añadiremos el prefijo "**dp**" (dato prueba) antes del nombre del parámetro de la petición.

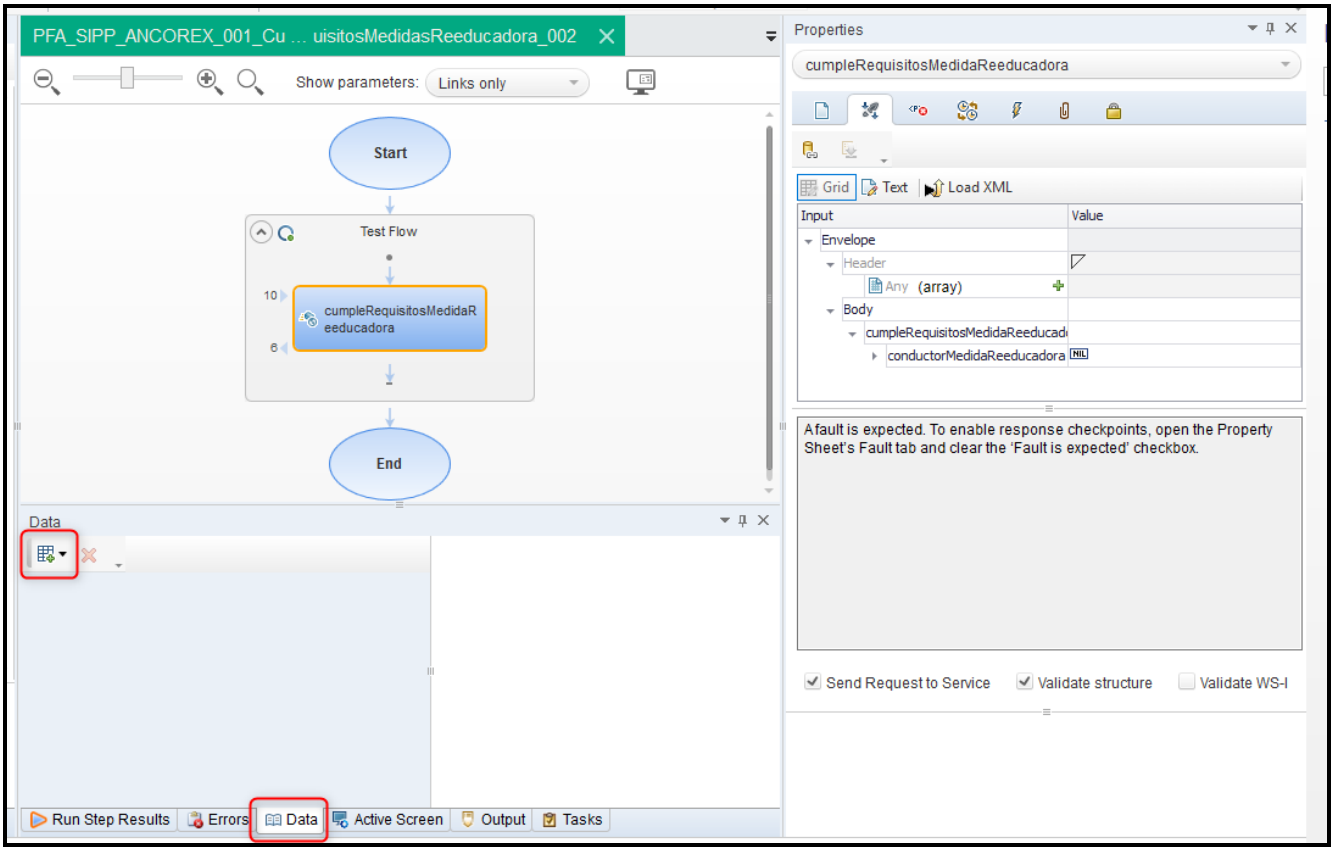
Se incluirán además las columnas necesarias para la validación de checkpoints, como son "**r**esultados **e**sperados" a los parámetros de dichas columnas le incluiremos el prefijo "**re**" antes del nombre del parámetro.

Input	Value
conductorMedidaReeducador	NIL
numPermiso	NIL
esProfesional	NIL
tipoCurso	NIL
fechaInicioCurso	NIL
fechaFinDeCurso	NIL
codCentro	NIL
codCurso	NIL
comprobarDuplicidad	NIL
numCiclo	NIL

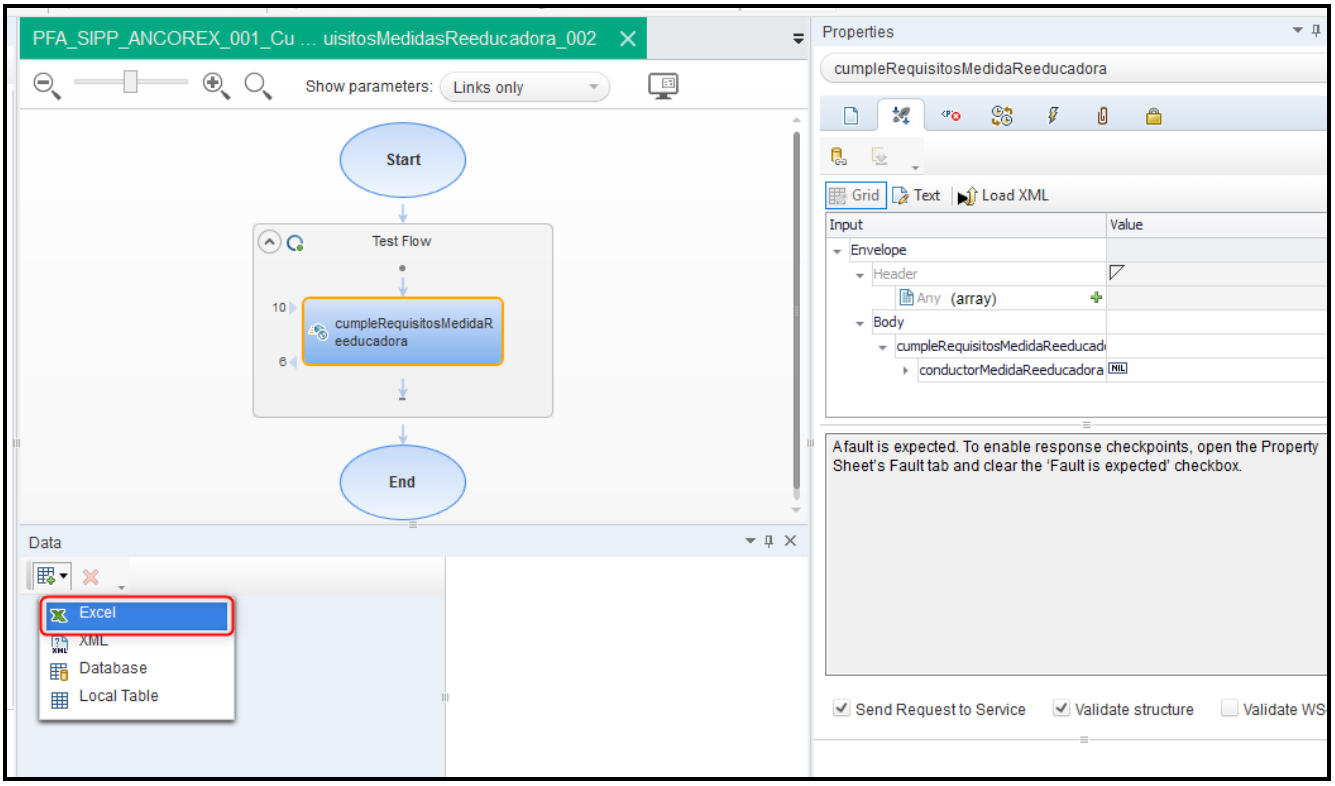
DENT_SIPP_ANCOREX_PRE_001_cumpleRequisitosMedidaReeducadora_002 [Modo de compatibilidad] - Excel

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	dpnumPermiso	dpesProfesional	dptipoCurso	dpfechaInicioCurso	dpfechaFinDeCurso	dpcodCentro	dpcodCurso	dpcomprobarDuplicidad	dpnumCiclo	recodigo	repuntos	refaultcode	refaultstring	
2	50185080	N		01/06/2018	03/06/2018	28194	2001A16001	N	0			Server	El campo tipoCurso debe s	
3	50185080	N	B1	01/06/2018	03/06/2018	28194	2001A16001	N	0			Server	No tiene anotada la perdida	
4		N	A	01/06/2018	03/06/2018	28194	2001A16001	N	0			Server	Error al unificar el document	
5				01/06/2018	03/06/2018							Server	El campo 'esProfesional' de	
6	50185080	N	B1		03/06/2018	28194	2001A16001	N	0			Server	Error en el formato de la fec	
7	51062803	N	A	01/06/2018	03/06/2018	28194	2001A16001	N	0			Server	maximo saldo	

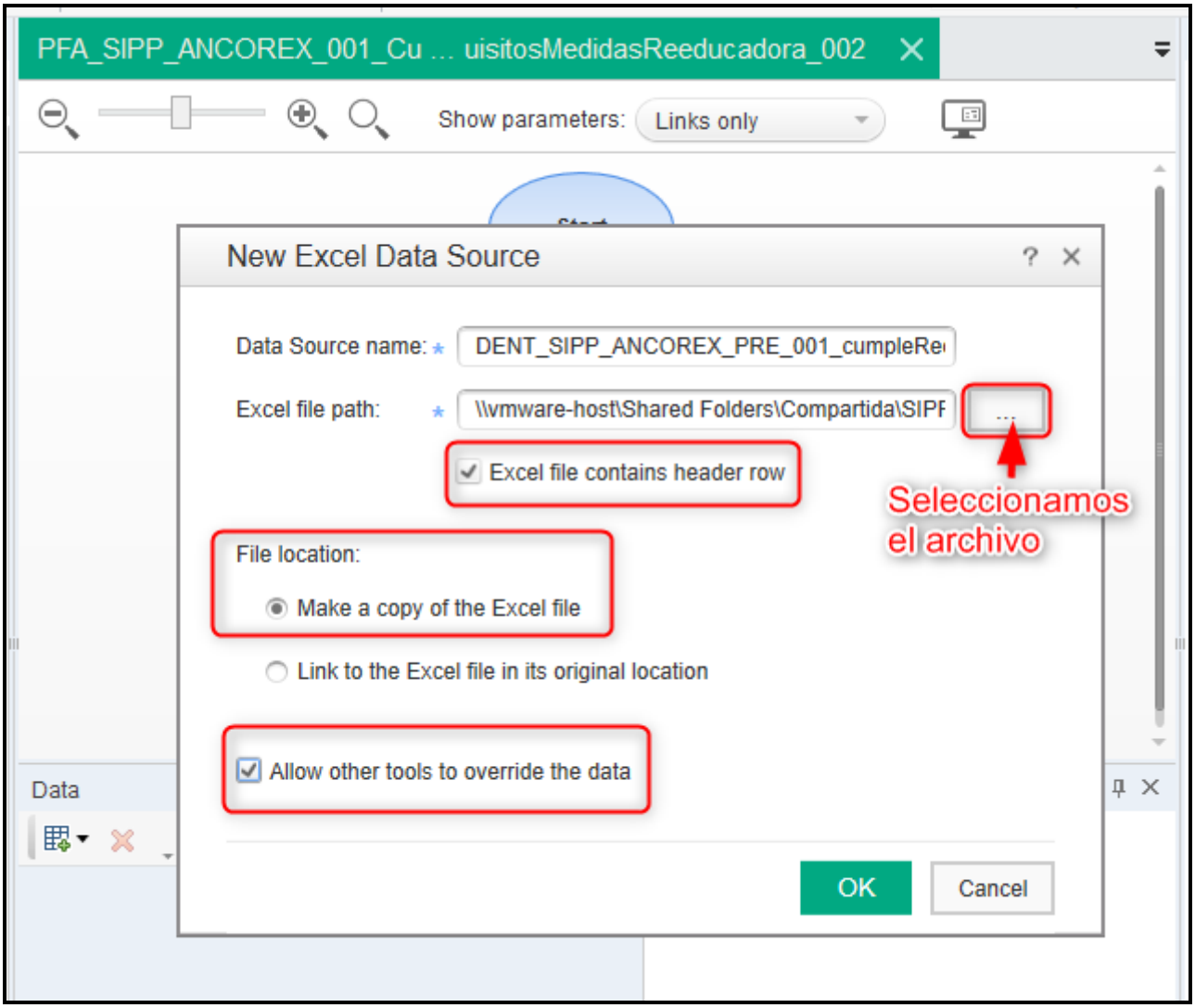
- Nos situamos en el servicio deseado y en la pestaña "Data" clicamos en el icono "New Data Source":

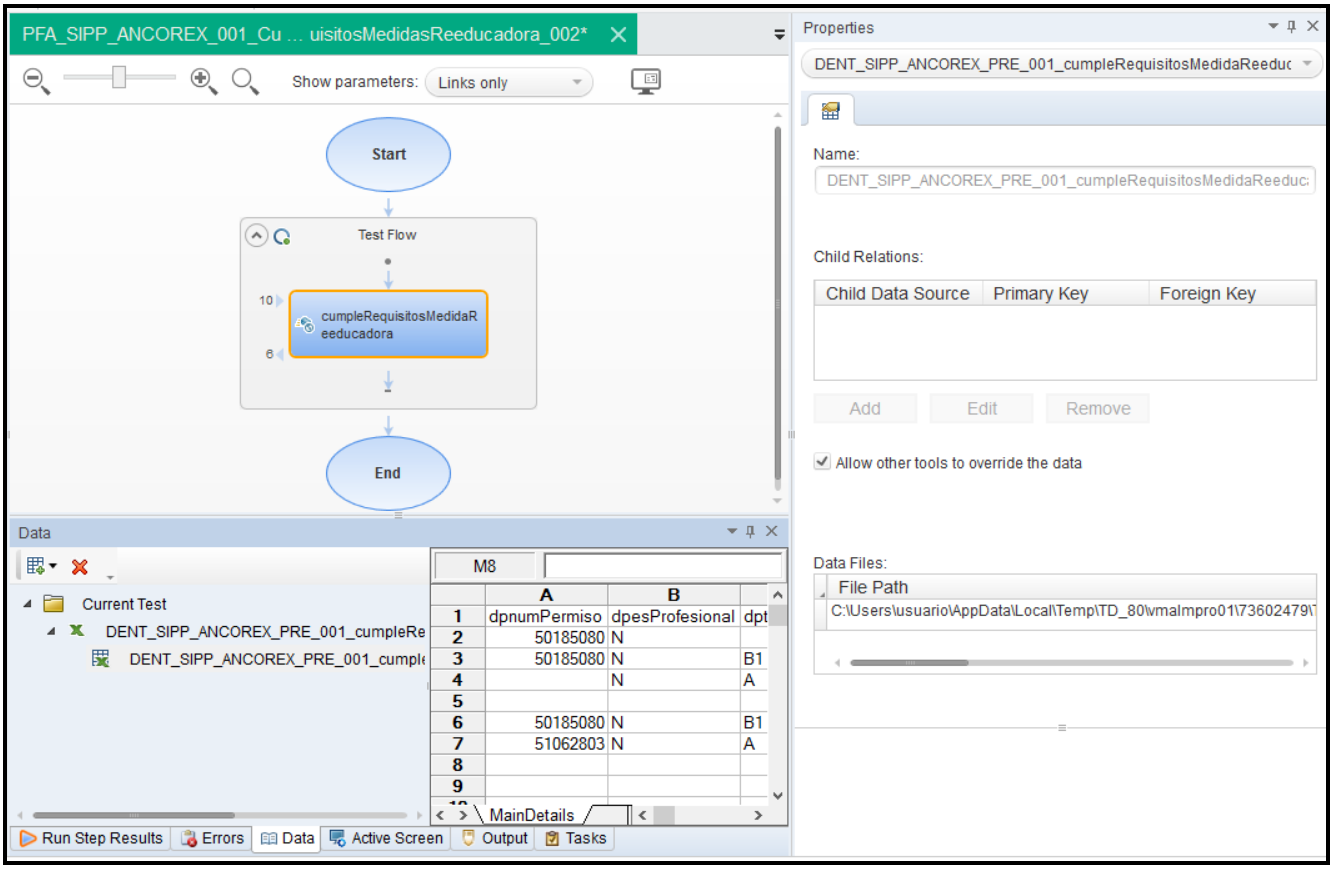


- Ahora seleccionamos Excel

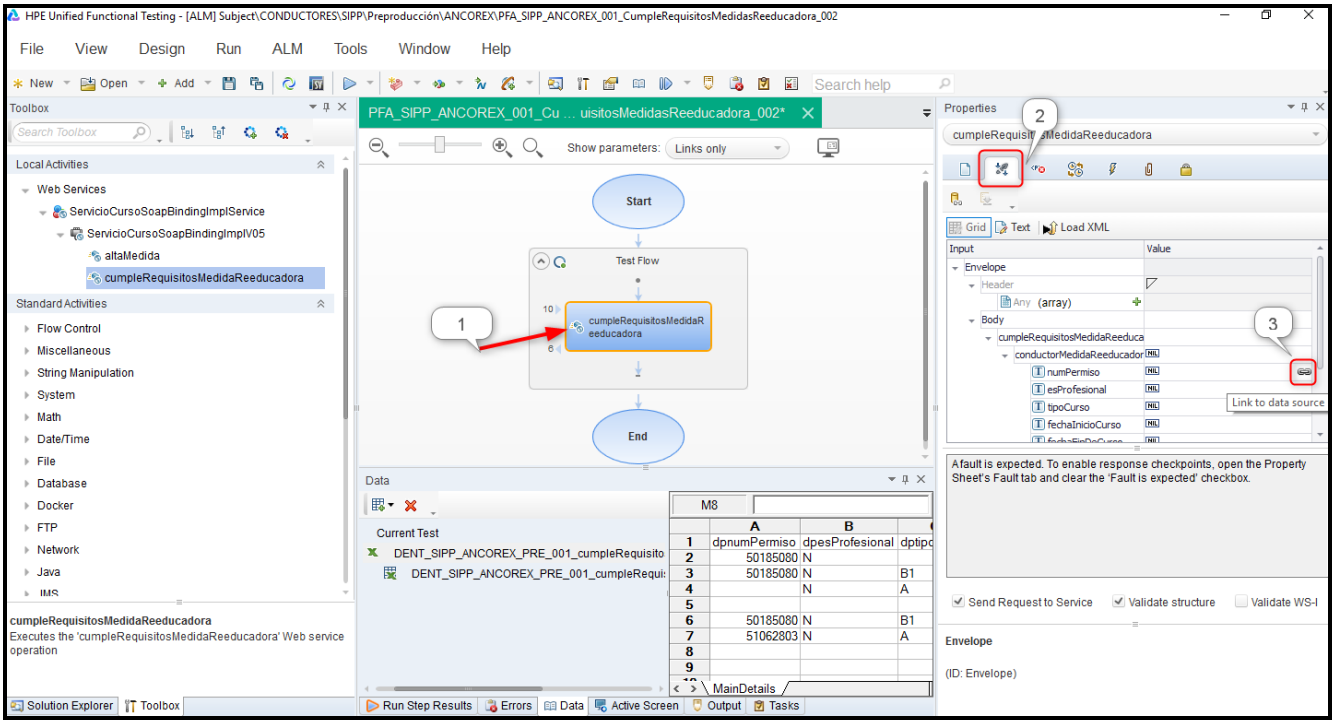


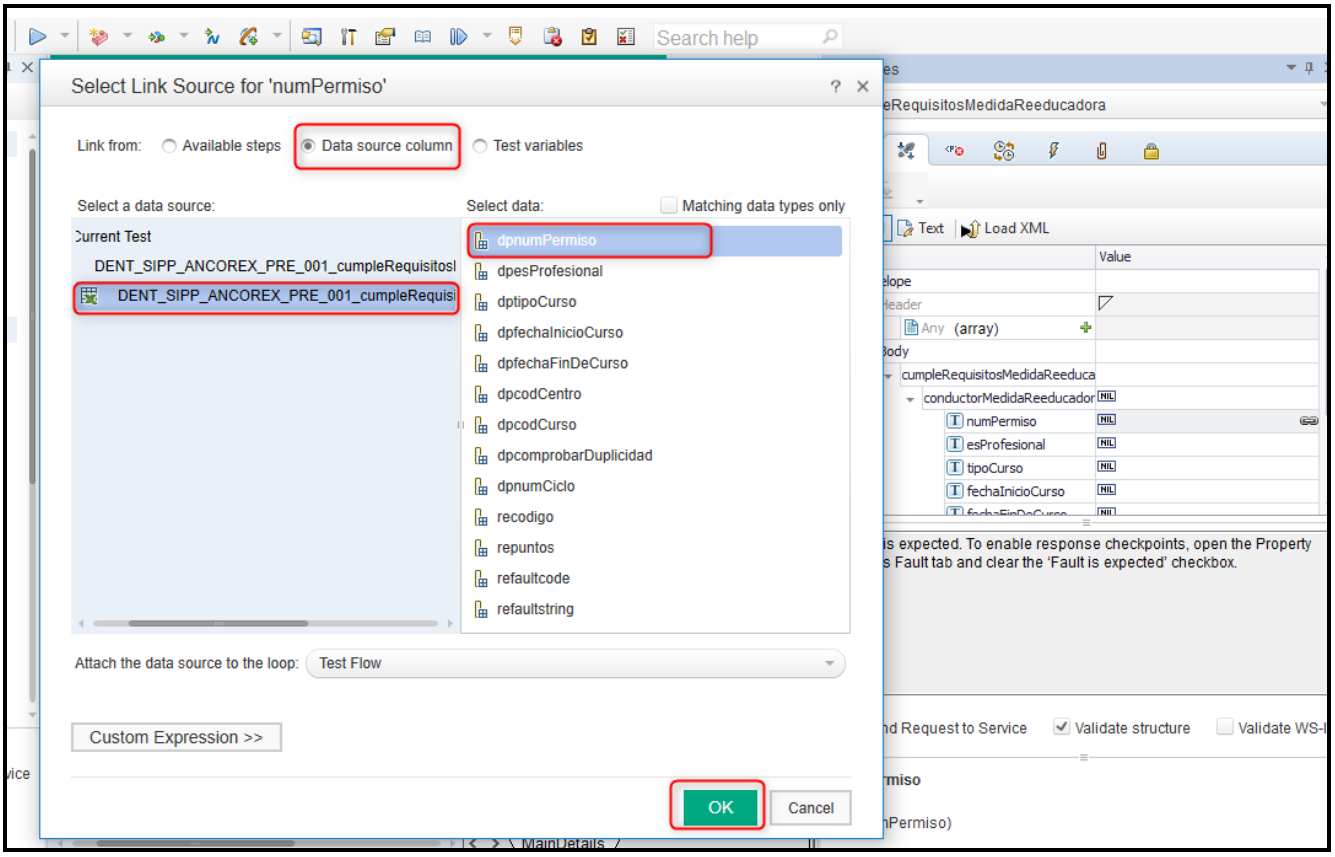
- Seleccionamos el archivo de la ubicación donde lo tengamos y marcamos las opciones "Excel file contains header row", "Make a copy of the Excel file" y "Allow other tools to override the data" y pulsamos en OK para guardar.


PFA_SIPP_ANCOREX_001_Cu ... uisitosMedidasReeducadora_002
Show parameters:
Links only
New Excel Data Source
Data Source name:
DENT_SIPP_ANCOREX_PRE_001_cumpleRe
Excel file path:
\\vmware-host\Shared Folders\Compartida\SIPF
Excel file contains header row
Seleccionamos el archivo
File location:
Make a copy of the Excel file
Link to the Excel file in its original location
Allow other tools to override the data
OK
Cancel
Data

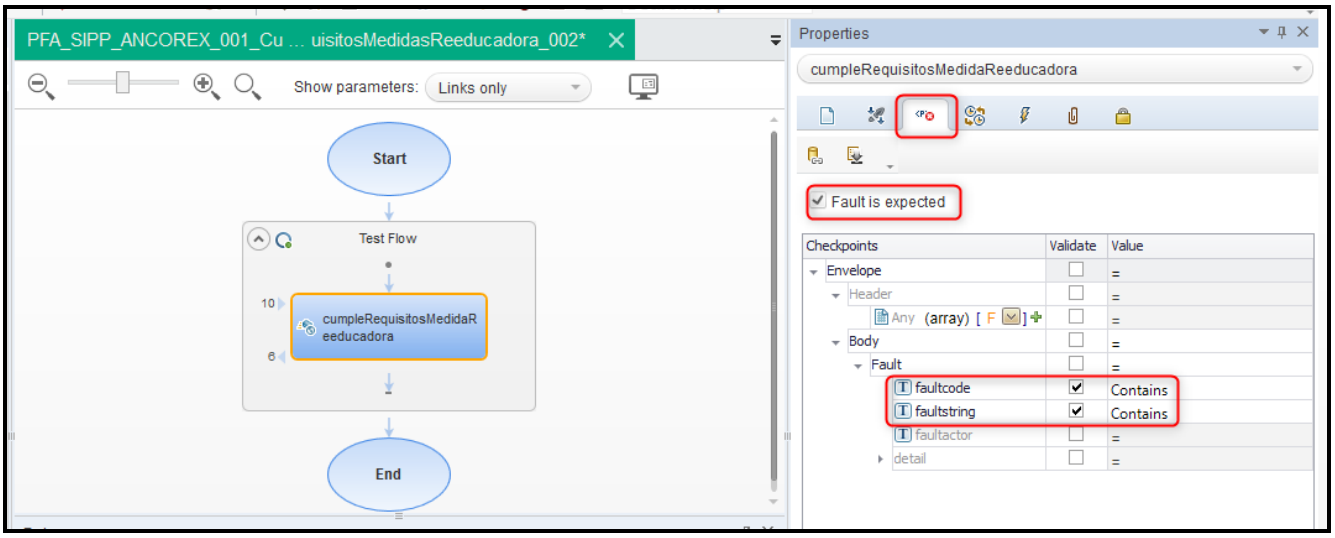


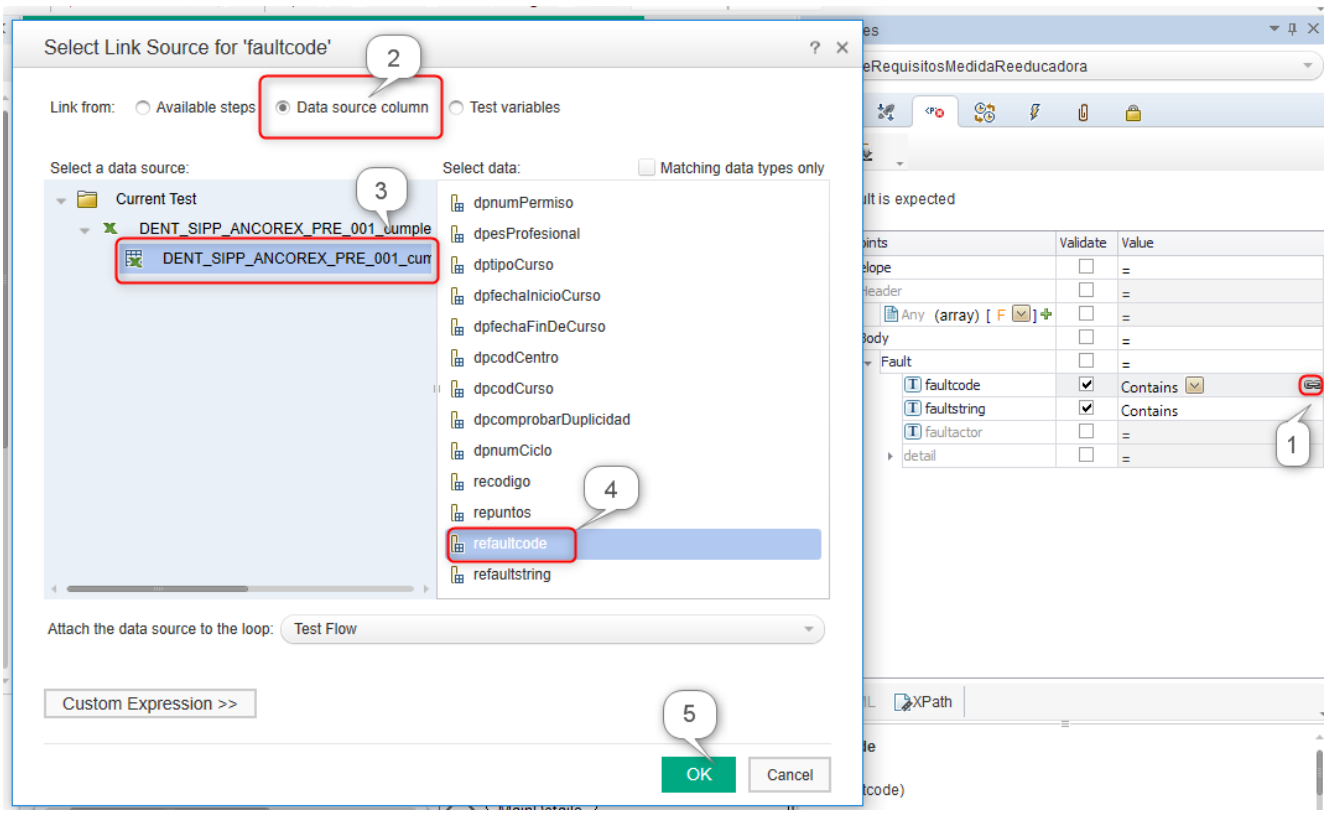
- Ahora solo nos queda seleccionar la fuente de cada parámetro de la hoja Excel. Para ello clicamos en el servicio – Inputs/Checkpoints y vamos asignando la fuente de datos de cada parámetro:


HPE Unified Functional Testing - [ALM] Subject\CONDUCTORES\SIPP\Preproducción\ANCOREX\PFA_SIPP_ANCOREX_001_CumpleRequisitosMedidasReeducadora_002
File View Design Run ALM Tools Window Help
Toolbox
Local Activities
Web Services
ServicioCursoSoapBindingImplService
ServicioCursoSoapBindingImplV05
altaMedida
cumpleRequisitosMedidaReeducadora
Standard Activities
Flow Control
Miscellaneous
String Manipulation
System
Math
Date/Time
File
Database
Docker
FTP
Network
Java
cumpleRequisitosMedidaReeducadora
Executes the 'cumpleRequisitosMedidaReeducadora' Web service operation
PFA_SIPP_ANCOREX_001_Cu ... uisitosMedidasReeducadora_002*
Show parameters: Links only
Start
Test Flow
cumpleRequisitosMedidaReeducadora
End
1
2
3
Properties
Grid
Text
Load XML
Input
Value
Envelope
Header
Any (array)
Body
numPermiso
esProfesional
tipoCurso
fechaInicioCurso
Link to data source
A fault is expected. To enable response checkpoints, open the Property Sheet's Fault tab and clear the 'Fault is expected' checkbox.
Send Request to Service
Validate structure
Validate WS-I
Envelope
(ID: Envelope)
Data
Current Test
DENT_SIPP_ANCOREX_PRE_001_cumpleRequisito
DENT_SIPP_ANCOREX_PRE_001_cumpleRequi
M8
A
B
dpnumPermiso
dpesProfesional
dptipo
50185080
N
B1
A
MainDetails
Solution Explorer
Toolbox
Run Step Results
Errors
Data
Active Screen
Output
Tasks

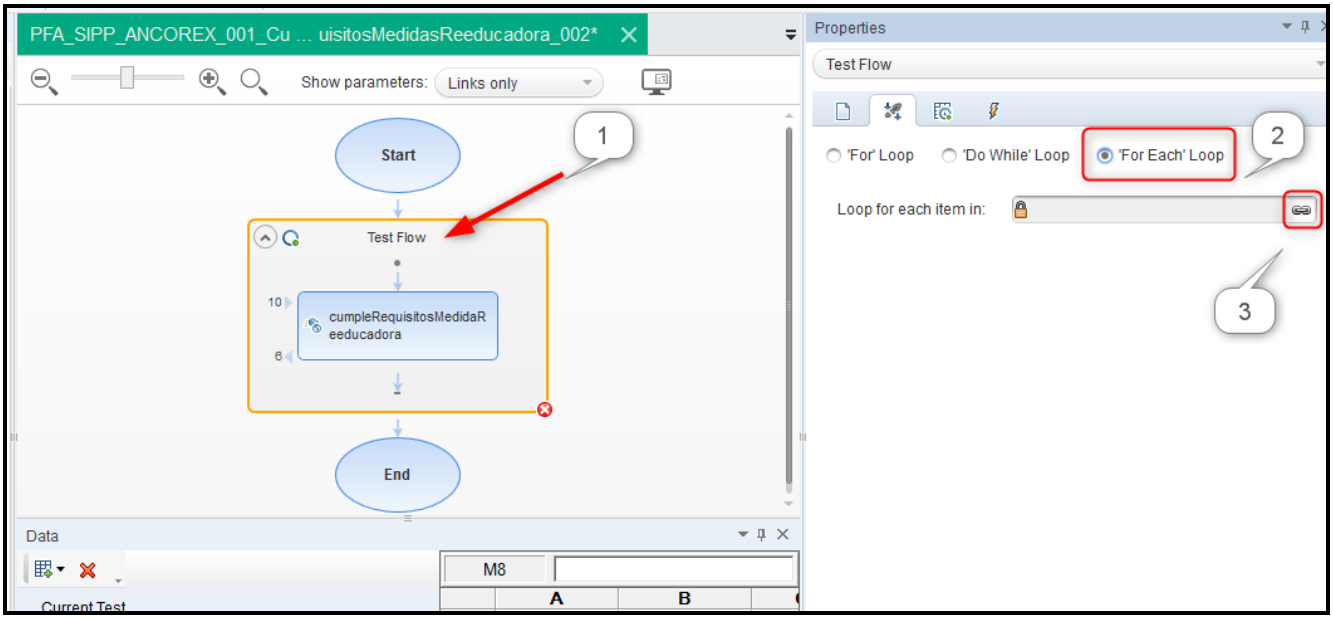

Search help
Select Link Source for 'numPermiso'
Link from:
Available steps
Data source column
Test variables
Select a data source:
Select data:
Matching data types only
Current Test
DENT_SIPP_ANCOREX_PRE_001_cumpleRequisitosI
DENT_SIPP_ANCOREX_PRE_001_cumpleRequisi
dpnumPermiso
dpesProfesional
dptipoCurso
dpfechaInicioCurso
dpfechaFinDeCurso
dpcodCentro
dpcodCurso
dpcomprobarDuplicidad
dpnumCiclo
recodigo
repuntos
refaultcode
refaultstring
Attach the data source to the loop:
Test Flow
Custom Expression >>
OK
Cancel
Text
Load XML
Value
Any (array)
cumpleRequisitosMedidaReeduca
conductorMedidaReeducador
numPermiso
esProfesional
tipoCurso
fechaInicioCurso
Validate structure
Validate WS-I

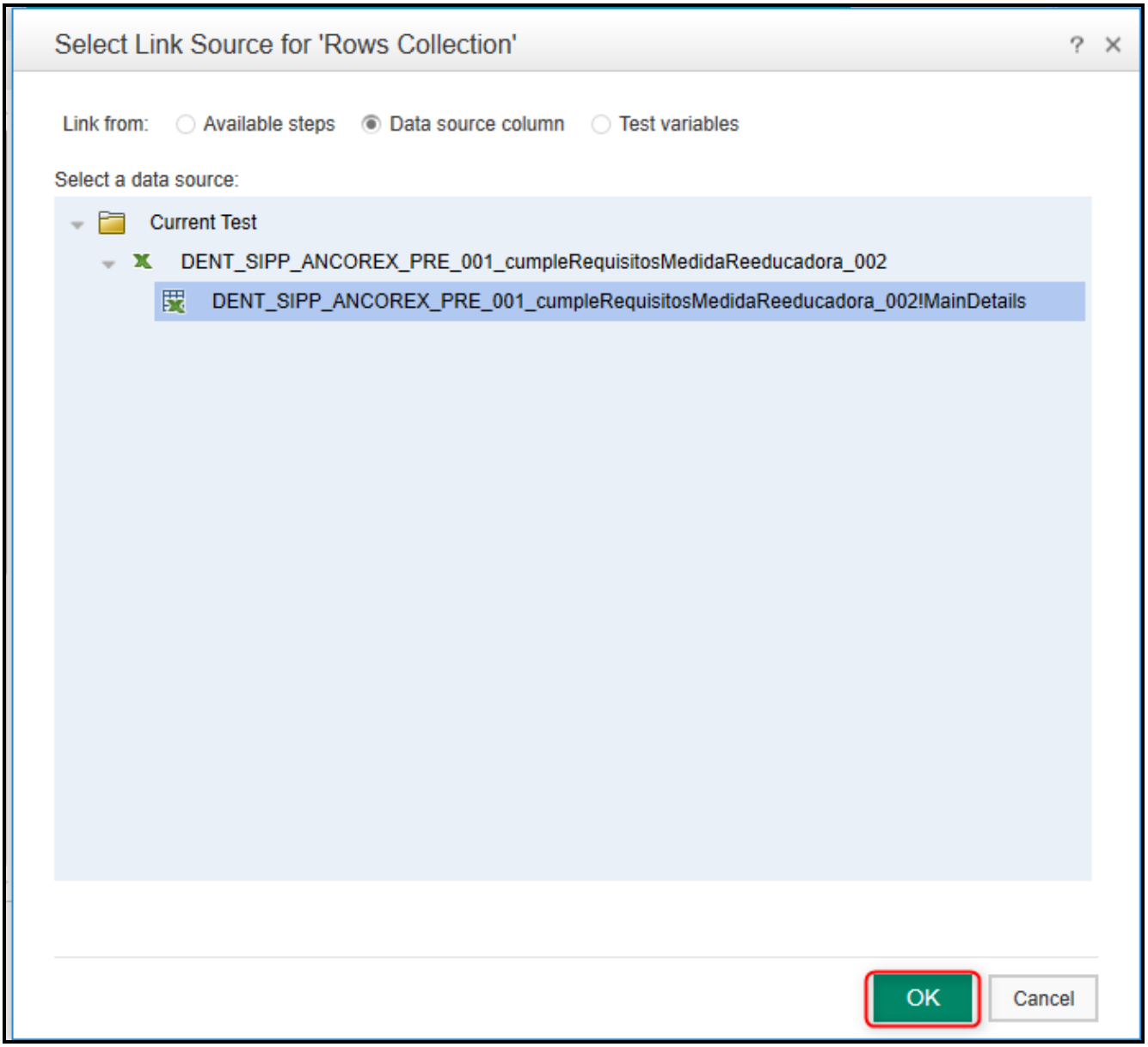
- Una vez tengamos los parámetros de entrada asignados, solo nos queda parametrizar los resultados esperados. En el caso del test que estamos ilustrando se trata de un caso negativo, donde esperamos que haya errores SOAP. Para ello se selecciona la pestaña "SOAP fault" y se marca la casilla "Fault is expected". Además se marcarán los parámetros de fallo que se quieran comprobar, en este caso "faultcode" y "faultstring" en los que comprobaremos que "contengan" el texto pasado como parámetro y que en este caso será el indicado en el Excel en las columnas "refaultcode" y "refaultstring"





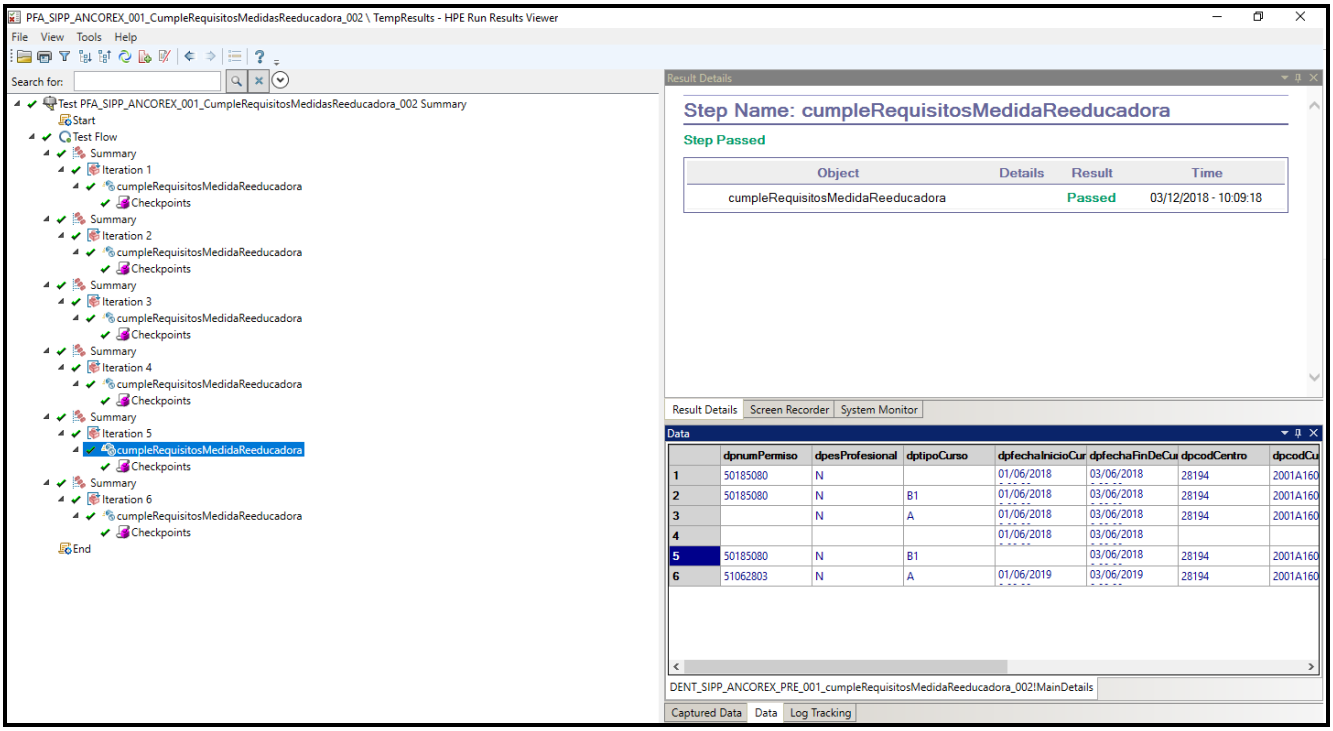
- Al seleccionar una tabla de datos como parámetros de entrada/salida debemos indicar que se use la misma en cada ejecución del loop. Para ello, clicamos sobre “Test Flow”, seleccionamos la opción “’For Each’ Loop” y asignamos nuestra tabla de datos como parámetro.


PFA_SIPP_ANCOREX_001_Cu ... uisitosMedidasReeducadora_002*
Show parameters:
Links only
Start
Test Flow
cumpleRequisitosMedidaReeducadora
End
Data
M8
A
B
Current Test
Properties
Test Flow
'For' Loop
'Do While' Loop
'For Each' Loop
Loop for each item in:
1
2
3



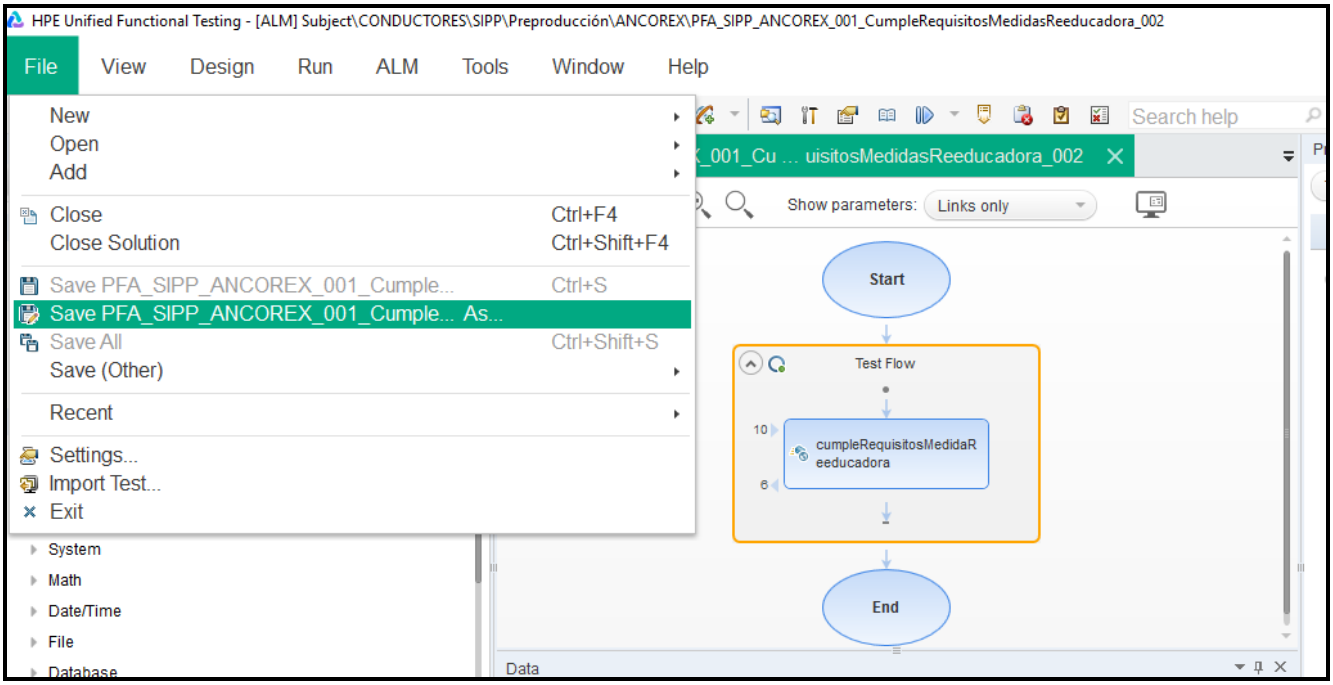
- Una vez que tengamos el caso completo con los steps y los datos de entrada/salida que queramos solo nos quedaría la ejecución completa del test y si está correcta ya solo nos queda guardar el caso y almacenarlo en ALM.

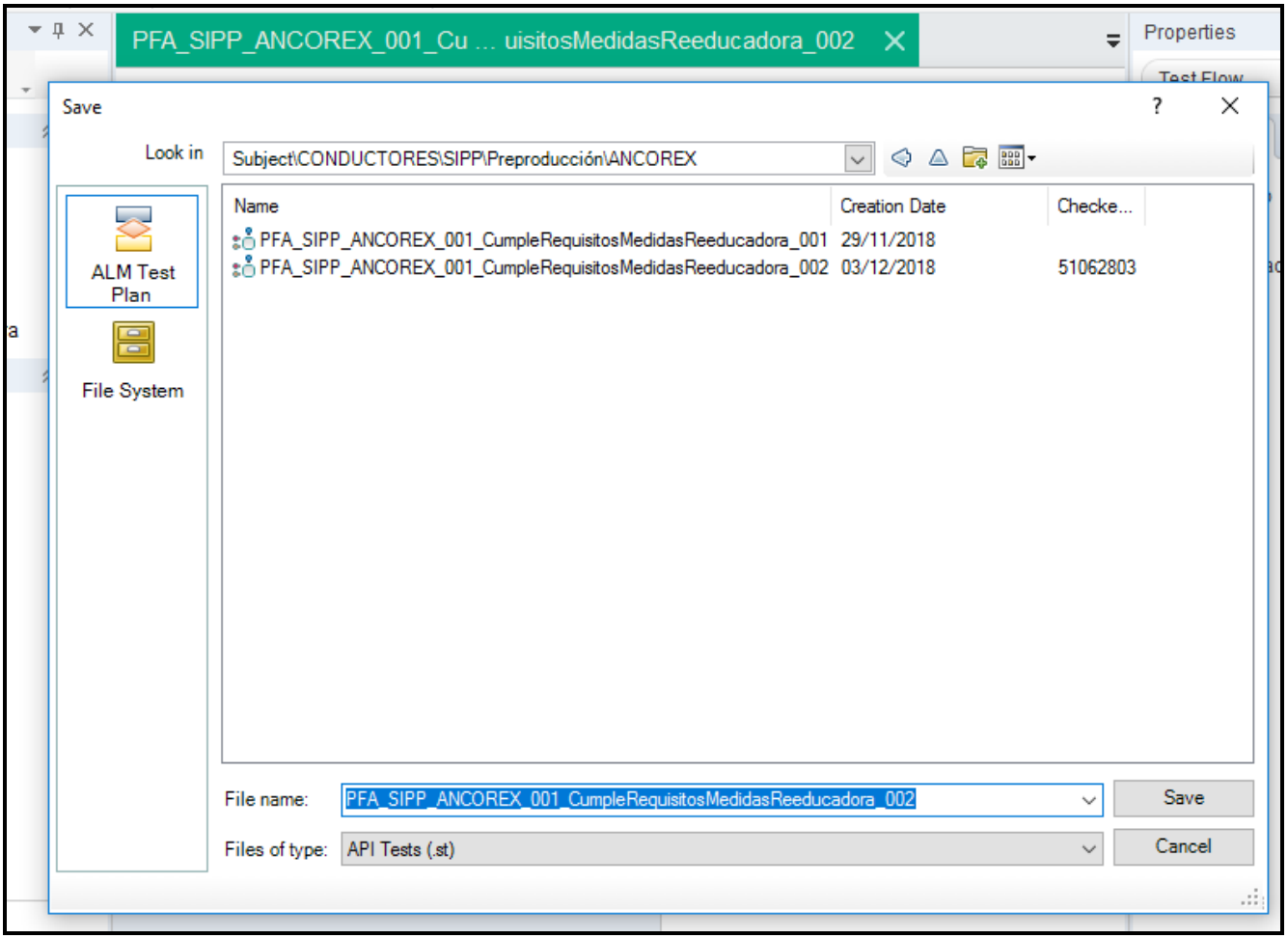
Como teníamos 6 filas de datos en la tabla de datos vemos que la ejecución contiene 6 iteraciones:

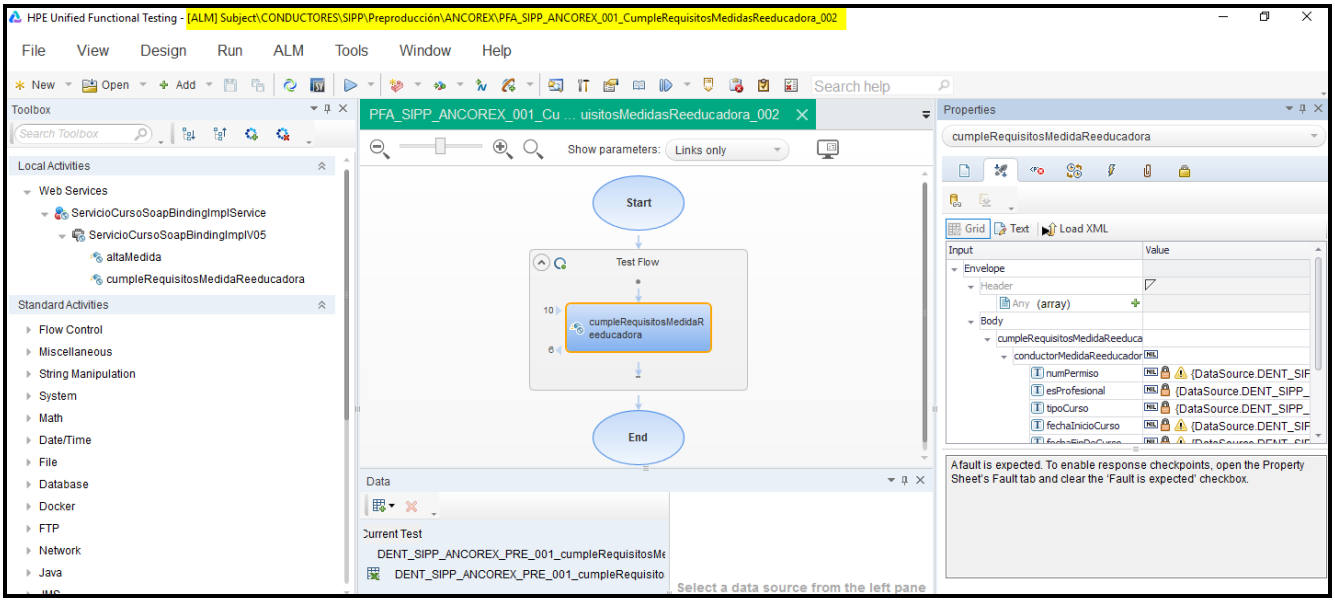


10.2.4 Almacenar el caso en ALM

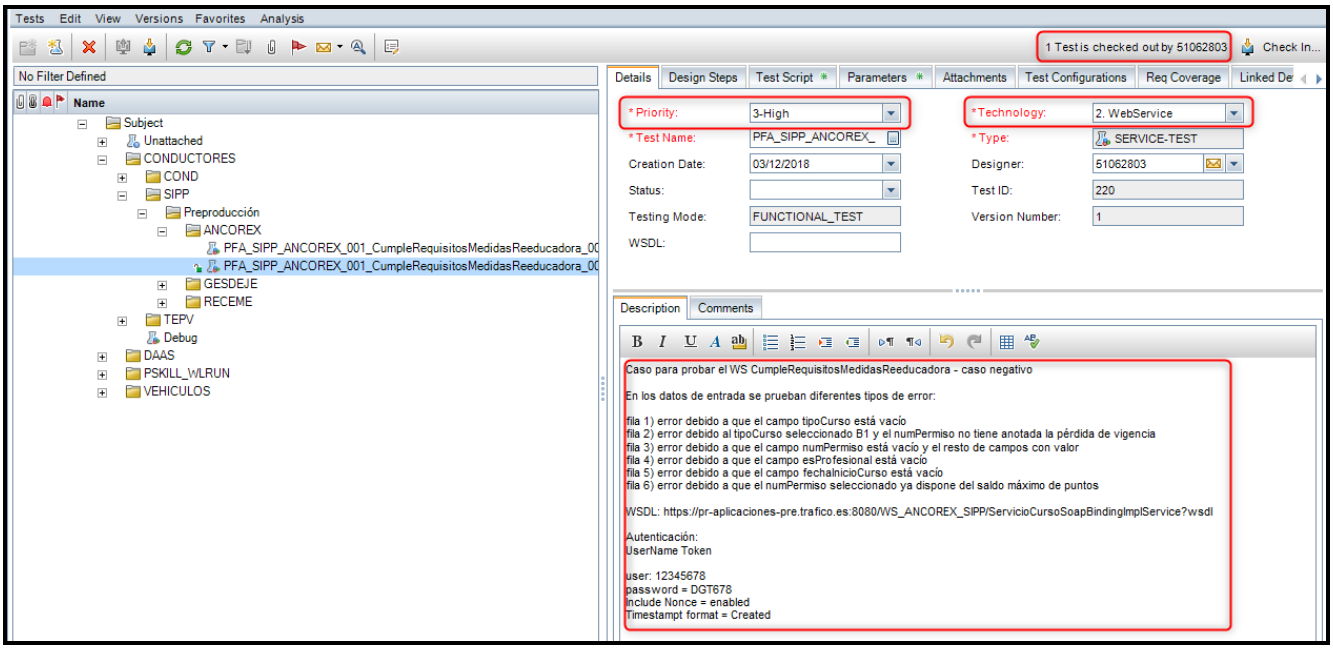
Para almacenar un script en ALM solo tendremos que usar la opción “Save as” y almacenar nuestro caso en la ruta de ALM adecuada.


HPE Unified Functional Testing - [ALM] Subject\CONDUCTORES\SIPP\Preproducción\ANCOREX\PFA_SIPP_ANCOREX_001_CumpleRequisitosMedidasReeducadora_002
File
View
Design
Run
ALM
Tools
Window
Help
New
Open
Add
Close Ctrl+F4
Close Solution Ctrl+Shift+F4
Save PFA_SIPP_ANCOREX_001_Cumple... Ctrl+S
Save PFA_SIPP_ANCOREX_001_Cumple... As...
Save All Ctrl+Shift+S
Save (Other)
Recent
Settings...
Import Test...
Exit
Search help
Show parameters:
Links only
Start
Test Flow
cumpleRequisitosMedidaReeducadora
End
System
Math
Date/Time
File
Database
Data


PFA_SIPP_ANCOREX_001_Cu ... uisitosMedidasReeducadora_002
Properties
Save
Look in
Subject\CONDUCTORES\SIPP\Preproducción\ANCOREX
ALM Test Plan
File System
Name
Creation Date
Checke...
PFA_SIPP_ANCOREX_001_CumpleRequisitosMedidasReeducadora_001 29/11/2018
PFA_SIPP_ANCOREX_001_CumpleRequisitosMedidasReeducadora_002 03/12/2018 51062803
File name:
PFA_SIPP_ANCOREX_001_CumpleRequisitosMedidasReeducadora_002
Save
Files of type:
API Tests (.st)
Cancel


HPE Unified Functional Testing - [ALM] Subject\CONDUCTORES\SIPP\Preproducción\ANCOREX\PFA_SIPP_ANCOREX_001_CumpleRequisitosMedidasReeducadora_002
File View Design Run ALM Tools Window Help
Toolbox
Local Activities
Web Services
ServicioCursoSoapBindingImplService
ServicioCursoSoapBindingImplV05
altaMedida
cumpleRequisitosMedidaReeducadora
Standard Activities
Flow Control
Miscellaneous
String Manipulation
System
Math
Date/Time
File
Database
Docker
FTP
Network
Java
PFA_SIPP_ANCOREX_001_Cu ... uisitosMedidasReeducadora_002
Show parameters: Links only
Start
Test Flow
End
Data
Current Test
Properties
cumpleRequisitosMedidaReeducadora
Grid
Text
Load XML
Input
Value
Envelope
Header
Body
numPermiso
esProfesional
tipoCurso
fechaInicioCurso
A fault is expected. To enable response checkpoints, open the Property Sheet's Fault tab and clear the 'Fault is expected' checkbox.

Aquí vemos el caso almacenado en ALM. En la pestaña Details añadimos la descripción del caso, para ello debemos hacer las modificaciones, por defecto el campo TestName y Type se rellenan automáticamente al guardar desde UFT, pero los campos Priority y Technology no, por lo que habrá que completarlos:



Una vez que tenemos nuestro Test podemos crear los siguientes tests usando la opción “Save as” y modificando los datos, o los Security Settings o las requests de modo que se vayan creando el resto de casos que aplicarían a un mismo WSDL. Renombraríamos cada caso de acuerdo a la nomenclatura especificada y modificaremos los datos y aserciones correspondientes en cada caso.

Será necesario realizar estas acciones para cada operación que queramos automatizar.

11 Ejecución desde ALM

La ejecución de los casos de prueba automáticos se realiza desde el módulo Test Lab de ALM, bajo lo que se denomina Test Set (batería de pruebas). Una batería de pruebas se define como un conjunto de casos de prueba que se ejecutan con un objetivo común.

11.1 Baterías de pruebas

11.1.1 Diseño de las Baterías de Pruebas

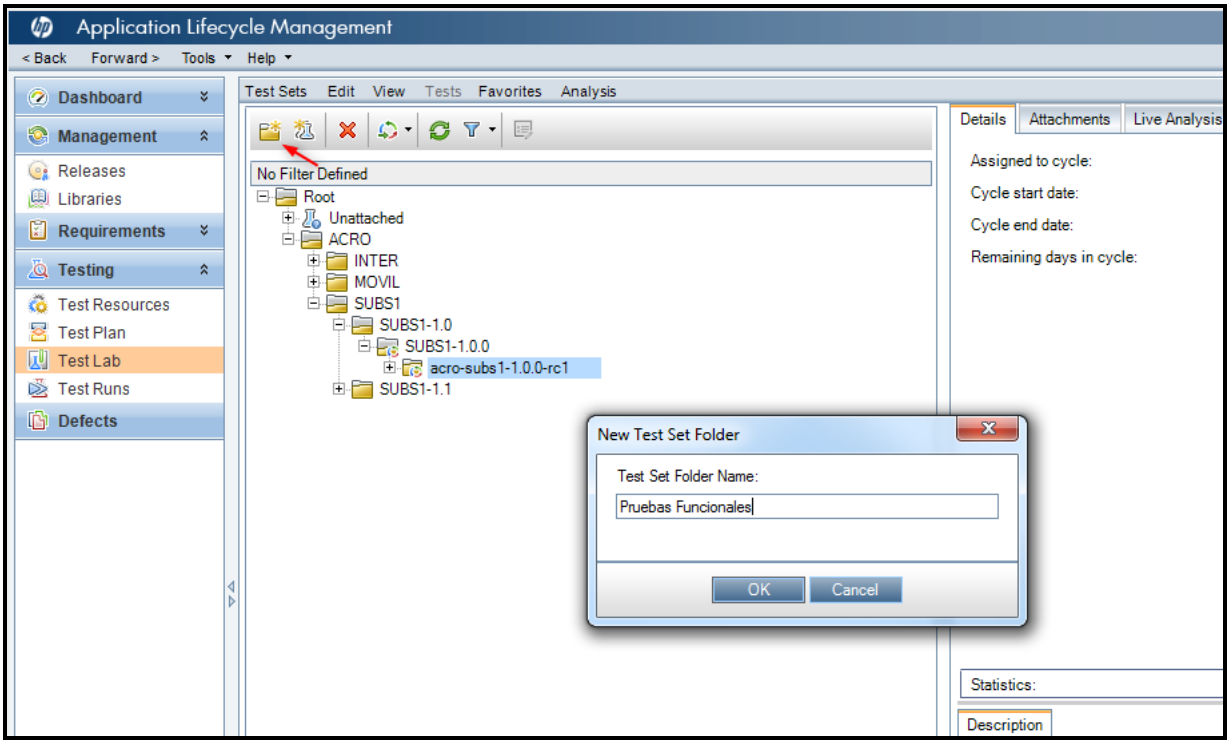
El diseño de las baterías de prueba es propio del aplicativo que se automatiza, así como la organización de las mismas en el Test Lab. A continuación se detallan los pasos necesarios para el diseño genérico de baterías de pruebas.

<u>Creación del árbol de baterías de pruebas:</u>

En el módulo **Test Lab** de ALM, posicionarse sobre una carpeta a la altura de la cual se desea crear una nueva.

Pulsar el botón **New Folder** situado en la esquina superior izquierda del módulo. Aparece la ventana **New Folder**.

En el campo **Folder Name** introducir un nombre de carpeta. Pulsar el botón **OK**.

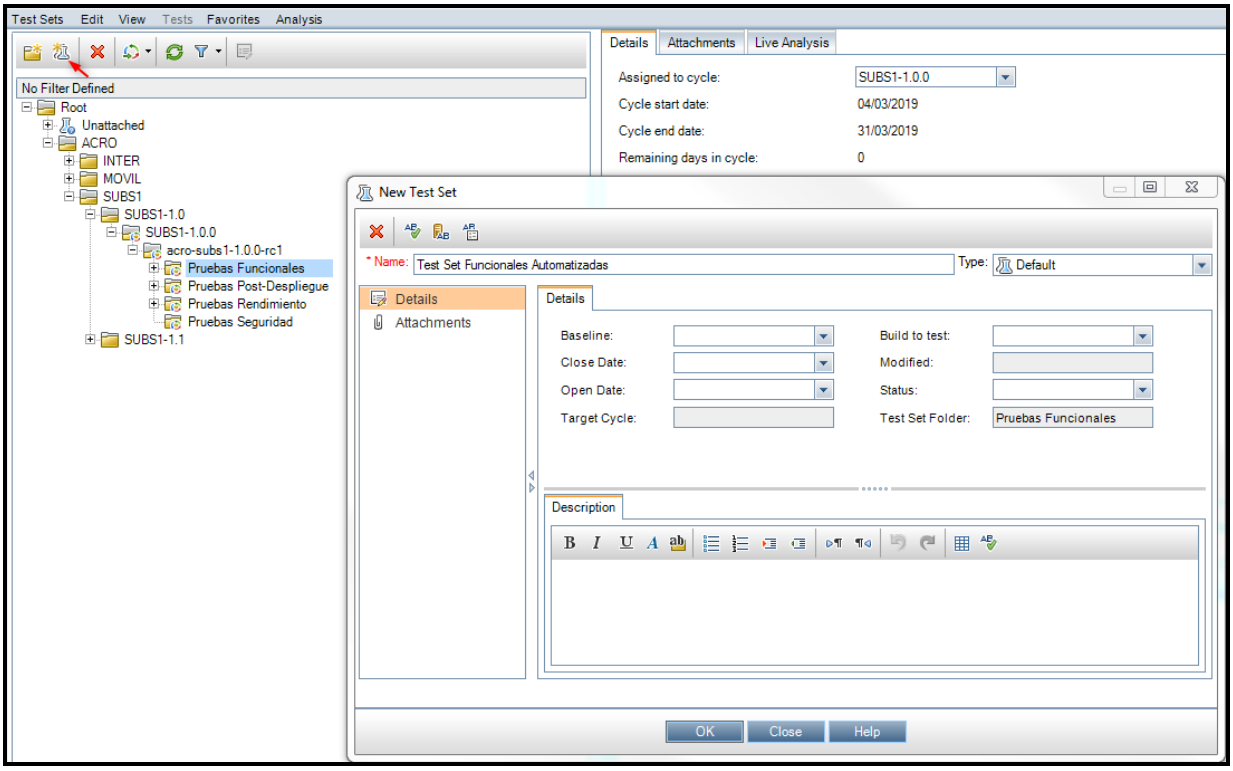


Creación de baterías de pruebas

En el módulo **Test Lab** de ALM, posicionarse sobre una carpeta del árbol en la cual se quiere crear una batería de pruebas.

Pulsar el botón **New Test Set** situado en la esquina superior izquierda del módulo. Aparece la ventana **New Test Set.**

En el campo **Test Set Name** introducir un nombre de batería de pruebas y en el campo **Description** introducir una descripción para la batería. Pulsar el botón **OK**.

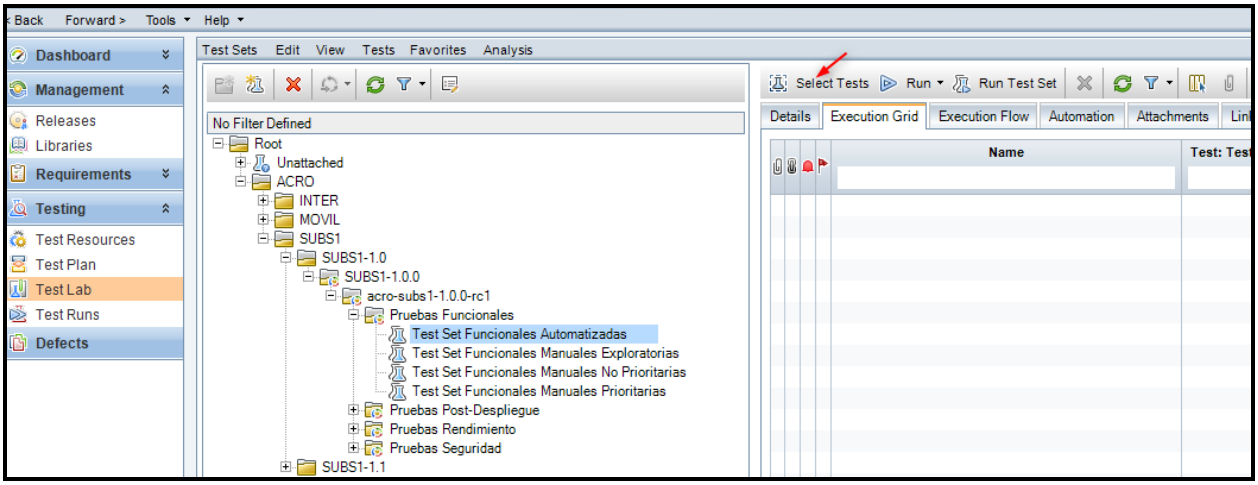


Incorporación de casos de prueba a las baterías

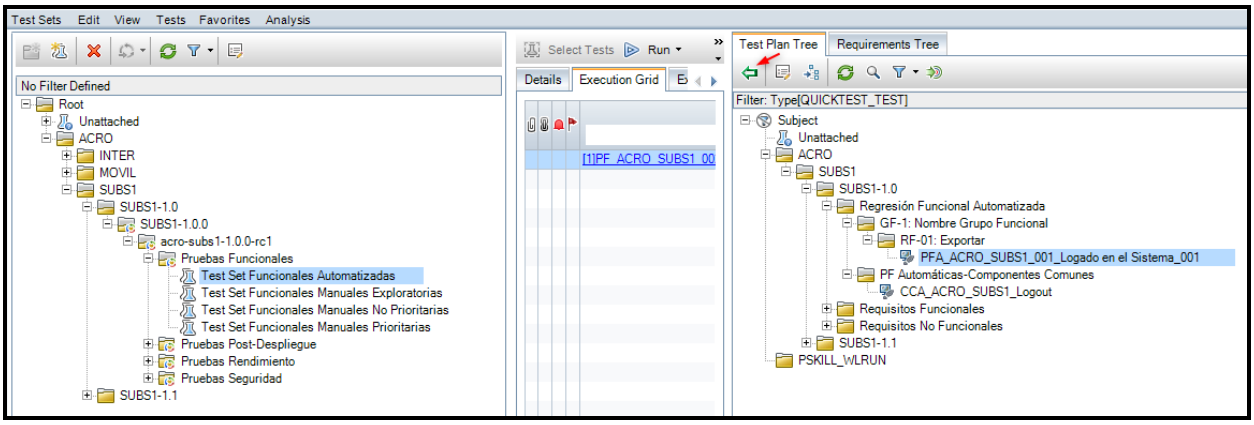
En el módulo **Test Lab** de ALM, posicionarse sobre una batería de pruebas a la cual se quieren incorporar casos de prueba.

En el panel derecho seleccionar la pestaña **Execution Grid**.

Pulsar el botón **Select Tests** que aparece encima de la pestaña **Execution Grid** (si se encuentra deshabilitado significa que ya está pulsado). A la derecha aparece un panel nuevo.



- En el nuevo panel seleccionar la pestaña **Test Plan Tree**.
- En el árbol seleccionar el caso de prueba que se desea incorporar a la batería. Pulsar el botón **Add Tests to Test Set** (botón de flecha verde a la izquierda). En el **Execution Grid** aparece el caso de prueba seleccionado.



- Repetir los pasos anteriores por cada una de las pruebas automáticas que se desean incorporar a la batería.

11.1.2 Ejecución de las Baterías de Pruebas Automáticas

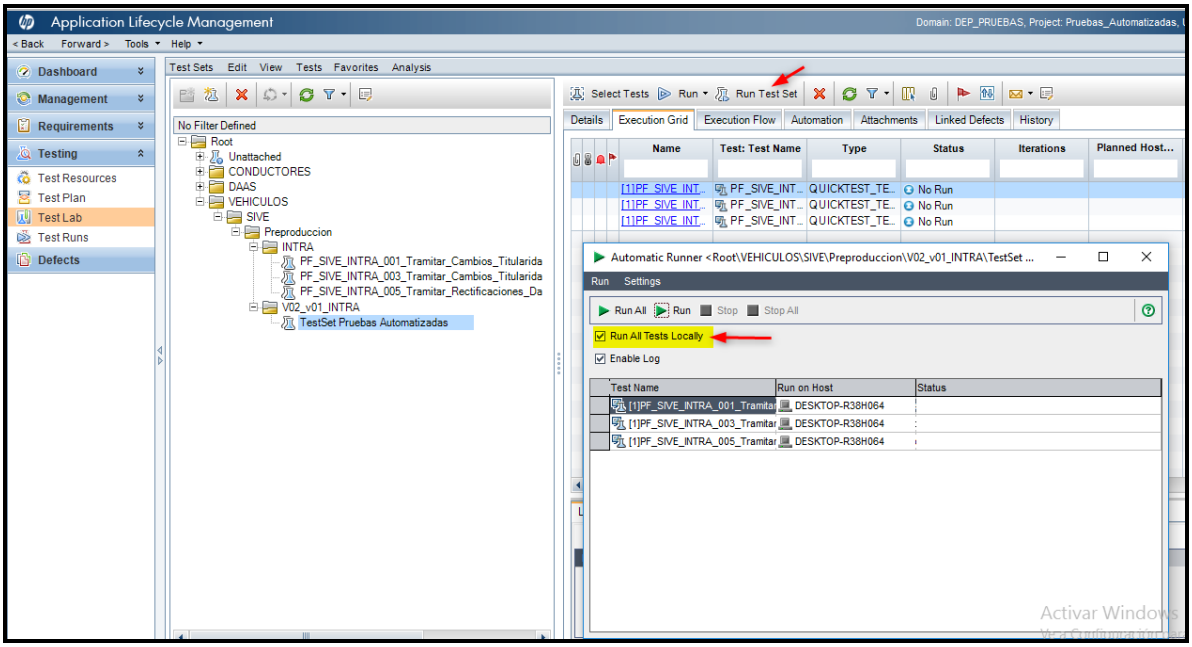
Los pasos necesarios para la ejecución de una batería de pruebas son los siguientes:

En el módulo **Test Lab** de ALM, posicionarse sobre una batería de pruebas.

Pulsar el botón **Run Test Set**. Aparece la ventana **Automatic Runner.**

En la columna **Run on Host** hay que indicarle la máquina en la cual se quieren ejecutar las pruebas automáticas. Existen dos posibilidades:

- Ejecutar en la máquina local. Seleccionar el check **Run All Tests Locally**. Todas las pruebas automáticas se ejecutarán en la máquina local en la cual se encuentra el cliente de ALM.



- Ejecutar en una máquina remota. En la columna **Run on Host**, seleccionar el botón de los tres puntos, aparece la ventana **Select Host**. Seleccionar una máquina que tenga UFT instalado. Las pruebas automáticas se ejecutarán en la máquina remota seleccionada.

Pulsar el botón **Run All**. ALM se conectará mediante un agente remoto a UFT, abrirá cada uno de los test automáticos de la batería y los ejecutará devolviendo los resultados de ejecución a ALM.

11.1.3 Ejecución de Baterías de Pruebas de Forma Desatendida

Se puede planificar la ejecución de scripts de forma desatendida en una fecha y hora específicas, siempre que se den estas condiciones:

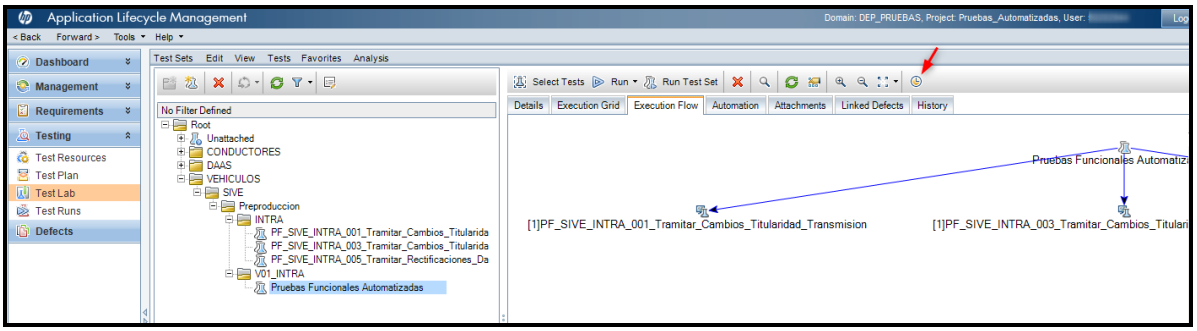
- La sesión del PC de UFT no esté bloqueada y ni se bloquee durante la ejecución.
- ALM permanezca abierto.
- UFT esté cerrado, o bien, pueda cerrarse sin que aparezca ningún mensaje pop-up para pedir confirmación de guardar cambios, o similares, porque estos mensajes dejarán bloqueada la herramienta UFT.

De esta manera no consumimos recursos de tiempo y licencias para la ejecución durante las horas de trabajo y se pueden ejecutar los casos durante la noche.

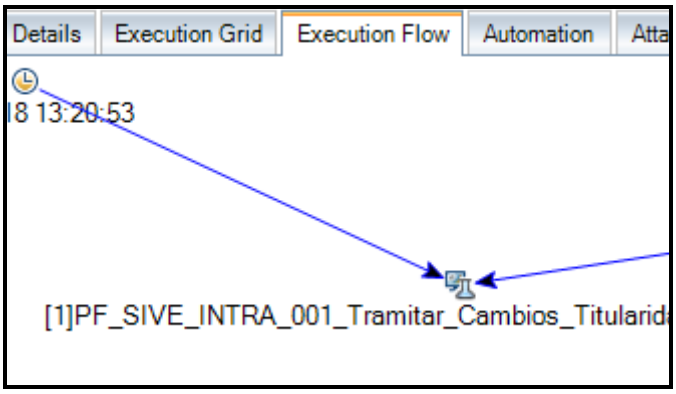
Para poder realizar la planificación de una ejecución, se han de seguir los siguientes pasos:

En el módulo "Test Lab", seleccionar el Test Set donde están los scripts a ejecutar, y seleccionar la solapa "Execution Flow", para ver el diagrama del flujo de la ejecución.

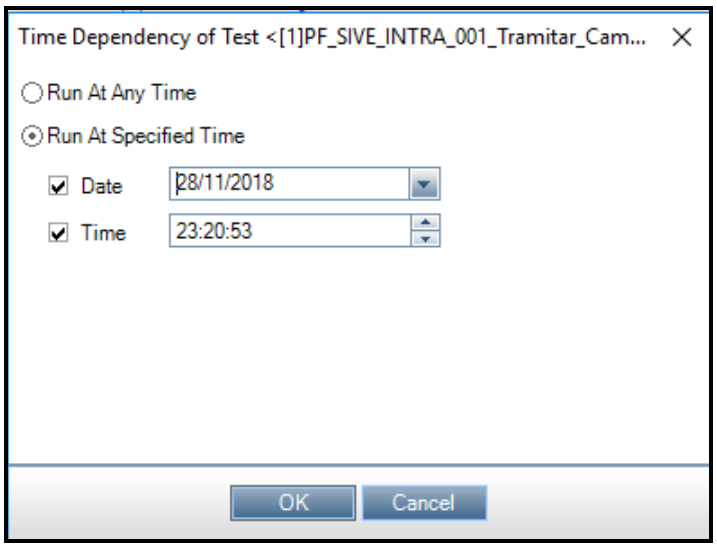
Añadir una dependencia temporal pulsando sobre el reloj "Add Time Dependency to Flow"



Arrastrar el puntero del ratón desde ese reloj hasta el primer script del flujo a planificar. Aparece una flecha que los engancha.



Una vez enganchados, hacer doble clic sobre el reloj. Aparece el cuadro de texto para especificar tanto la fecha como la hora a la que se lanzará la ejecución del script de forma desatendida. (Es posible poner más de una dependencia temporal.)



Una vez que tenemos establecida la fecha y hora, pulsamos el botón “Run Test Set” para que aparezca la ventana del “Automatic Runner”.

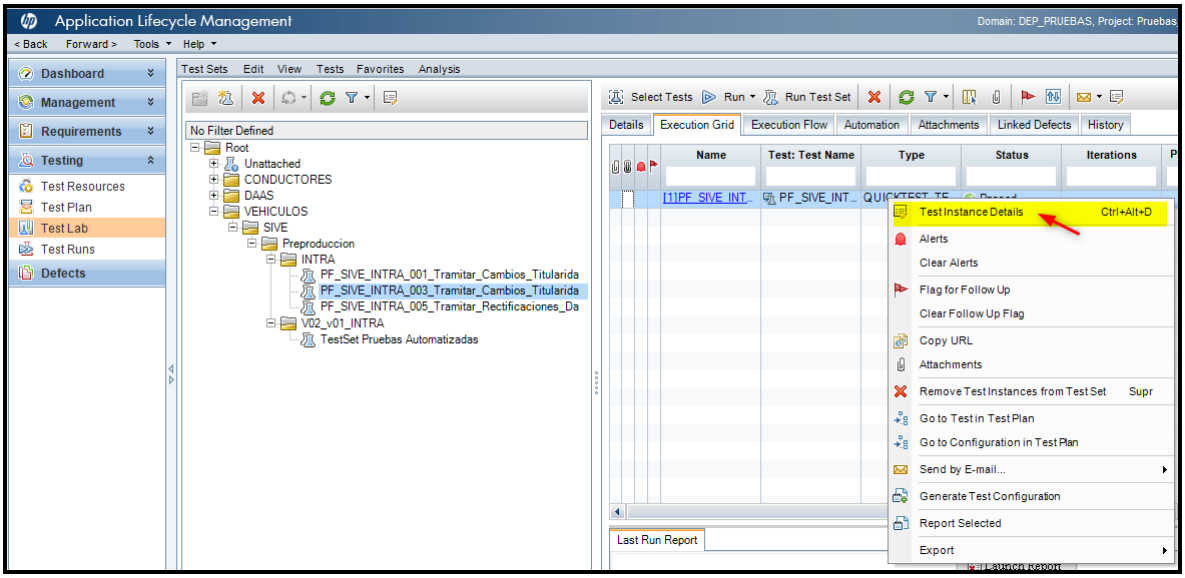
Pulsamos el botón “Run All” y esperamos que todos los tests se pongan en estado Waiting. Esta ventana se puede minimizar y permite seguir trabajando con ALM.

11.1.4 Análisis de resultados de ejecución

Después de la ejecución de una batería de pruebas se deben analizar los informes de resultados donde podrán aparecer los defectos encontrados en el aplicativo bajo prueba.

Cada ejecución de un script genera un informe de resultados que se adjunta a dicha ejecución. Los pasos para visualizar dicho informe son:

- En el módulo **Test Lab** de ALM, posicionarse sobre una batería de pruebas.
- En la pestaña **Execution Grid** seleccionar el test automático que se quiere consultar.
- Pulsar el botón derecho y seleccionar **Test Instance Details**.

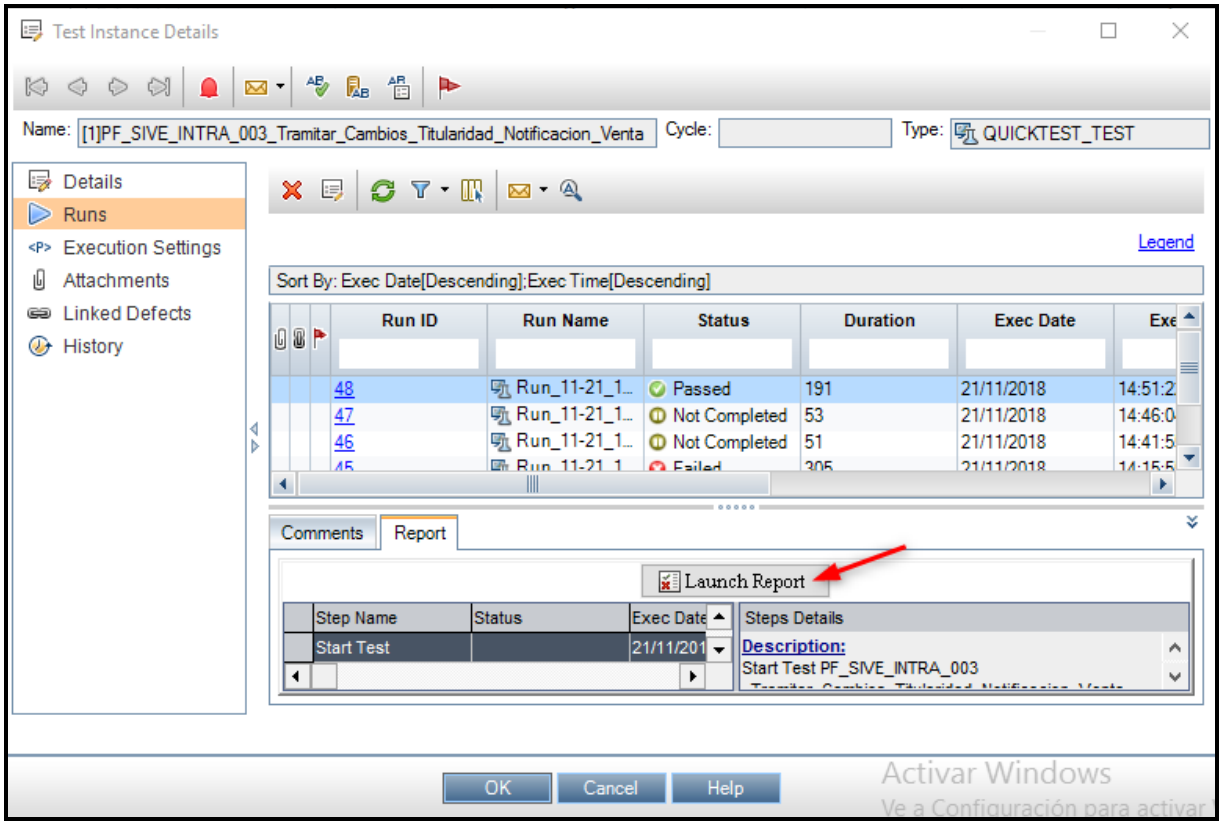


- Aparece la ventana **Test Instante Details**. Seleccionar **Runs**. Se presentará en la ventana el histórico de ejecuciones de este test automático.
- Seleccionar la ejecución que se desea analizar.

A partir de este punto hay pequeñas diferencias en el caso de analizar prueba GUI y pruebas API que detallamos a continuación:

11.1.4.1 Pruebas GUI

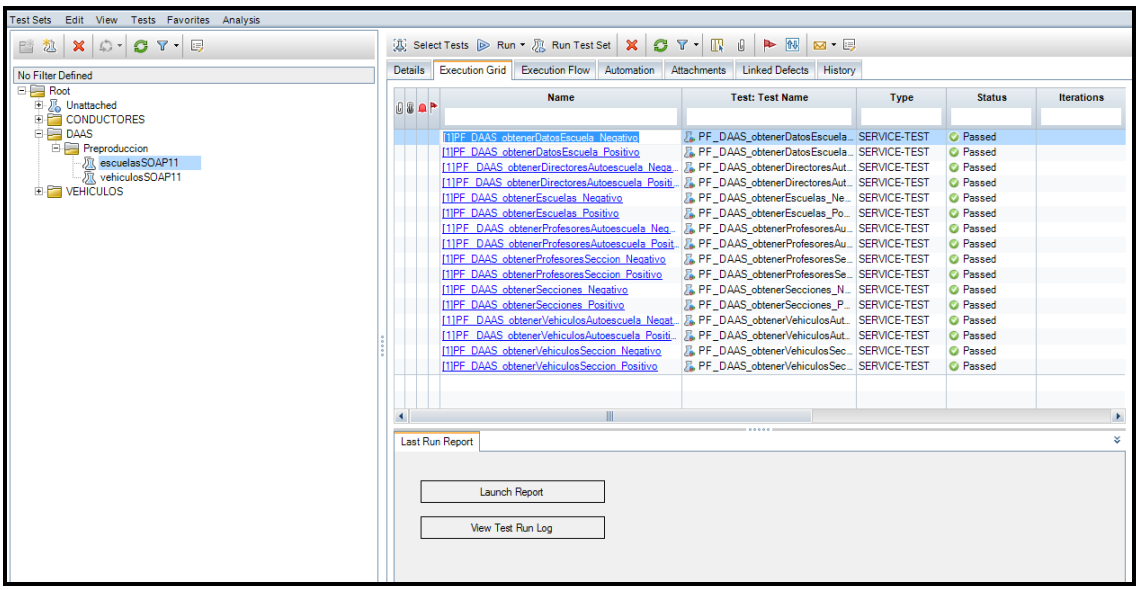
- Pulsar el botón **Launch Report** que aparece en la parte inferior de la ventana. Aparece el informe de resultados de la ejecución seleccionada.



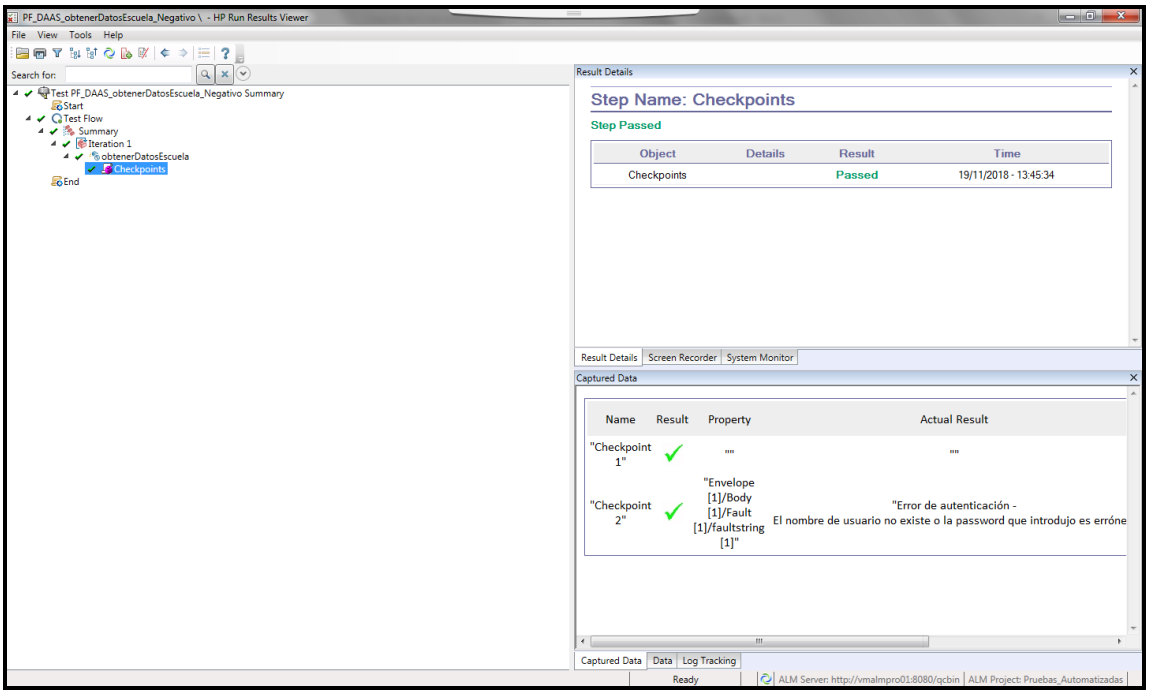
El informe de resultados contiene información de la ejecución de cada uno de los pasos que componen el script automático, informando de si ha sido correcto (icono de v verde) o ha fallado (icono de aspa roja). Para localizar los posibles errores se debe expandir el árbol de resultados y seleccionar el paso que ha fallado, dicho paso mostrará dos tipos de información, un pantallazo de la aplicación bajo prueba en el momento en el cual se detectó el error y un detalle que indica la causa del fallo. Con esta información y la comprobación manual sobre la aplicación bajo prueba se detecta la existencia o no del error.

11.1.4.2 Pruebas API

En la pestaña “Last Run Report” hay 2 botones para acceder al Reporte (Launch Report) y para ver el log de la ejecución (View Test Run log).



Al pulsar en el primer botón se nos abre el reporte del caso de UFT que contiene la petición enviada así como el estado de los checkpoints, y un resumen de la ejecución.



Al pulsar en el segundo botón se nos muestra el log de la ejecución.

vtd_user: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
2018-11-19 13:45:33,979 [13] INFO  - Step 'Test Flow' : Step 'Test Flow' started
2018-11-19 13:45:34,003 [13] INFO  - Step 'Iteration 1' : Step 'Iteration 1' started
2018-11-19 13:45:34,003 [13] INFO  - Step 'obtenerDatosEscuela' : Step 'obtenerDatosEscuela' started
2018-11-19 13:45:34,010 [13] INFO  -
Request:

<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/">
  <Body>
    <obtenerDatosEscuela xmlns="http://trafico.es/servicios/daas/escuelas">
      <codigoEscuela>GC0238</codigoEscuela>
    </obtenerDatosEscuela>
  </Body>
</Envelope>

2018-11-19 13:45:34,307 [13] INFO  - URL: https://preapl-p3.trafico.es:8080/WS_DAAS/services/escuelas
2018-11-19 13:45:34,409 [13] INFO  - Returning the response to the previous channel
2018-11-19 13:45:34,409 [13] INFO  -

Response with security:

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:date="http://exslt.org/dates-and-times" xmlns:dgt="http://www.datapower.com/extensions/functions" xmlns:xsi="http://www.w3.org/2001/XMLSche
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>soapenv:0009</faultcode>
      <faultstring>Error de autenticación - El nombre de usuario no existe o la password que introdujo es errónea</faultstring>
      <detail>Rejected by policy.</detail>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>


2018-11-19 13:45:34,435 [13] INFO  -
Response:

<soapenv:Envelope xmlns:date="http://exslt.org/dates-and-times" xmlns:dgt="http://www.datapower.com/extensions/functions" xmlns:xsi="http://www.w3.org/2001/XMLSche
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>soapenv:0009</faultcode>
      <faultstring>Error de autenticación - El nombre de usuario no existe o la password que introdujo es errónea</faultstring>
      <detail>Rejected by policy.</detail>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>

2018-11-19 13:45:34,463 [13] INFO  - Structural Validation: True
2018-11-19 13:45:34,479 [13] INFO  - Checkpoint 1: "Error de autenticación - El nombre de usuario no existe o la password que introdujo es errónea" Contains "El nc
2018-11-19 13:45:34,494 [13] INFO  - Step 'obtenerDatosEscuela' ended successfully
```

11.2 Problemáticas en la Ejecución de Baterías de Pruebas (Test Sets)

En este apartado se explican algunos problemas asociados a la ejecución de las baterías de pruebas que se solucionan con una construcción adecuada de los scripts.

Para la correcta ejecución secuencial de los scripts dentro de una batería de pruebas es muy importante que se haya definido el mismo punto de Inicio-Fin de la prueba para todos los scripts de la batería.

Por ejemplo: En una aplicación Web se puede definir que el punto de Inicio-Fin sea la aplicación cerrada (implica el navegador cerrado). Otro punto de Inicio-Fin podría ser la página inmediatamente después del login.

11.2.1 Construcción de Scripts Independientes

La construcción de los scripts automáticos con UFT debe ser lo más independiente posible, de forma que no tengan dependencias en su ejecución.

No siempre es posible evitar estas dependencias, pero se debe hacer un esfuerzo por evitarlas.

Si se construyen scripts que van a tener dependencias, es decir, que van a tener que seguir un orden establecido en las baterías de pruebas se corre el riesgo que ante un fallo de uno de los scripts los siguientes no puedan ser ejecutados o también finalicen con fallos.

Por el contrario, si todos los scripts de la batería de pruebas son independientes, el orden de ejecución no es importante dentro de la batería y un fallo en la ejecución de uno de ellos no afecta al resto.

11.2.2 Errores no controlados

A la hora de ejecutar las baterías de pruebas (Test Sets) en ALM se pueden producir errores no controlados durante la ejecución de un script que provoquen que los siguientes scripts de la batería de pruebas no puedan ejecutarse por no quedar la aplicación en la posición de Inicio-Fin esperada.

Por ejemplo: Si en una aplicación Web se ha definido que la posición de Inicio-Fin sea la aplicación cerrada (es decir ningún Navegador abierto) se debe controlar al inicio de ejecución de cada script que se cumple con esta posición Inicio-Fin y de no ser así realizar las acciones necesarias para conseguirlo (para este ejemplo cerrar todos los navegadores abiertos).