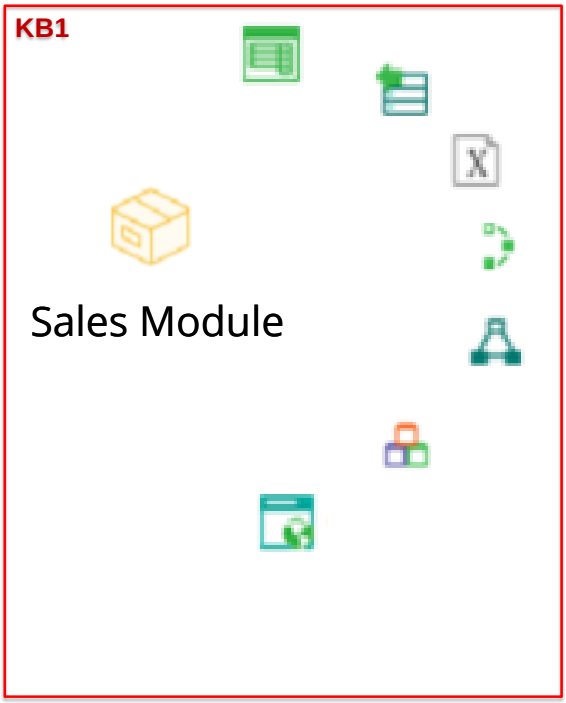
Módulos y Objetos Externos

GeneXus™

A continuación veremos algunos objetos GeneXus que son útiles a la hora de encapsular funcionalidad y organizar los objetos de nuestra base de conocimiento.

Módulos

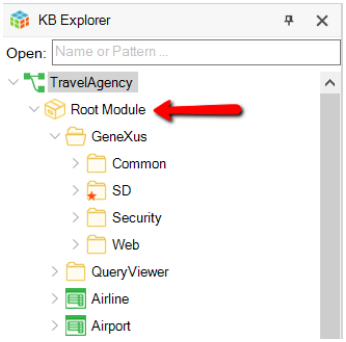
Veamos en primer lugar los módulos.

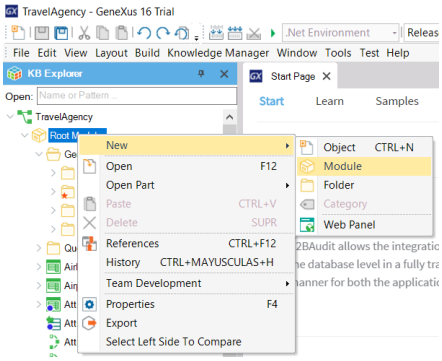


Los módulos son objetos GeneXus contenedores, que nos permiten agrupar objetos de nuestra KB facilitando el entendimiento de la misma, su mantenimiento y la integración de objetos con otras KBs.

¿Qué es un módulo?

- Objeto GeneXus que nos permite agrupar objetos de la KB y encapsular sus funcionalidades
- Diseñado para facilitar el entendimiento, mantenimiento e integración de objetos entre KBs
- Los módulos y los folders nos permiten crear jerarquías





Cuando creamos una base de conocimiento, se crea el módulo Root Module y por defecto todos los objetos que creamos quedan en ese módulo.

Tanto los módulos como los folder nos ayudan a organizar objetos, sin embargo hay diferencias conceptuales entre un módulo y un folder. Los módulos nos ayudan a encapsular y modularizar partes de la KB, pudiendo establecer qué objetos son visibles desde otros objetos y cuáles no, como veremos más adelante.

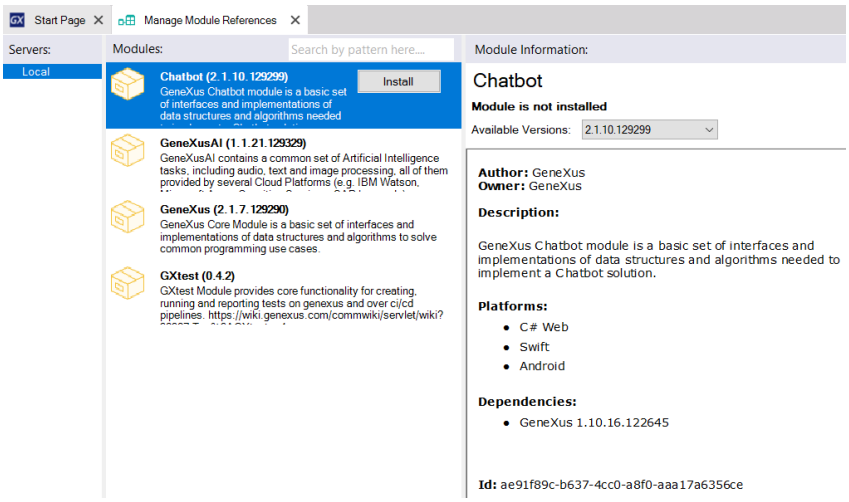
Los folders en cambio, son contenedores que sólo nos ayudan a organizar objetos , separándolos según algún criterio. Junto con los módulos crean un árbol jerárquico donde la raíz es siempre es un módulo como en el caso del Root Module. Esto lo podemos apreciar en el KB Explorer.

Los módulos pueden tener folder como hijos pero los folders no pueden tener módulos como hijos.
Como regla general podemos decir que se pueden usar los módulos para encapsular y los folders para organizar los objetos dentro del módulo.

Para agregar un objeto a un módulo, podemos arrastrarlo dentro del KB Explorer hacia el módulo, o dar botón derecho sobre el módulo y luego New Object, o cambiar el valor de la propiedad Module/Folder del objeto.

Agregando módulos ya creados a la KB

- Knowledge Manager / Manage Module References



Los módulos que son empaquetados y que fueron compartidos con nosotros, los vemos a través del menú Knowledge Manager / Manage Module References.

Para cada módulo disponible, podemos ver su información y decidir si lo instalamos o no en nuestra KB. Si lo instalamos quedará guardado bajo el nodo References del KB Explorer, a diferencia de los objetos creados por nosotros, que por defecto quedan guardados en el Root Module.

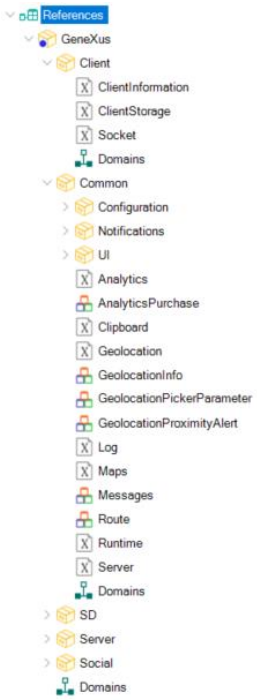
Los objetos de estos módulos no los podremos modificar (los veremos como Read-Only) y ya están compilados, por lo que cuando damos F5 no será necesario especificarlos, generarlos, etc.

Sin embargo, podremos utilizarlos libremente desde los objetos de nuestra aplicación, utilizando las funcionalidades que ofrecen.

Cualquier integrante de la comunidad puede crear, compartir o incluso vender sus módulos a través del Marketplace.

Uno de estos módulos es el módulo GeneXus, también conocido como GeneXus Core.

GeneXus Module



El módulo GeneXus es distribuido automáticamente e instalado en todas las KBs y como todo módulo externo a nuestra aplicación, lo encontramos en el nodo References.

Está compuesto por una serie de sub-módulos que contienen un conjunto de APIs con sus respectivos dominios y SDTs, que nos permiten interactuar con distintas tecnologías, dispositivos, sensores, aplicaciones, etc.

Estas API's son implementadas como Objetos Externos, que veremos a continuación.

Más información sobre módulos

https://wiki.genexus.com/commwiki/servlet/wiki?22411

Por más información sobre el objeto módulo, puede seguir el siguiente link del Wiki: https://wiki.genexus.com/commwiki/servlet/wiki?22411

Objetos Externos

Veamos ahora qué son los Objetos Externos.

Los objetos externos son objetos GeneXus que nos permiten acceder a recursos externos a nuestra KB, como si fueran un objeto más de la KB.

Por esa razón, son cada vez más frecuentes e importantes en nuestras aplicaciones tanto webs como para dispositivos móviles. Veamos cómo usarlos.

¿Qué recursos podemos utilizar como objetos externos en nuestra KB?

- Native objects from programming languages:
  - .NET Assemblies (.dll)
  - Java Classes (.class)

- Resources from several external sources:
  - Enterprise Java Beans (EJB)
  - Stored Procedures in a DBMS
  - Web Services (WSDL=SOAP, OpenApi=REST) published in a Web Server
  - SAP BAPI modules
  - JSON files
  - XML schemas

- External Objects available in GeneXus
  - APIs for Smart Devices: Access to HW (camera, GPS, microphone, etc.) y SW (calendar, contacts, notifications, etc.)
  - APIs to Web: Access to events, communication with server, clipboard, maps, social networks, etc.

- External Objetos published in GeneXus Marketplace

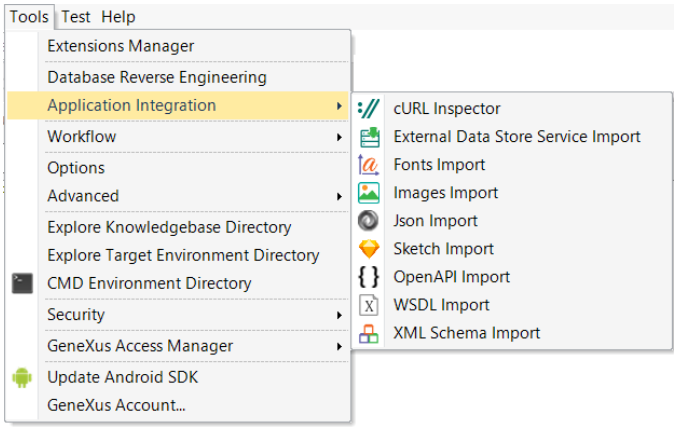
Se puede importar a nuestra KB, distintos tipos de recursos. Por ejemplo, si tenemos algo programado en .NET, podemos generar una DLL e importarla a la KB como un objeto externo. Luego podremos invocar desde nuestra aplicación a las funciones incluidas en la DLL, como si fueran procedimientos programados en GeneXus.
Lo mismo sucede con las clases creadas en Java.

También podemos importar recursos almacenados en otras fuentes externas, como programas Java Beans, o procedimientos almacenados en una base de datos, webservices (tanto SOAP como REST), módulos de SAP, archivos JSON generados por cualquier aplicación, o esquemas de XML.

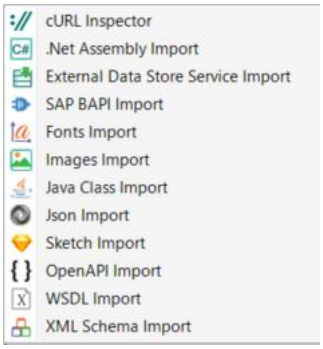
GeneXus también provee un conjunto de Objetos Externos que están en el RootModule o en el módulo GeneXus, que nos permiten el acceso a una variedad de recursos, como por ejemplo APIs para poder interactuar con el hardware o aplicaciones nativas de dispositivos móviles, o APIs para acceder al servidor, a eventos, o a aplicaciones de Windows como el bloc de notas, o sitios externos como el uso de mapas, redes sociales, etc.

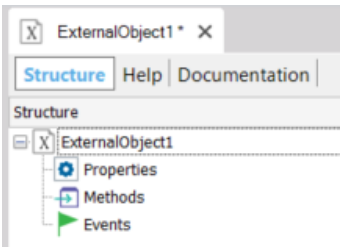
También disponemos de objetos externos publicados en el Marketplace de GeneXus, que podemos incorporar a nuestra aplicación.

Crear un objeto externo usando el wizard



En la versión Full:



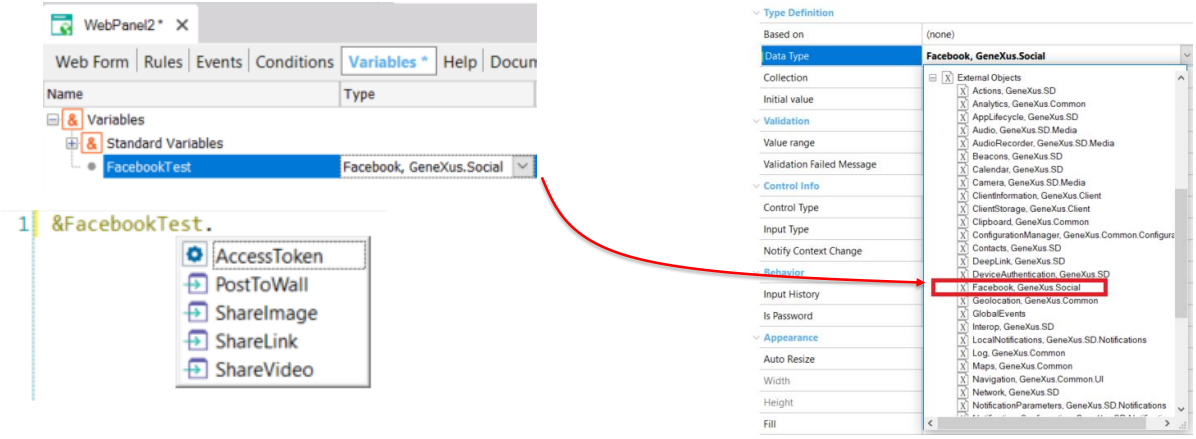


Para crear un objeto externo, la mejor forma es hacerlo a través de un wizard. Si vamos al menú Tools y elegimos Application Integration, vemos los distintos recursos a importar y se ejecutará un wizard específico para ese recurso. Al finalizar el wizard, se creará un objeto externo que quedará automáticamente asociado al recurso, todas las propiedades del objeto externo quedarán ajustadas según el tipo de recurso importado.

También podemos crear un objeto externo con New Object como a cualquier otro objeto GeneXus, pero en ese caso deberemos configurar sus propiedades, métodos y eventos en forma manual.

¿Cómo usamos a un objeto externo?

- 1) Creamos una variable cuyo tipo sea el Objeto Externo
- 2) Invocamos los métodos o asignamos las propiedades disponibles

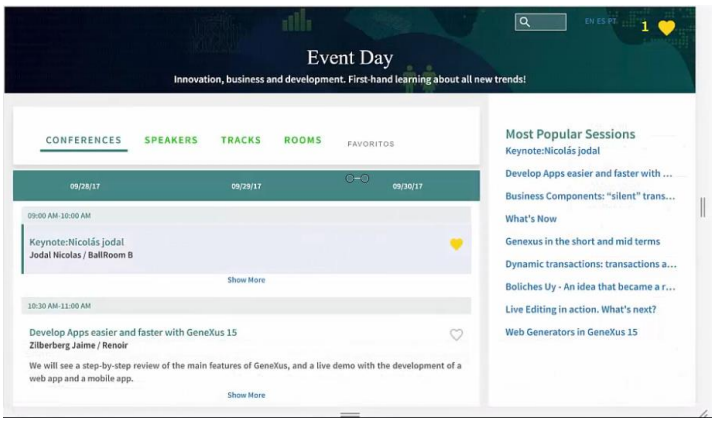


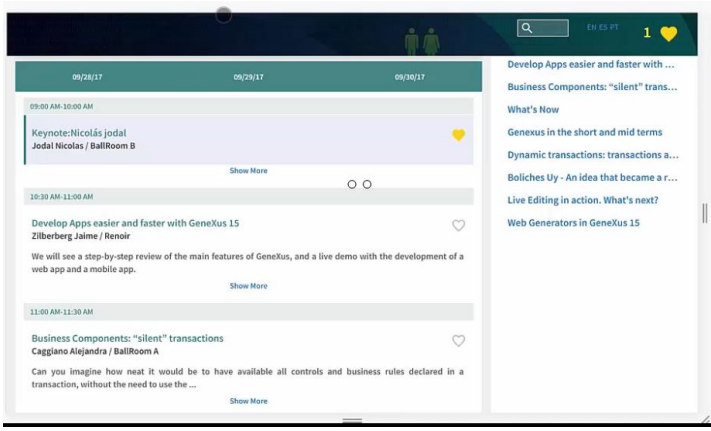
Más informácion sobre objetos externos: https://wiki.genexus.com/commwiki/servlet/wiki?5669

Una vez creado el objeto externo basado en las propiedades correspondientes al recurso externo que deseamos utilizar, el mismo quedará disponible como cualquier otro tipo de datos en la Base de conocimiento.

La forma de usarlo es igual que con cualquier tipo de datos extendido: definiendo una variable de ese tipo y luego llamando a los métodos y/o configurando las propiedades que necesitemos.

Ejemplo de uso de un objeto externo con JavaScript





Otra cosa que podemos hacer con los External Objects es interactuar con JavaScript, por ejemplo para vincular eventos implementados en un JavaScript externo con un evento GeneXus.

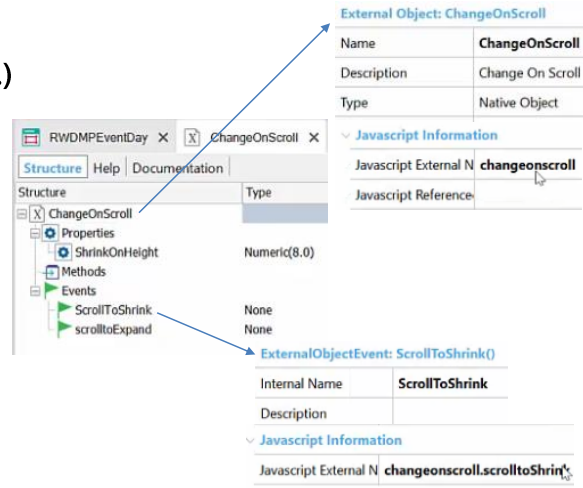
Veamos esto con una aplicación de ejemplo.

Vemos que cuando hacemos scroll en la aplicación, la barra superior se achica.
Esto se está implementando con un evento JavaScriptChangeOnScroll programado en forma externa, vamos a ver como logramos vincular ese JavaScript con GeneXus.

Ejemplo de uso de un objeto externo con JavaScript (cont.)

changeonscroll.js

```
var changeonscroll = {
    shrinkOnHeight: 30
};
$(function() {
   $(window).on('scroll', function() {
    var distanceY = window.pageYOffset || document.documentElement.scrollTop;
    var shrinkOn = changeonscroll.shrinkOnHeight;
    if (distanceY > shrinkOn ) {
        gx.fx.obs.notify("changeonscroll.scrolltoShrink");
        }
     else {
        gx.fx.obs.notify("changeonscroll.scrolltoExpand");
        }
   });
});
```




```
Event Start

    Form.HeaderRawHTML = !"<link href='https://fonts.googleapis.com/css?family=Source+S

    Form.HeaderRawHTML += GetChangeOnScrollScript()
    changeonscroll.ShrinkOnHeight = 20
```

El JavaScript que tenemos es muy sencillo, básicamente cuando define que se llegó a cierta altura
de scroll dispara un evento Shrink y sino dispara un evento Expand.

¿Cómo vamos a hacer para vincular este JavaScript con GeneXus?

Lo hacemos con un objeto externo Changeonscroll, que básicamente está asociado con un JavaScript
externo que se llama Changeonscroll y aquí están los eventos que este JavaScripts está disparando.
En este caso tienen el mismo nombre que en el JavaScripts pero podríamos cambiarlo.

Estos eventos se implementaron en la Web Master Panel (o master page) donde está incluido el objeto externo Changeonscroll con el evento ScrolltoExpand y el evento ScrolltoShrink.
En un caso estamos ocultando el componente y en el otro caso lo estamos mostrando, ya que básicamente queremos que se vea o se oculte según qué tan abajo estamos en la barra de scroll.

Para incluir el JavaScript en la aplicación, lo estamos haciendo de la misma forma que lo hicimos
siempre que es agregar el script en el código HTML.

Por más información sobre objetos externos

https://wiki.genexus.com/commwiki/servlet/wiki?5669

Por más información sobre los objetos externos, puede seguir el siguiente link en pantalla.

training.genexus.com
wiki.genexus.com
training.genexus.com/certifications