

Módulos y Objetos Externos

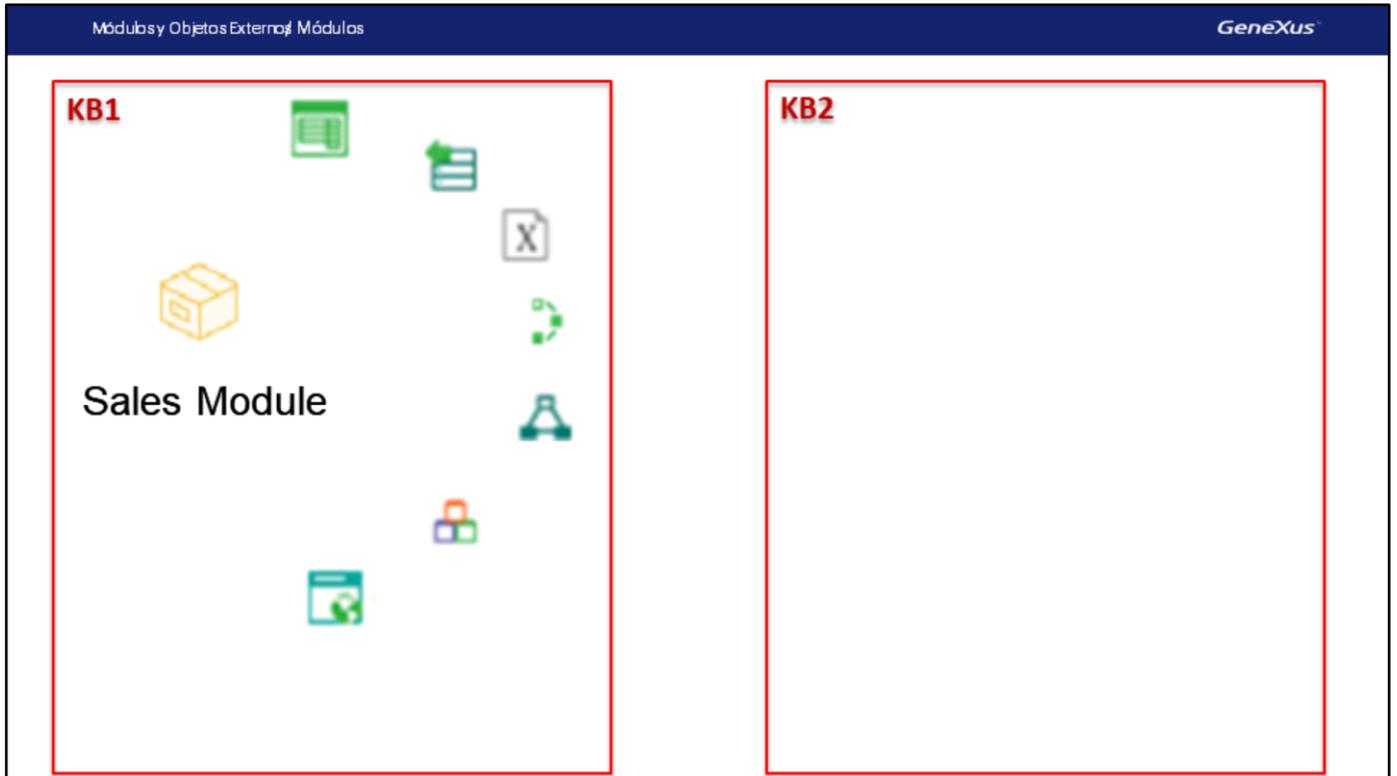
Introducción

GeneXus 16

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

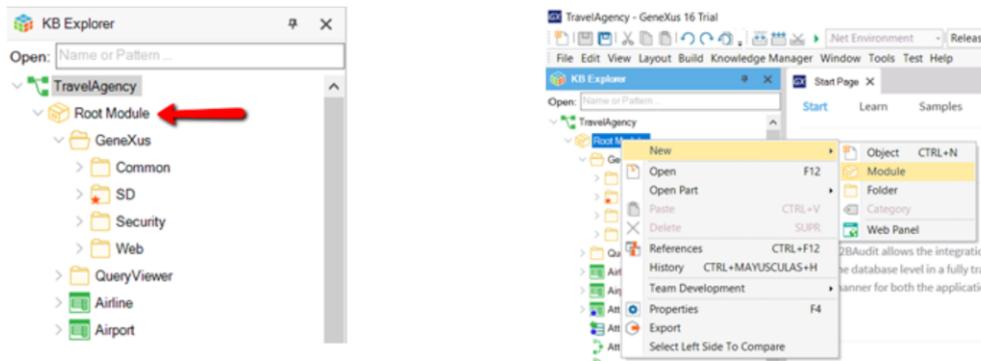
Veamos en primer lugar los módulos.



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 folders 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

The screenshot shows the 'Manage Module References' window in the Knowledge Manager. The interface is divided into two main sections: a list of available modules on the left and a detailed view of the selected 'Chatbot' module on the right.

Module List (Left):

- Chatbot (2.1.10.129299)**: GeneXus Chatbot module is a basic set of interfaces and implementations of data structures and algorithms needed. **Install** button.
- GeneXusAI (1.1.21.129329)**: GeneXusAI contains a common set of Artificial Intelligence tasks, including audio, text and image processing, all of them provided by several Cloud Platforms (e.g. IBM Watson).
- GeneXus (2.1.7.129290)**: GeneXus Core Module is a basic set of interfaces and implementations of data structures and algorithms to solve common programming use cases. **(Highlighted with a red box)**
- GXtest (0.4.2)**: GXtest Module provides core functionality for creating, running and reporting tests on geneXus and over c/cd pipelines. <https://wiki.geneXus.com/commwiki/servelet/wik?>

Module Information (Right):

Chatbot
Module is not installed
 Available Versions: 2.1.10.129299

Author: GeneXus
Owner: GeneXus

Description:
 GeneXus Chatbot module is a basic set of interfaces and implementations of data structures and algorithms needed to implement a Chatbot solution.

Platforms:

- C# Web
- Swift
- Android

Dependencies:

- GeneXus 1.10.16.122645

Id: ae91f89c-b637-4cc0-a8f0-aaa17a6356ce

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.

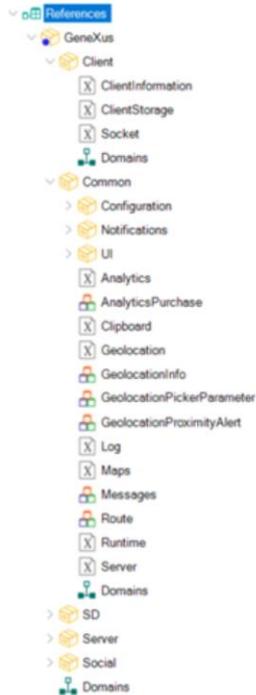
Los objetos de estos módulos no los podremos modificar (los veremos como Read-Only) y ya están compilados, por lo que cuando damos F5 no será necesario especificarlos, generarlos, etc.

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 Referencias.

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?

- Objetos nativos creados y compilados en un lenguaje de programación:
 - Assemblies de .NET (.dll)
 - Clases de Java (.class)
- Recursos almacenados en distintas fuentes externas:
 - Enterprise Java Beans (EJB), de un servidor JEE
 - Stored Procedures creados en un DBMS
 - Web Services (WSDL, OpenApi) publicados en un servidor Web
 - Módulos BAPI de SAP
 - Archivos JSON
 - Esquemas de XML
- Objetos Externos disponibles en GeneXus
 - APIs para Smart Devices: acceso al HW (cámara, GPS, micrófono, etc.) y SW (calendario, contactos, notificaciones, etc.)
 - APIs para acceso a eventos, comunicaciones con el servidor, bloc de notas, mapas, redes sociales, etc.
- Objetos Externos publicados en el Marketplace de GeneXus

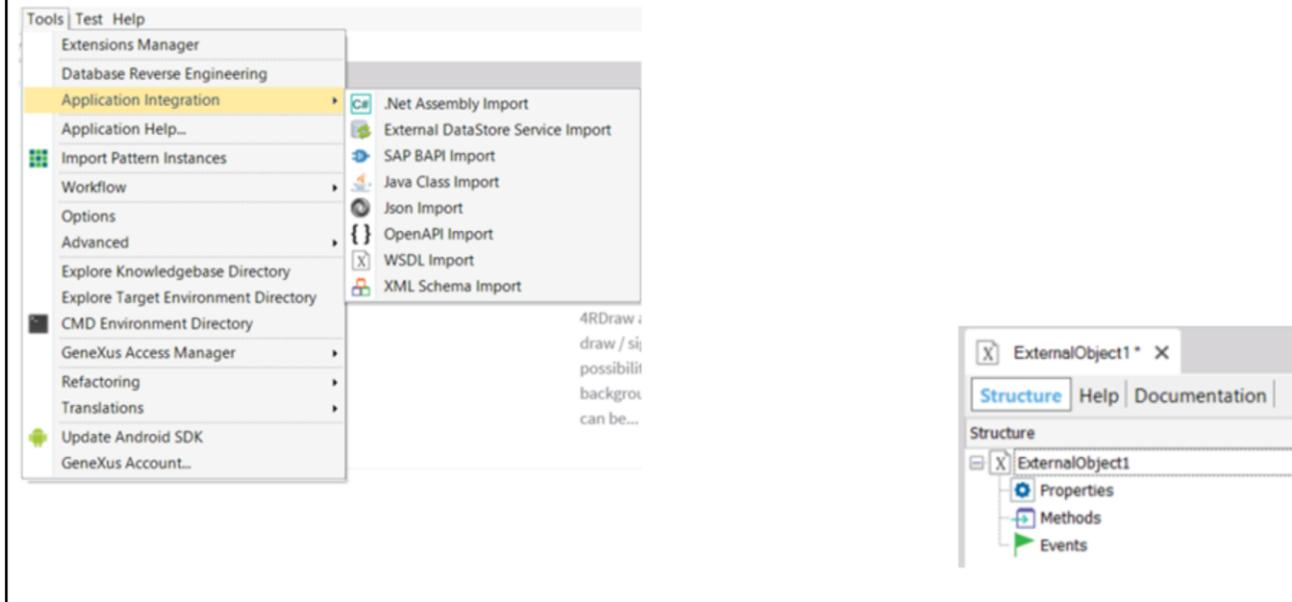
Se puede importar a nuestra KB, distintos tipos de recursos. Por ejemplo, si tenemos algo programado en .NET, podemos generar una DLL e importarla a la KB como un objeto externo. Luego podremos invocar desde nuestra aplicación a las funciones incluidas en la DLL, como si fueran procedimientos programados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



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

The screenshot illustrates the process of defining an external object variable in GeneXus. On the left, the 'Variables' pane shows a variable named 'FacebookTest' with the type 'Facebook, GeneXus.Social'. A dropdown menu is open, showing a list of methods: 'AccessToken', 'PostToWall', 'ShareImage', 'ShareLink', and 'ShareVideo'. On the right, the 'Type Definition' pane shows a list of external objects, with 'Facebook, GeneXus.Social' highlighted in red. A red arrow points from the variable's type dropdown to the highlighted item in the list.

Más información 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.

Por más información sobre los objetos externos, puede seguir el siguiente link en pantalla: <https://wiki.genexus.com/commwiki/servlet/wiki?5669>

GeneXus™

The power of doing.

Videos

training.genexus.com

Documentation

wiki.genexus.com

Certifications

training.genexus.com/certifications