



GAM Components

Nicolas Adrién | GeneXus Training

Integration with GeneXus

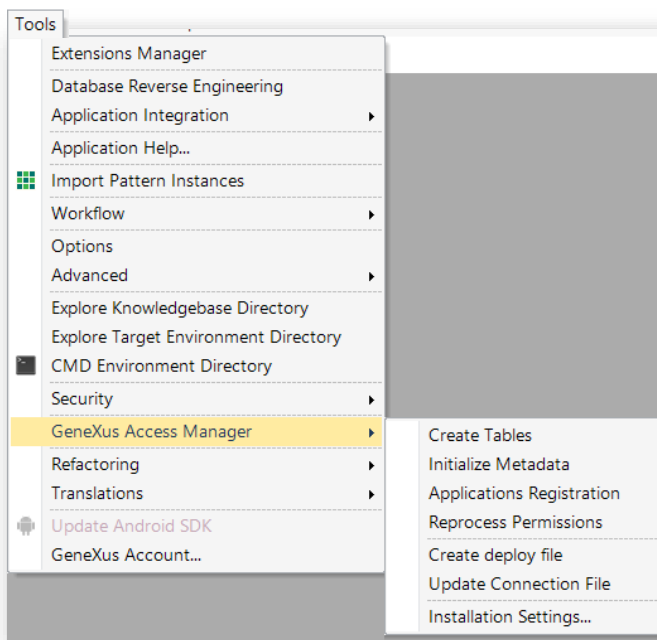


Backend

Data Store

Frontend

GAM soporta plataformas para ejecutar su ambiente tanto web como móvil. Se puede definir el lenguaje a utilizar en el Backend, los sistemas de gestión de bases de datos, y la plataforma a utilizar en el FrontEnd. Todo desde la UI de GeneXus. Mas adelante entraremos en profundidad en cada uno de estos.



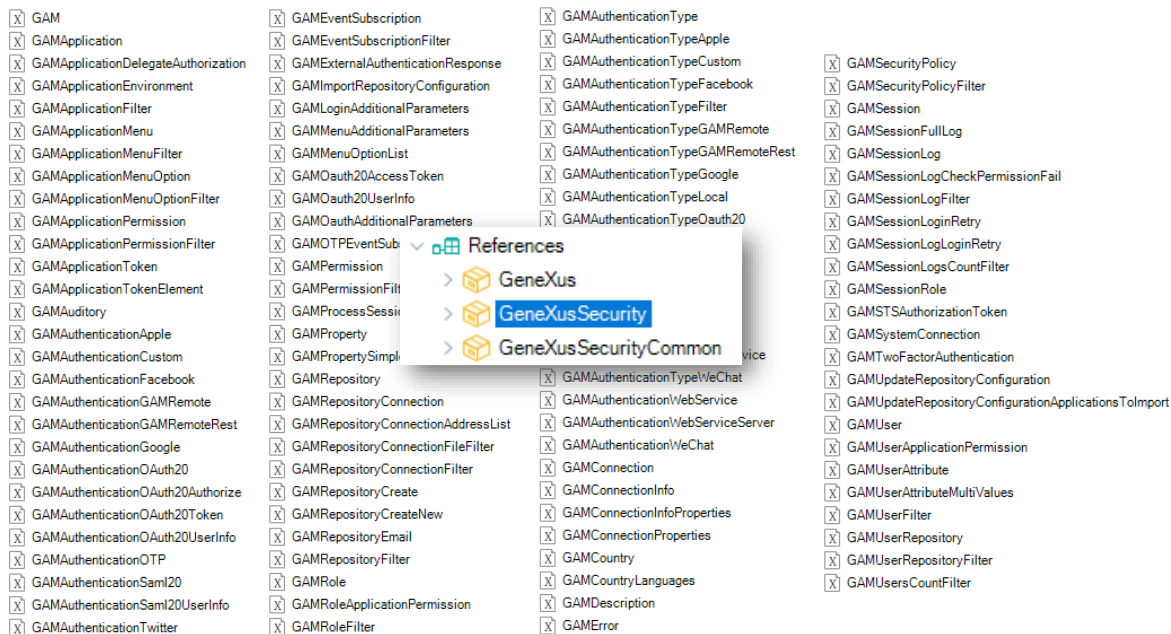
Bajo el menú Tools/GeneXus Access Manager, podemos acceder a las distintas funcionalidades que nos ofrece el GAM dentro de GeneXus.

Desde allí podemos:

- Crear las tablas que utiliza GAM
- Inicializar los metadatos
- Registrar las aplicaciones que utilizaran GAM
- Reprocesar los permisos de las mismas
- Crear el archivo de deploy
- Modificar el archivo de conexiones

Y de ultimo,

- Configurar la instalación del mismo



En el ítem References del menú contextual de la KB, dentro de GeneXusSecurity podemos encontrar todos los External Object que nos proporciona GAM, como vemos en pantalla.

Veamos la estructura de uno a modo de ejemplo.

Structure	Type	Is Collection	Description
GAMUser			
Properties			
GUID	GAMGUID, GeneXusSecurityCommon	<input type="checkbox"/>	User GUID (Identificator)
NameSpace	GAMRepositoryNameSpace, GeneXusSecurityCommon	<input type="checkbox"/>	User name space
AuthenticationTypeName	GAMAuthenticationTypeName, GeneXusSecurityCommon	<input type="checkbox"/>	Authentication type name
Name	GAMUserIdentification, GeneXusSecurityCommon	<input type="checkbox"/>	User name (nickname)
Login	GAMUserLogin, GeneXusSecurityCommon	<input type="checkbox"/>	Login
E-Mail	GAMEMail, GeneXusSecurityCommon	<input type="checkbox"/>	Email
ExternalId	GAMUserIdentification, GeneXusSecurityCommon	<input type="checkbox"/>	External identification
Password	GAMPASSWORD, GeneXusSecurityCommon	<input type="checkbox"/>	Password
FirstName	GAMDescriptionShort, GeneXusSecurityCommon	<input type="checkbox"/>	First name
LastName	GAMDescriptionShort, GeneXusSecurityCommon	<input type="checkbox"/>	Last name
Methods			
Get	GAMUser, GeneXusSecurity	<input type="checkbox"/>	Return current user entity
GetByGUID	GAMUser, GeneXusSecurity	<input type="checkbox"/>	Get User by user GUID
GUID	GAMGUID, GeneXusSecurityCommon	<input type="checkbox"/>	
Errors	GAMError, GeneXusSecurity	<input checked="" type="checkbox"/>	
GetByExternalId	GAMUser, GeneXusSecurity	<input type="checkbox"/>	Get User by user external identification
ExternalId	GAMUserIdentification, GeneXusSecurityCommon	<input type="checkbox"/>	
Errors	GAMError, GeneXusSecurity	<input checked="" type="checkbox"/>	
GetByLogin	GAMUser, GeneXusSecurity	<input type="checkbox"/>	Get User by user login
AuthenticationTypeName	GAMAuthenticationTypeName, GeneXusSecurityCommon	<input type="checkbox"/>	
UserName	GAMUserIdentification, GeneXusSecurityCommon	<input type="checkbox"/>	
Errors	GAMError, GeneXusSecurity	<input checked="" type="checkbox"/>	
Undelete	GAMBoolean, GeneXusSecurityCommon	<input type="checkbox"/>	It allows to recover the User that was logically deleted (GAMUser.Delete).
Errors	GAMError, GeneXusSecurity	<input checked="" type="checkbox"/>	

Profundizando por ejemplo en el External Object GAMUser, vemos que su estructura (al igual que todos los demás) es conformada por propiedades y métodos.

Dado que este External Object es bastante grande, se simplificó el contenido a mostrar en esta diapositiva para que se vea el contenido explicado. Desde GeneXus pueden ver la totalidad de propiedades y métodos que éste contiene.

Podemos notar que para cada propiedad y método, a su vez tenemos el tipo, si es colección o no, y descripción que nos sirven de ayuda al momento de querer utilizarlos.

Veamos un ejemplo de uso.

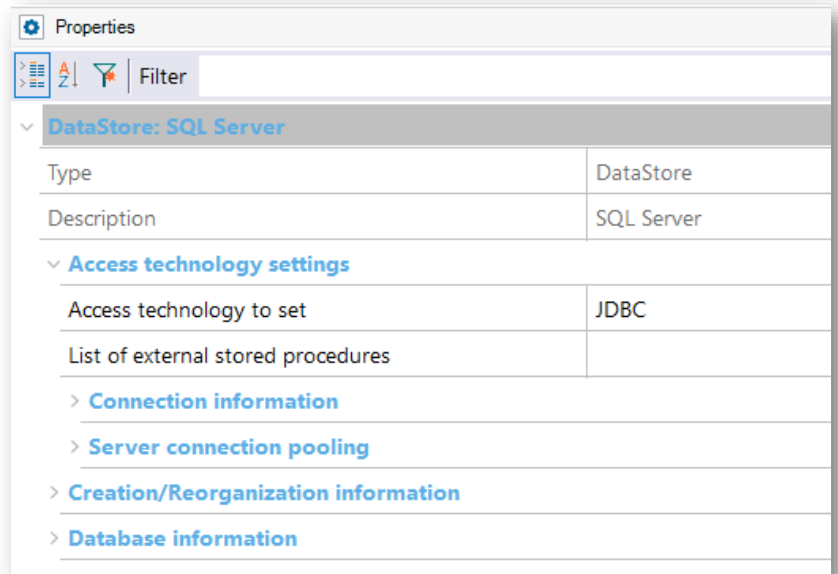
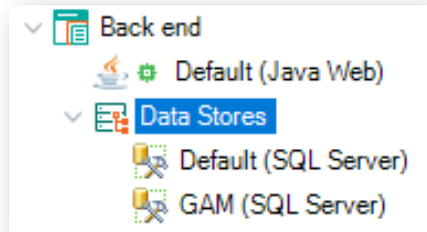


```
&User.Name = "John"
```

```
&Name = &User.GetName()
```

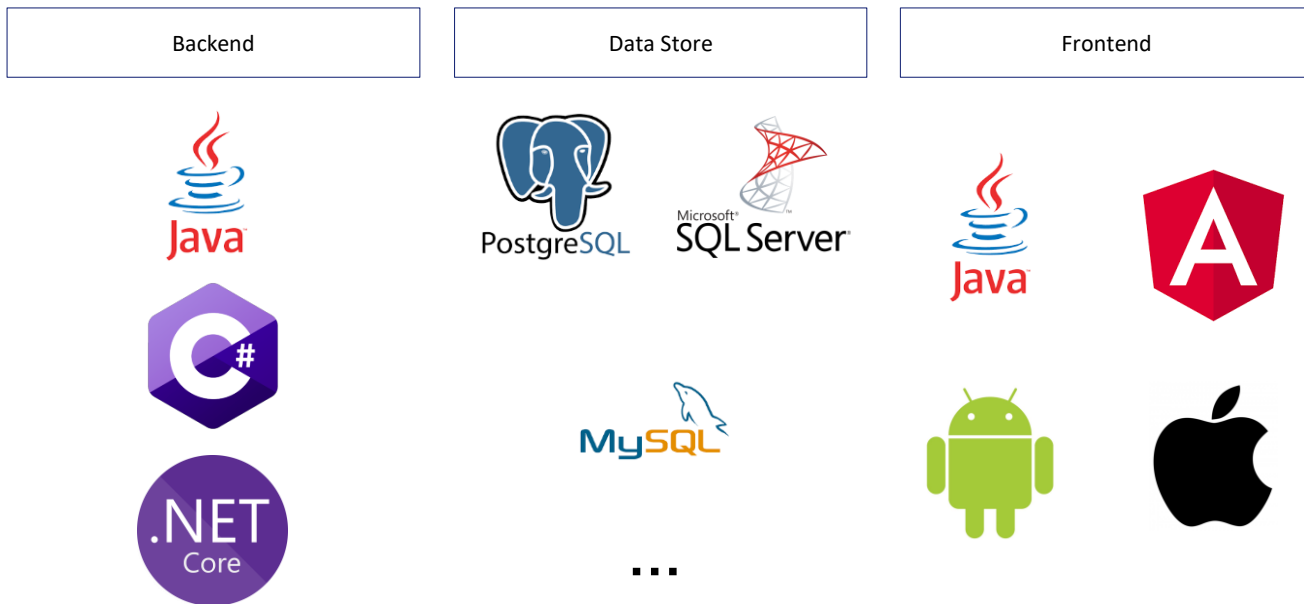
Para utilizar el External Object debemos crear una variable de este tipo (en nuestro caso GAMUser) y para usarlo en el código, simplemente se asignan valores a sus atributos o propiedades, o se utilizan sus métodos para obtener los datos ya cargados.

Para utilizar el resto de los External Object, se sigue el mismo flujo y metodología.



Al momento de habilitar GAM en la KB, se genera un Data Store para GAM para almacenar su información junto al de la Aplicación, donde es posible configurar la tecnología a utilizar, la información de conexión y servidor, etc.

Platforms



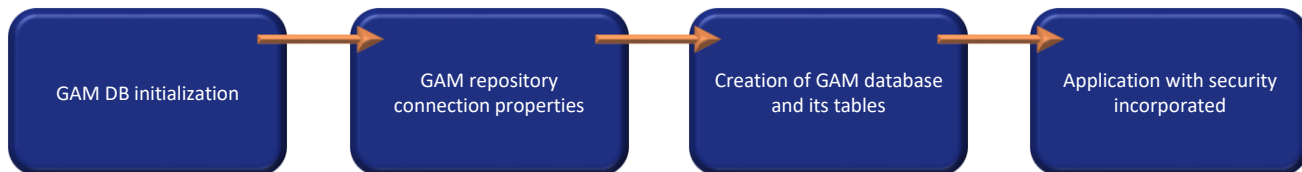
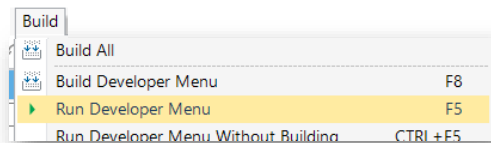
Entrando mas en detalles sobre las distintas plataformas que soporta GAM, tenemos que:

Para Backend, se puede generar bajo los lenguajes Java, C# y .NET Core.

Para el Data Store, se ofrecen muchas opciones donde se pueden destacar PostgreSQL, SQL Server, MySQL, entre otras.

Finalmente, para Frontend se puede utilizar versiones web, como Java o Angular, y versiones móviles, como lo son Android y Apple.

Databases



Después de correr el Developer Menu, se establece una conexión con la base de datos especificada en el Data Store de GAM, usando las propiedades de conexión de este, verificando la existencia de algunas tablas y la versión de GAM.

Dado que estas tablas no existen, se crean las tablas de la base de datos GAM. Entonces, sucede lo siguiente:

Primero se asignan algunas propiedades relacionadas con la conexión al repositorio GAM, con sus valores predeterminados. Luego, se crea la base de datos GAM y todas sus tablas. Antes de crear las tablas, se le pregunta al usuario si desea crear la estructura de la base de datos GAM.

Backend

The screenshot displays the 'Users' management interface in the GeneXus Access Manager Backend. The top navigation bar includes 'GENEXUS ACCESS MANAGER', 'USERS', 'ROLES', and 'SETTINGS'. A user profile 'Administrator' is visible in the top right. The main content area features a search bar with the text 'Try a Search' and a '+ ADD' button. Below this is a table with the following data:

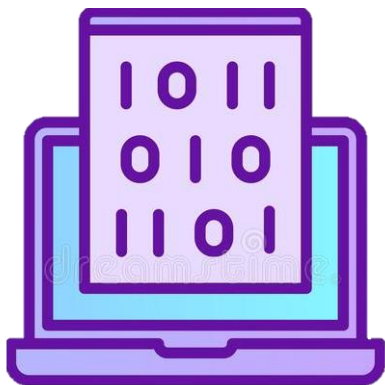
User Name	First Name	Last Name	Authentication	
admin	Administrator	User	local	EDIT

At the bottom of the table, there are navigation links: 'FIRST / PREV / NEXT'.

El Backend de GAM es una Aplicación GAM definida automáticamente con la creación de los metadatos GAM, durante el proceso de inicialización.

El propósito de esta aplicación es configurar y gestionar todos los conceptos relacionados con GAM.

Backend



Compiled



XPZ

Existen dos versiones del Backend.

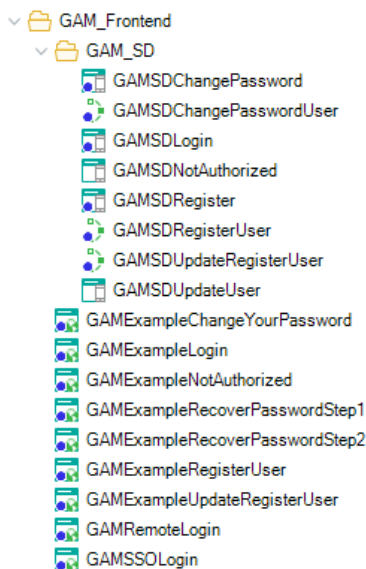
La primera es la compilada. Por defecto se crea al habilitar GAM en la KB.

La segunda versión, es por XPZ. Un XPZ se refiere a los archivos utilizados por el Manager de KBs para intercambiar objetos entre bases de conocimiento. El nombre proviene de la extensión de estos archivos. En este caso, se puede incorporar GAM a la KB importando este xpz que viene en la instalación de GeneXus.

Estos objetos son útiles ya que son ejemplos de uso de la API GAM.

Pueden ser modificados a voluntad del desarrollador GeneXus si no se cumplen algunos requisitos (para ello se encuentra disponible la API GAM).

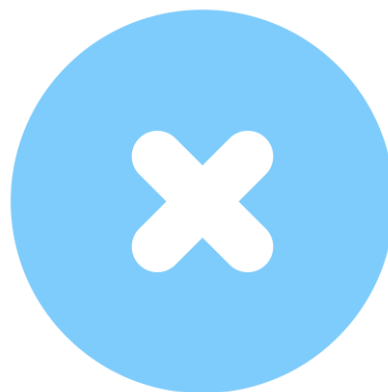
Frontend



Al momento de habilitar GAM, se importan a la KB todos los objetos de ejemplo del Frontend.

Los ejemplos de GAM son objetos web y paneles SD que utilizan la API GAM, y su propósito es ayudar al usuario GeneXus a aprender a usar esta API. Otro propósito importante de estos objetos es ayudar al usuario a comenzar a utilizar GAM, ya que incluyen objetos para login, registro, cambio de contraseñas, redirección en caso de error de Autorización, entre otros.

Todos estos objetos se consolidan en la KB durante el proceso de activación de GAM y se ubican en la carpeta Ejemplos de GAM; cada vez que se instala un nuevo build o upgrade de GAM, el usuario puede decidir si desea actualizar dichos ejemplos con el look and feel de la aplicación, por ejemplo.



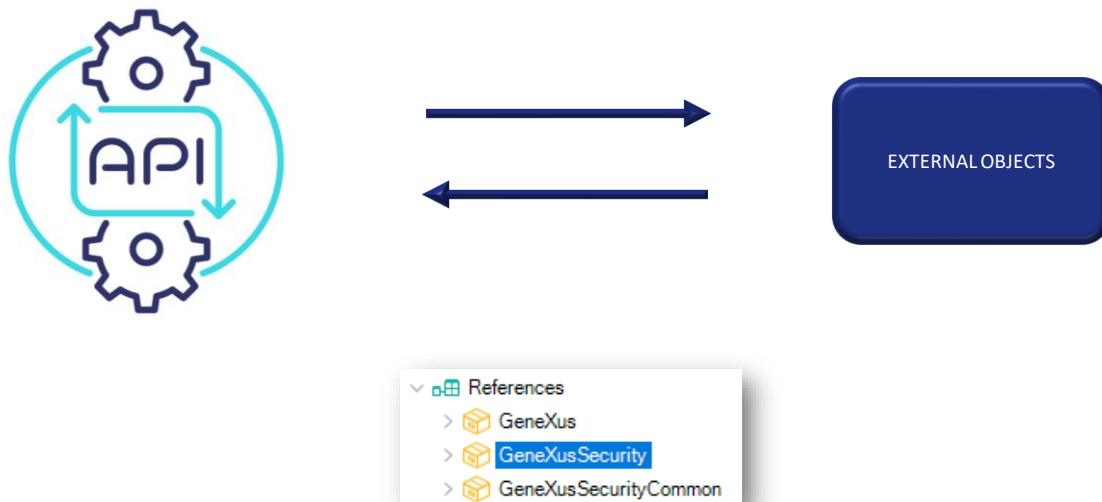
GAMRegisterUser

GAMSDRegisterUser

Una buena práctica consiste en crear copias de los ejemplos para ser utilizados en la aplicación. De esta forma, ante una actualización de GeneXus (y por lo tanto de GAM), los cambios introducidos no serán sobrescritos.

Otra recomendación es eliminar aquellos paneles que no serán utilizados. Un ejemplo de esto es el auto-registro que veremos mas adelante. Si en la aplicación los usuarios serán dados de alta por un administrador, se recomienda eliminar los paneles GAMRegisterUser y GAMSDRegisterUser.

API



GAM proporciona una API que permite a los usuarios manejar tipos de datos y métodos para agregar seguridad a las aplicaciones GeneXus (tanto aplicaciones Web como aplicaciones mobile).

Cuando la seguridad integrada está habilitada en la KB, se incorporan objetos externos para permitir que el usuario interactúe con la API de GAM. Los objetos externos son la forma de acceder a la API de GAM. Todos ellos se distribuyen en un módulo llamado GeneXusSecurity, como dijimos anteriormente.

Las APIs son las utilizadas por los paneles de ejemplo. A partir de estas, es posible realizar todas las acciones disponibles en el Backend de GAM.

Settings



```
<?xml version="1.0" encoding="UTF-8"?>
```

▼ **Integrated Security**

Integrated Security Level	Authentication	66a6306</key>
Application ID	08d3dc04-2dd4-401e-8635-2160f4505da7	



application.gam

```
<?xml version="1.0" encoding="utf-8"?>
<Application>
  <Id>08d3dc04-2dd4-401e-8635-2160f4505da7</Id>
</Application>
```

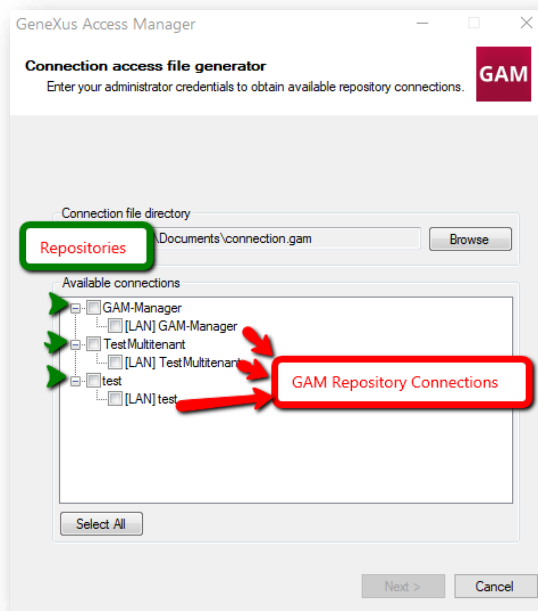
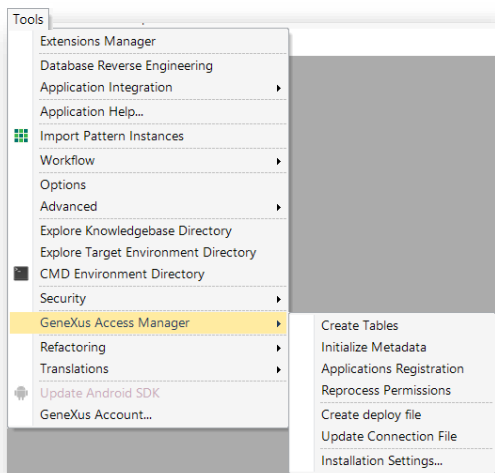
GAM genera distintos archivos con configuraciones para la conexión a repositorios. Estos archivos son connection.gam y application.gam.

connection.gam es un archivo de configuración de GAM que contiene la clave asociada con cada Repositorio al que se desea conectar. La información de conexión asociada a la clave tiene que existir en la base de datos de GAM (debe haber sido creada previamente usando la API de GAM). En el caso de Java, dicho archivo se copia en la raíz de la aplicación web en el servidor de servlets (esto es a partir del upgrade 3 de Evolution 2). En el caso de NET va en el directorio virtual (o sea, el directorio web).

application.gam es un archivo de configuración de GAM que contiene el id de la Aplicación WEB GAM de la KB, especificado en la propiedad Application Id, y se utiliza para que GAM identifique que aplicación web está corriendo. Durante el tiempo de creación de prototipos, se transfiere automáticamente a la carpeta WEB-INF en el caso de las aplicaciones Java y al directorio virtual en el caso de las aplicaciones NET. Al llevar la aplicación a producción, se debe incluir este archivo en la implementación. El ID que contiene, debe existir en una aplicación definida en la Base de datos GAM, y es configurable desde el IDE de GeneXus a través de las propiedades del Environment en la sección Integrated Security como vemos en pantalla.



Settings - Tools



Existen herramientas de GAM para modificar la información de los archivos de configuración para distintos ambientes, como puede ser Testing, PreProduccion o Producción, con el fin de facilitarnos la edición de los mismos.

Para el archivo **connection.gam**, existe la herramienta **GamDeployTool** que puede ser ejecutada desde el IDE de GeneXus, seleccionando la opción **GAM - Update Connection File**.

Esta herramienta esta destinada para el uso de los administradores del Repositorio de GAM Manager.

Al usarla, se agregan entradas en determinadas tablas de la base de datos GAM dependiendo de la selección del usuario, quien puede seleccionar entre las conexiones de Repositorio GAM existentes de cada Repositorio.

Al finalizar el proceso, el archivo se genera en la dirección especificada, para luego ser copiado a la aplicación web.

Settings



client.cfg

```
EnableIntegratedSecurity=1
IntegratedSecurityLoginWeb=GAMEExampleLogin
IntegratedSecurityNotAuthorizedWeb=GAMEExampleNotAuthorized
```

```
DataSource1=GAM
DataSource2=DEFAULT
```



web.config

```
<add key="DataStore1" value="GAM" />
<add key="DataStore2" value="Default" />

<add key="EnableIntegratedSecurity" value="1" />
<add key="IntegratedSecurityLoginWeb"
value="gameexamplelogin,objects" />
<add key="IntegratedSecurityNotAuthorizedWeb"
value="gameexamplenotauthorized,objects" />
```

Otros tipos de archivos de configuración son el **client.cfg** y **web.config**.

Estos dependen del generador que se esté utilizando:

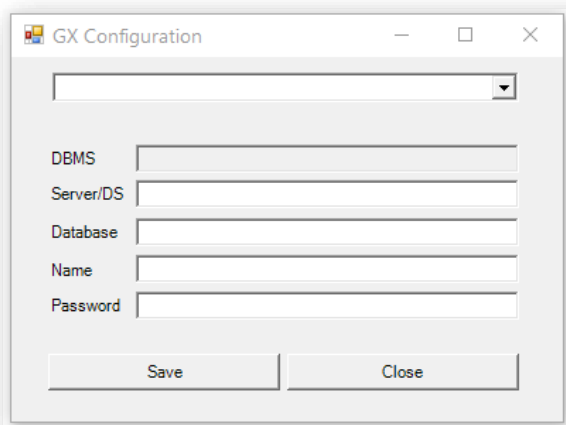
En aplicaciones Java, se utiliza client.cfg (donde aplica para aplicaciones web y de 2 niveles).

En aplicaciones .Net, se utiliza web.config

Para el **client.cfg** y **web.config**, al momento de habilitarse GAM en una base de conocimiento, se setean en ellos las líneas que vemos en pantalla.

Corresponden a establecer en 1 la seguridad integrada, y definir los objetos de ejemplo de Login y No Autorizado.

Además, se agrega un DataSource nuevo que corresponde a GAM, dejando en segundo lugar el DEFAULT de la aplicación.



En cuanto a las herramientas para estos dos últimos archivos, tenemos:

Para el archivo **web.config**, existe una tool realizada por GeneXus llamada GxConfig, la cual facilita la creación del archivo a través de la interfaz grafica que vemos en pantalla.

Para el archivo **client.cfg**, existen distintas maneras de facilitar la configuración del mismo que varían según la versión de GeneXus que se está utilizando.



GX_GAMCONNECTIONKEY

connection.gam

En varios
escenarios
, una

práctica
común es
leer la
información
de
configuración
de las

variables de
entorno, en
lugar de los
archivos de
configuración

•

Desde el **upgrade** #5 de GeneXus 17, la clave de conexión GAM se puede especificar a través de una variable de entorno llamada **GX_GAMCONNECTIO NKEY.**

Esto permite, por ejemplo, que al realizar un deploy a Docker se pueda configurar cual será la Key de conexión que se quiere usar para esa instancia del contenedor.

Cuando se intenta acceder a cualquier API del GAM desde una aplicación lo primero que se hace es chequear si está configurada esa variable de ambiente para obtener los datos de conexión del GAM. Si no está, se busca el archivo connection.gam para obtener esta Key.

En la Wiki de GeneXus se describe un documento que detalla los escenarios en los que se utiliza la práctica de utilizar variables de ambiente, sus ventajas y cómo hacerlo.

GeneXus™

training.genexus.com

wiki.genexus.com

training.genexus.com/certifications