
GeneXus™
by Globant

DEPLOYMENT

Nicolas Adrién

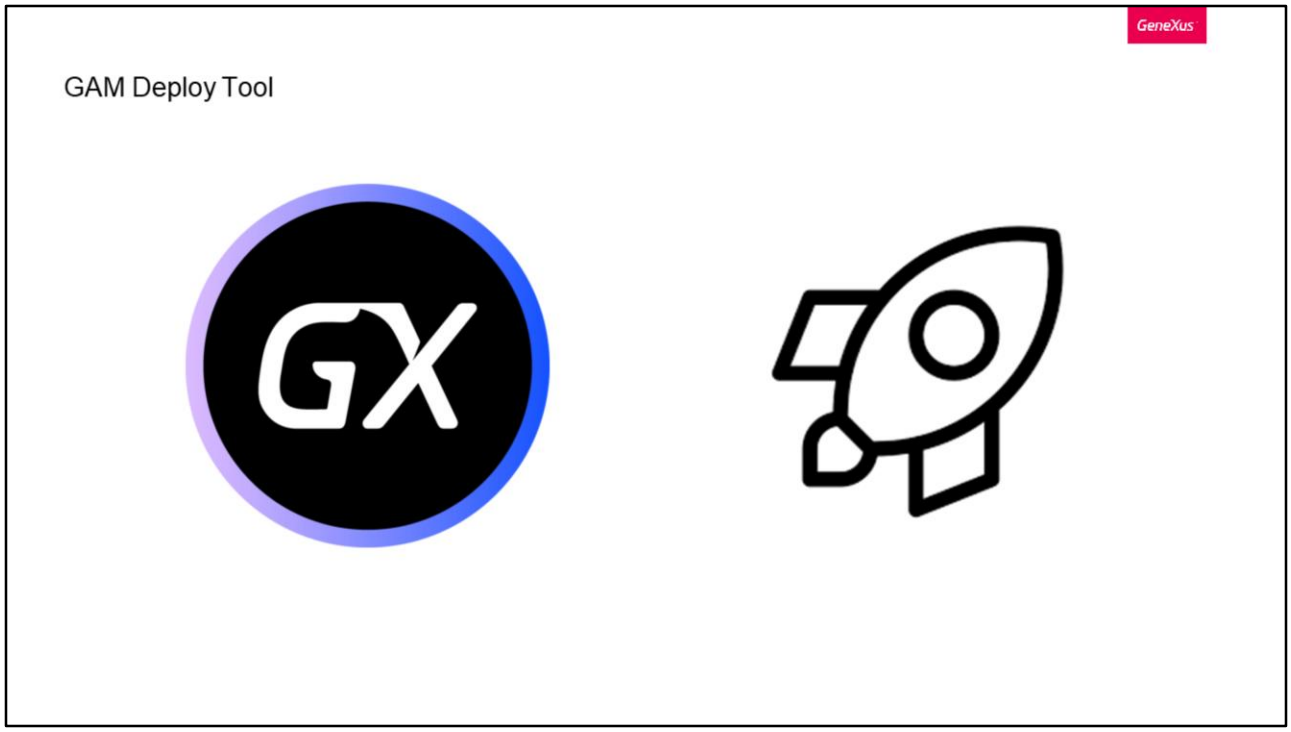
GeneXus

En este video hablaremos sobre el pasaje entre ambientes y upgrades, problemas comunes con esto, y aprenderemos a usar la herramienta GAM Deploy Tool.

En ciertas ocasiones puede pasar que un desarrollador no tenga acceso a la base de datos y desea realizar un pasaje de GAM entre ambientes.
Aquí nos podemos preguntar cómo hacemos para realizar las reorganizaciones e impactos de metadata.

Para solucionar este problema, GAM nos proporciona distintos scripts que ya vienen incluidos en la carpeta de instalación de GeneXus.
Dentro de la carpeta Platforms, se deberán encontrar los ambientes que tenemos definidos en nuestras KBs.
Allí, vamos a encontrar los siguientes archivos.

Con el primer archivo podremos crear toda la base de datos de GAM.
Los restantes, consisten en migrar entre las distintas versiones de GAM, como indican sus nombres. Algo importante con esto, es que en caso de utilizar estas migraciones, las mismas se deben ejecutar en orden para que todo funcione correctamente.
Esto forma parte de la estructura de la base de datos, donde en esos archivos tendremos todas las sentencias SQL correspondientes que deben ser ejecutadas en un cliente de la base de datos de GAM.



GAM Deploy Tool es una herramienta que tiene como propósito ayudarnos con los despliegues de aplicaciones utilizando GAM, ya que cuenta con el conocimiento de la estructura y relaciones de la base de datos de GAM, brindando la posibilidad de importar datos de manera gradual manteniendo la consistencia del modelo de datos.
La herramienta está orientada para llevar a producción la información que requiere ser actualizada en la Base de Datos GAM, y se encuentra disponible desde GeneXus 16 U5.

Nos centraremos en la versión por línea de comandos.

GeneXus

GAM Deploy Tool command line

JAVA: /tomcatproduction/webapps/XXX/WEB-INF/classes/com/kbname#

java -cp "../../../lib/*" genexus.security.api.agamdeploytool "<Action> <Corresponding Flags>"

.NET Framework:

agamdeploytool.exe "<Action> <Corresponding Flags>"

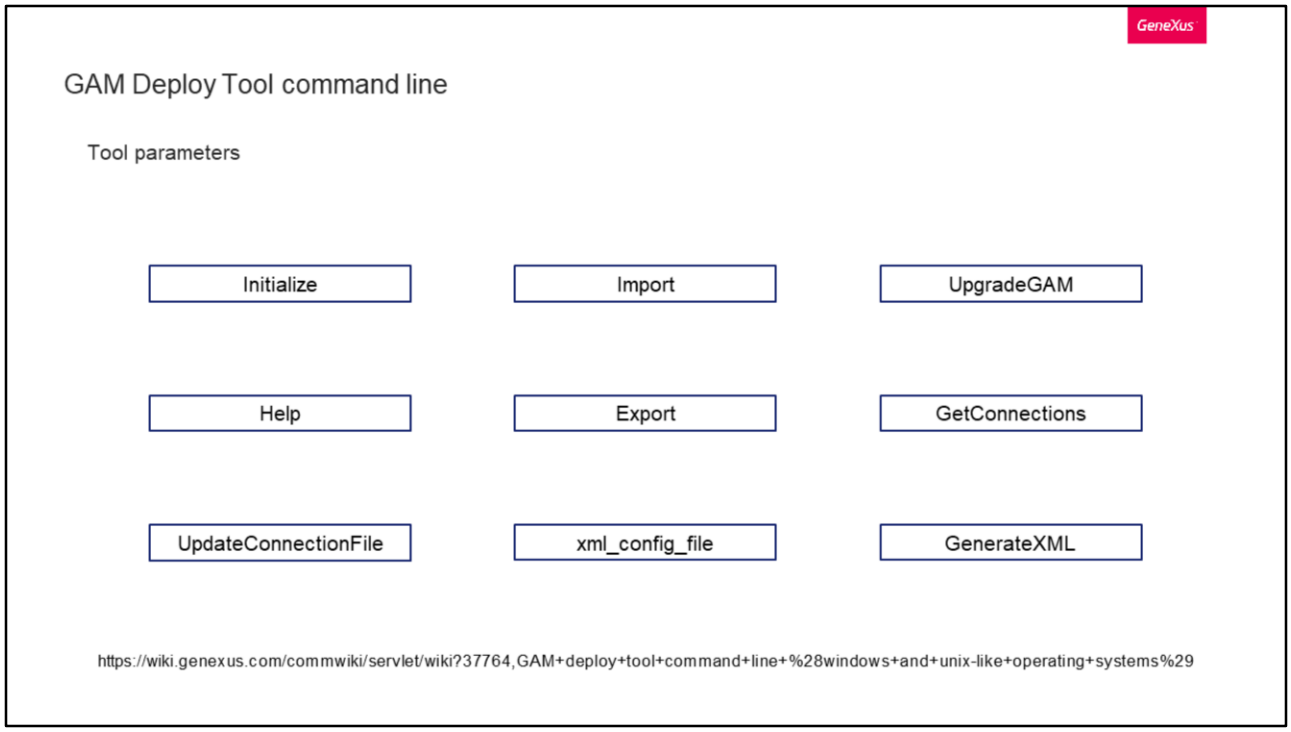
.NET:

dotnet agamdeploytool.dll "<Action> <Corresponding Flags>"

Las llamadas a la herramienta deben tener el siguiente formato, según el ambiente en el que se encuentren.

Lo ideal es ejecutar la herramienta posicionados debajo del entorno de implementación. En el caso de .NET, en el directorio virtual \bin. En el caso de Java, bajo su webapp como vemos en el comando.

La herramienta no va a solicitar la configuración de conexión de la base de datos GAM, como el servidor, puerto, usuario o contraseña, porque esa información se toma de los archivos de configuración como lo es el client.cfg en el caso de Java y web.config en el caso de .NET.



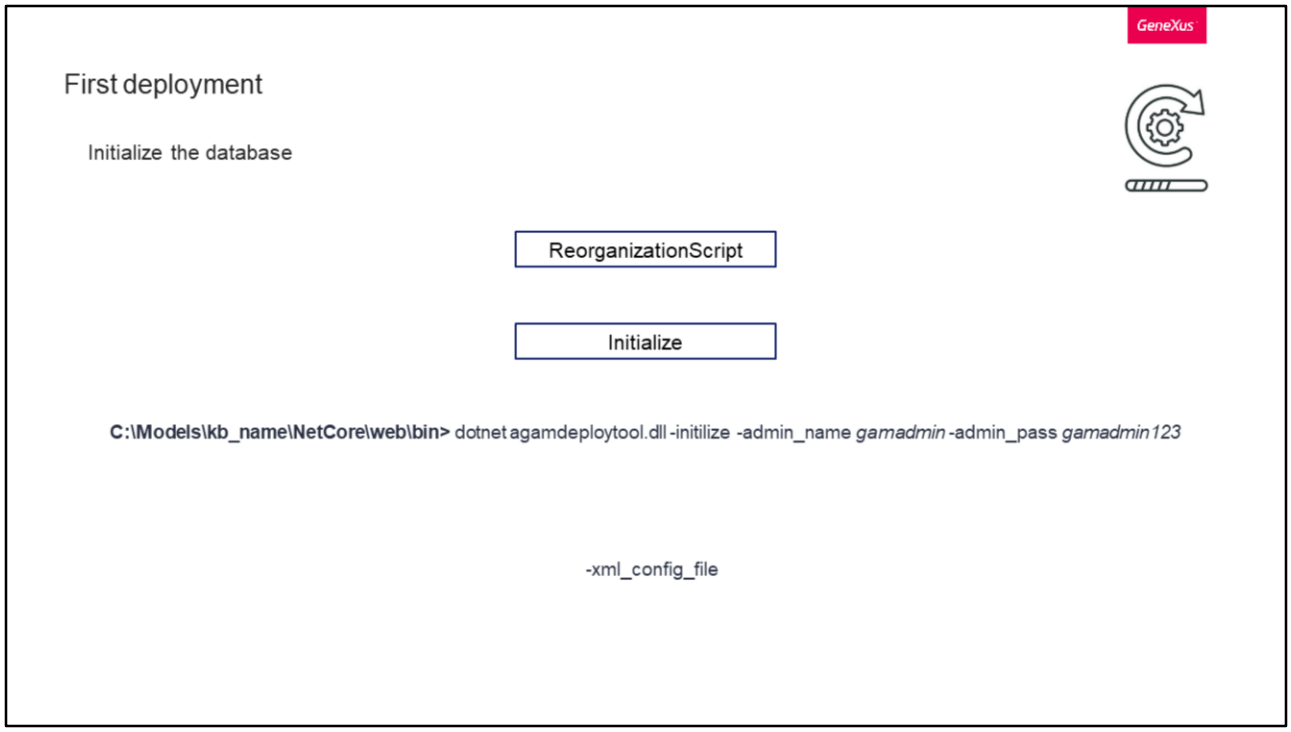
Esta herramienta nos brinda distintas acciones que podemos hacer con ella. Veamos en detalle cuales son.

- Inicializar: Nos permite inicializar la base de datos GAM con sus metadatos (solo repositorio 1)
- Importar: Importa un paquete con sus configuraciones y datos
- Actualizar GAM: Actualiza la versión de la base de datos GAM. En caso de que se necesite hacer una reorganización, se la debe realizar previamente y luego si ejecutar esta acción de la herramienta
- Ayuda: Como todo comando, con esto veremos las acciones disponibles que tiene la herramienta
- Exportar: Exporta los datos de una base de datos GAM y los almacena en un paquete .gpkg.
- Obtener Conexiones: Obtiene las conexiones agrupadas por Repositorio. Esta función nos es de utilidad para obtener los GUID de nuestros repositorios y los nombres de conexión que se envían como parámetros en la opción UpdateConnectionFile. Esta ultima actualiza o crea el archivo connection.gam con los datos de conexión que obtiene
- xml_config_file: Esta es una bandera especial que solo recibe un archivo XML en el que se cargan todos los parámetros a ingresar en la herramienta, incluida la acción que queramos

Finalmente tenemos a Generar XML, la cual genera un XML de ejemplo y lo muestra en la salida estándar. Este XML se puede usar como entrada para la herramienta,

cambiando los valores de etiqueta correspondientes.

Dentro de cada acción por supuesto tenemos variadas banderas a incluir. El detalle de estas las pueden encontrar en la Wiki de GeneXus con toda la documentación necesaria.



Ahora veamos como deberíamos hacer para desplegar el GAM y sus datos por primera vez, y para posteriores despliegues.

Comencemos con el primer despliegue.
Obviamente utilizaremos la herramienta GAM Deploy Tool.

Al inicio del video veíamos que GAM nos proporcionaba distintos scripts en la carpeta de instalación de GeneXus. Como en nuestro caso queremos inicializar la base de datos, ejecutaremos el ReorganizationScript. Con él crearemos la base de datos de GAM vacía.
A su vez también vimos las distintas opciones que nos brinda la herramienta que estamos viendo, donde una de ellas es Initialize. Con esta inicializamos la base de datos de GAM de la siguiente manera.
En la ejecución debemos indicar el nombre de usuario y contraseña de GAM.

Un paréntesis a mencionar es que este y todos los restantes ejemplos son en base a .NET, que es lo que utilizarán en el práctico.

A su vez, tenemos la posibilidad de usar la siguiente flag, donde a través de esta podremos indicar la ruta a un archivo XML que tenga todos los parámetros configurados. Si establecemos esta flag, todas las demás se ignoran automáticamente y solo se tienen en cuenta los parámetros del archivo. Un ejemplo de archivo para

esta flag es el siguiente.

GeneXus

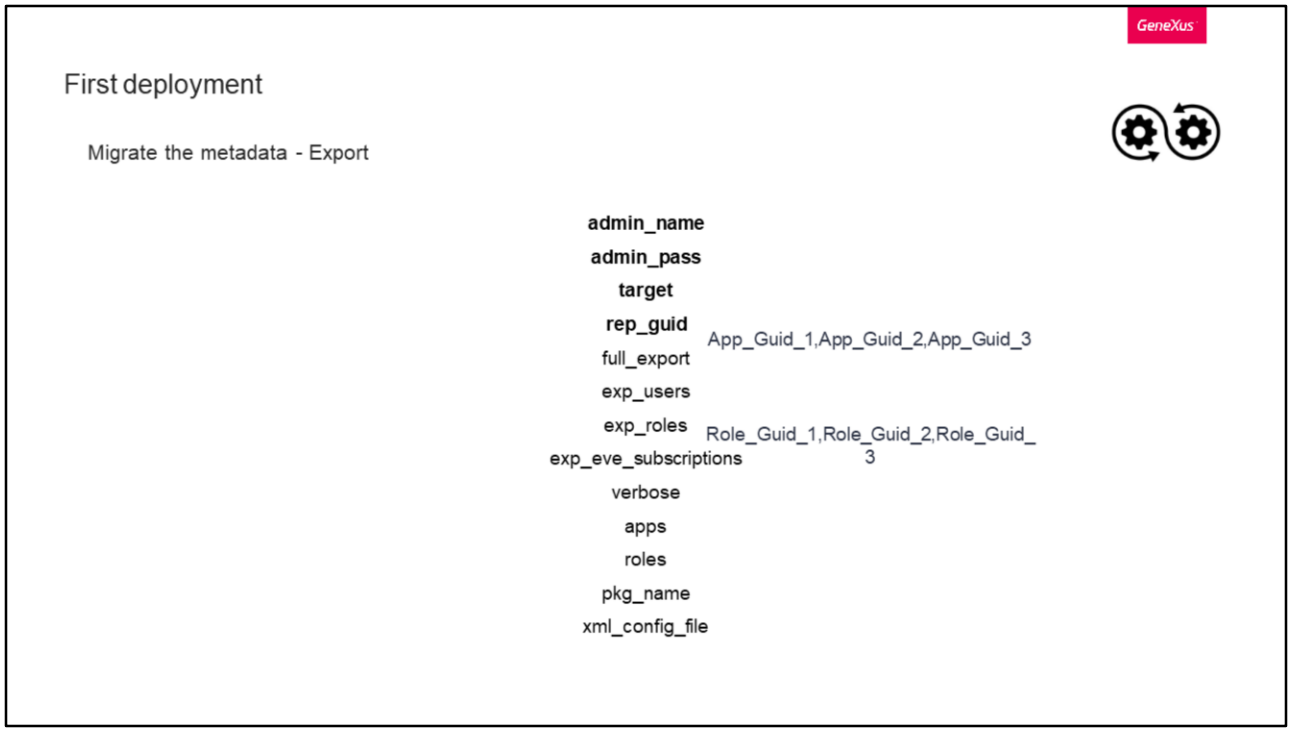
```
<?xml version="1.0" encoding="utf-8"?>
<GamDeployTool>
    <GeneralSettings>
        <ProcessType>Import</ProcessType>
        <GamAdminName>gamadmin</GamAdminName>
        <GamAdminPass>gamadmin123</GamAdminPass>
        <Verbose>true</Verbose>
    </GeneralSettings>

    <DbmsSettings>
        <DbmsCode>18</DbmsCode>
    </DbmsSettings>

    <ImportSettings>
        <PackageFilePath>C:\Temp\fullgx\GAM_package_GAMPackage403.gpkg</PackageFilePath>
        <ImportType>Full</ImportType>
        <GenerateDefaultConnection>false</GenerateDefaultConnection>
        <ImportRepositoryAction>update</ImportRepositoryAction>
        <RepositoryGuid>0d777d89-790e-4a19-84c5-e92f5e9e5920</RepositoryGuid>
    </ImportSettings>

</GamDeployTool>
```

DB2 for iSeries => 9
DB2 Universal Database => 5
Informix => 11
MySQL => 18
Oracle => 7
PostgreSQL => 15
SQL Server => 12

En el tag ProcessType tenemos los valores: Import, Export, GenerateConnectionFile, RestartGamDb, y UpdateSchema.

Dentro de las configuraciones de conexión a la base de datos, tenemos el código del DBMS. Allí, tenemos los siguientes valores según el sistema de base de datos que estemos utilizando.

Finalmente tenemos a la configuración para ejecutar un Import. Obviamente esto solo se toma en cuenta si ProcessType tiene el valor Import.
Aquí dentro primero tenemos la ruta de donde cargar el paquete de deploy, luego el tipo de importación (si es Full o Custom), un booleano para indicar si se debe generar una conexión por defecto para el repositorio, una flag que puede indicar si se quiere actualizar un repositorio existente (como está ahora con el valor update, o si queremos crear un nuevo repositorio que en ese caso deberíamos poner el valor CreateNew), y finalmente tenemos un identificador de repositorio que aplicaría al repositorio que queremos actualizar si en la flag anterior ingresamos el valor update.



Una vez ejecutado el Initialize, tendremos el repositorio por defecto creado. Pero ahora tenemos que pasar nuestros propios repositorios, junto a usuarios, roles, etc.
Para hacer esto, tenemos que primero exportar los datos y luego importarlos en el ambiente de Producción o Pre-Producción al que queremos migrar los datos.

Además del usuario y contraseña, en target especificaremos el directorio de destino de la exportación, y en rep_guid introduciremos el identificador del repositorio a exportar.
Miremos los siguientes campos:

- full_export: Booleano que en caso de ser verdadero, le estamos indicando que la exportación sea completa, incluyendo usuarios, roles y aplicaciones.
- exp_users, roles y eve_subscriptions: indicaran si queremos exportar usuarios, roles y las suscripciones a eventos respectivamente. Esta flag solo se toma en cuenta si full_export tiene valor Falso.
- apps y roles: contendrán el listado de aplicaciones y roles respectivamente que queremos que se exporten. Estos deben tener el siguiente formato.
- pkg_name: es el nombre del paquete que estamos exportando
- xml_config_file: que como ya dijimos, es un xml con toda esta configuración de parámetros.

Algo a destacar es que las flags en negrita son obligatorias para hacer el export.

Luego de haber hecho el Export, tenemos que hacer el Import. Veamos las flags disponibles para esto.

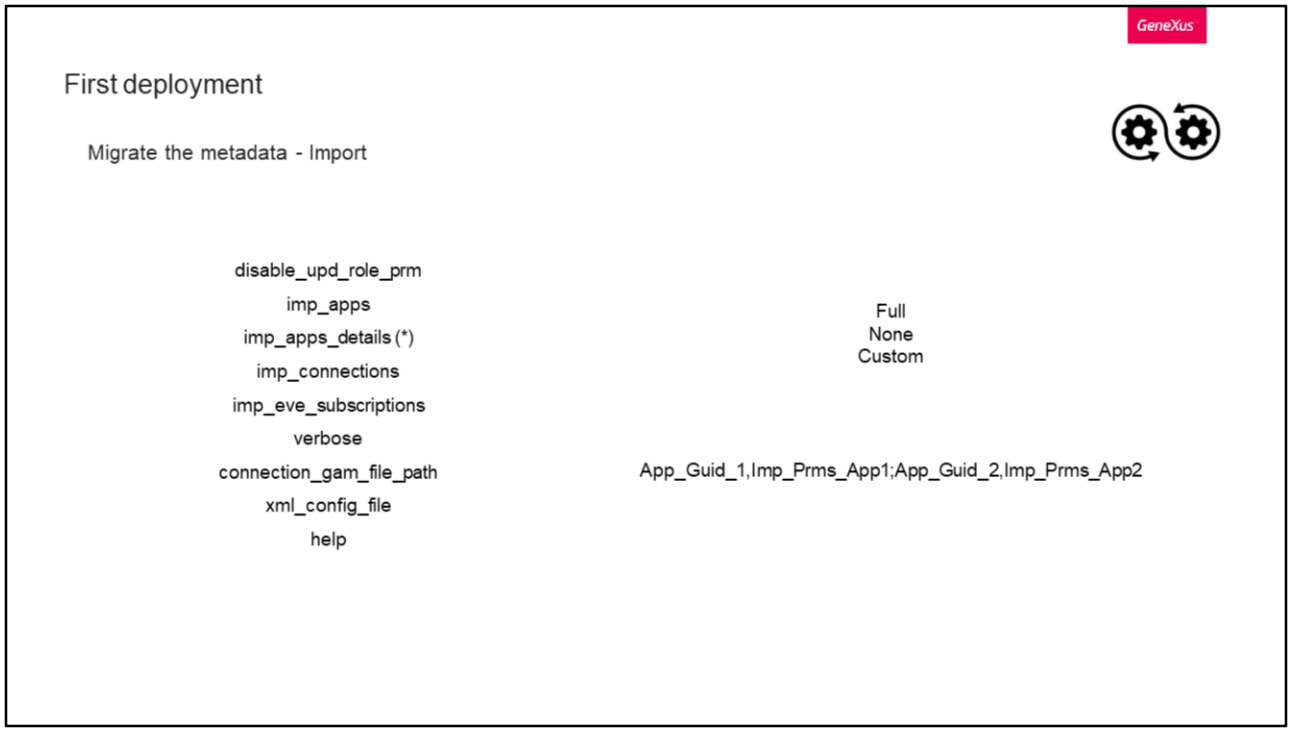
En file_path_package debemos introducir la ruta de nuestro paquete exportado anteriormente.
Luego tenemos los datos del administrador, que en este caso además del nombre y contraseña, podemos ingresar el nombre de usuario y el identificador del rol de dicho administrador.
upd_rep es un booleano para indicar si vamos a hacer una actualización del repositorio existente, en caso de que si, en la siguiente flag indicamos su identificador. En caso de que no, tenemos las siguientes banderas:

- new_rep_create, que es un booleano que indica si se va a crear un nuevo repositorio. Luego tenemos el nombre, namespace, identificador, nombre y contraseña del nuevo administrador del repositorio, y nombre de usuario y contraseña de la conexión del repositorio.

Después tenemos a todas las que indican que es lo que queremos importar, las cuales corresponden a los tipos de autenticación, políticas de seguridad, usuarios y roles. Todas booleanas. En caso de que queramos importar todo, podemos usar la flag siguiente que es imp_full. (Esta es solo valida desde GeneXus 16 U6).

Al igual que en la placa anterior, las flags en negrita son las obligatorias.
Las que tienen un asterisco tambien lo son pero solo cuando la flag new_rep_create

tiene el valor True.



Siguiendo con las banderas, con disable_upd_role_prm indicamos si se deben importar los permisos de los roles que ya existen en la Base de Datos, por defecto, está en False.
imp_apps es el nivel con el que se importan las aplicaciones, y para esto tenemos los siguientes valores:

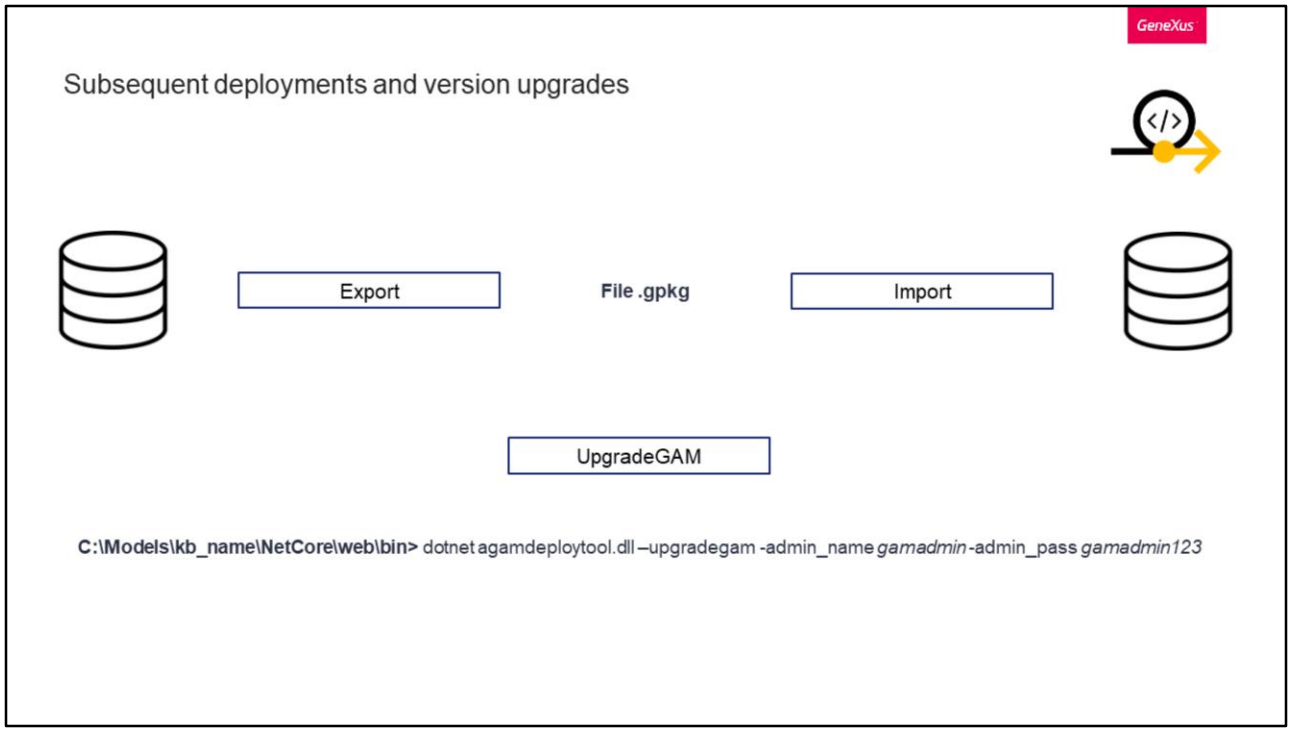
- Completo, donde todas las aplicaciones se importan con todos los permisos
- Ninguno, que indica que nada se importa en relación con las aplicaciones
- O custom, donde se configuran según la flag imp_apps_details

Esta ultima flag, representará entonces al listado de pares Identificador de aplicación y booleano indicando si se importan los permisos de esa aplicación.
Después tenemos a imp_connections que importa las conexiones del paquete. Para los nuevos repositorios, siempre se crea una nueva conexión independientemente del valor de esta bandera; pero en caso de actualización, las conexiones no se actualizan a menos de que tengan el mismo nombre de usuario de conexión que una conexión ya existente.
Para cerrar la parte de importación, tenemos a imp_eve_subscriptions, que importará las suscripciones a eventos.

Del resto de flags, podemos destacar a connection_gam_file_path, con la cual indicaremos el destino de donde se generará el archivo connection.gam.

La flag imp_apps_details tiene asterisco dado que es obligatoria solo cuando la flag

imp_apps tiene el valor Custom.



Una vez que ya hicimos el primer despliegue, veamos como se hacen los siguientes, o como migrar de versiones de GAM.

En caso de que solo tengamos que actualizar nuestra propia metadata, debemos realizar el mismo proceso anterior de primero Exportar, lo cual traerá de la base de datos lo necesario generando un archivo .gpkg, y luego Importar, seleccionando por supuesto lo que queremos migrar a través de las distintas opciones que nos brinda la funcionalidad.

Ahora, si solo queremos actualizar la versión de GAM, simplemente basta con ejecutar la acción de UpgradeGAM, donde esta nos actualizara el GAM junto a su metadata. Su ejecución, es análoga a la de inicializar, donde se requiere el nombre y contraseña del administrador.

GeneXus™
by Globant

training.genexus.com
wiki.genexus.com