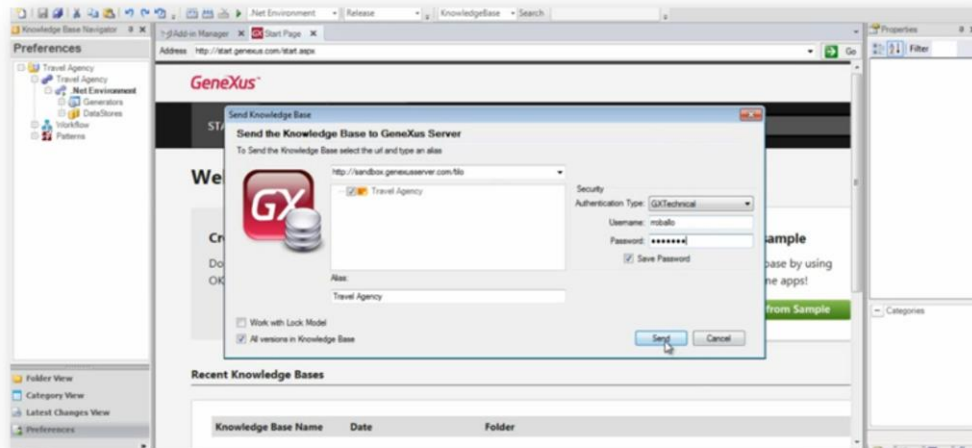


GeneXus X Evolution 3

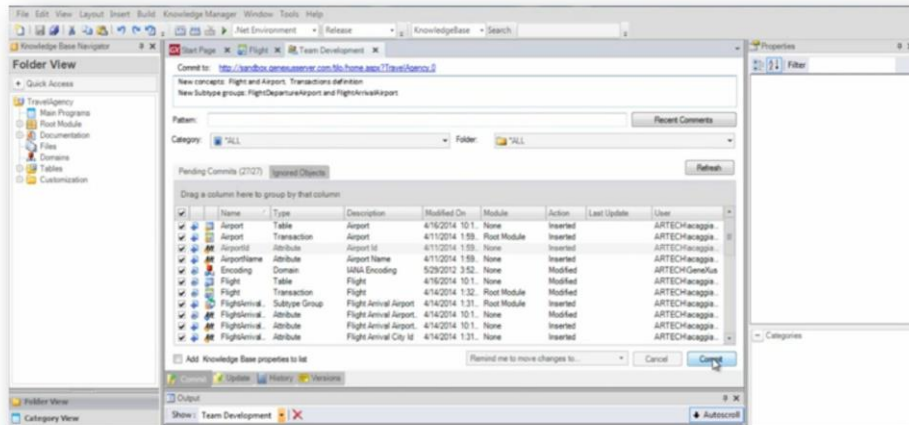
GeneXus Server

Envío de la KB a GeneXus Server



Lo primero que debemos hacer es subir a un servidor la base de conocimiento creada, con el fin de que a partir de ese momento GXserver se encargue de administrarla automáticamente.

Envío de modificaciones

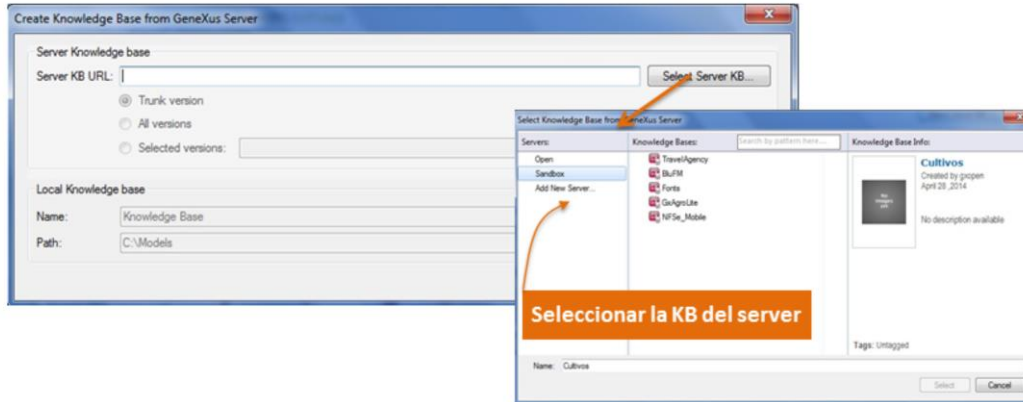


La acción “Commit” permite actualizar las nuevas definiciones y cambios hechos por los desarrolladores, en la KB ubicada en el servidor.

Explicaremos ahora más detalladamente, en qué consisten las acciones y conceptos básicos relacionados con el uso de GXserver.

Nuevo desarrollador se une al equipo

File / New / Knowledge Base from Server



Empecemos por suponer que un nuevo desarrollador, Mike, desea integrarse a este desarrollo. Como 1er paso deberá sincronizarse con la KB del server, esto significa que deberá crearse una copia local desde el server. Para eso, elegirá la opción **File/ New /Knowledge Base from GXserver**:

Mike debe conocer previamente la dirección del server, y debe ingresarla <hacerlo>, luego selecciona la KB correspondiente...e indica un nombre para la KB local que está creando a partir de esta del server.. vamos a poner TavelAgencyMike.

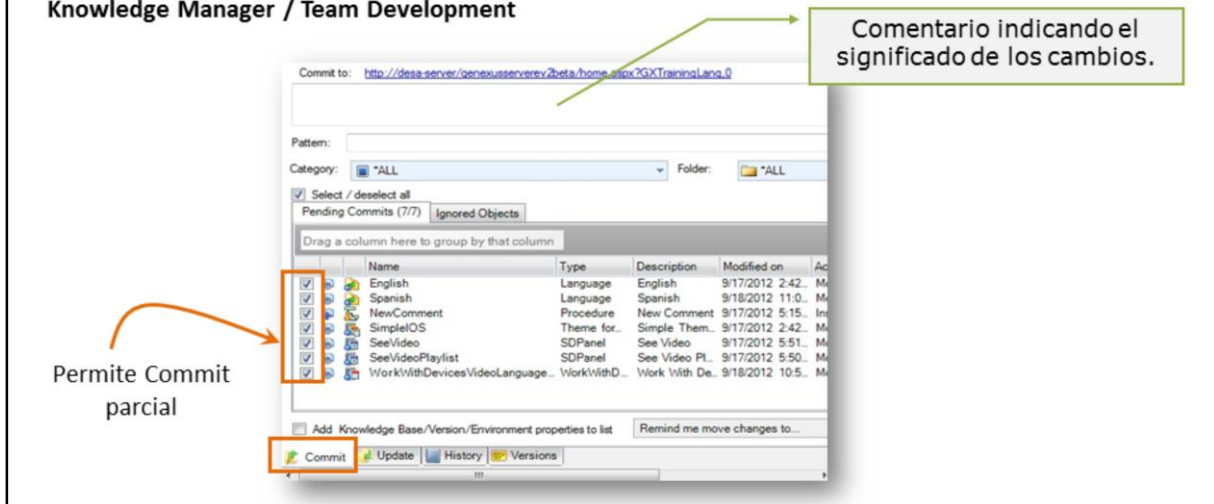
Y presiona luego el botón Create. De esta forma Mike recibe una copia local de la KB actual en el server y queda también sincronizado con ella.

Cada desarrollador trabaja sobre una copia personal de la KB (no se requiere ningún tipo de conexión y no hay tampoco interacción en esa instancia entre los desarrolladores).

De ahora en más Mike trabajará en su **KB local**.

Envío de cambios al servidor

Knowledge Manager / Team Development



Ahora supongamos que Peter, que es uno de los desarrolladores que ha estado trabajando en la aplicación para la agencia de viajes, define en su KB local, una nueva transacción de nombre Employee, para registrar a los empleados de la agencia Define EmployeeId, EmployeeName, EmployeeAddress y EmployeePhone.

También define dos reglas para que controlar que sea obligatorio que se ingrese el nombre y el teléfono del empleado:

```
Error('The name can not be empty') if EmployeeName.isEmpty();
Error('The phone can not be empty') if EmployeePhone.isEmpty();
```

Peter decide que esta funcionalidad está completa y quiere compartirla, es decir, agregarla a la solución completa en el servidor, entonces tiene que ejecutar la acción: "commit".

Para ello selecciona el menú **Knowledge Manager / Team Development**

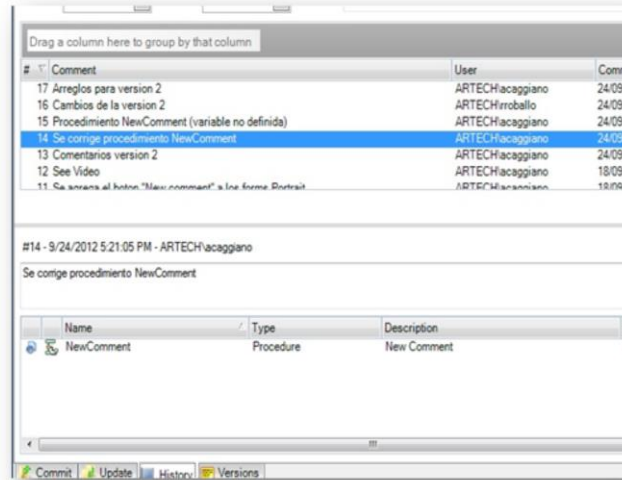
..y desde esta **solapa Commit** se observa la lista de objetos que han sido modificados, o que están definidos en su kb local y aún no han sido integrados en el servidor.

Es posible seleccionar algunos de los objetos para enviar al servidor, es decir que es posible realizar un commit parcial.

Cuando se realiza una operación de commit, se debe agregar un comentario que describe los cambios que se están subiendo. Este comentario pasará luego a formar parte del log en el servidor.

En el servidor, el conocimiento enviado es consolidado con el existente y el desarrollador que ejecutó la operación "commit" recibe un "feedback".

Registro de envíos



Drag a column here to group by that column

#	Comment	User	Commit
17	Arreglos para version 2	ARTECHacaggiano	24/09/20
16	Cambios de la version 2	ARTECHirrobello	24/09/20
15	Procedimiento NewComment (variable no definida)	ARTECHacaggiano	24/09/20
14	Se corrige procedimiento NewComment	ARTECHacaggiano	24/09/20
13	Comentarios version 2	ARTECHacaggiano	24/09/20
12	See Video	ARTECHacaggiano	18/09/20
11	Se anexa al footer "NewComment" a la forma Portrait	ARTECHacaggiano	18/09/20

#14 - 9/24/2012 5:21:05 PM - ARTECHacaggiano

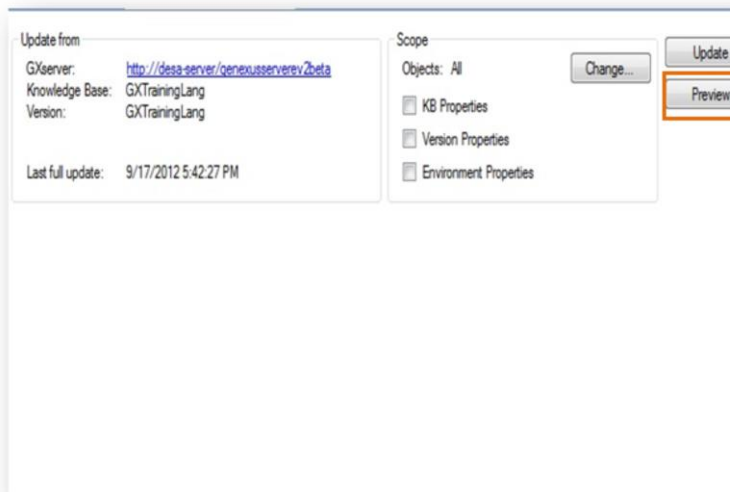
Se corrige procedimiento NewComment

Name	Type	Description	Access
NewComment	Procedure	New Comment	Modifiable

Commit Update History Versions

En la solapa History, vemos el registro de los envíos realizados, su comentario y el conjunto de objetos

Update: Recepción de modificaciones



¿Cómo hace Mike para recibir en su kb local los cambios realizados por Peter o por cualquier otro desarrollador involucrado en este desarrollo?

Para recibir los cambios del server deberá desplegar el cuadro de diálogo: **Knowledge Manager / Team Development**..... y deberá seleccionar la solapa Update:

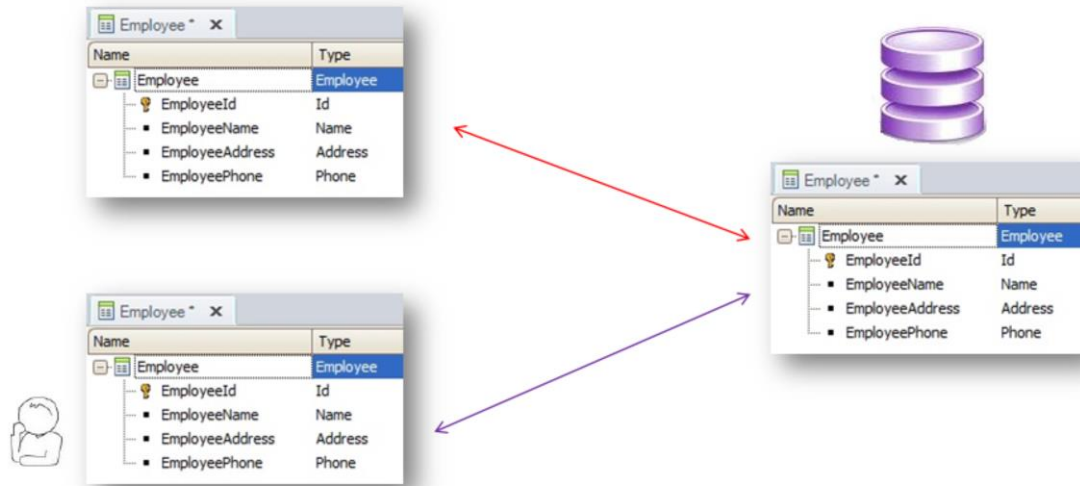
En esta solapa **Update**, si selecciona el botón **Preview** podrá ver tanto los objetos nuevos, como aquellos que sufrieron cambios.

Puede elegir recibir todas las modificaciones, o seleccionar solo algunas.

Al presionar luego el botón Update Selected, recibe lo que eligió. Ahora Mike ve en su KB local también a la transacción Employee

Así que en este momento Mike tiene localmente las mismas definiciones que se encuentran en la kb centralizada en GXServer.

Ejemplo



Peter y Mike están sincronizados con la misma KB administrada en GXserver, y tienen inicialmente la misma definición de la transacción Employee.

¿Qué sucederá si ahora tanto Mike como Peter deciden seguir agregando nuevos atributos a la transacción Employee

Ejemplo

Peter



Name	Type
Employee	Employee
EmployeeId	Id
EmployeeName	Name
EmployeeAddress	Address
EmployeePhone	Phone
EmployeePhoto	Image

Commit



Update



Name	Type
Employee	Employee
EmployeeId	Id
EmployeeName	Name
EmployeeAddress	Address
EmployeePhone	Phone
EmployeeSalary	Numeric(4,0)
EmployeePhoto	Image

Name	Type
Employee	Employee
EmployeeId	Id
EmployeeName	Name
EmployeeAddress	Address
EmployeePhone	Phone
EmployeePhoto	Image

Por ejemplo supongamos que Mike decide agregar el atributo EmployeeSalary, y Peter decide definir el atributo EmployeePhoto.

Peter intenta realizar primero la acción de Commit. Así que va a Knowledge Manager / Team Development, ingresa el comentario y presiona Commit.

Mike no sabe lo que Peter hizo, y como él mismo ha hecho una nueva definición decide integrarla al Server y realiza también la acción Commit..... y presiona Commit..... ¿pero qué sucede?

GeneXus le avisa que en GXserver hay una nueva definición sobre ese objeto que está intentando subir y por lo tanto antes de efectuar su commit deberá hacer una operación de update para recibir esas modificaciones en su kb local.

Así que realiza la operación Update y ahora en su transacción Employee tiene incluido el atributo EmployeePhoto. O sea, se realizó el merge correspondiente.

Ejemplo

Peter



Name	Type
Employee	Employee
EmployeeId	Id
EmployeeName	Name
EmployeeAddress	Address
EmployeePhone	Phone
EmployeePhoto	Image



Name	Type
Employee	Employee
EmployeeId	Id
EmployeeName	Name
EmployeeAddress	Address
EmployeePhone	Phone
EmployeeSalary	Numeric(4,0)
EmployeePhoto	Image

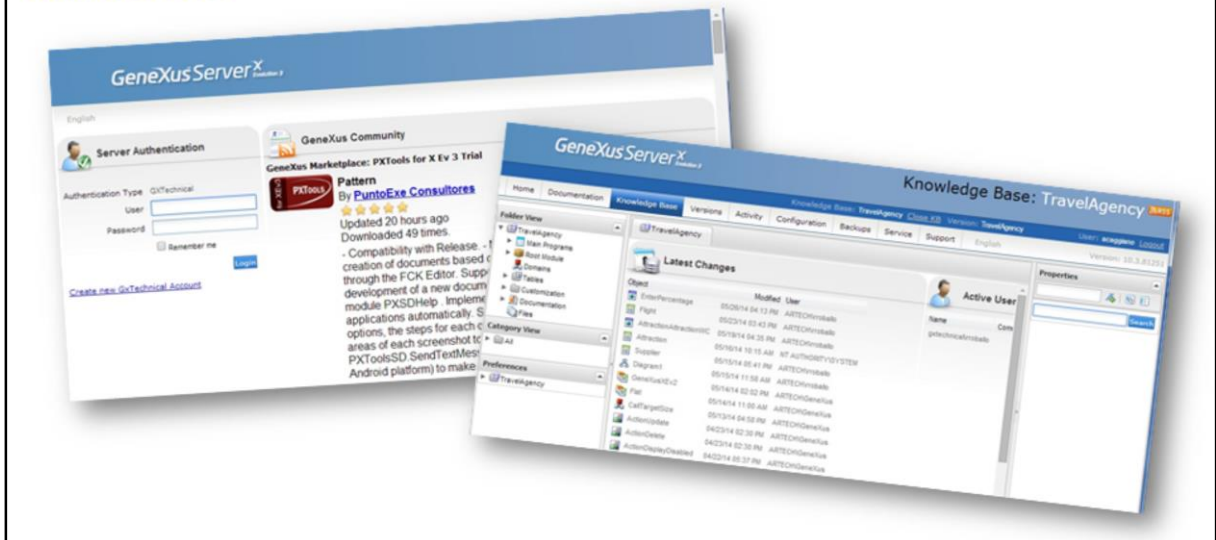
Commit



Name	Type
Employee	Employee
EmployeeId	Id
EmployeeName	Name
EmployeeAddress	Address
EmployeePhone	Phone
EmployeeSalary	Numeric(4,0)
EmployeePhoto	Image

Ahora sí Mike puede efectuar su commity finalmente en GXserver queda la definición completa de la transacción Employee.

Consola web



¿Es posible ver de alguna forma la KB que se va construyendo en el server? Si

A través de la url de ubicación de GXserver, es posible ingresar a su consola web y visualizar las bases de conocimiento publicadas.

Una vez autenticados con nuestro usuario de Gxtechnical, podemos ver la lista de kbs publicadas.

Ingresamos a nuestra Kb y podemos ver datos estadísticos y el registro de los comentarios ingresados.

Para ver el estado de la KB, elegimos la solapa Knowledge Base. Desde aquí podemos comprobar que la transacción Employee tiene efectivamente todas las definiciones integradas.

Vemos entonces la definición completa de su estructura, y también la definición de sus reglas.

Para salir de esta vista de la Kb presionamos Logout.

Estos son entonces los conceptos y acciones básicas en el uso de GXserver.

GeneXus[™]