Business Process Management Suite

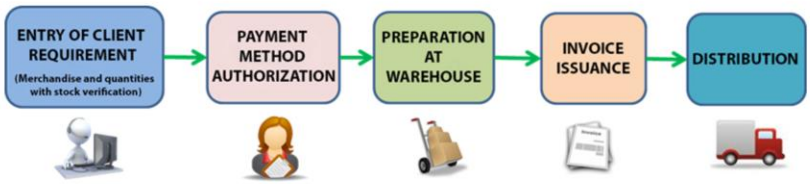
GeneXus™ 15

Brief theoretical introduction

- What is a business process?

  A business process is a set of logically related tasks which are performed to achieve a business result.

- Example



- What is BPM?

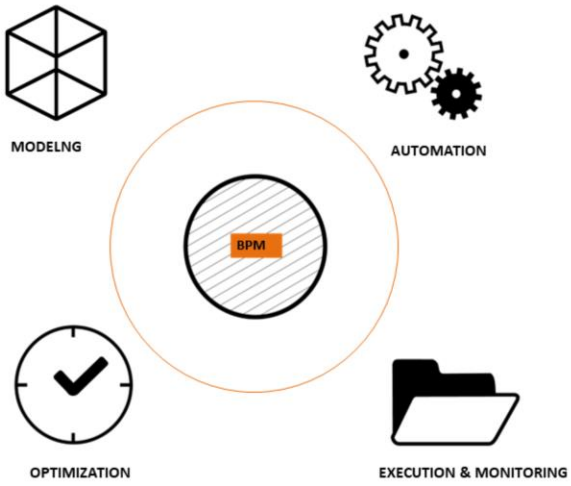
  The BPM (Business Process Management) is:
  - a methodology
  - a set of management tools

Un **proceso de negocio** es un conjunto de tareas relacionadas lógicamente llevadas a cabo para lograr un resultado de negocio.

En el ejemplo se está mostrando un proceso de negocio que se sigue en una empresa que vende mercadería “al por mayor”. Se pueden observar claramente las tareas consecutivas que se siguen para llevar a cabo el proceso de venta.

BPM (Business Process Management) es una metodología que se enfoca en administrar y optimizar en forma continua las actividades y procesos de negocio de la organización, incluyendo prácticas y políticas de gestión, métricas y procesos de control, así como herramientas de software que soportan todas estas actividades.

BPM Life Cycle

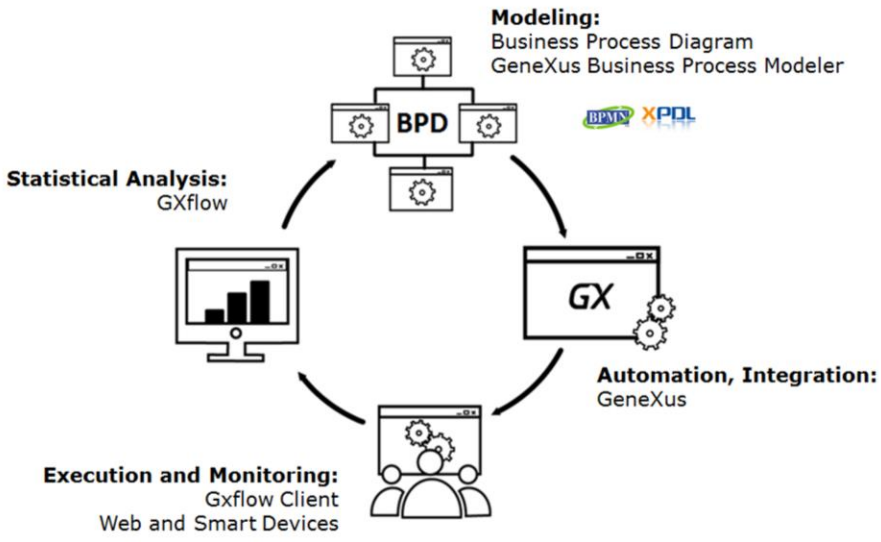


El **ciclo de BPM** comienza en la etapa de **modelado**, donde se describen los procesos de negocio de la organización. Generalmente se utilizan herramientas gráficas para comprender mejor los procesos.

Luego viene la etapa de **automatización** de esos procesos de negocio para lo cual utilizamos herramientas de software.

A continuación comienza la **ejecución y monitoreo** de estos procesos donde se analiza qué tan eficientes son, dónde están los cuellos de botella y que mejoras podemos se pueden implementar para mejorarlos en la siguiente etapa para obtener los resultados esperados.

Dado que es un ciclo, la metodología orientada a procesos es un **proyecto que no tiene fin** y el foco es la **mejora continua** de la organización.


Modeling:
Business Process Diagram
GeneXus Business Process Modeler
BPD
BPMN
XPDL
Statistical Analysis:
GXflow
GX
Automation, Integration:
GeneXus
Execution and Monitoring:
Gxflow Client
Web and Smart Devices

1. Productividad:

Algo que ayuda mucho en todo esto es tener la posibilidad en todo momento de saber en qué estado está un proceso y quiénes tienen las tareas en un momento dado. Todo esto redunda en un aumento de la productividad dado que se ahorra mucho tiempo de gestión y operativa en determinar cómo se debe continuar con determinado proceso y obviamente esto trae como consecuencia un ahorro en dinero.

2. Procesos de negocio optimizados

El trabajar sobre los proceso de negocio de la empresa hace que hagamos un análisis introspectivo y estemos continuamente optimizándolos.

3. Mayor visibilidad y control:

Otro tema importante es que se gana en control pues en cualquier momento podemos saber el status de los procesos y actuar proactivamente por ejemplo para delegar trabajo o reasignar tareas que aún no se han procesado.

4. Mejora en la comunicación:

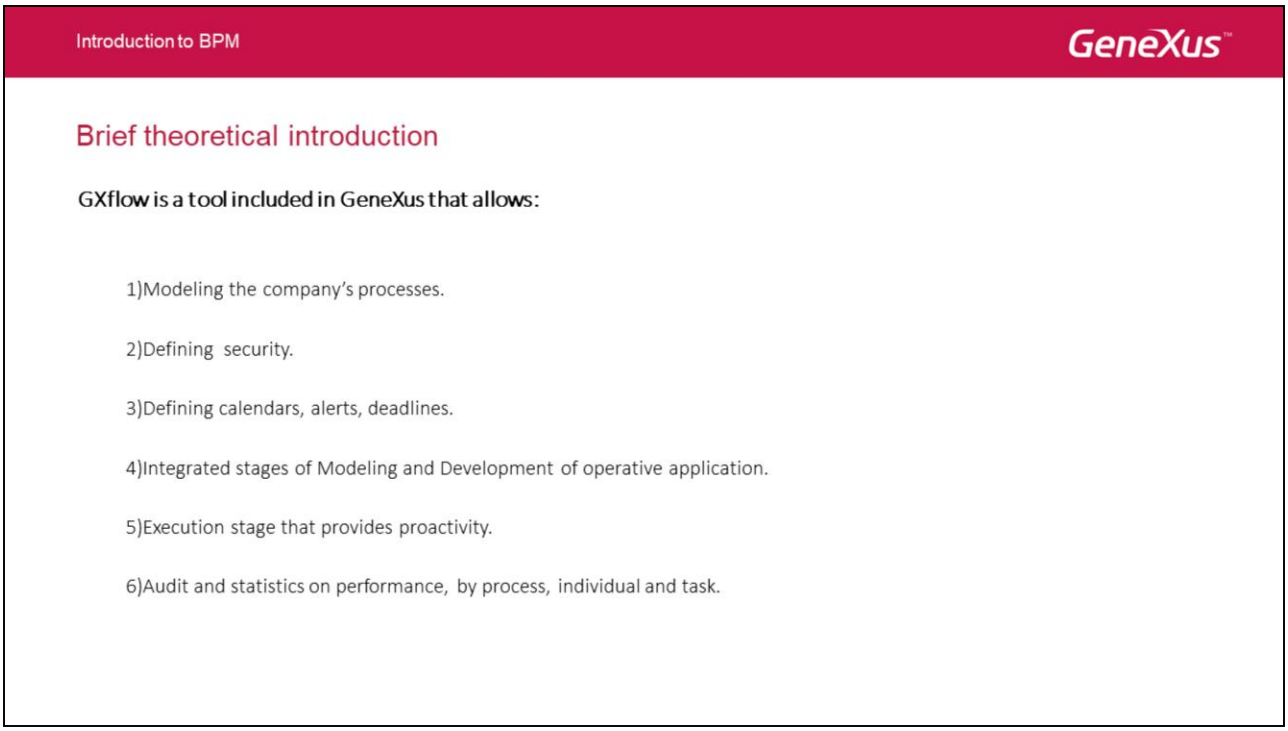
El hecho de tener bien definido como se deben hacer las cosas y que cada uno tiene claro el rol que debe cumplir pero además conoce como se desarrolla todo mejora enormemente la comunicación entre los actores y agilita mucho el pasaje de información en las distintas etapas.

5. Flexibilidad ante cambios

El hecho de tener claro como están definidas todas las etapas de un proceso nos permite evaluar el impacto de cualquier cambio que se haga en el mismo, ya sea un cambio para optimizar o un cambio obligado porque hay que agregar nuevas tareas por cambios en regulaciones. El hecho de poder medir este riesgo ante los cambios le da a la empresa mayor

flexibilidad y de esta forma puede ir haciendo evolucionar sus procesos a los cambios que le dicten los mercados con los cuales interactúa.

Desde el punto de vista del desarrollo la ganancia viene por el lado de que vamos a “declarar” más y programar menos, vamos a separar toda la lógica del control de flujos de nuestra programación lo cual hará que sea mucho más entendible lo que hacemos y por lo tanto mucho más fácil de mantener.



GXflow es una herramienta integrada a GeneXus que nos permite y Brinda:

1) **Modelar los procesos de la empresa:** Diagramar procesos nos permite cambiar el orden de las tareas, agregar y remover tareas, y/o cambiar las condiciones para la ejecución, sin tener que cambiar el código de los objetos.
2) **Definir seguridad:** Los Roles son definidos con las tareas que cada uno puede realizar. Esto evita tener que incluir código en los objetos con propósitos de seguridad.
3) **Definir calendarios, alertas, y deadlines.**

4) Etapas de Modelado y Desarrollo de aplicación operativa integradas: En GeneXus 15, relacionar objetos GeneXus desarrollados que implementan la aplicación operativa con los diagramas que modelan procesos es fácil y simple. Veremos cómo los objetos son arrastrados a los diagramas y qué conveniente es tener modelado de procesos integrado al desarrolla de la aplicación operativa.

5) Etapa de ejecución que brinda proactividad: En ejecución, cada usuario verá inmediatamente las tareas que tiene para hacer (sin ser necesario buscar tareas pendientes en la aplicación)

6) Auditoría: GXflow permite ver el estado de cada usuario, el tiempo que un usuario precisa para realizar cada tarea, etc.

Brief theoretical introduction

Steps for working with GXflow:

MAY BE IN PARALLEL

- Create GeneXus objects to describe reality and processes
- Create business process diagrams to model processes
- Relate GeneXus objects to business process diagrams elements
- Execute process

Basic concepts to create business process diagrams

- How do we create a business process diagram?

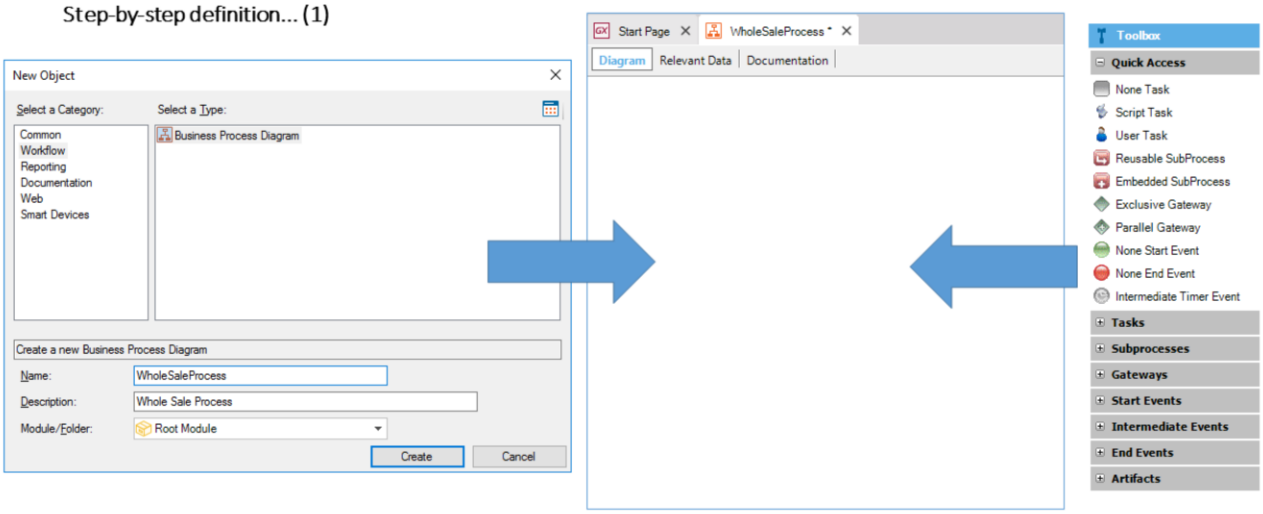
  - By creating a **Business Process Diagram** object using the GeneXus IDE
  - By using the Business Process Modeler (stand alone tool)

- How many business process diagrams will we define in our KB?

  As many as the company has.

Basic concepts to create business process diagrams

Step-by-step definition... (1)

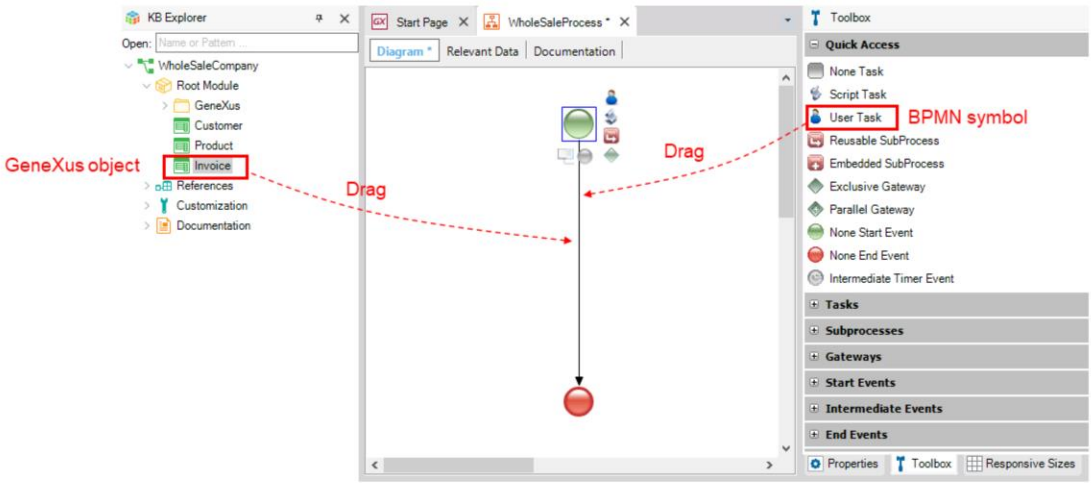


Luego de crear el objeto Business Process Diagram, incluiremos en el mismos los símbolos que formarán el diagrama.

La notación que se utiliza para los diagramas es BPMN (Business Process Modeling Notation o Notación para el Modelado de Procesos de Negocio)

Basic concepts to create business process diagrams

Step-by-step definition...(2)

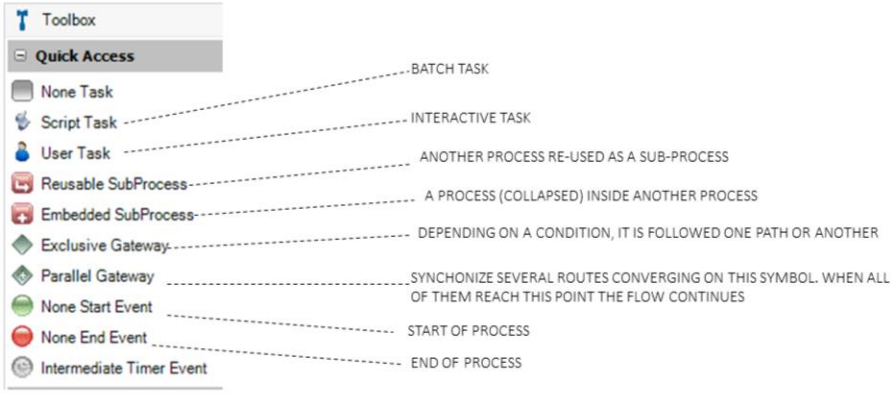


Cuando desde el “KB Explorer” arrastramos un objeto transacción o web panel al diagrama, estamos agregando una “tarea/actividad interactiva” al flujo, **que permanecerá con ese objeto asociado en etapa de ejecución.** Dicha “tarea/actividad interactiva” agregada al diagrama, será automáticamente nominada con el mismo nombre que el de la transacción o web panel arrastrado.

Por otro lado, si agregamos una “tarea/actividad interactiva” al diagrama arrastrando el símbolo correspondiente desde la toolbox disponible para crear diagramas de procesos de negocios, **necesitaremos asociar esa tarea a un objeto transacción o web panel definido en la KB.** Para esto, luego de agregar la tarea al diagrama y seleccionarla, simplemente tenemos que editar sus propiedades (F4 para abrir el diálogo Properties) y completar la propiedad **Web Application** con el objeto correspondiente, y la propiedad **Name** con el nombre elegido para la “tarea/actividad interactiva”.

Basic concepts to create business process diagrams

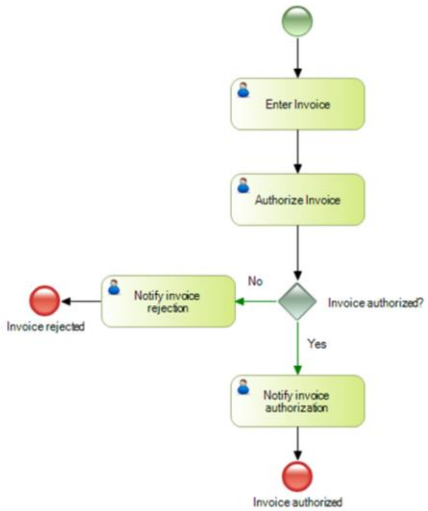
Description of symbols



GeneXus

Basic concepts to create business process diagrams

Step-by-step definition ... (3)



Aquí podemos ver que al diagrama de proceso creado le hemos agregado tareas interactivas y una condición.

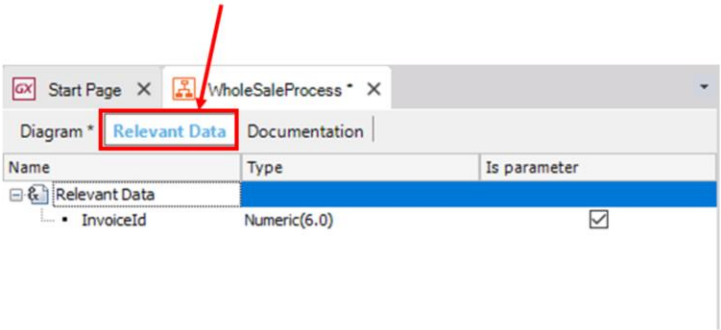
Haciendo doble click en las flechas verdes que salen de la condición, vemos que el editor de condiciones se abre para editar cada una (y entonces cuando cualquiera de las dos condiciones se cumple, un flow específico de actividades le seguirá).

Ahora veremos el concepto de “DATOS RELEVANTES” (relevant data), que es básico para entenderla información que podemos incluir en las condiciones.

GeneXus

Basic concepts to create business process diagrams

RELEVANT DATA ≈ GLOBAL VARIABLE IN DIAGRAM



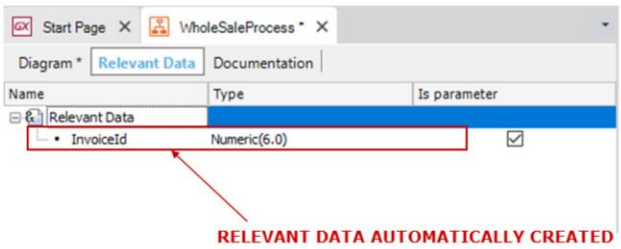
El concepto de “Datos Relevantes” se utiliza para especificar “variables globales” en un proceso

Este concepto permite administrar el pasaje de información entre las tareas y que la información sea conocida en todo el flujo

Basic concepts to create business process diagrams

RELEVANT DATA (Contd.)

- When we drag a transaction from "KB Explorer":



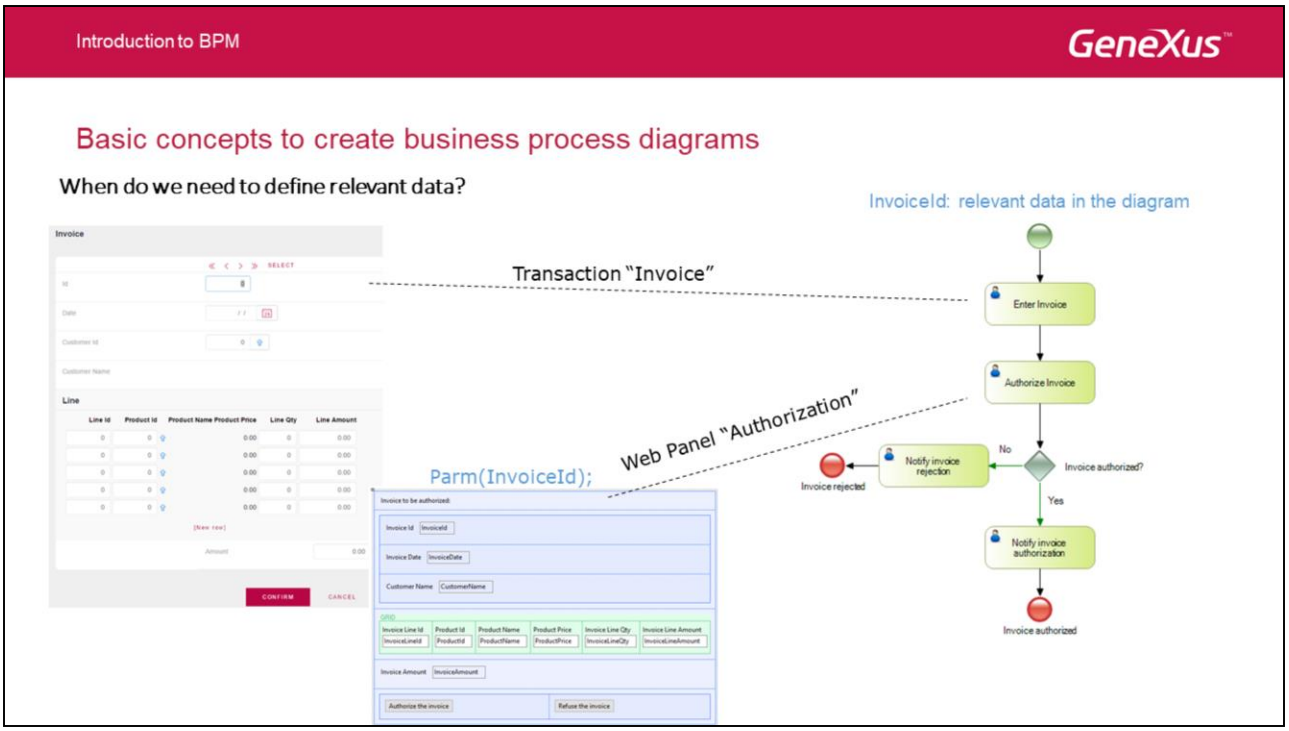
- Same name and data type that Transaction Primary Key

Cuando se arrastra desde "KB Explorer" una transacción a un diagrama de proceso, automáticamente se crea un dato relevante con el mismo nombre e igual tipo de dato que la clave primaria de la transacción.

Para los datos relevantes definidos automáticamente con el mismo nombre y tipo que las llaves primarias en las transacciones (como InvoiceId en este ejemplo), hay una relación automática entre el dato relevante y el atributo llave primaria.

Esto significa que cuando modelamos el diagrama, "los datos relevantes son los datos globales conocidos en este contexto", y en los objetos GeneXus que desarrollamos asociados con las actividades del diagrama "recibimos, en la regla Parm, el atributo llave primaria" (con la misma información o en el contexto del diagrama, o en el contexto del desarrollo de funcionalidad, respectivamente). En otras palabras, hay un mapeo automático entre los datos relevantes con el mismo nombre y tipo que un atributo llave primaria y ese atributo llave primaria.

Incluso cuando la mayoría de los datos relevantes en un diagrama de proceso coincide con llaves primarias, también hay casos donde necesitamos definir explícitamente otros datos relevantes. Veremos ejemplos de esto.



Al principio, explicamos que cuando trabajamos en GeneXus con workflow, seguimos estos pasos básicos:

•Crear objetos GeneXus

•Crear diagramas de procesos de negocios

• Asociar objetos GeneXus a diagramas de procesos de negocios

• Ejecutar proceso

Así es como generamos un diagrama de proceso en nuestra KB. Previamente, habíamos definido algunos objetos GeneXus que habíamos arrastrado al diagrama, y desarrollamos otros objetos y los incluimos en el diagrama también. Más adelante veremos cómo asignar roles a diferentes actividades en el diagrama. Por ahora, la idea es que tenemos una transacción “Invoice” en la KB que fue arrastrada al diagrama como la primera actividad interactiva del procesos, y más tarde definiremos que esa tarea interactiva puede ser ejecutada por los vendedores de la compañía.

Cuando un vendedor ingresa una venta (a través de la transacción “Invoice”), la entrada va a incluir el cliente, la mercancía requerida, la cantidad, el método de pago pedido por el cliente, y la venta será guardada con un número interno (diferente del número de factura formal), y allí es donde termina la primera actividad. Luego, esa venta será evaluada por un supervisor (que evaluará el cliente involucrado, la cantidad, y el método de pago) que aceptará o rechazará la venta. Esta segunda actividad interactiva fue agregada al diagrama arrastrando el web panel “Authorization”.

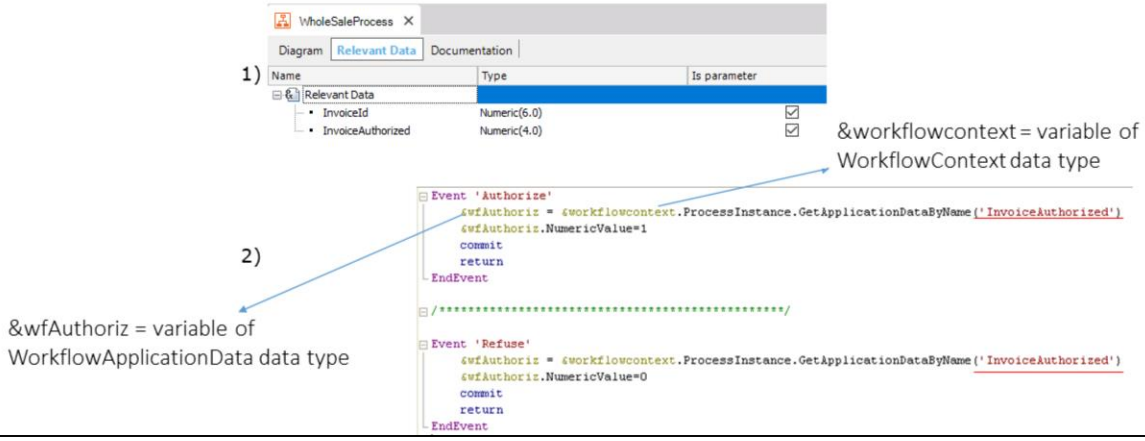
Entonces, a nivel del diagrama de proceso, tenemos los datos relevantes InvoiceId (con el mismo nombre y

tipo de datos que el atributo *InvoiceId*), que significa que los datos son conocidos a lo largo del flow de actividades. En lo que concierne a los objetos GeneXus relacionados al diagrama, el web panel “Authorization” implementa la segunda actividad del proceso, y recibe el atributo *InvoiceId* por parámetro. En el form del web panel “Authorization” se verán los datos de la factura, además de el límite de crédito y el balance del límite de c´redito del cliente, y estará la posibilidad de ver la historia de facturas anteriores para ese cliente. Este web panel ofrecerá al supervisor dos botones: “Authorize” (Autorizar) y “Refuse” (Rechazar).

Basic concepts to create business process diagrams

Step-by-step definition ... (4)

We create a new relevant data: **InvoiceAuthorized**



Hemos definido entonces los datos relevantes en el diagrama del proceso (llamado InvoiceAuthorized), y luego, en el web panel “Autorization”, los datos relevantes son leídos en la variable &wfAuthoriz del tipo de datos WorkflowApplicationData. El tipo de datos WorkflowApplicationData significa “datos relevantes”.

La variable &WorkflowContext también es usada para acceder a los datos de la instancia del proceso, a través de la siguiente consulta:

&WorkflowContext.ProcessInstance.GetApplicationDataByName('InvoiceAuthorized').

El método WorkflowProcessInstance significa “instancia del proceso”. Si entonces analizamos la primera línea de código en cualquiera de los dos eventos, estamos recuperando, en la variable &wfAuthoriz (del tipo “datos relevantes” (relevant data)), el valor del dato relevante llamado “InvoiceAuthorized” (el nombre del dato relevante va entre comillas, como definimos en el diagrama del proceso) perteneciente a la instancia del proceso que se está ejecutando.

Luego, en la segunda línea de código en ambos eventos, **el valor 0 o 1, como es el caso, es asignado a la variable &wfAuthoriz (del tipo “relevant data”)** (usando la propiedad Numeric del tipo de datos WorkflowApplicationDatatype, porque un valor numérico es asignado).

Por último, un commit y un return son realizados en cada evento, porque el valor del dato relevante ya está recuperado con el valor deseado, y la idea es completar esta actividad.

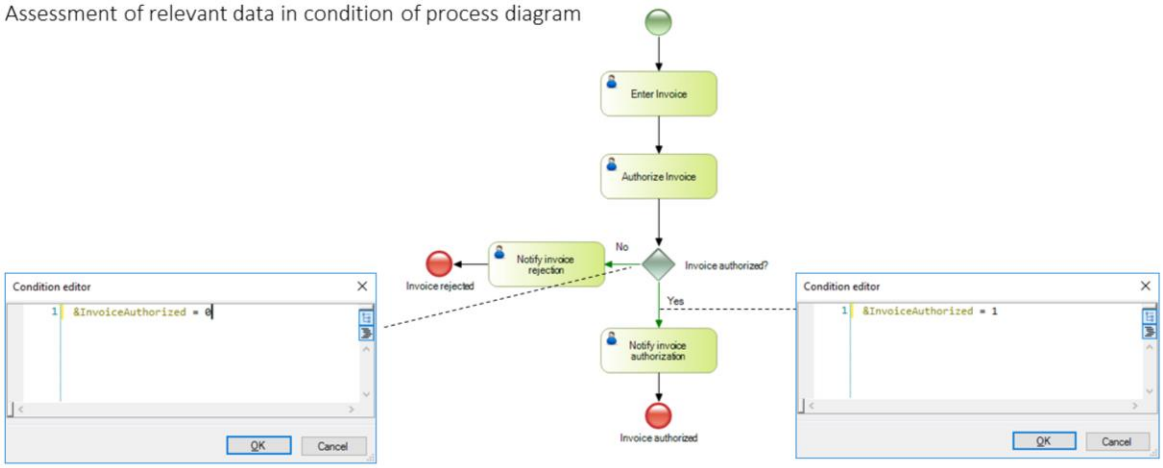
Así es como leemos y cargamos datos relevantes en objetos GeneXus asociados con las actividades en un diagrama de proceso, usando tipos de datos, propiedades, y métodos del workflow.

Sin embargo, para datos relevantes con el mismo nombre y tipo que las llaves primarias, esto no es necesario (ya que los atributos primarios son recibidos directamente por parm, y ya hay mapeo automático con los datos relevantes correspondientes).

Basic concepts to create business process diagrams

Step-by-step definition ... (5)

Assessment of relevant data in condition of process diagram



El valor del dato relevante &InvoiceAuthorized está disponible en todo el diagrama. (Throughout the diagram we have the relevant data &InvoiceAuthorized...)

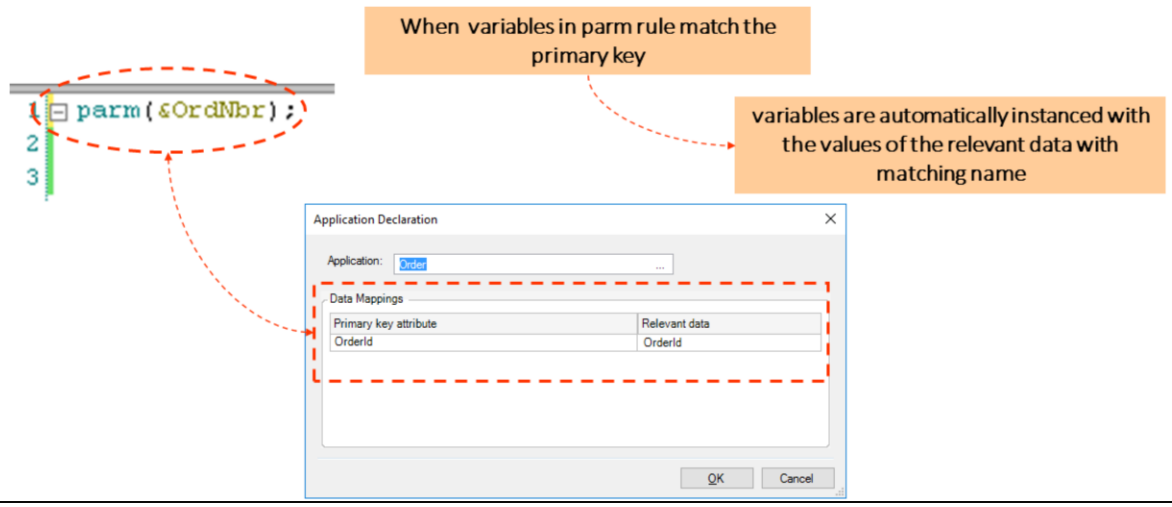
Observemos que en cada uno de los caminos verdes saliendo del diamante de condición, el valor del dato relevante &InvoiceAuthorized es evaluado.

Incluso si no incluimos el & (ampersand) cuando definimos el nombre del dato relevante, el diálogo para definir datos relevantes claramente muestra el ícono &, indicando que hay variables involucradas.

Entonces, para referenciar a datos relevantes en el flow, referenciamos con &, como se muestra en las dos capturas correspondientes al editor de condiciones.

Basic concepts to create business process diagrams

Relevant Data mapping in Transactions with parameters

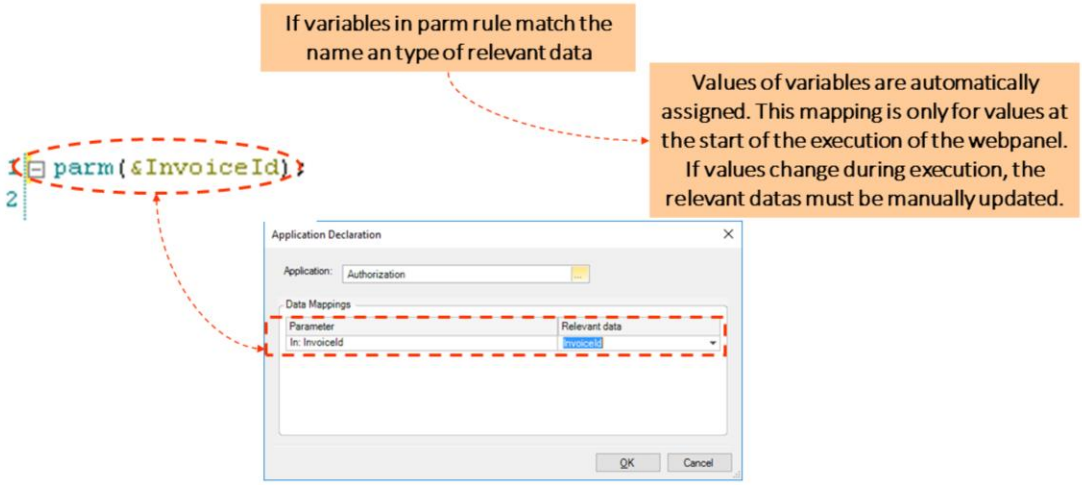


Cuando desde un diagrama de procesos se invoca a un objeto transacción, ¿cómo se realiza el mapeo de los datos relevantes en una transacción cuando recibe parámetros?

Cuando las transacciones reciben por parámetro la clave primaria de la transacción y la reciben directamente en el atributo o en variables basadas en la clave primaria (con el mismo nombre y tipo) , se realiza un mapeo automático con los datos relevantes, es decir, al momento de la invocación se asigna automáticamente el parámetro con el valor del dato relevante que tiene igual nombre.

Basic concepts to create business process diagrams

Relevant Data mapping in Web panels with parameters



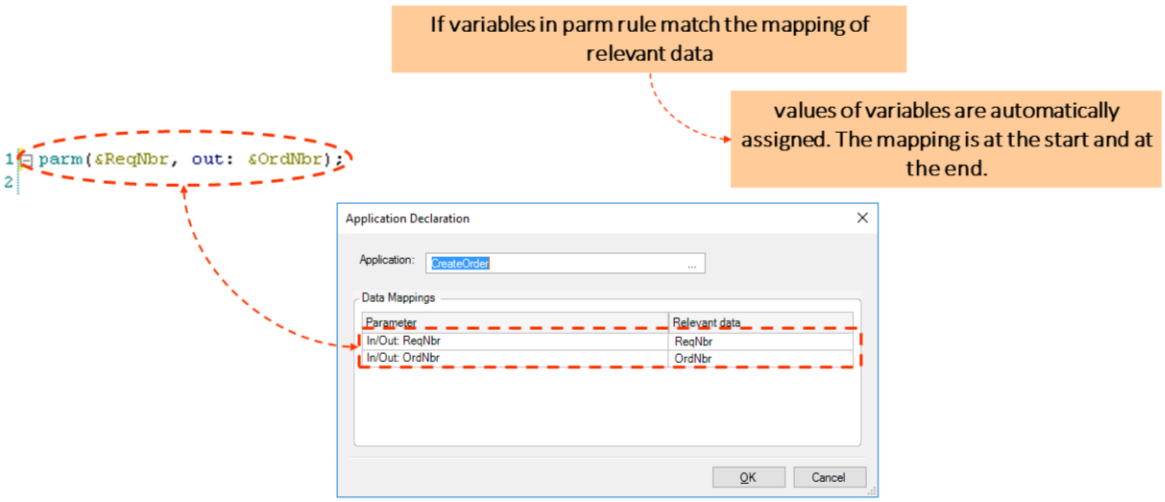
Cuando desde un diagrama de procesos se invoca a un web panel, ¿cómo se realiza el mapeo de los datos relevantes en webpanels cuando reciben parámetros?

Si las variables o atributos en regla parm coinciden en nombre con el dato relevante, se asignan automáticamente los valores de estas variables o atributos. Este mapeo es solo para los valores a la entrada del webpanel (es decir cuando se invoca al mismo), a la salida no se actualiza el valor del dato relevante, es decir, si durante la ejecución del webpanel cambió el valor de la variable, hay que actualizar manualmente al dato relevante del diagrama.

Nota: Las variables tienen que tener el mismo nombre que los atributos.

Basic concepts to create business process diagrams

Relevant Data mapping in Procedure with parameters



Cuando desde un diagrama de procesos se invoca a un objeto Procedimiento ¿cómo se realiza el mapeo de los datos relevantes en procedimientos cuando reciben parámetros?

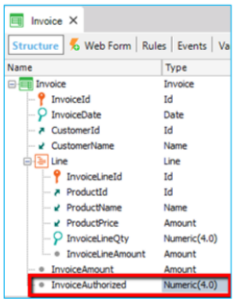
Si las variables o atributos en la regla parm coinciden con el mapeo de los datos relevantes (es decir las variables o atributos coinciden en nombre y tipo con los datos relevantes), se asignan automáticamente los valores de las variables o atributos con los valores de los datos relevantes (al comienzo de la ejecución del procedimiento) y viceversa (al final de la ejecución del mismo).

El mapeo es a la entrada y a la salida, es decir, cuando desde el diagrama se invoca al objeto procedimiento, los valores de los datos relevantes de cargan automáticamente en las variables o atributos de la regla parm del procedimiento, y cuando finaliza el procedimiento, los valores de las variables o atributos (que eventualmente pueden haber cambiado) se cargan automáticamente a los datos relevantes del diagrama.

GeneXus

Basic concepts to create business process diagrams

Another possible solution to solve steps (4) and (5) – Step 1

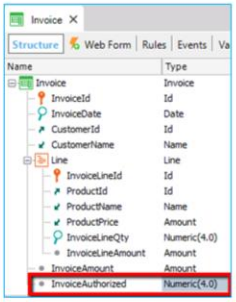


1)Define *InvoiceAuthorized* attribute in "Invoice" transaction

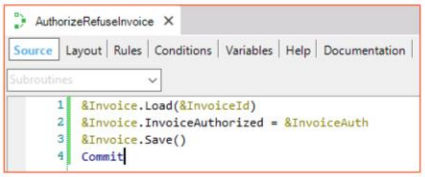
Creamos al atributo InvoiceAuthorized en la transacción Invoice y lo utilizamos para almacenar, en cada factura si la misma fue autorizada o no.

Basic concepts to create business process diagrams

Another possible solution to solve steps (4) and (5) – Step 2



2) Web panel "Authorization" invokes procedure AuthorizeRefuseInvoice who saves the value

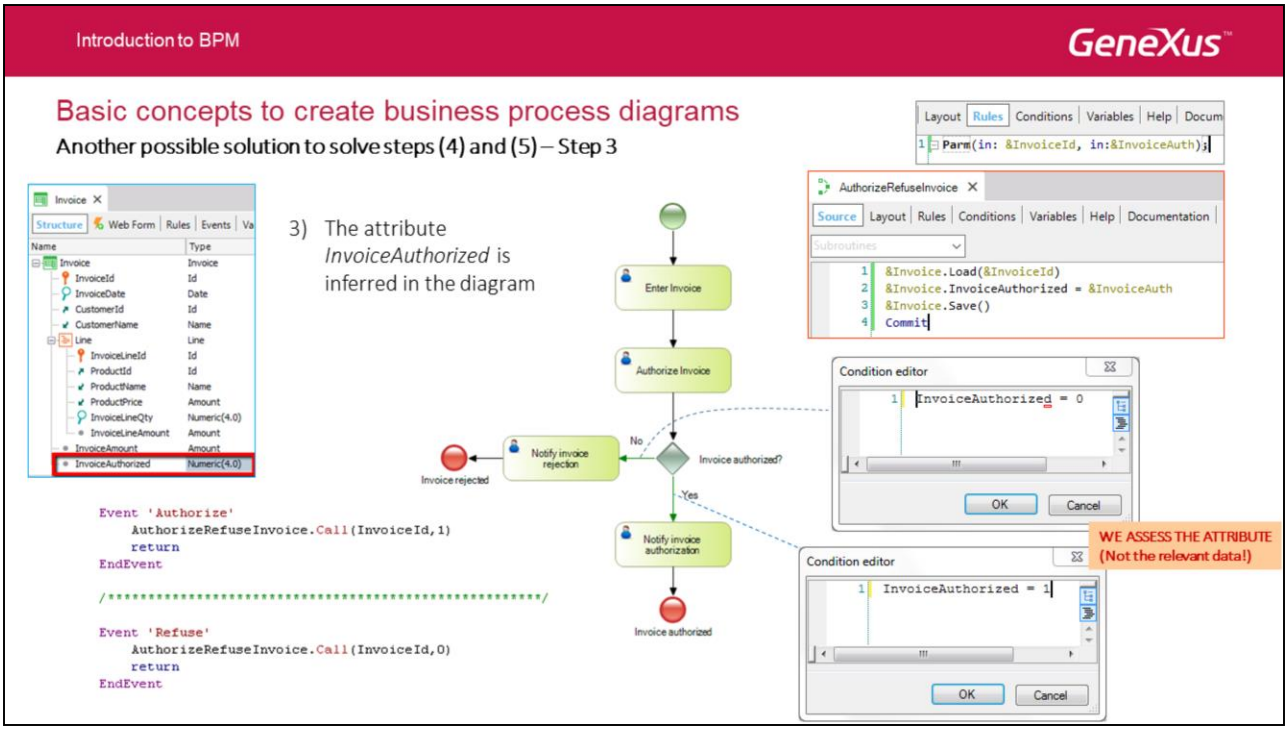



```
Event 'Authorize'
    AuthorizeRefuseInvoice.Call(InvoiceId,1)
    return
EndEvent

/*****************************************************/

Event 'Refuse'
    AuthorizeRefuseInvoice.Call(InvoiceId,0)
    return
EndEvent
```

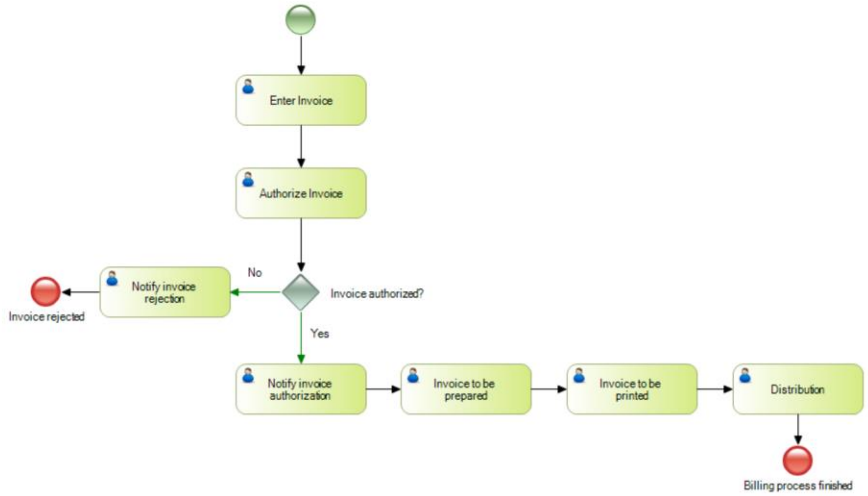
Desde el webpanel Authorization Invocamos al procedimiento AuthorizeRefuseInvoice quien guarda el valor 1 o 0, en el atributo InvoiceAuthorized.



Con esta solución, no necesitamos definir el dato relevante InvoiceAuthorized, porque en un atributo tenemos la información de si la factura fue autorizada/rechazada…y en el diagrama de procesos podemos inferir atributos a través de llaves primarias, y evaluar el valor de los atributos directamente.

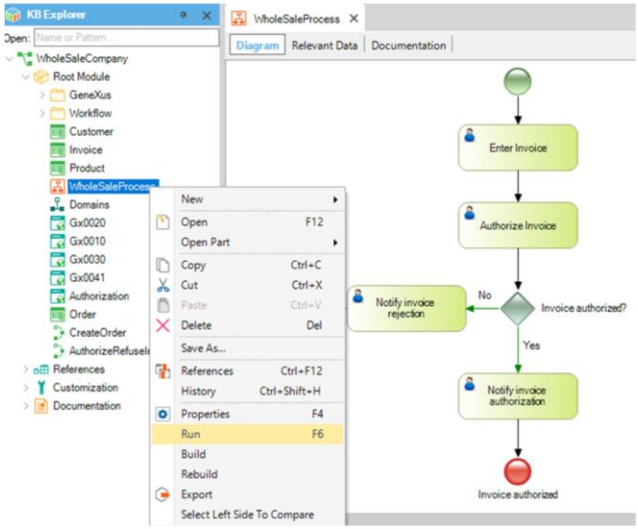
Basic concepts to create business process diagrams

Complete process diagram, corresponding to the process we are creating:



Basic concepts to create business process diagrams

To execute a process we must select the Run option



Para ejecutar un proceso con el prototipador de GXflow se debe seleccionar la opción Run sobre el mismo. Al hacerlo el proceso de Build de GeneXus se encargará de realizar todas las acciones necesarias para poder ejecutar el proceso.

Algunas de estas acciones son:

-**Creación o reorganización de las tablas de workflow**: estas tablas son requeridas por el motor de workflow. Serán creadas si se detecta que no existan en la base de datos de la aplicación o reorganizadas en caso de que las tablas existentes correspondan a una versión anterior de GXflow.

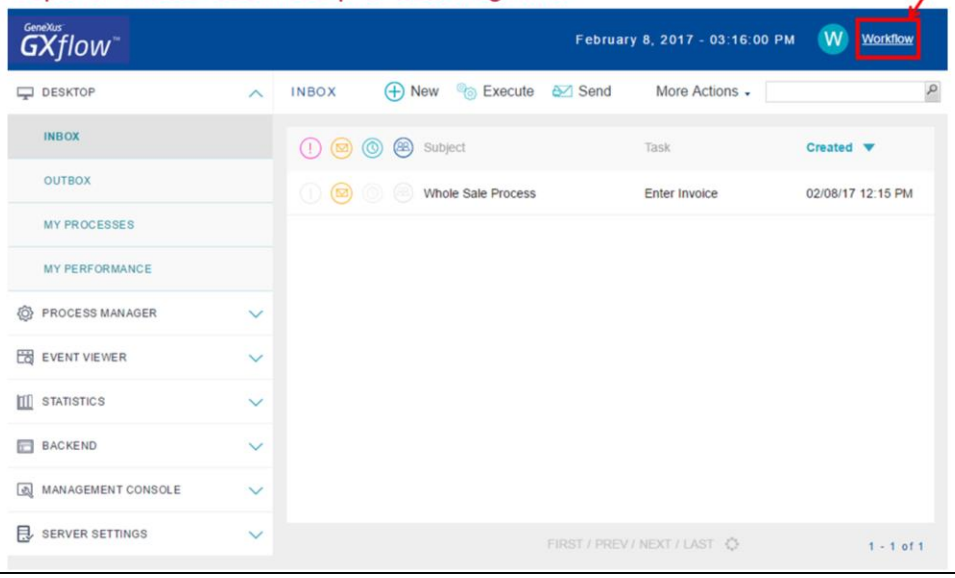
-**Deploy de procesos:** disponibiliza los procesos para que puedan ser ejecutados por el motor de workflow. Esta operación se ejecuta solamente si se crearon o modificaron diagramas.

-**Deploy del runtime de workflow:** los binarios y recursos que componen el motor y cliente de GXflow son integrados con los binarios y recursos de nuestra aplicación. Esta operación solamente se ejecutará la primera vez que se hace un build teniendo diagramas de procesos o cuando se detecte que en la instalación de GeneXus se tiene una versión más nueva del runtime de GXflow.

-Adaptación de objetos asociados a tareas de procesos: para cada uno de estos objetos se genera el código necesario para que puedan ser invocados por el motor de workflow.

Basic concepts to create business process diagrams

Execution:



Al ejecutar el diagrama se abre una ventana en el navegador, mostrando la ventana principal del Cliente de Workflow. Esta pantalla similar a una bandeja de entrada de mails. Allí identificaremos a la tarea pendiente de ser ejecutada y el proceso al cual pertenece.

La tarea pendiente es para el usuario **Workflow**, que es el usuario administrador utilizado por defecto en la prototipación. Para cambiar el usuario, debemos cambiar el modo de ejecución.

Hay dos métodos de ejecución:

- Prototyper
- StandardClient

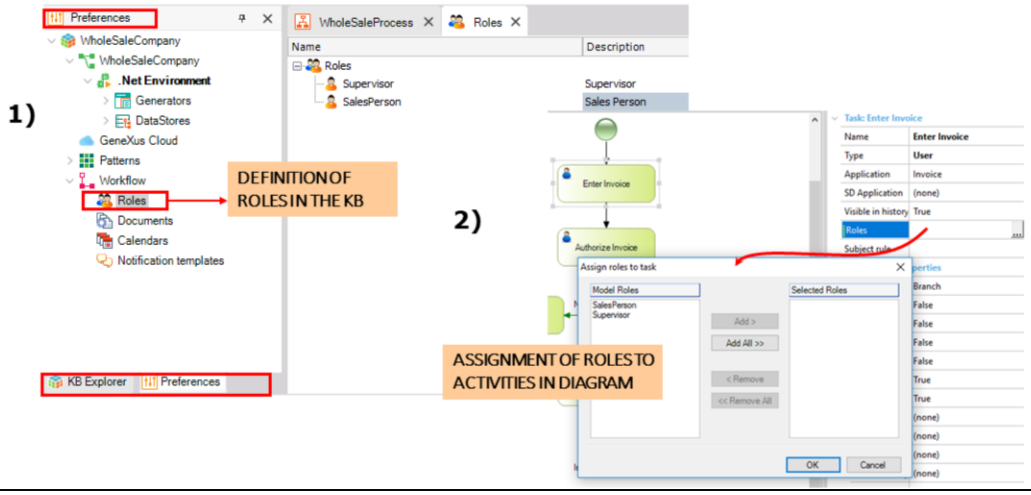
En la sección Preferences→Environment→Properties podemos configurar el modo que deseemos.

El método “Prototyper” está, por supuesto, orientado a la etapa de prototipación, y el método “StandardClient” está orientado a la etapa de producción. Las pruebas de seguridad no son importantes durante la etapa de prototipación, entonces cuando ejecutamos la aplicación con el método “Prototyper”, todos las instancias de prueba anteriores son borradas y existe la posibilidad de ejecutar todas las actividades sin controlar los roles.

Cuando ejecutamos la aplicación en el método “StandardClient”, nada es abortado al comienzo, y se deben crear usuarios (para el Login), cada uno con su lista de roles correspondiente, porque en este caso, la seguridad sí será controlada.

Basic concepts to create business process diagrams

Role definitions:



La definición de roles se hace al nivel de la KB, en la sección de Preferences.

Seleccionando cada actividad en el diagrama y presionando F4, en la propiedad **Roles** podemos asignar la lista de roles que van a ejecutar esa actividad.

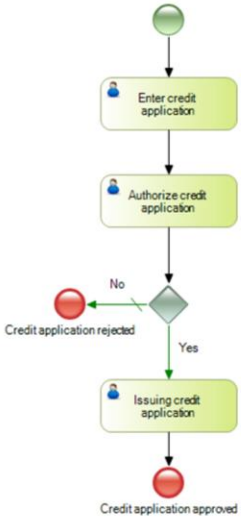
Una vez que los roles son asignados, para chequear la seguridad del proceso que modelamos, tenemos que cambiar el modo de ejecución. Para eso, en el grupo Workflow de las propiedades del Environment, seleccionamos la propiedad Execution Mode y asignamos el valor StandardClient.

Module 2

Process Modeling

GeneXus

What is business process modeling?



It is a representation of a **business process**

The business process may be:

- Analyzed
- Executed
- Improved

El modelado de procesos es la actividad de representar los procesos de negocio de una empresa con el objetivo de que estos puedan ser analizados, ejecutados y mejorados.

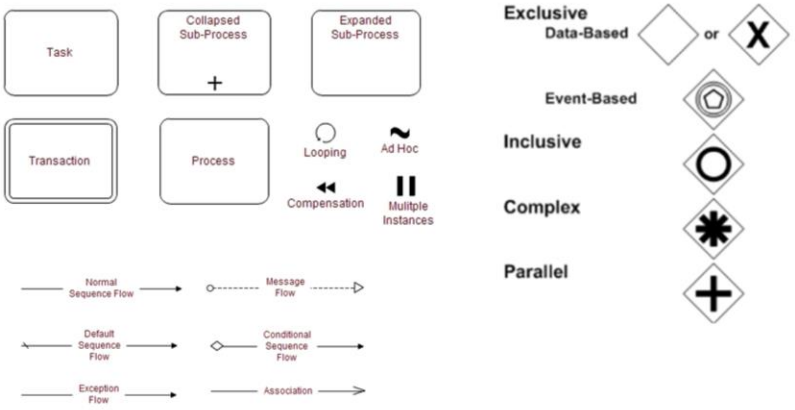
GeneXus

BPMN: Business Process Modeling Notation

- Graphic notation to represent a process flows
- Standard maintained by the OMG
- Currently in its version 2.0

Para llevar a cabo esta actividad veremos el estándar **BPMN (Business Process Modeling Notation)**, este estándar define **símbolos y reglas** que nos permiten representar gráficamente un proceso de negocio.

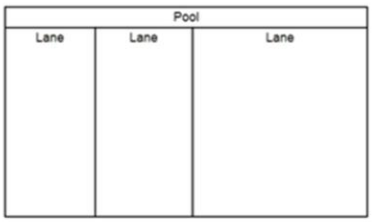
Symbols

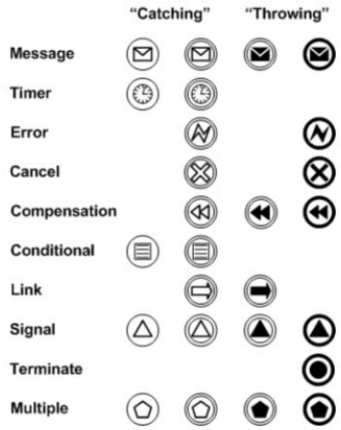


Los símbolos definidos por el estándar BPMN son **monocromáticos**, las herramientas como GeneXus que se basan en el estándar además agregan colores de forma que sea más claro para el usuario.

GeneXus

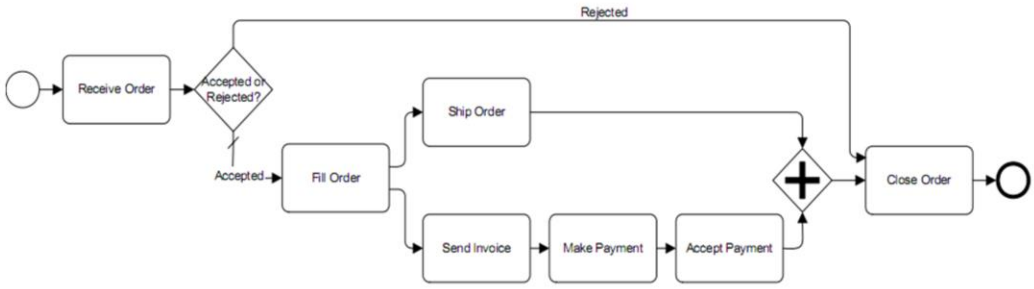
Symbols





GeneXus

Symbols



Activities

Entonces veamos como representar estas actividades, para esto contamos con el símbolo **Actividad**.

Una **actividad** representa cierto trabajo a ser realizado dentro del proceso de negocio.

Una actividad toma cierto tiempo para ser ejecutada y utiliza uno o más recursos de la organización.

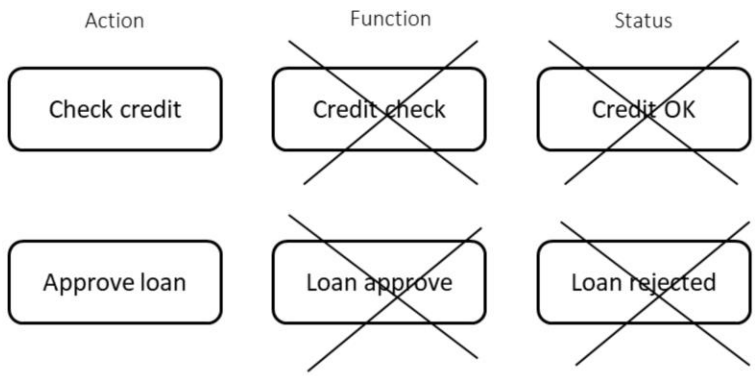
Requiere cierto tipo de entrada y usualmente produce algún tipo de salida.

Se representa gráficamente con un rectángulo redondeado.

Una actividad puede ser atómica (llamada Tarea) o compuesta (llamada Subproceso).

Activities

- They represent actions implemented in the process.
- They DO NOT represent statuses or functions.
- Their names must be of the type Verb – Noun.



Las actividades representan **acciones** que se llevan a cabo en el proceso.

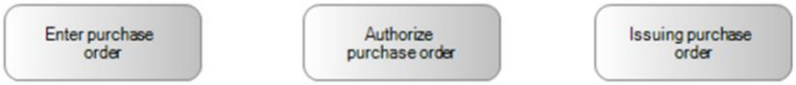
No representan estados del proceso y no son funciones.

Sus nombres deben ser siempre de la forma: Verbo (en infinitivo) - Sustantivo.

A través de los ejemplos podemos ver **cómo describir** las actividades de los procesos.

Activities

Example: Purchase process



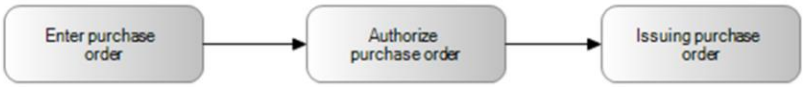
Supongamos que queremos representar un proceso de compra, en el cual se debe ingresar una orden de compra, autorizarla y emitir la misma. Por lo tanto, tenemos que representar tres actividades:

Ingresar Pedido de Compras

Autorizar Pedido de Compras

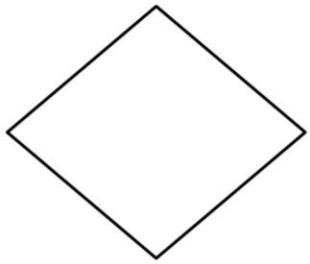
Generar Orden de Compras

Sequence Connectors



Para indicar el orden de precedencia de estas actividades utilizamos el símbolo **Conector de Secuencia**.

Gateways



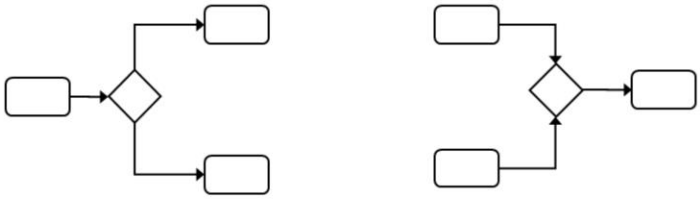
Hasta este punto el proceso que definimos es lineal, es decir que se procesan las tres actividades una a continuación de la otra sin excepciones. Pero en nuestro ejemplo luego de la tarea de autorización hay dos caminos posibles, uno es que se genere la orden de compras y el otro es que el proceso termine en base a la decisión tomada durante la autorización del pedido.

Los Gateways o Compuertas son elementos de modelado que controlan como un proceso Diverge o Converge. Cuando el flujo no necesita ser controlado entonces los gateways no son necesarios.

Todos los Gateways se representan con una figura de diamante, utilizándose distintos íconos en su interior para distinguir el tipo de Gateway.

Gateways

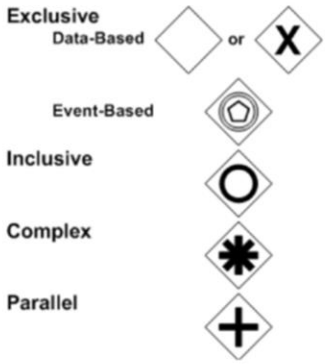
They are used to control branching and merging of flow paths in a process



Para esto vamos a utilizar el símbolo **Gateway**, ya que nos permite ramificar o unificar el proceso.

Gateways

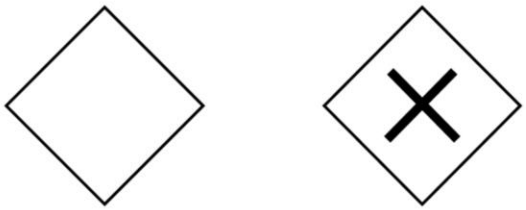
BPMN defines several types:



Exclusive branching

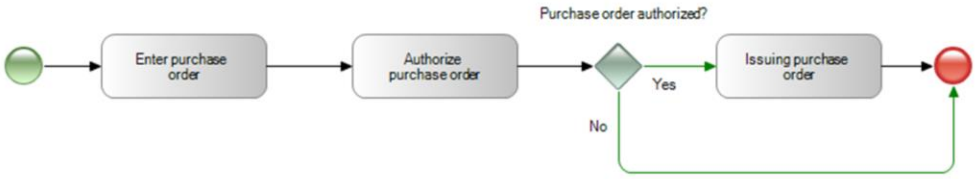
Exclusive gateway

It is used when the path of the process flow depends on the assessment of a condition of the true / false type.



En nuestro proceso vamos a utilizar un gateway de tipo **Exclusive** para determinar si se genera la orden de compras o no.

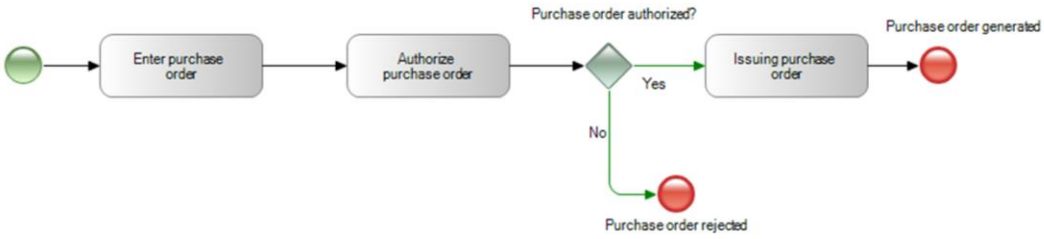
Gateways



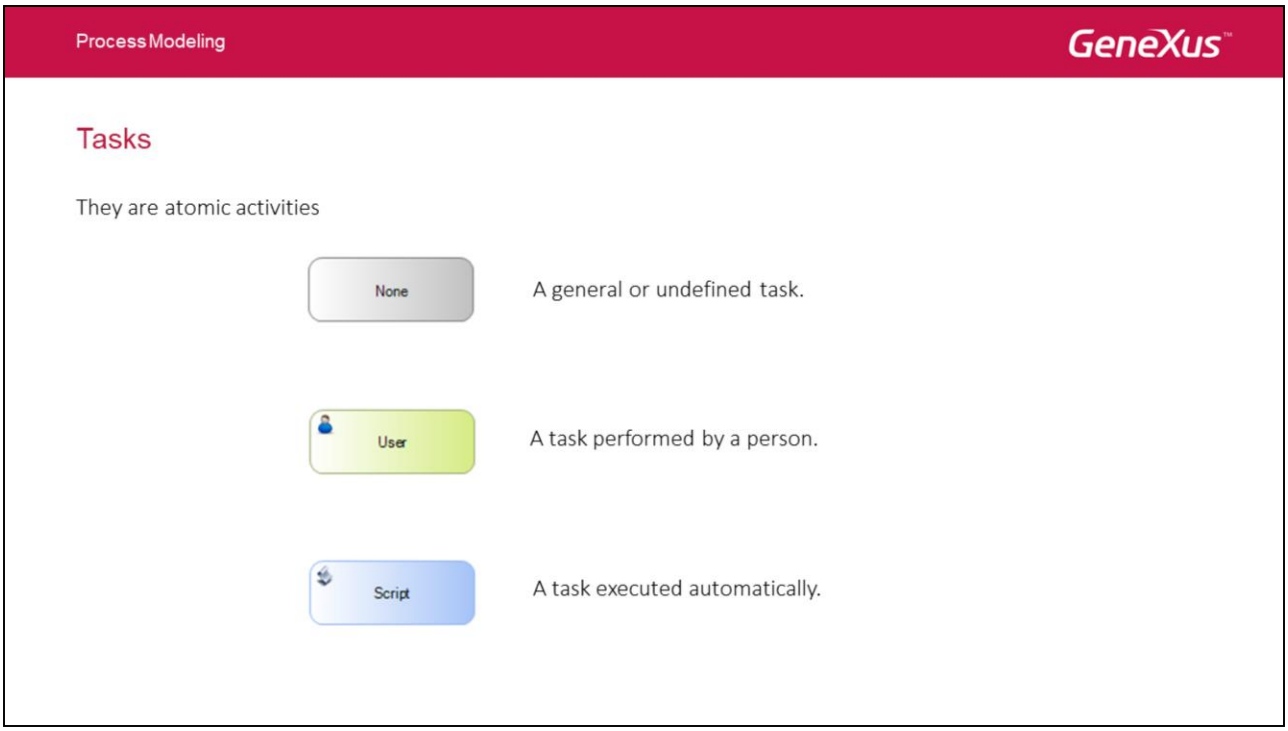
Lo colocamos luego de la tarea de autorización y antes de la generación de la orden de compra.

Como vemos estamos utilizando el mismo símbolo de fin para ambos caminos, BPMN establece que lo correcto es utilizar uno para cada camino.

Gateways



De esta forma podemos conocer cómo terminó el proceso, en este caso si el pedido fue autorizado o rechazado.



Las tareas son actividades atómicas (indivisibles).

Los diferentes tipos de tareas para modelar procesos son los siguientes:

None

Tarea genérica que se puede utilizar si aún no se tiene más información acerca de la naturaleza de la tarea

User

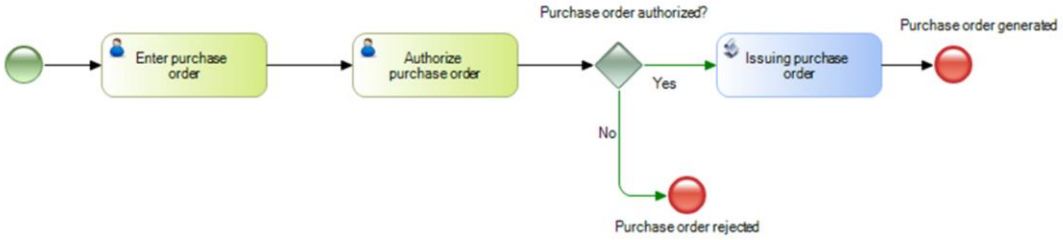
Tarea ejecutada por una persona en forma interactiva. Los usuarios pueden ver sus tareas en una interface similar a un Inbox.

Script

Tarea ejecutada de forma automática (batch), que no requiere la intervención humana. La tarea es ejecutada automáticamente por el workflow cuando está pronta para ser ejecutada. Cuando la ejecución del programa está completada, se completa la tarea.

Example

Purchase process



Ejemplo: **Proceso de compras**

1. Un **funcionario** ingresa un pedido de compras
2. El pedido es autorizado por el **jefe del funcionario**
3. Si el pedido es autorizado, **se genera automáticamente** la orden de compras o de lo contrario el proceso termina

Este ejemplo está compuesto por **tres actividades que deben ejecutarse en ese orden para obtener un resultado** que en este caso es la generación de la orden de compras.

En nuestro ejemplo tanto la tarea de ingreso del pedido de compras como la tarea de autorización deben ser realizadas por usuarios por lo tanto les asignamos el tipo **User.** Para el caso de la generación de la orden como es automática no requiere la interacción de un usuario, por lo tanto le asignamos el tipo **Script**.

Task Properties

Las tareas tiene asociadas diferentes propiedades de configuración. A continuación veremos algunas de ellas.

Task Properties

Application / SD Application

- Object that will be invoked upon executing the task.

Consult Application

- It shows the resulting information from the execution.

Preview Application

- It shows the basic data for the user to decide whether to take the task.

- **Application:** Permite indicar el objeto GeneXus que se invocará al ejecutar la tarea.

- **Consult Application:** Permite indicar el objeto GeneXus que se invocará desde la historia del proceso para mostrar la información resultante de la ejecución. Es accedida desde el log histórico (History Log).

- **Preview Application:** Permite indicar el objeto GeneXus que se invocará desde la opción Preview de una tarea para decidir si se debe o no tomar la tarea.

Task Properties

Roles

- Functional roles that can execute the task

Subject rule

- It allows changing the process instance subject

Request comments

- Forces the user to enter a comment in order to complete the task.

Send email

- Sends an email to those in charge, every time that a task instance is created.

-**Roles:** Se indican los roles que tendrán permisos para ejecutar esa tarea. La definición de los roles se realiza desde Preferences/Workflow/Roles.

- **Subject Rule:** Permite modificar el asunto de la instancia de proceso. Se pueden utilizar caracteres, atributos, datos relevantes. Por ejemplo se puede colocar "Customer registration number“ + CustomerId + &relevantData

- **Request comments:** Indica si el usuario tendrá que ingresar comentarios antes de finalizar la tarea.

- **Send email:** Indica que se envíe un email a los responsables de una tarea cada vez que se cree una instancia de ella, teniendo la posibilidad de usar la plantilla por default de envío de email o crear una personalizada.

Una actividad puede ser atómica o compuesta.

El tipo atómico se conoce como **Tarea**.

El tipo compuesto de una actividad es llamado **Subproceso**.

Gráficamente se diferencian por el símbolo de más (+) que utiliza el Subproceso

Sub-processes

- The are composite activities.
- There are two types:
  - Reusable: they enable us to reuse a process defined in an another diagram.
  - Embedded: they enable us to define the sub-process embedded within the diagram containing it.

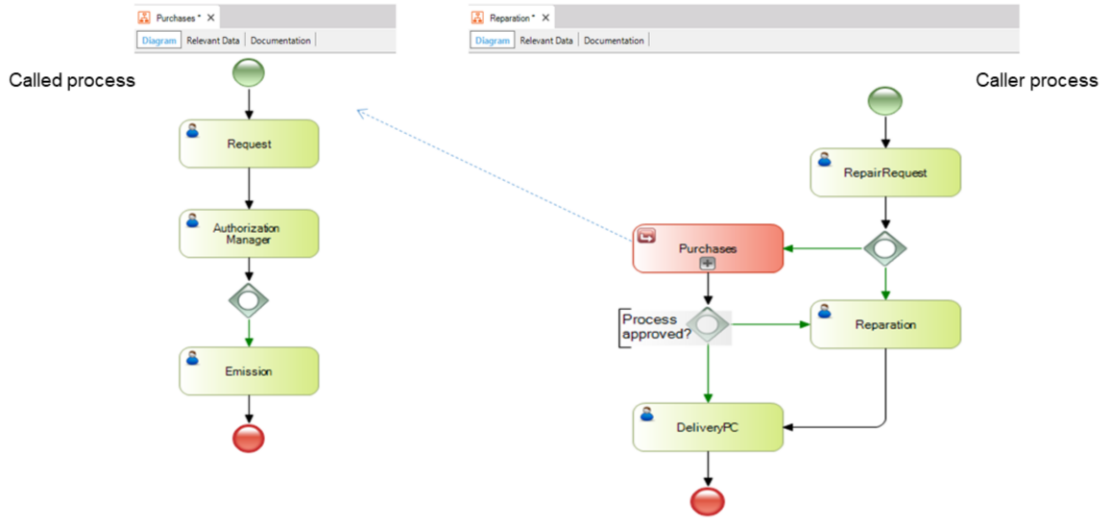
La finalidad de los sub-procesos es poder encapsular un proceso como una tarea de otro proceso. Esto implica dos ventajas: la mejora de la visibilidad de las definiciones de procesos y la reutilización de los mismos.

Hay dos tipos:

Reusable: permiten reusar un proceso definido en otro objeto Business Process Diagram

Embebido: permiten definir el subproceso embebido dentro del diagrama que lo contiene

Reusable Subprocess



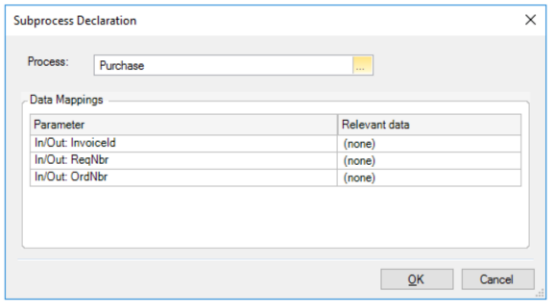
Un subproceso reusable (proceso llamado) es un proceso que es invocado desde otro proceso (proceso llamador).

Los datos relevantes del proceso llamado son independientes de los datos del proceso llamador pero existe un mecanismo de pasaje de parámetros entre ellos.

Mapping of relevant data between subprocesses

It allows communicating the data of a process with data of the other one in two ways:

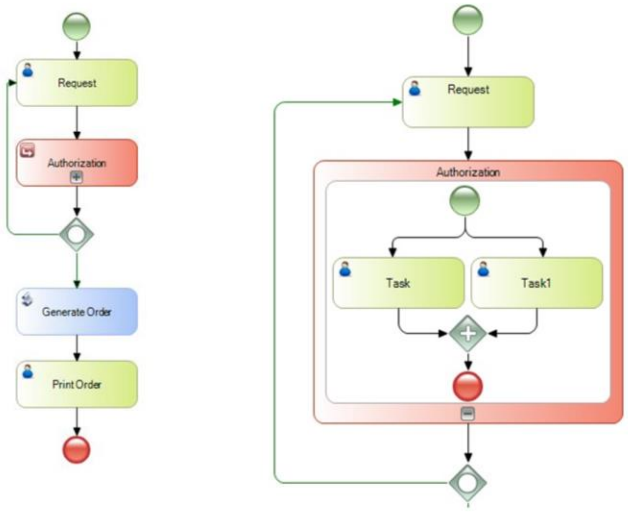
- From the route that arrives at the subprocess
- From the dialog box to select the subprocess



Cuando se llama a un subproceso muchas veces es necesario intercambiar información o recibir información de ese proceso. Para la comunicación de los datos se utilizan los datos relevantes.

Para acceder a la configuración de que datos relevantes que se van a intercambiar se puede hacer desde: la ruta que llega al subproceso o desde el diálogo de selección del subproceso.

Embedded Subprocesses



Subprocesos embebidos

¿Cuándo usarlo? Por ejemplo, en caso de tener un diagrama complejo se puede detectar un grupo de actividades que puedan ser encapsuladas en un subproceso embebido, teniendo como resultado un diagrama más compacto con una visión global y entendible del proceso o si queremos ver un detalle en particular se puede hacer una expansión inline (+) y conocer las actividades embebidas, lo que nos dará una ganancia en la documentación del proceso.

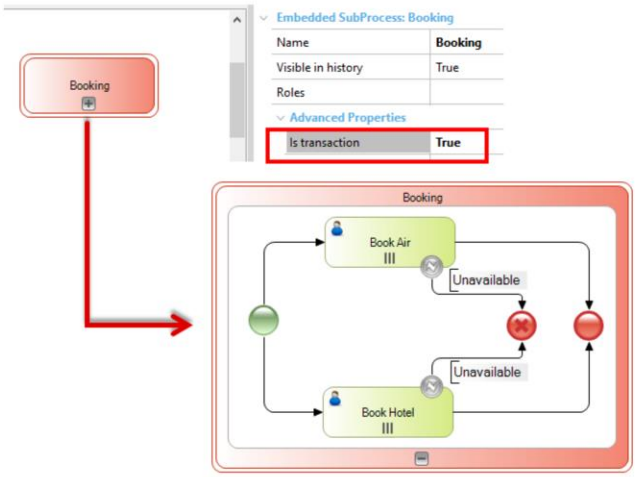
Otra opción es cuando a un conjunto de tareas se quiere hacer algo, por ejemplo poner un timer, en este caso en vez de crear otro proceso para agrupar esas tareas se hace dentro del mismo.

Sus principales características son:

- Nos permiten definir un subproceso dentro de un proceso padre y no es reusable por otros procesos.
- Los datos relevantes del proceso padre estarán disponibles para el subproceso sin necesidad de realizar mapeos.
- No pueden iniciarse con eventos de tipo triggers (por ejemplo un timer) ya que sólo el proceso padre decide cuando deben iniciarse.

Tip: Para editar un subproceso embebido se puede hacer dentro del proceso padre o mediante el doble clic en el subproceso, esto abre el subproceso en una nueva pestaña y es útil para editar procesos complejos o que se anidan en varios niveles de subprocesos.

Transactional Subprocess

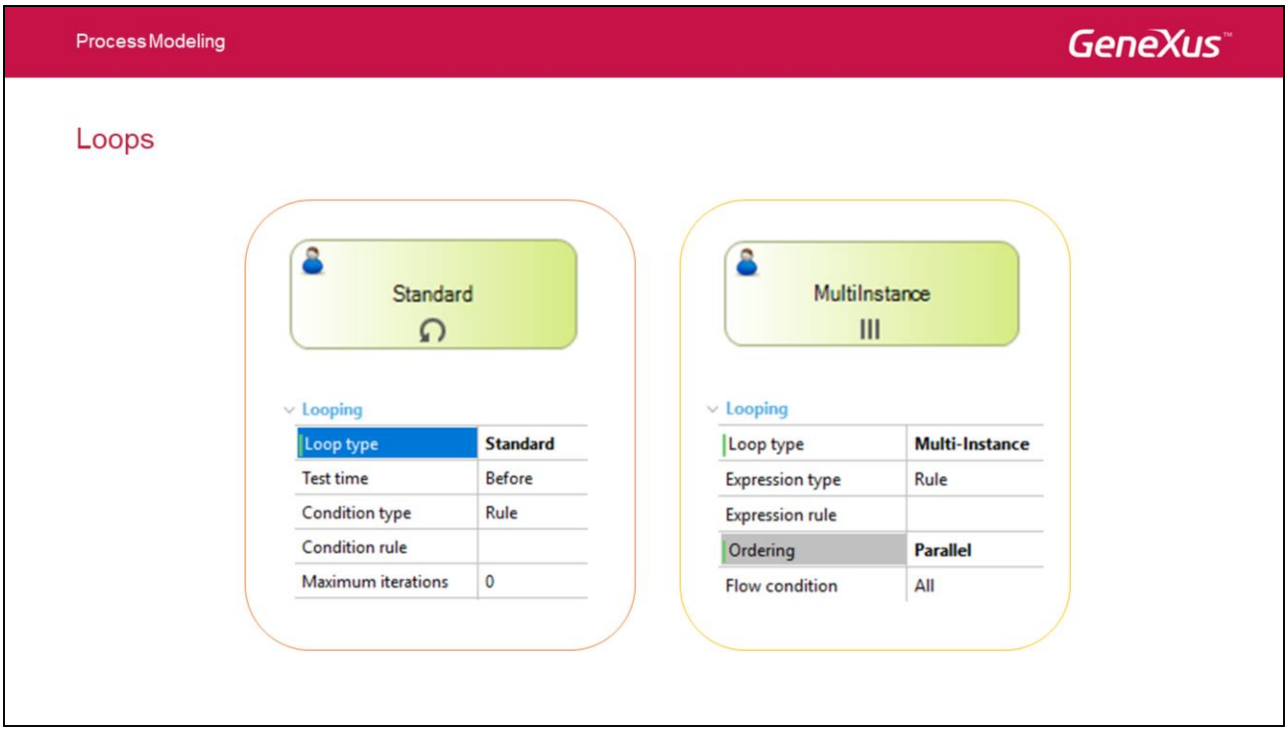


Subproceso transaccional

Un subproceso embebido puede ser considerado como una unidad lógica (transacción), es decir, que se deben completar todas las tareas que están dentro del subproceso, en caso contrario el proceso se deshace, deshaciendo todas las actividades que lo componen (rollback).

Para definir a un subproceso transaccional hay que habilitar la propiedad 'Is transaction' del subproceso y **se identificará por tener un borde doble**.

Como se muestra en el ejemplo tenemos un subproceso que hace reservas de vuelos y hoteles. Si ocurre un error de falta de disponibilidad en cualquiera de las reservas, el flujo avanza hacia un evento de finalización de tipo Cancel. Esto accionará el rollback del proceso y cualquier actividad de reserva que haya sido completada quedará como "undone".



Existen varias maneras de definir bucles en el modelado de los procesos.

Los bucles pueden ser definidos en una sola actividad (ya sea tarea o subproceso), pare ello hay que modificar la propiedad *Looping type* que puede tomar los siguientes valores:

Standard

Una actividad de tipo bucle **estándar** tiene asociada una expresión booleana que se evalúa después de cada ciclo del bucle. Si la expresión es todavía cierta, entonces el ciclo continuará.

Existen dos variantes del bucle, que reflejan las construcciones de programación ”While” y “Until”.

En un bucle while la expresión se evalúa antes de ejecutar la actividad, lo que significa que la actividad puede ejecutarse cero veces.

En un bucle until la expresión se evalúa después de que la actividad se ha ejecutado, lo que significa que la actividad se llevará a cabo al menos una vez.

El tipo de bucle se configura con la propiedad “Test Time” (Before = While, After = Until).

La condición (booleana) puede ser definida mediante una regla o mediante un procedimiento (para casos complejos).

Multi-Instance

Un bucle Multi-Instance refleja la construcción de programación “for each”.

La expresión de un bucle Multi-Instance es una expresión numérica que se evalúa sólo una vez antes de realizar la actividad. El resultado de la evaluación de la expresión es un número entero que especificará el número de veces que la actividad deberá repetirse.

Al igual que con el bucle estándar la expresión puede definirse con una regla o con un procedimiento. Hay también dos variantes del bucle Multi-Instance dependiendo si las diferentes instancias ejecutan en forma secuencial o en paralelo, lo cual se configura en la

propiedad Ordering.

Example

Purchase process

Proceso de compras

1. Un funcionario ingresa un pedido de compras
2. Cada ítem del pedido es autorizado por separado por el jefe del funcionario
3. Si al menos uno de los ítems del pedido fue autorizado entonces se genera la orden de compras correspondiente

Es necesario un mecanismo para declarar que una actividad será ejecutada potencialmente más de una vez

El número de veces que será ejecutada no es conocido durante el diseño

Standard Looping

- The activity will be executed sequentially, provided that a condition of the true/false type is fulfilled.

- Properties involved:
  - Condition rule
  - Test time

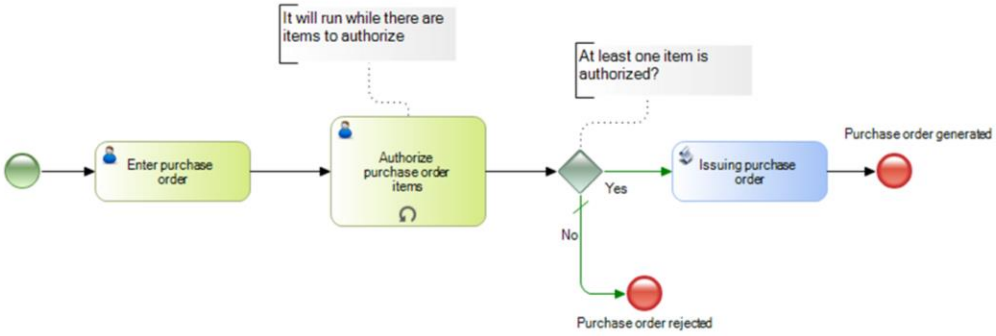
Otras propiedades:

- ***Condition rule***: permite especificar la condición que debe cumplirse para que se siga ejecutando la actividad (implementación)
- ***Test time****:* especifica el momento en que se evalúa la condición

  -Before: la evaluación se hace antes de ejecutar la actividad

  -After: primero se ejecuta la actividad y luego se evalúa la condición

Standard Looping



Process Modeling

GeneXus

Multi-instance

- The activity is executed the number of times defined by the assessment of a numeric expression.
- Activities may be executed in a sequential or a parallel manner.

Properties involved:

- **Expression rule**
- **Ordering**
  - Sequential
  - Parallel
- **Flow condition** (only if Ordering = Parallel)
  - None
  - One
  - All
  - Complex

Otras propiedades:

- **Expresion rule**: expresión numérica que indica el número de veces que se ejecutará la actividad (implementación)
- **Ordering**

  -Sequential: se ejecutará en forma secuencial

  -Parallel: se ejecutará en forma paralela

- **Flow condition** (aplica solamente si *Ordering = Parallel)*

  Determina cómo se unifican los caminos luego de que se ejecutan las diferentes instancias de una actividad

  Valores:

  None: No hay unificación. Luego de que cada instancia termina, se continúa con el flujo que sigue a la misma, por lo que este flujo se repite por cada instancia.

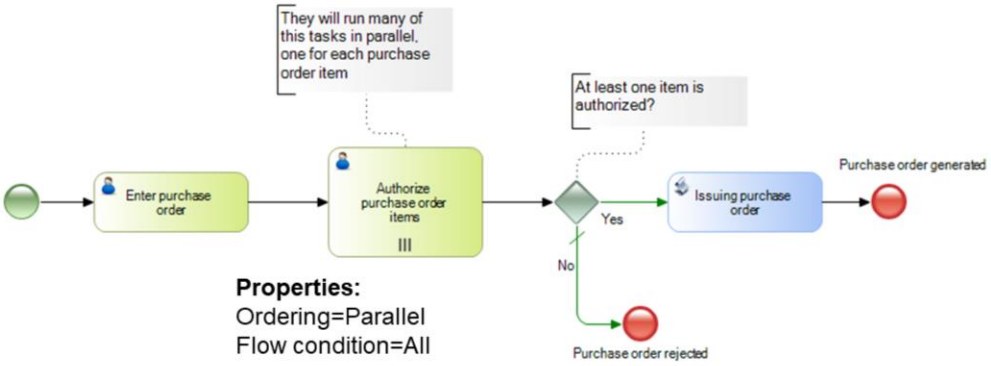
  One: No hay unificación. El proceso continúa luego de que la primera de las instancias termina de ejecutarse y se ejecuta sólo el flujo que parte de ella.

  All: Similar a una parallel Gateway, el proceso continúa solamente si todas las instancias de la actividad finalizaron

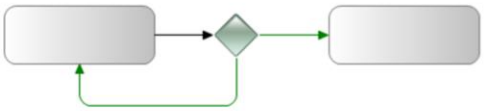
  Complex: permite especificar una condición que indicará cuando puede continuar el proceso. La condición se evalúa cada vez

que finaliza una instancia.

Multi-instance

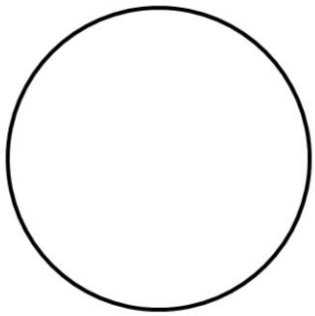


Loops



Los bucles también pueden crearse utilizando conectores de secuencia que conectan con elementos “anteriores” del diagrama.

Events



Hasta aquí no hemos mencionado dónde comienza y termina el proceso para esto vamos a utilizar el símbolo **Evento.**

BPMN Events

They represent occurrences taking place during the process

Types of events:

Los eventos representan **sucesos que ocurren durante el proceso**.

Hay tres tipos de eventos:

Inicio: Permite indicar dónde inicia el proceso. Un evento inicial **NO** puede tener conectores entrantes.

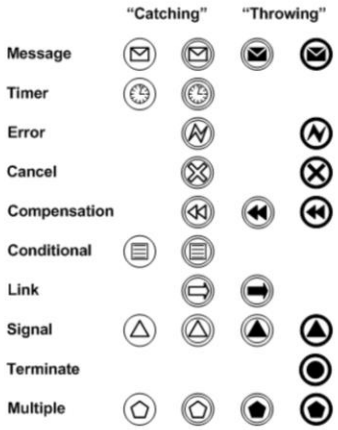
Intermedio: Permite indicar un suceso entre el inicio y el fin de un proceso.

Fin: Permite indicar dónde termina el proceso. Un evento final **NO** puede tener conectores salientes

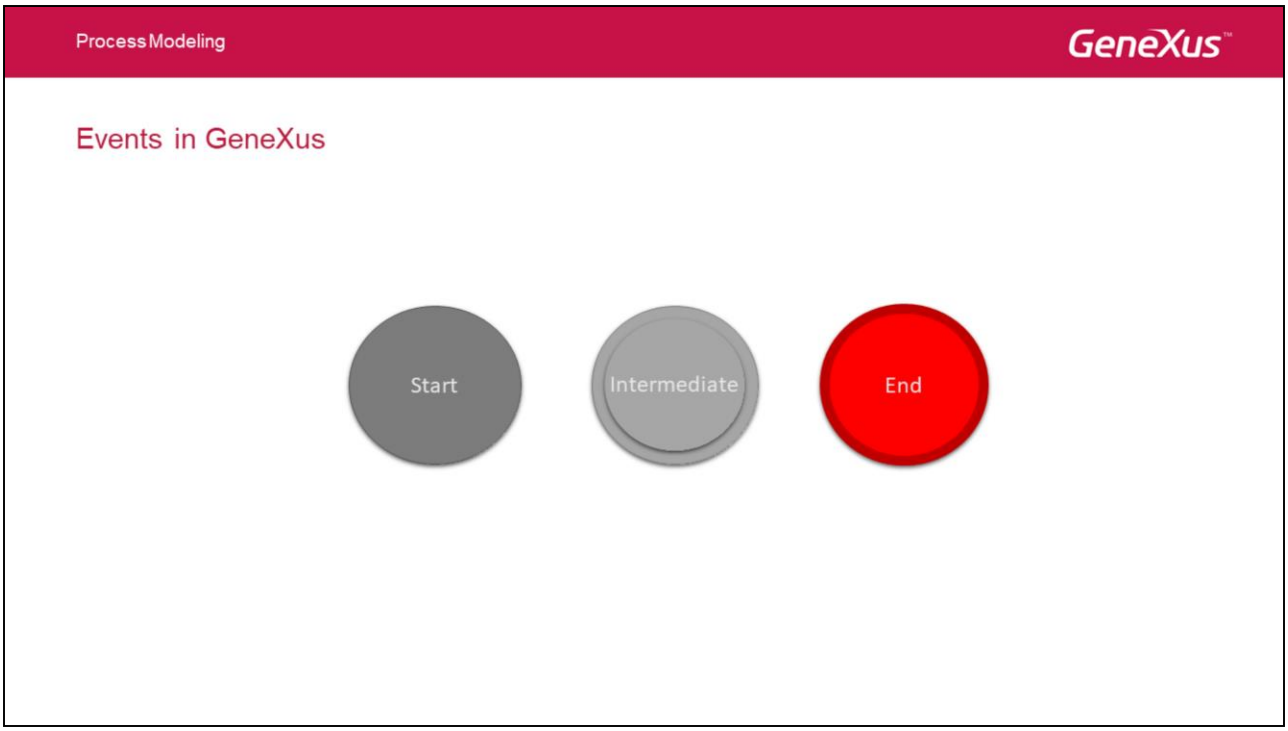
Estos se diferencian por el trazo, los de **inicio tienen trazo fino**, los **intermedios tienen doble trazo fino** y los de **fin tienen trazo grueso.**

BPMN Events

For each event type, BPMN defines sub-types to specify the nature of an event.



Para cada uno de estos tres tipos BPMN define **subtipos** que permiten especificar la **naturaleza del evento**.



Los eventos se representan con un círculo. Es algo que “pasa” durante el curso del proceso de negocio. Estos eventos afectan al flujo del proceso y suelen tener una causa (trigger) o un impacto (resultado). Los eventos representados con un círculo con centro abierto permiten a los marcadores internos diferenciar diferentes triggers y resultados.
Hay tres tipos de eventos, basados en cuando afectan al flujo:

Start

Como su nombre lo indica, un evento Inicial indica por donde va a iniciar un proceso. Su utilización es opcional. En caso de **NO** utilizarlo, cualquier actividad que **NO** le lleguen conectores se considerará como **Inicial**.
Se recomienda su utilización pues hace mas legible el diagrama. BPMN establece que los eventos de inicio se representan con un círculo de trazo fino (el color es agregado por GeneXus).
Un evento inicial **NO** puede tener conectores entrantes.

Intermediate

El evento Intermedio indica que algo pasa (un evento) en algún lugar entre el inicio y el final de un proceso.
Afecta el flujo del proceso, pero **NO** Inicia o termina (directamente) el proceso. BPMN establece que los eventos intermedios se representan con un círculo de trazo doble (el color es agregado por GeneXus).

End

El evento de Fin indica que un proceso va a terminar.
Su utilización es opcional. En caso de **NO** utilizarlo, cualquier actividad que **NO** tenga conectores salientes marca el fin de ese camino en el proceso (puede haber varios caminos paralelos).
Un evento Final **NO** puede tener conectores salientes. BPMN establece que los eventos de fin se representan con un círculo de trazo grueso (el color es agregado por GeneXus).

Process Modeling GeneXus

Events in GeneXus

Los eventos pueden se pueden clasificar como “Catch” o “Throw” dependiendo si atrapan el evento o si lo disparan.

Tipos de Evento

None

Es un evento que no tiene un tipo definido.

En un evento intermedio puede ser utilizado para marcar un hito en el proceso.

Timer

Cuando es utilizado como un elemento independiente permite especificar un “delay” en el proceso. Esto es, una espera por un tiempo determinado.

Cuando es utilizado como adjunto a una actividad, permite modelar cuanto quiero esperar por esa actividad (deadline).

Conditional

Este evento es utilizado para esperar hasta que cierta condición sea verdadera.

Signal

Este evento es utilizado para enviar o recibir señales dentro o fuera del proceso (en este último caso el alcance es la jerarquía de procesos).

Error

El evento error interrumpe el proceso o requiere corrección.

Cuando es intermedio se utiliza adjunto a una actividad y permite definir un camino alternativo

si existe la posible ocurrencia de un error al ejecutar dicha actividad. En caso de ser de tipo final interrumpirá el proceso.

Events in GeneXus

Message

Este evento envía o recibe un mensaje desde otra entidad de negocio o participante (otra Pool).

Compensate

Este evento es utilizado para deshacer actividades en el caso de que un subproceso transacción sea cancelado o necesite deshacerse (rollback).

En un evento catch solamente pueden estar adjuntos a actividades. Cuando es throw se puede colocar en el flujo normal del proceso

Link

El evento link siempre se utiliza de a par: (throw-catch), es necesario que ambos eventos tengan el mismo nombre para hacer el vínculo entre ellos. Su objetivo es crear un flujo de secuencia virtual.

Solo puede haber un evento link del tipo catch y existir muchos eventos link del tipo trhow con el mismo nombre.

Son utilizados únicamente dentro de un mismo proceso. Para comunicación entre procesos se recomienda el uso de señales.

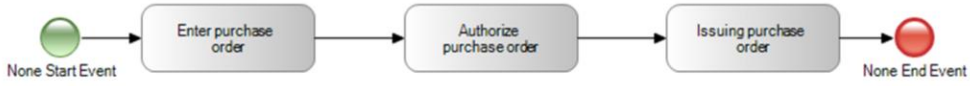
Cancel

Sirve para cancelar un subproceso transaccional. Un evento de finalización cancel indica que se debe realizar un rollback a dicho subproceso.

Terminate

Un evento “terminate” da por terminado el proceso. Esto es, a diferencia del evento final “none”, no solo termina el camino actual sino todo los caminos paralelos que puedan existir.

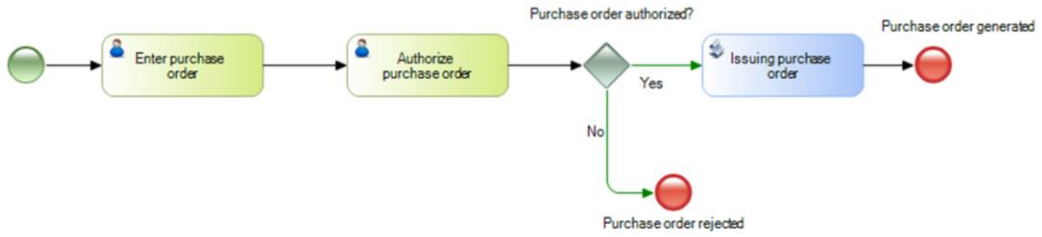
Start and End Events



Entonces agregamos a nuestro proceso el inicio y fin para indicar dónde comienza y dónde termina.

Process Termination

- The “None” termination event is enough to indicate the end of a process, when we have **only one active path**



Process Termination

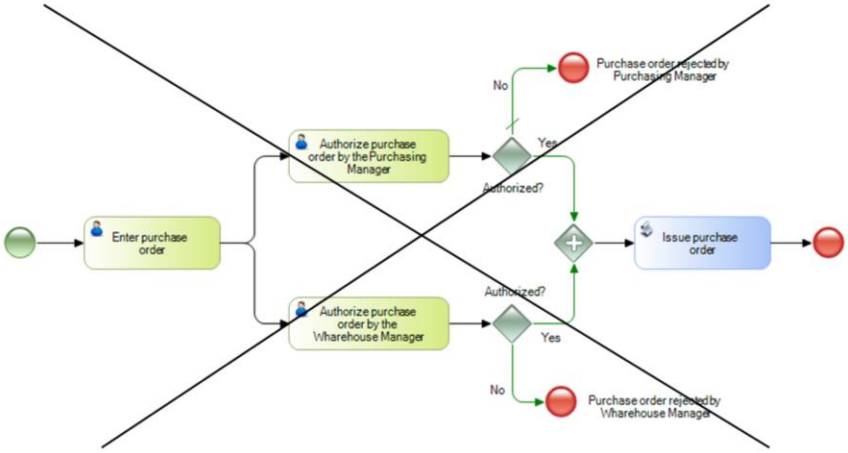
- What happens when we have two parallel authorizations?

¿Qué pasa si tenemos dos autorizaciones en paralelo?

Si el flujo del proceso llega a determinado punto, se requiere que el proceso finalice, independientemente de si se tienen otros caminos paralelos que todavía están activos.

En este ejemplo si uno de los gerentes no autoriza la orden de compra, el proceso debe terminar, independientemente de si el otro gerente la aprobó o no.

Process Termination



La solución usando None End Events no funciona, porque este símbolo solamente finaliza el flujo que llega al mismo y el resto de los flujos continúan activos.

En el ejemplo, la orden de compra se emitiría igual si uno de los gerentes no la aprueba.

Process Termination

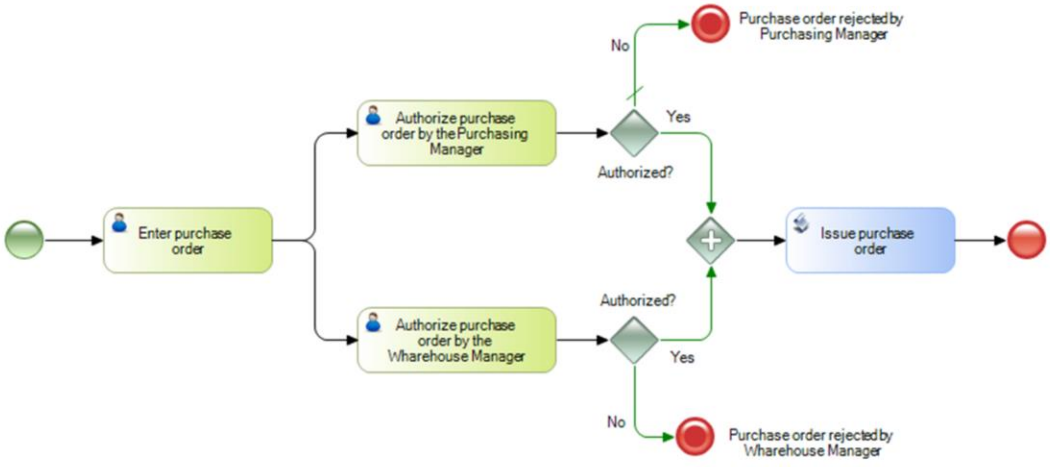
Terminate End Event

- Finalizes a process by interrupting all activities that remain active at that point.

Terminate End Event:

- Finaliza un proceso interrumpiendo todas las actividades que estén activas en ese momento
- Si el proceso se utiliza como subproceso, NO se finaliza el proceso padre

Process Termination

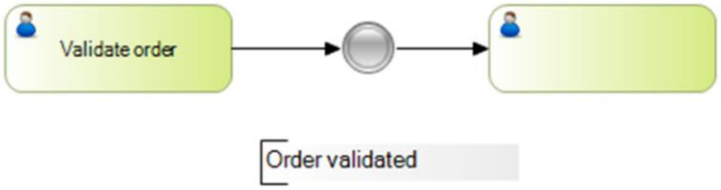


Attached Intermediate Events

Es posible adjuntar eventos al borde de una actividad (tarea o subproceso) para modelar excepciones al flujo normal del proceso.

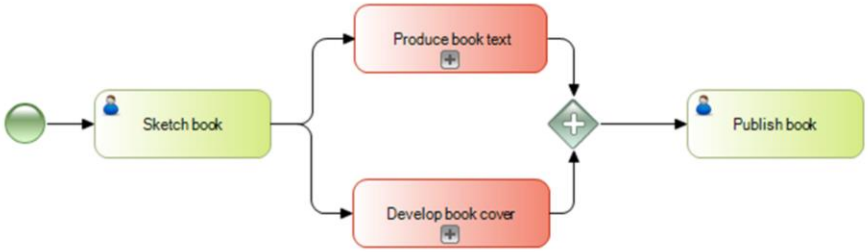
Esto permite que si durante el transcurso de tiempo en el que la actividad esta activa, ocurre dicho evento, se puede disparar un flujo alternativo con la posibilidad de interrumpir la actividad.

Indication of Milestones



Signals

- They provide communication between processes



Signals

Example: Communication between processes

- Subprocess *Produce book text*

- Subprocess *Develop book cover*

Signals

Intermediate Event Signal (BPMN)

Throw
(to trigger the signal)

Catch
(to catch the signal)

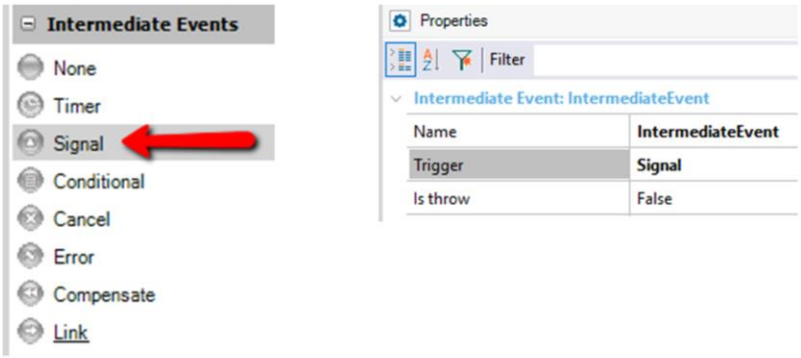
Signal Intermediate Event: Permite señalizar la ocurrencia de un evento que podrá ser detectado en cualquier parte del proceso, subprocesos e incluso procesos ancestros.

Existen dos tipos:

- Trow (dispara la señal)
- Catch (atrapa la señal)

Signals

Intermediate Event Signal (GeneXus)



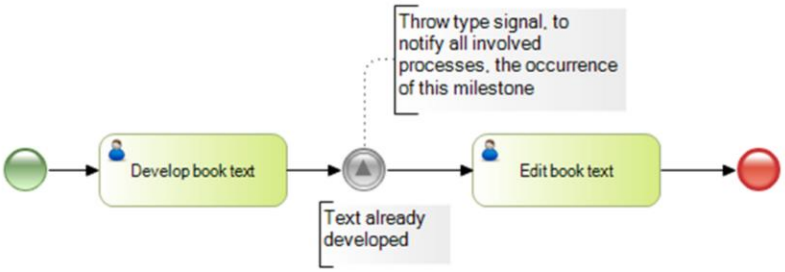
En GeneXus, utilizamos un Evento Intermedio del tipo Señal y configuramos sus propiedades:

- **Trigger** = *Signal*
- **Name**: Las señales se identifican por su nombre
- **Is throw**: Si es un *Throw* hay que configurar la propiedad *Is Throw* = *true*

Signals

Example: Communication between processes

- Subprocess *Produce book text*

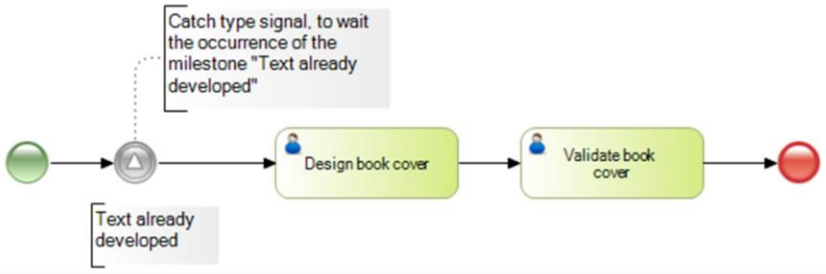


Se dispara una señal, para indicar que el hito de que el texto ya ha sido desarrollado.

Signals

Example: Communication between processes

- Subprocess *Develop book cover*

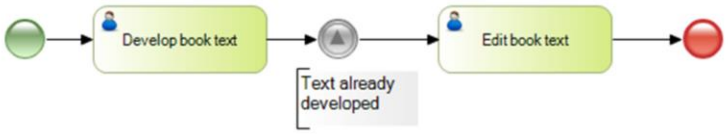


Se captura la señal para ver si ocurrió el hito de que el texto ya ha sido desarrollado.

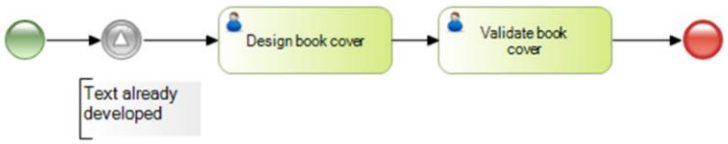
Signals

Example: Communication between processes

- Subprocess *Produce book text*



- Subprocess *Develop book cover*



Los procesos se comunican entre sí utilizando las señales.

Event Properties

Intermediate Event: IntermediateEvent

Name	IntermediateEvent
Trigger	Signal
Is throw	False

Intermediate Event: IntermediateEvent

Name	IntermediateEvent
Trigger	Conditional
Condition type	Rule
Condition rule	

Intermediate Event: IntermediateEvent

Name	IntermediateEvent
Trigger	Timer
Submit to calendar	False
Time unit	Minutes
Lapse expression type	Rule
Lapse expression rule	

Todos los eventos tienen la propiedad Nombre y Trigger que indica cual es el tipo de evento.

Adicionalmente y dependiendo del tipo de evento pueden existir otras propiedades.

Event Signal

La propiedad “Is throw” indica si esta disparando o esperando por el evento.

El nombre del evento se utiliza para discriminar entre diferentes señales. Esto es, si tengo una evento de señal de tipo “catch” y nombre “Signal” va a esperar por un evento de señal de tipo “throw” y de mismo nombre.

Event Conditional

Un evento condicional tiene asociada una condición. Dicha condición se puede expresar con una regla o mediante el llamado a un procedimiento.

Las reglas de condición de un evento condicional se evalúan por el motor de workflow cuando se modifican los datos relevantes de la instancia de proceso a la que pertenece el evento o de un dato relevante compartido en una jerarquía de procesos.

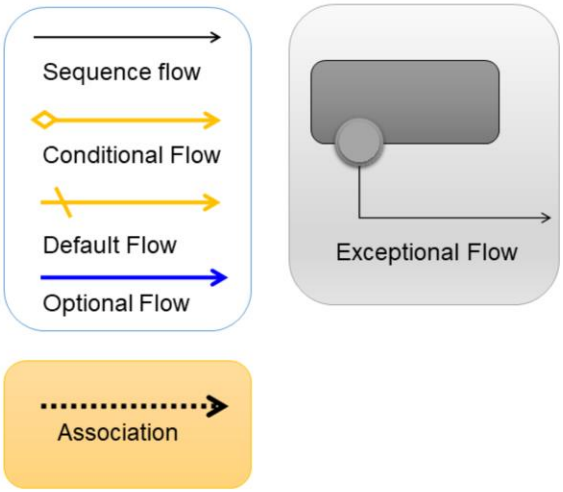
Como consecuencia, la expresión a definir tiene que estar basada en datos relevantes ya que si utilizara atributos la regla nunca se llegaría a evaluar.

Los eventos condicionales siempre son de tipo “catch”.

Event Timer

Un evento de tiempo tiene asociada una expresión numérica (o un procedimiento) que indicará el tiempo que debe pasar para que ocurra el evento (contado desde el momento de su creación). Se puede especificar la unidad de tiempo y si esta sujeto a un calendario particular.

Types of Connectors



Los elementos del diagrama de procesos se conectan entre ellos en un diagrama para crear el esqueleto básico de la estructura de un proceso de negocio.

Hay tres tipos de conectores que hacen esta función. Estos conectores son:

Flujo Secuencial

El flujo secuencial se representa por una línea sólida con una cabeza de flecha sólida y se usa para mostrar el orden (la secuencia) en el que las diferentes actividades se ejecutarán en el Proceso.

Flujo Condicional

El flujo condicional se representa por una línea sólida con una cabeza de flecha sólida. Si el flujo parte desde una actividad entonces en el comienzo de la línea se utiliza un mini rombo. Si el flujo parte desde una compuerta entonces el rombo no se utiliza porque el tipo de flujo queda implícito.

Un flujo condicional tiene asociado una regla de condición que es evaluada en tiempo de ejecución para determinar si dicho flujo debe continuar o no.

Flujo Default

Este flujo se utiliza en exclusive o inclusive gateways y es utilizado para modelar que es el flujo que debe continuar cuando los restantes flujos condicionales salientes no evalúan como verdadero en tiempo de ejecución.
Para diferenciarlo se utiliza barra diagonal que lo corta transversalmente en el comienzo de la línea.

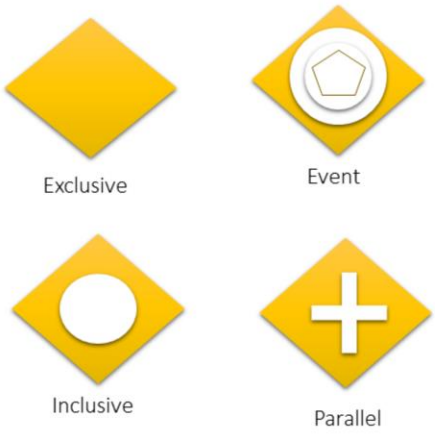
Flujo Opcional

El flujo opcional se representa por una línea de color azul. El usuario selecciona en tiempo de ejecución por qué ruta debe seguir el flujo.

Asociación

Una asociación se representa por una línea de puntos con una punta de flecha y se usa para asociar datos, texto y otros artefactos con los elementos del diagrama.

Types of Gateways



Existen diferentes tipos de gateways las cuales se pueden utilizar tanto para convergencia como para divergencia. A continuación se detallan las mismas:

- **Exclusive:** *Divergencia:* el flujo seguirá por solamente uno de los caminos que tienen origen en el Gateway, la elección se hará evaluando las condiciones que se definen para el Gateway (se tendrá una por cada camino).
Convergencia: cuando múltiples caminos llegan al Gateway, este simplemente deja avanzar el flujo sin evaluar ninguna condición ni realizar ninguna sincronización.

- **Event:** *Divergencia:* el flujo avanza por solamente uno de los caminos que tienen origen en el Gateway según la ocurrencia de un evento especificado.
Convergencia: tiene el mismo comportamiento que el Gateway Exclusive.

- **Inclusive:** *Divergencia:* el flujo seguirá por todos los caminos que tienen origen en el Gateway que cumplan las condiciones del Gateway (se tendrá una por cada camino).

Convergencia: realiza una sincronización de todos los caminos por donde el flujo del proceso efectivamente llegará al gateway.

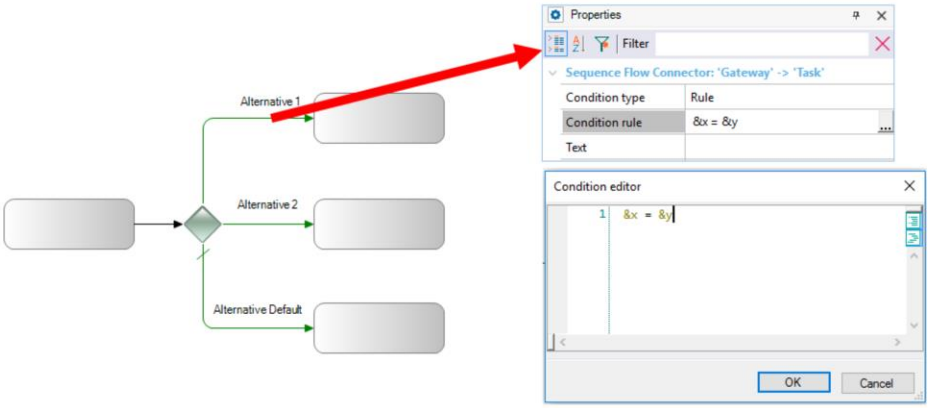
- **Parallel:** *Divergencia:* el flujo avanza por todos los caminos que tienen origen en el Gateway.

Convergencia: realiza una sincronización de todos los caminos que llegan al Gateway

Luego veremos en detalle la utilización de cada uno de ellos en diferentes ejemplos de

modelado de procesos

Exclusive Gateway



Flow -> One of the routes

Los **Exclusive** gateways (decisiones) son lugares dentro de un proceso de negocio donde la secuencia de flujo puede tomar dos o más caminos alternativos. Para una determinada instancia del proceso, sólo se puede tomar uno de los caminos.

Una decisión no es una actividad desde la perspectiva del proceso de negocio, es un tipo de puerta de enlace que controla la secuencia de flujo entre las actividades.

Puede ser pensado como una pregunta que se hace en ese momento en el proceso y que tiene un conjunto definido de respuestas alternativas (solo una de las cuales puede ser verdadera en una determinada instancia).

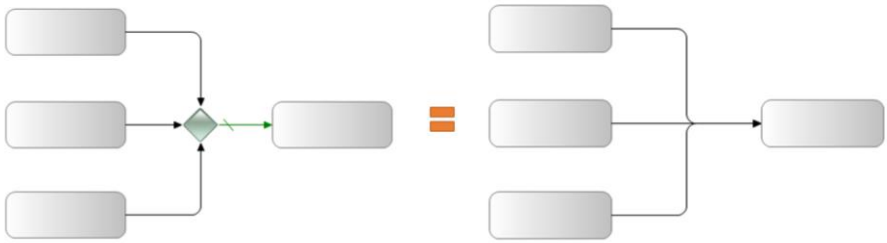
A cada conector saliente se le asocia una regla de condición.

Durante la ejecución del gateway se evalúan las reglas condicionales de cada una de los conectores salientes. La primera que evalúe como verdadera será la que determine el camino a seguir.

Si ninguna de las reglas evalúa en verdadero entonces se sigue por el camino default (el conector que esta marcada con una raya transversal).

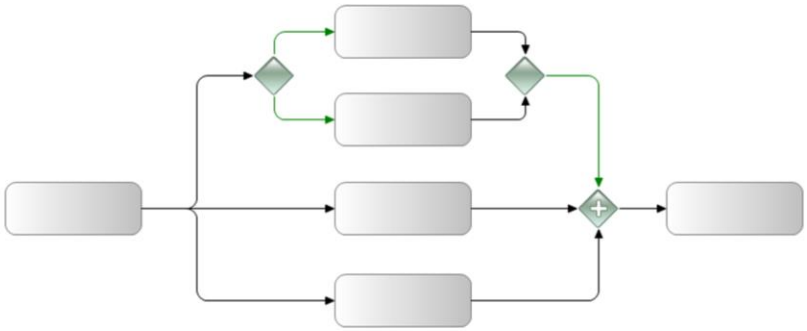
El conector default no es obligatoria, en ese caso se debe asegurar que en cada instancia al menos una de las reglas evaluará como verdadero.

Exclusive Gateway (Synchronization)



Los **Exclusive** gateway también pueden ser utilizadas para sincronización, aunque su utilización rara vez es necesaria para el modelado ya que generalmente puede ser modelado sin el mismo (como se ve en la otra figura) obteniendo el mismo comportamiento.

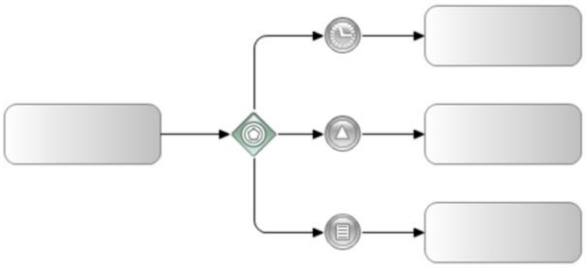
Exclusive Gateway (Synchronization)



Hay ciertas situaciones en las que un **Exclusive** gateway si es requerida para sincronizar.

En el ejemplo de la figura, si no se utilizara el **Exclusive** para sincronizar el resultado de un gateway anterior, **Parallel** gateway tendría cuatro conectores de secuencia de entrada. Sin embargo, sólo tres de las cuatro secuencias de flujo podrían pasar en cada vez, por lo tanto, el proceso quedaría atascado en ese **Parallel** gateway.

Event Gateway

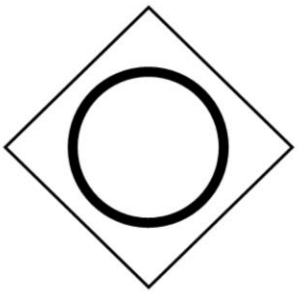


Flow -> One of the routes

A gateway of **Event** type is a particular case of an **Exclusive** gateway (in the sense that only one of the possible routes can be selected) where the decision of which route to take depends on the occurrence of an event and not on the evaluation of a condition rule.

It's worth mentioning that if none of the specified events occur, the process will be stuck. Therefore, always using a **Timer** event is recommended as one of the possible alternatives.

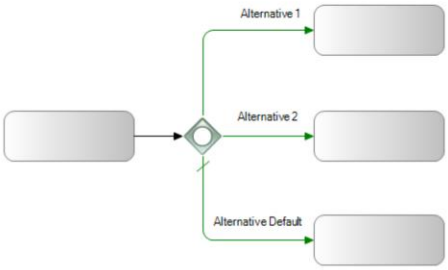
Inclusive Gateway



Un **Inclusive** gateway es similar al exclusive en el sentido que cada conector tiene asociada una regla condicional. La diferencia es que se evalúan todas las reglas y se sigue por todas aquellas que evalúan como verdadero.

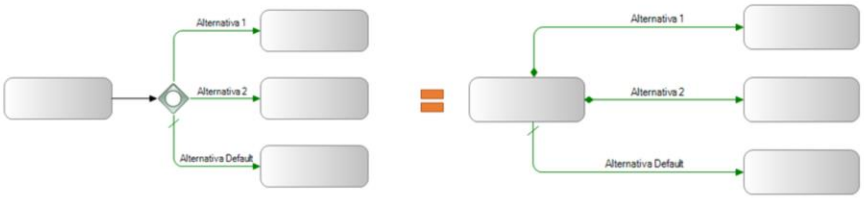
Se recomienda el uso de caminos default a fin de evitar que el proceso quede atascado en cualquier situación.

Inclusive Gateway (Branching)



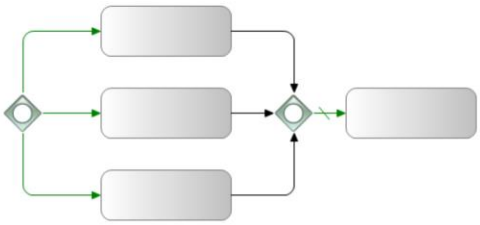
Flow -> All routes

Inclusive Gateway (Branching)



Se puede modelar este mismo comportamiento sin utilizar una **Inclusive** gateway. En su lugar podemos utilizar secuencias de flujo de tipo condicional, las cuales también tienen asociada una regla de condición.

Inclusive Gateway (Synchronization)

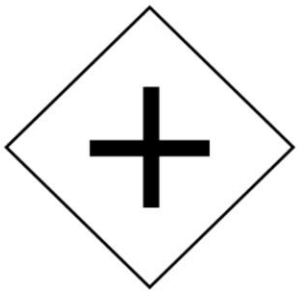


Un **Inclusive** gateway también puede ser utilizado para sincronizar.

En este caso el gateway espera por todas secuencias de flujo que fueron seleccionadas en una paso anterior.

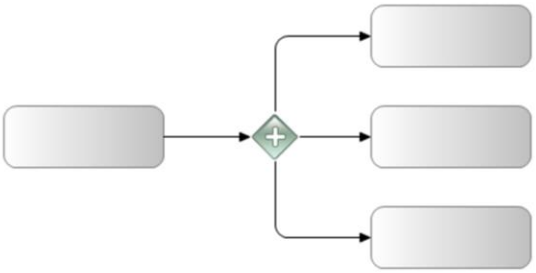
Actualmente el motor de gxflow solo soporta esta modalidad cuando el **Inclusive** gateway que sincroniza esta emparejada con otro **Inclusive** gateway como lo muestra la figura.

Parallel Gateway



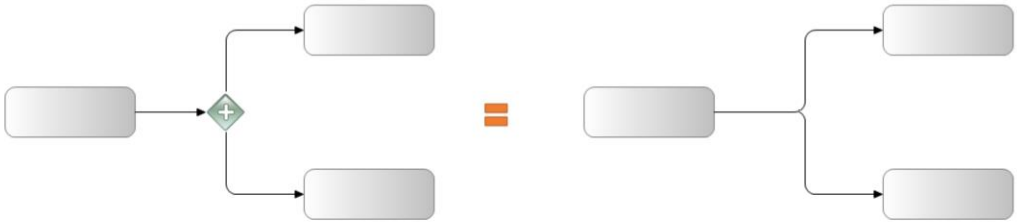
Un **Paralell** gateway permite crear flujos paralelos (o sincronizarlos cuando se utiliza en esa otra modalidad).

Parallel Gateway (Branching)



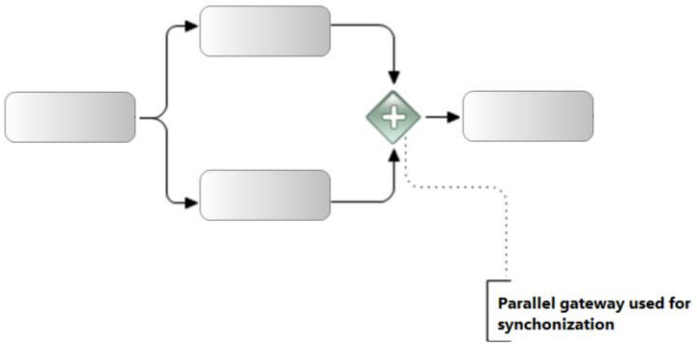
Simplemente divide el flujo en dos o más caminos paralelos. No evalúa ninguna condición.

Parallel Gateway (Branching)



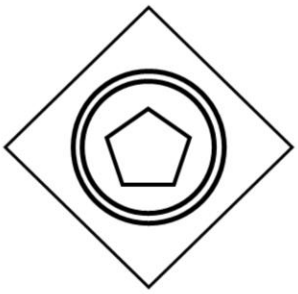
Se puede modelar este mismo comportamiento sin utilizar un **Paralell** gateway como muestra la figura. Sin embargo, el uso del gateway puede llegar a clarificar el diagrama en determinadas circunstancias.

Parallel Gateway (Synchronization)



Cuando el **Paralell** gateway es utilizado para sincronización, esta espera por todas las secuencias de flujo entrantes para entonces si poder seguir adelante.

Gateway Event



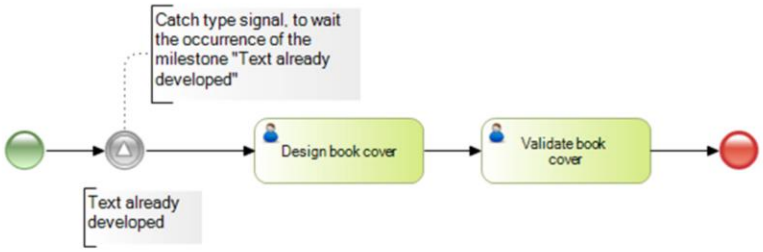
Un gateway de tipo **Event** es un caso particular de un **Exclusive** gateway (en el sentido que solo uno de los caminos posibles puede ser elegido) en que la decisión de que camino seguir depende de la ocurrencia de un evento y no de la evaluación de una regla de condición.

Dada una lista de posibles eventos se necesita esperar por la ocurrencia del primero de ellos

Cabe aclarar que si nunca ocurre ninguno de los eventos especificados, el proceso quedará estancado, por lo que se recomienda utilizar siempre un evento **Timer** como una de las alternativas.

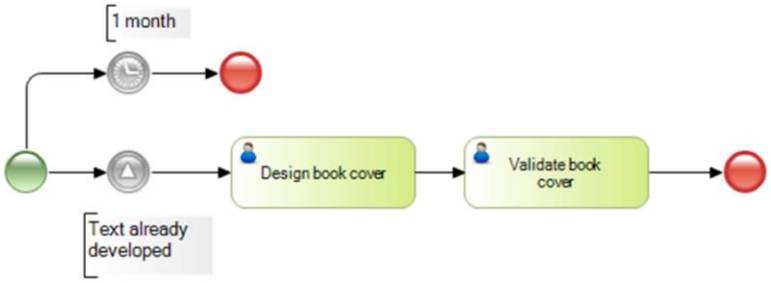
Gateway Event

The process below may be blocked indefinitely awaiting the arrival of the signal.

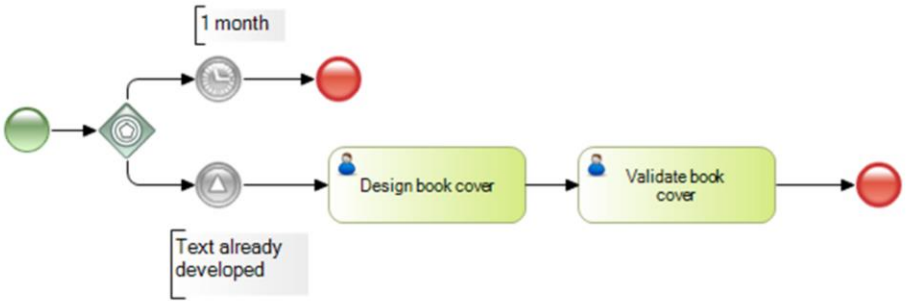


Event-based branching

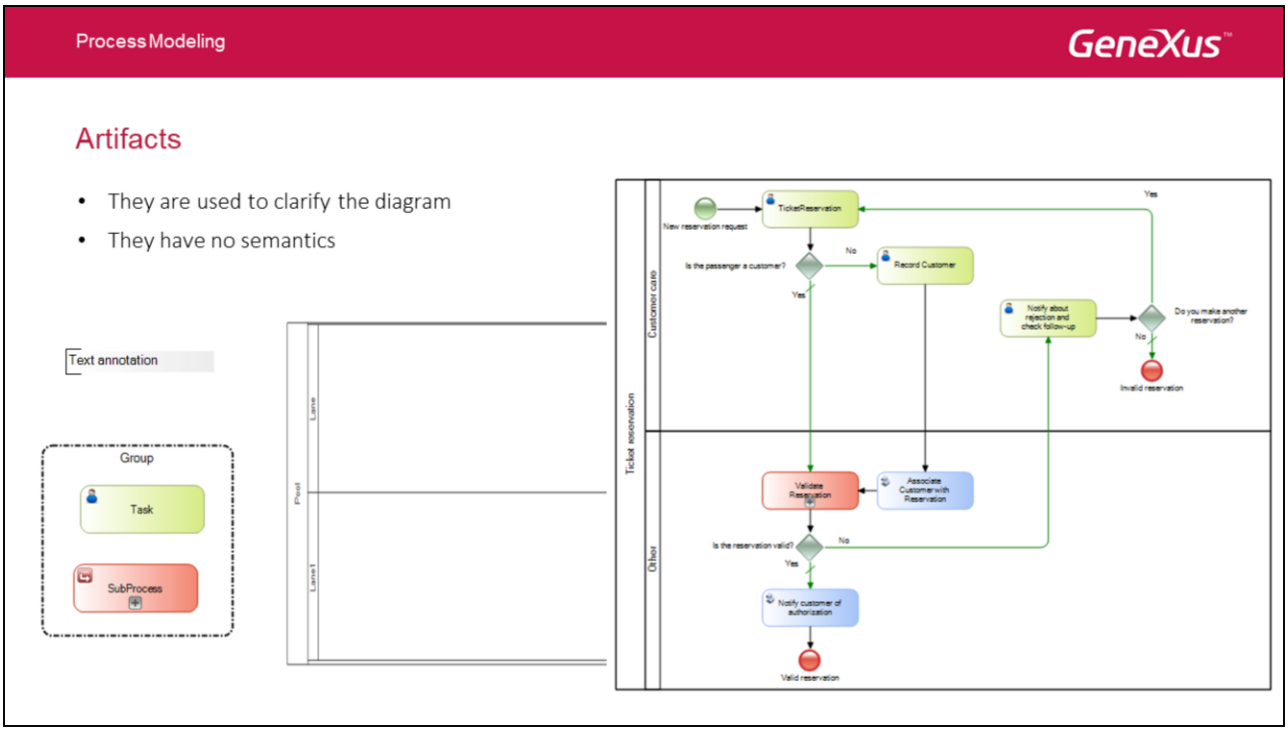
We avoid the blocking by defining a maximum waiting time of one month.



Event Gateway



Cuando utilizamos un gateway del tipo evento, el camino que continúe depende del tipo de evento que se capture.



Los Artifacts son símbolos que se usan dentro de los diagramas para dar mas claridad al mismo. Los artifacts disponibles son:

Text annotation: Las anotaciones de texto son un mecanismo para que el modelador pueda proporcionar información adicional para el lector del diagrama.

Gráficamente se representa con un rectángulo abierto con una línea de color negro sólido.

Un anotación de texto se puede conectar a un determinado elemento del diagrama utilizando una asociación (flecha punteada).

Group: Un grupo es un artefacto que proporciona un mecanismo visual para agrupar elementos de un diagrama de manera informal. Gráficamente se representa con un rectángulo de esquinas redondeadas formado por una línea punteada.

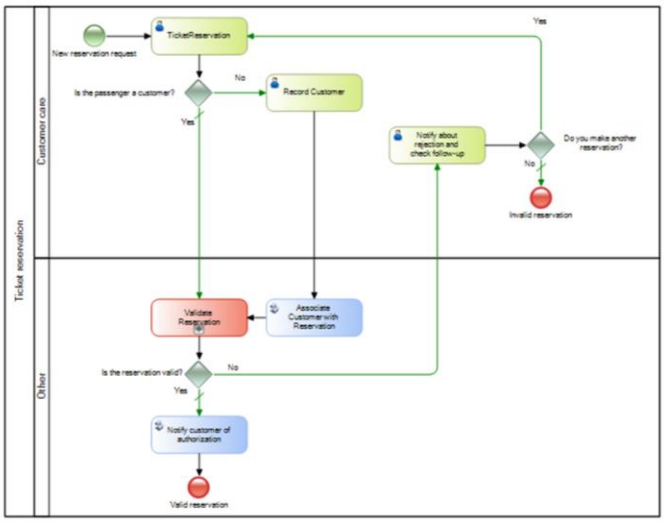
Pool: Representa el proceso de un participante y es utilizado cuando el diagrama involucra dos entidades de negocio o participantes separados. Las actividades dentro de los contenedores son considerados procesos autónomos, por lo que el flujo de mensajes se definen como el mecanismo para tener comunicación entre dos participantes.

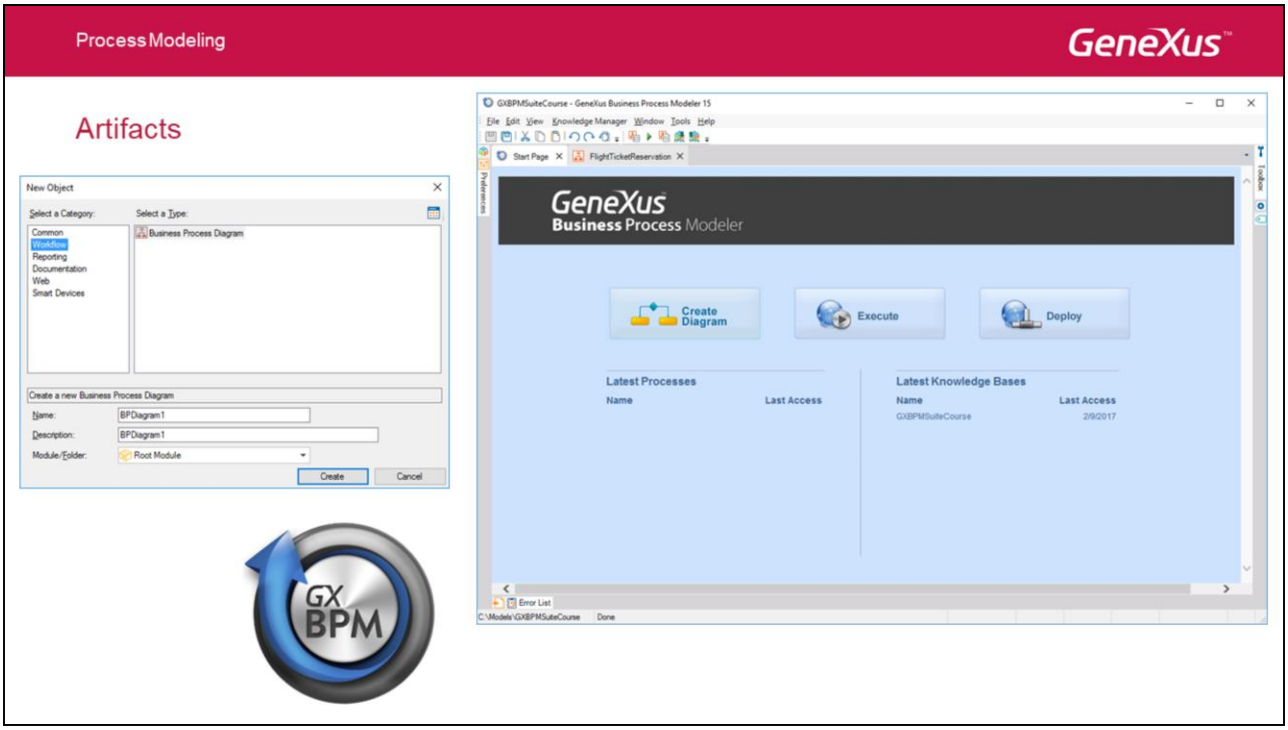
Lane: Permite asociar las actividades con una función o rol específicos de la empresa.

GeneXus

Artifacts

Pool and Lanes example





Para **modelar procesos de negocio** se puede utilizar el objeto Business Process Diagram en el IDE de GeneXus pero además se puede utilizar la herramienta **GeneXus Business Process Modeler** (genexus.com/gxbpm) que se distribuye de forma independiente de GeneXus. Esta herramienta está pensada para resolver dos escenarios, los cuales se describen a continuación.

Escenario 1: fase de análisis

El primero en la fase inicial del proyecto donde tenemos a los analistas, que son funcionales que entienden muy bien del negocio de la empresa y conocen los procesos de negocio pero no necesariamente tienen un perfil técnico. Y lo que precisan es una herramienta que les permita capitalizar ese conocimiento, documentarlo y formalizarlo de forma que luego pueda ser automatizado, en este caso con GXflow.

Escenario 2: modificación de procesos que ya están en producción

El segundo escenario se ubica en el otro extremo del proyecto, cuando el sistema está en producción o se implantando. Por ejemplo un ERP que se está implantando en una empresa y se necesita parametrizar determinadas reglas de negocio.

GeneXus Business Process Modeler provee todas las funcionalidades de edición del modelador de procesos incluido en la versión estándar de GeneXus.

Las funcionalidades más importantes son:

- Edición del flujo que define el proceso. Esto implica poder agregar o quitar cualquier tipo de elemento de la notación BPMN al diagrama además de poder conectar elementos.
- Edición de las propiedades de los elementos de un diagrama.
- Poder asociar el objeto GeneXus que se encargará de resolver la operativa de la tarea en ejecución (transacción, web panel o procedimiento).
- Edición de reglas del proceso (por ej.: reglas subject, de definición del lapso de un evento timer, condicionales, etc.)
- Creación de nuevos procesos. Solamente se permitiría crear nuevos diagramas de procesos de negocio pero no cualquier objeto GeneXus.
- En este escenario solamente se pueden editar diagramas de procesos, para los demás objetos GeneXus que se podrán asociar a los diagramas solo se podrán visualizar algunas de sus partes (por

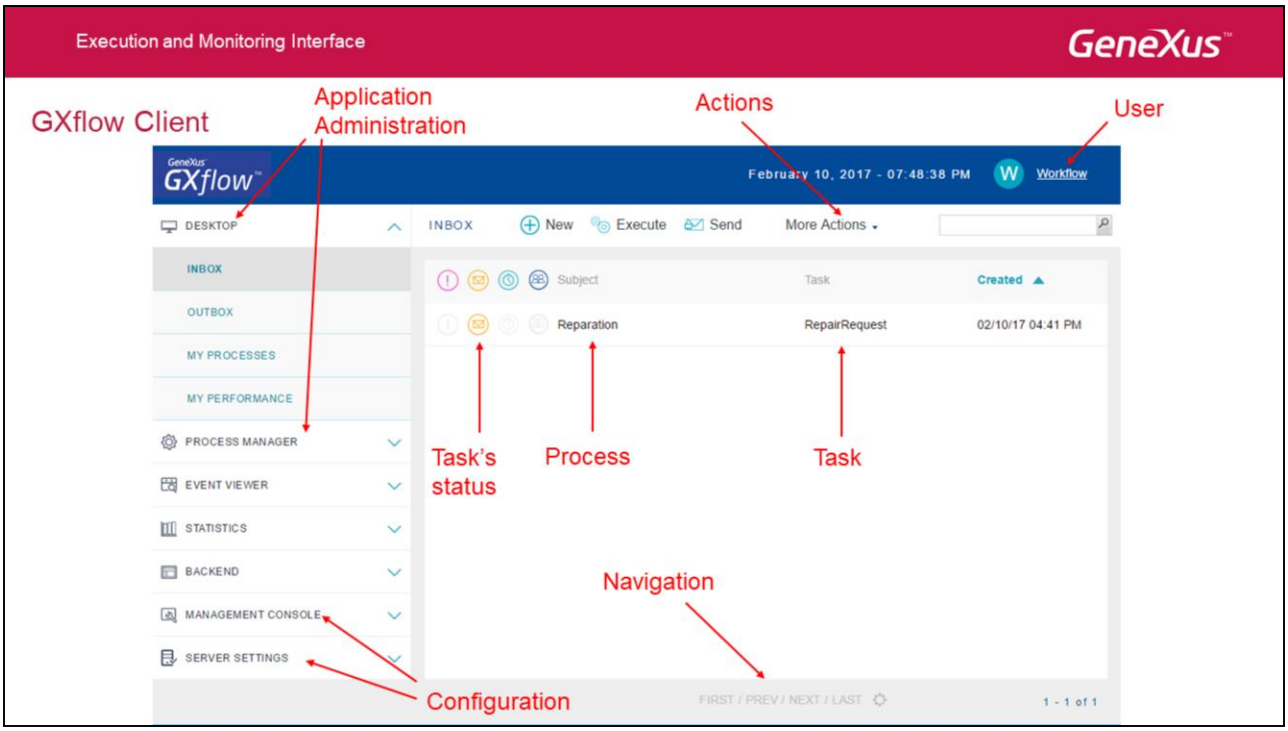
ej.: estructura de transacciones, regla parm y documentación) pero no editarlas.

GeneXus 15

Module 3

Execution and Monitoring Interface

En este módulo estudiaremos los componentes del cliente GXflow.



Esta es la interfaz del cliente GXflow, donde a la izquierda tenemos las aplicaciones y a la derecha a medida que se selecciona una aplicación podemos ver la información que contiene. Estas aplicaciones son:

Escritorio

- Bandeja de Entrada, Bandeja de Salida, Mis procesos, Mi rendimiento

Administrador de Procesos

- Procesos, Tareas, Eventos de Negocio, Definiciones de Procesos, Calendarios, Participantes, Plantillas de notificación

Visor de Eventos

- Eventos

Estadísticas

- Análisis de procesos, Análisis de tareas, Performance de procesos, Performance de tareas, Performance de equipos

Backend

- Menúes, Componentes, Acciones

Consola de Administración

- Modelo Organizacional: Usuarios, Roles, Definiciones de Unidades Organizacionales, Unidades Organizacionales

Ajustes del Servidor

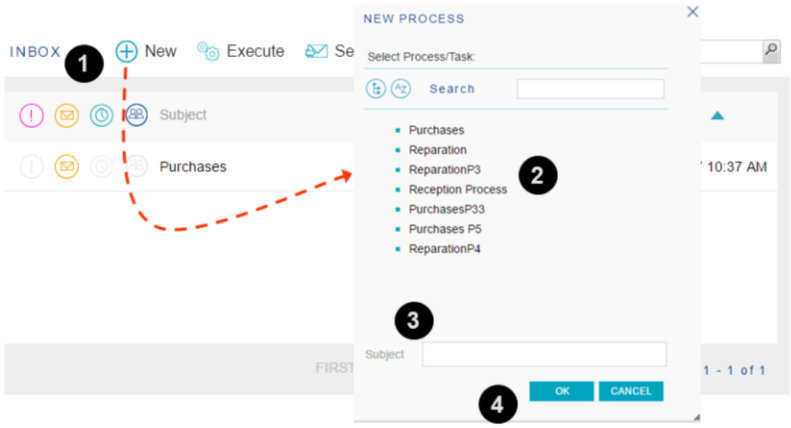
- Principal, Seguridad, Avanzados

Execution and Monitoring Interface

GeneXus

Inbox

Start of new process instances: Only users who have roles associated with the start element



1- New Button
2- Select process/task
3- Subject
4- Confirm

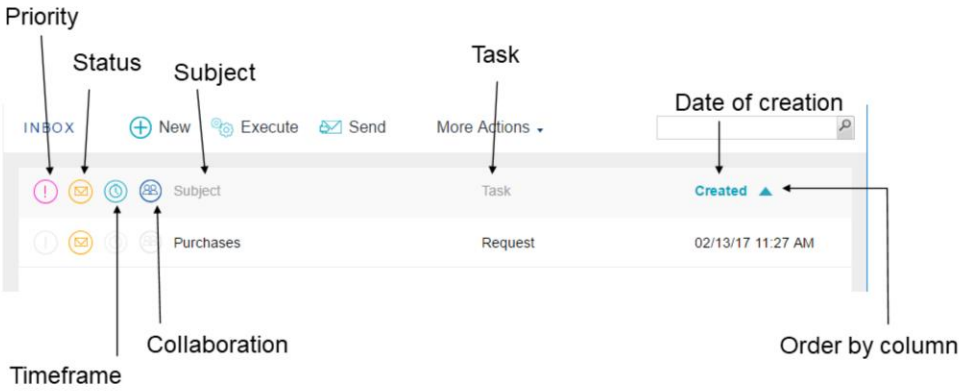
Esta es la bandeja de entrada donde los usuarios interactúan con las tareas ejecutándolas y completándolas.

Se podrán iniciar instancias de proceso cuyo símbolo inicial sea un evento Start, una tarea Interactiva o bien una tarea Batch; además, el usuario logueado debe contar con el rol asociado a alguno de estos símbolos.

Execution and Monitoring Interface

GeneXus

Inbox



La grilla que contiene las tareas a ejecutar está compuesta por las siguientes columnas:

Prioridad: Esta columna indica la prioridad de la instancia de proceso. Puede ser alta, normal o baja, si es normal no vemos ningún ícono.

Status: La tarea está pendiente de ejecución o no

Plazo de tiempo: Indica el tiempo que queda para que la tarea expire.

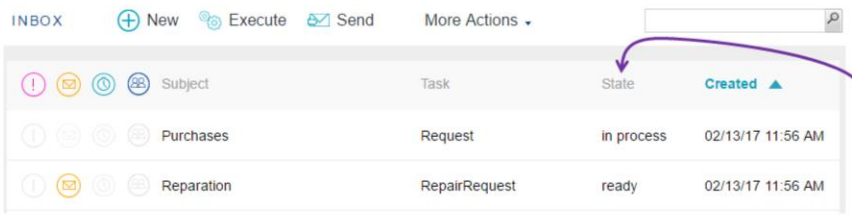
Colaboración: Indica que hay otros usuarios colaborando en la tarea.

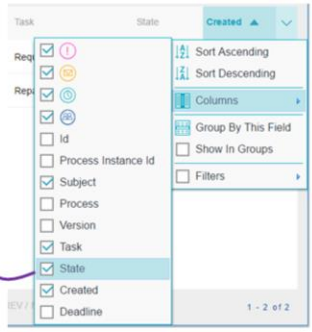
Asunto: Indica el asunto de la instancia de proceso.

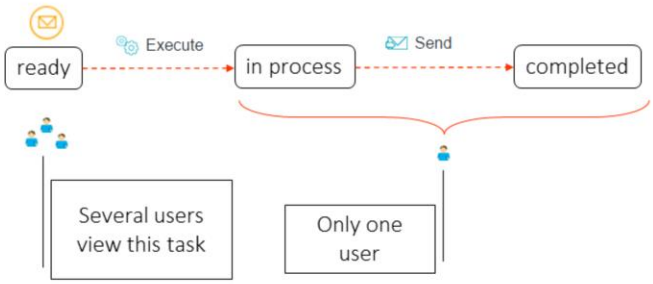
Tarea: Indica el nombre de la tarea que se está procesando.

Fecha de creación: Indica la fecha de creación de la tarea.

Implementation cycle of a task

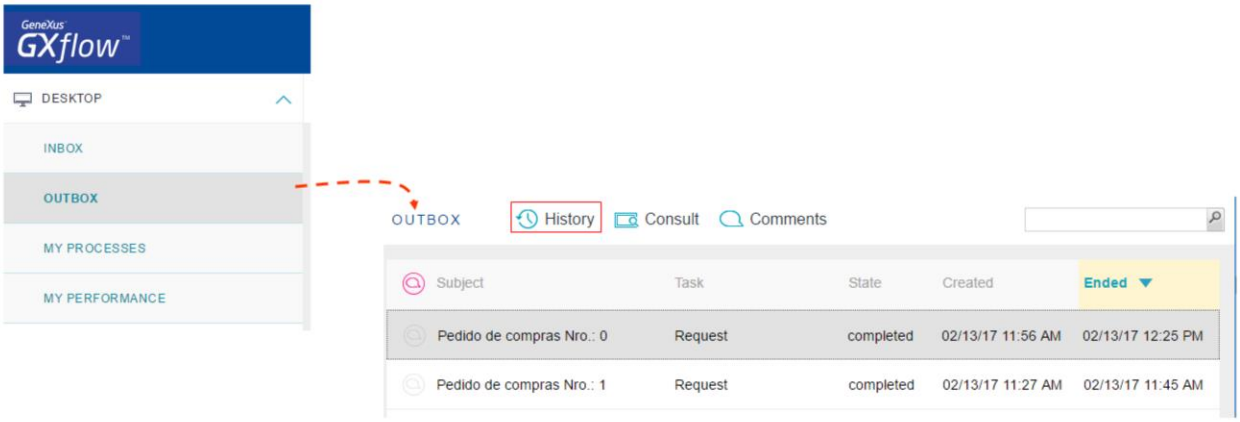






Outbox

- Tasks completed by the user



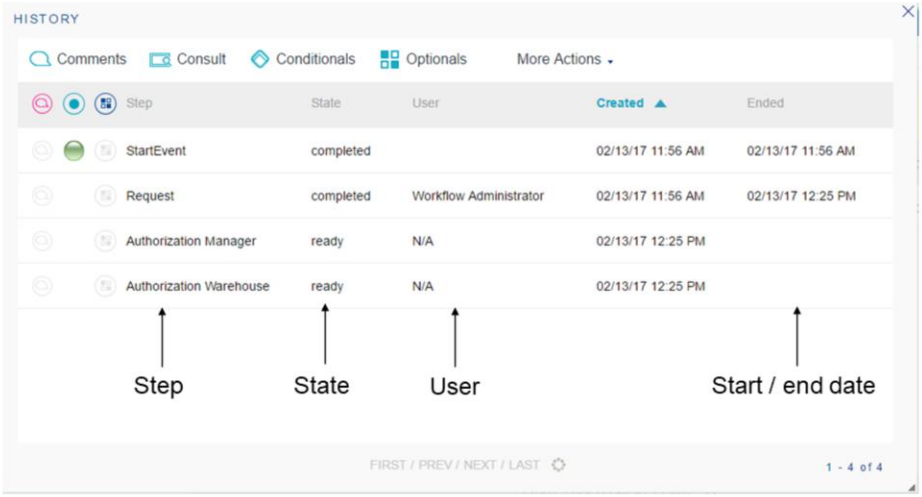
La bandeja de salida nos permite ver las tareas que el usuario logueado procesó.

También es posible consultar la tarea, si fueron insertados comentarios y el histórico del proceso.

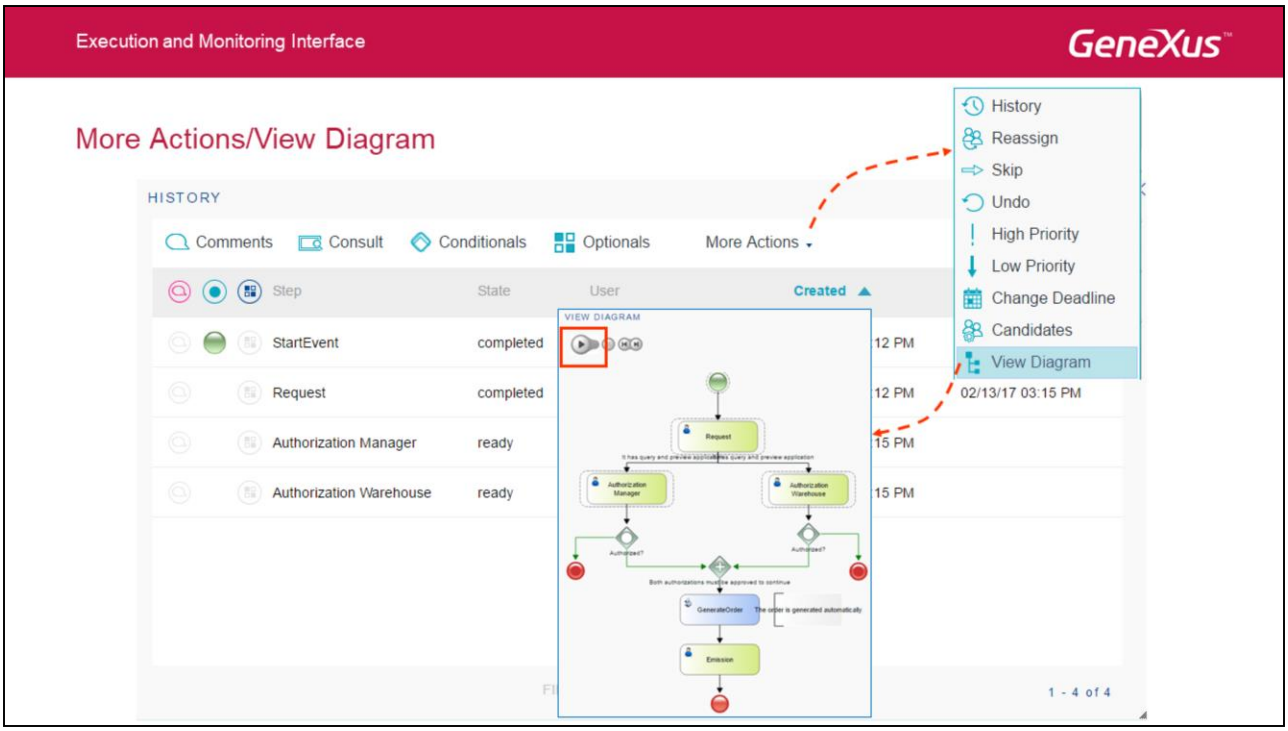
GeneXus

History Log

- Historical data of process execution



La historia de una instancia de proceso nos permite visualizar como se dio el flujo a medida que se fue procesando, por ejemplo que valor tomaron las condiciones, que camino opcional se eligió, que usuario procesó cada tarea, en que fecha se dio inicio y fin.



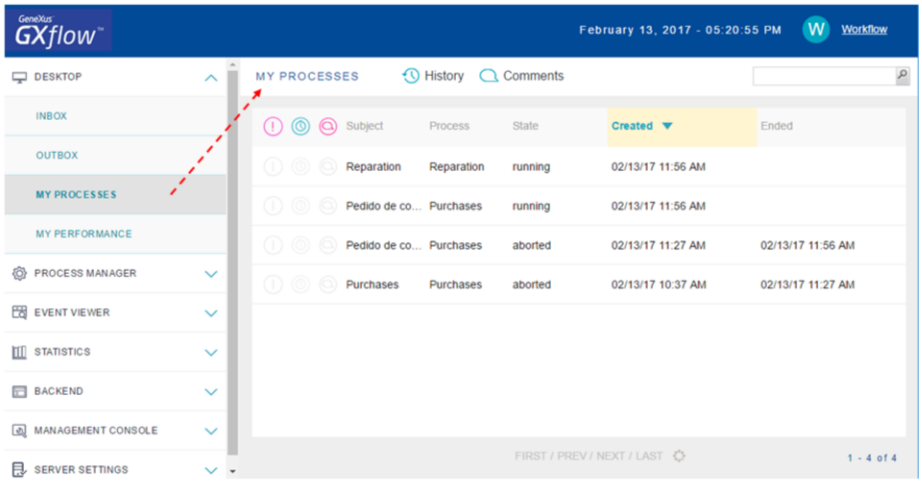
También es posible tener una representación gráfica del proceso y visualizar el orden en que se ejecutaron las diferentes tareas solo haciendo clic en el botón 'Ver diagrama' como se muestra en la imagen.

Al dar clic en el botón de reproducción el usuario inicializará la animación que mostrará paso a paso las tareas que se han generado para la instancia del proceso consultada.

Además si posiciona el puntero del mouse sobre alguna de las tareas aparecerá un pequeño recuadro de texto con información sobre la tarea.

My Processes

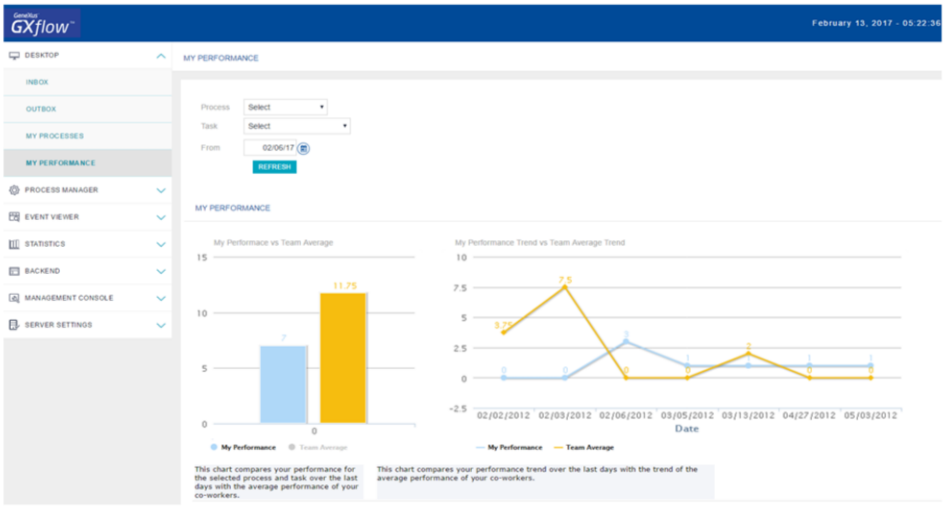
- Process instances created by the user



La aplicación de “Mis Procesos” nos permite visualizar las instancias que el usuario creó, comentarios que se hayan ingresado y la historia del proceso.

My Performance

- Compare user performance

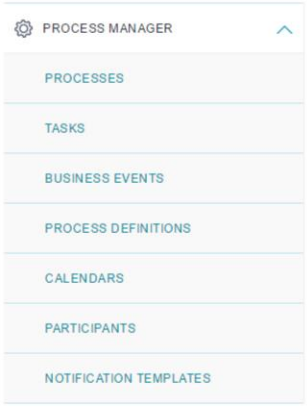


Esta aplicación permitirá al usuario comparar su rendimiento en base a una fecha para los procesos y tareas seleccionados con el rendimiento de sus compañeros de trabajo que compartan un mismo rol o tarea.

GeneXus

Process Manager

- Manages everything related to processes (instances, definitions, statuses, etc.)

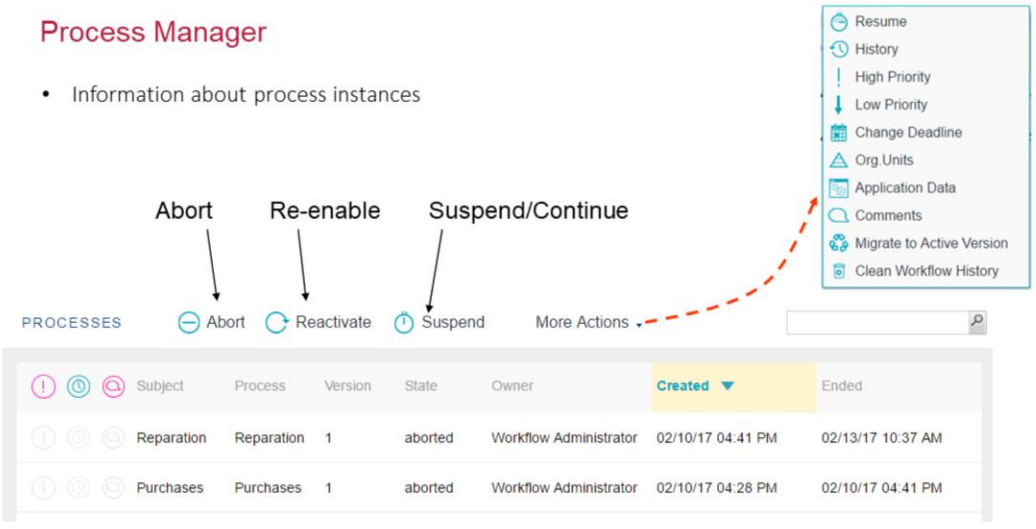


El administrador de procesos cuenta con las aplicaciones de administración que tienen que ver con los procesos de negocio.

GeneXus

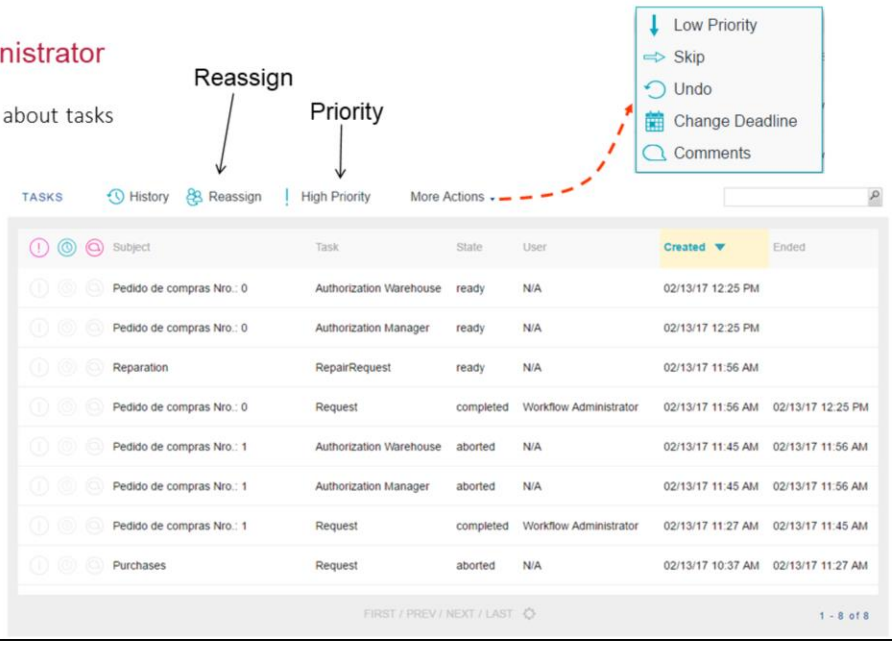
Process Manager

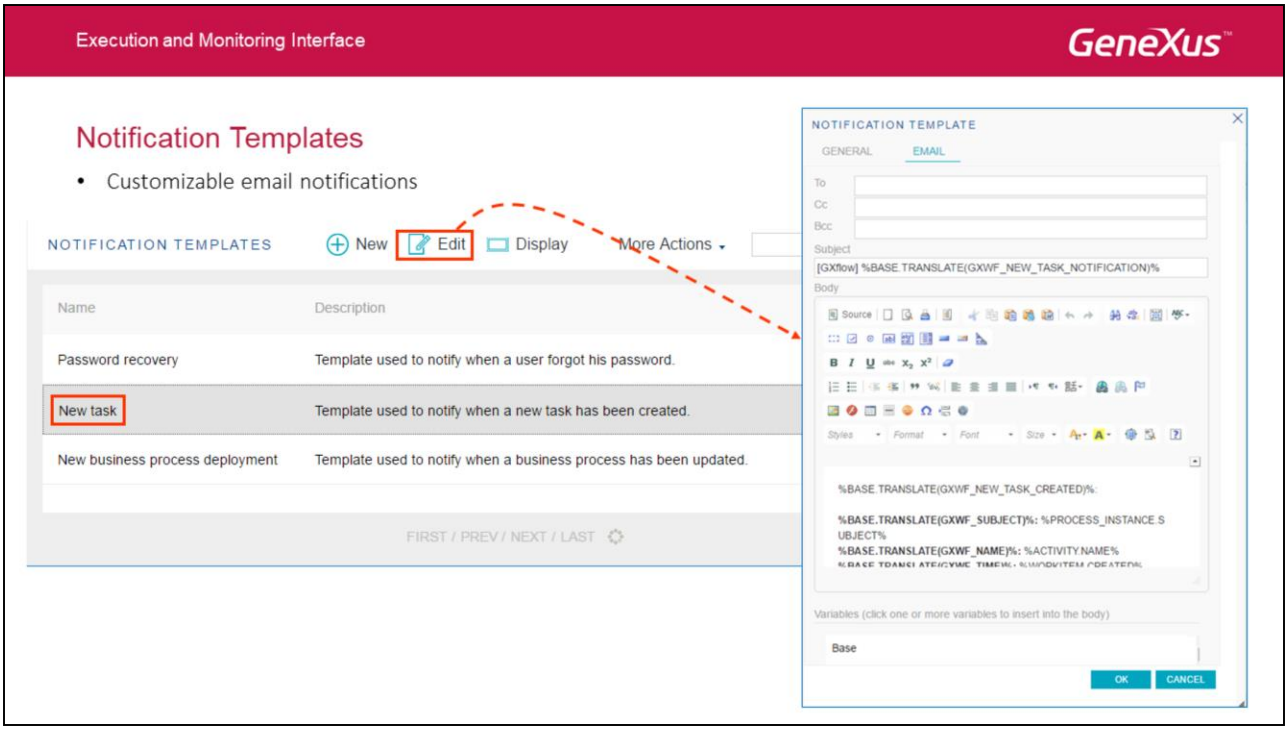
- Information about process instances



Tasks administrator

- Information about tasks

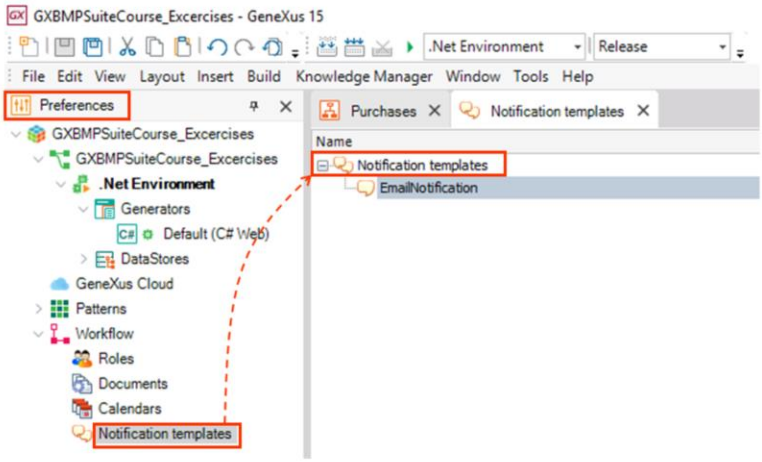




Existen plantillas predefinidas para la notificación de correo electrónico, las cuales puede ser personalizadas por medio de variables de proceso. El usuario administrador podrá editar el destinatario, asunto y cuerpo del mensaje como se muestra en la imagen.

GeneXus

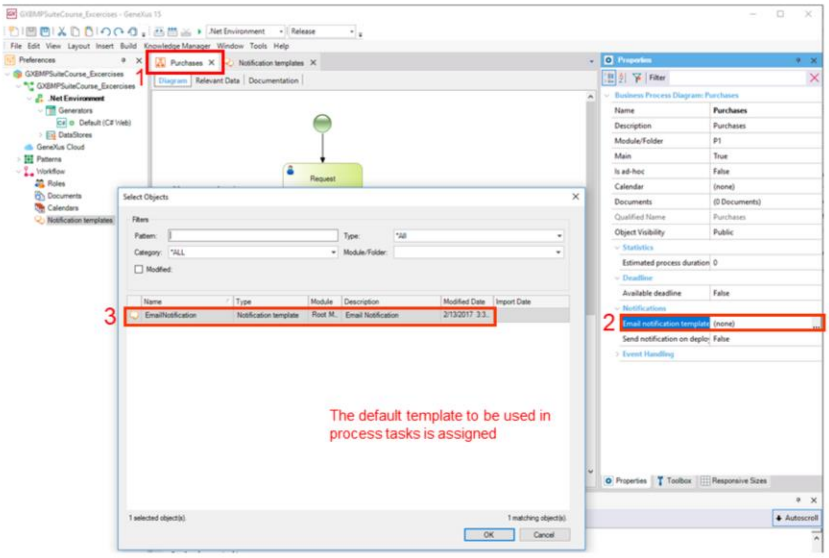
Creating a template in GeneXus



Otra opción es crear una plantilla nueva.

Para ello hay que ir a Preferences – Workflow - Notification templates, al momento de definir la plantilla hay que indicar el nombre y una descripción.

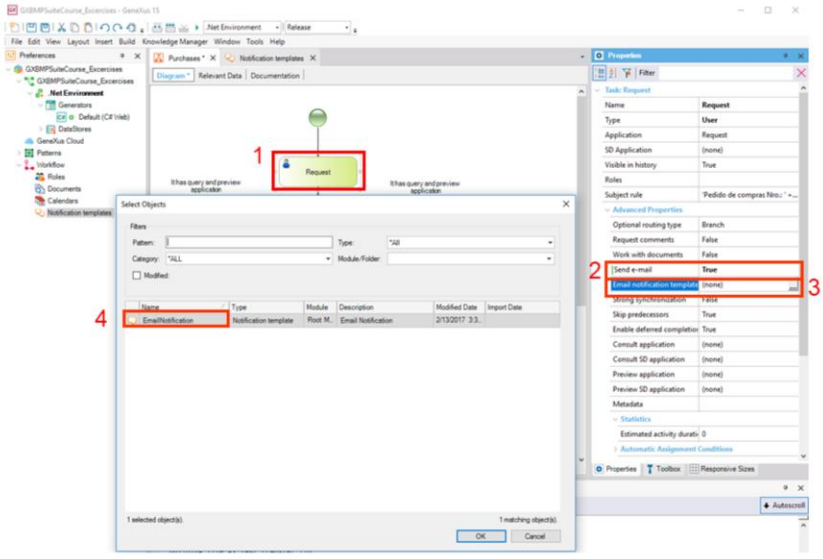
Associating the template at diagram level



Las plantillas de notificación se pueden asociar a nivel de diagrama, dejándolas disponibles para todas las tareas del proceso.

También es posible enviar una notificación al hacer un deploy del proceso, para ello solo hay que setear la propiedad «Send Notification on deploy» en True.

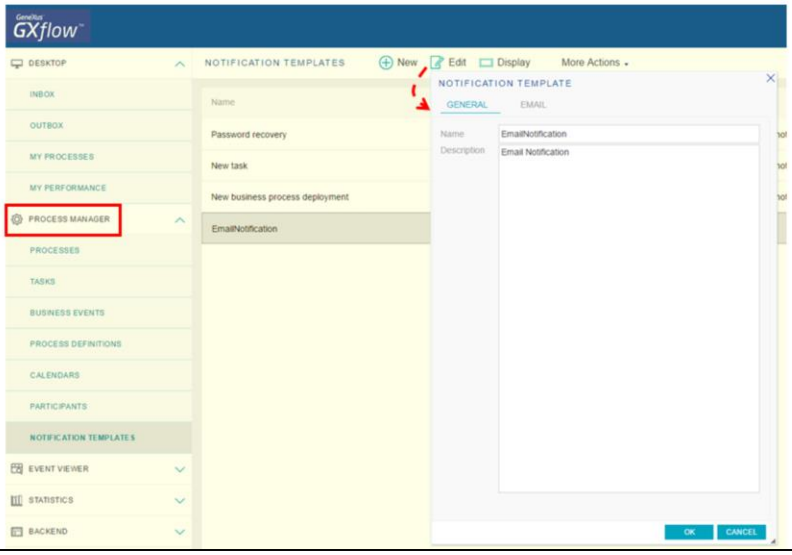
Associating the template at task level



Las plantillas de notificación se pueden asociar directamente a cada tarea, para ello hay que setear la propiedad «Send Email» en True, posteriormente se elige la plantilla.

GeneXus

Editing the template in GXflow Client

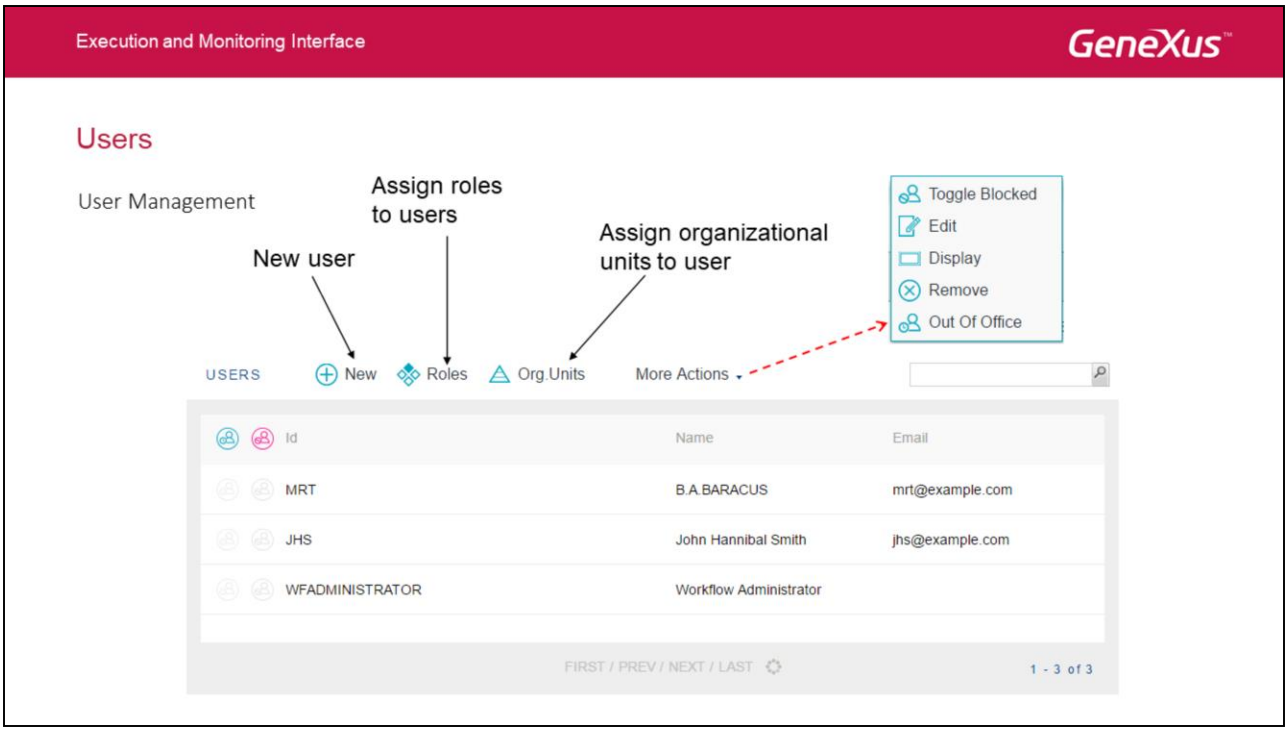


Una vez definida la plantilla a nivel de KB, se puede editar la misma en el Cliente de GXflow teniendo la posibilidad de configurar el contenido del mensaje, asunto y destinatarios.

Management Console

- Management of Users, Roles and Organizational Units

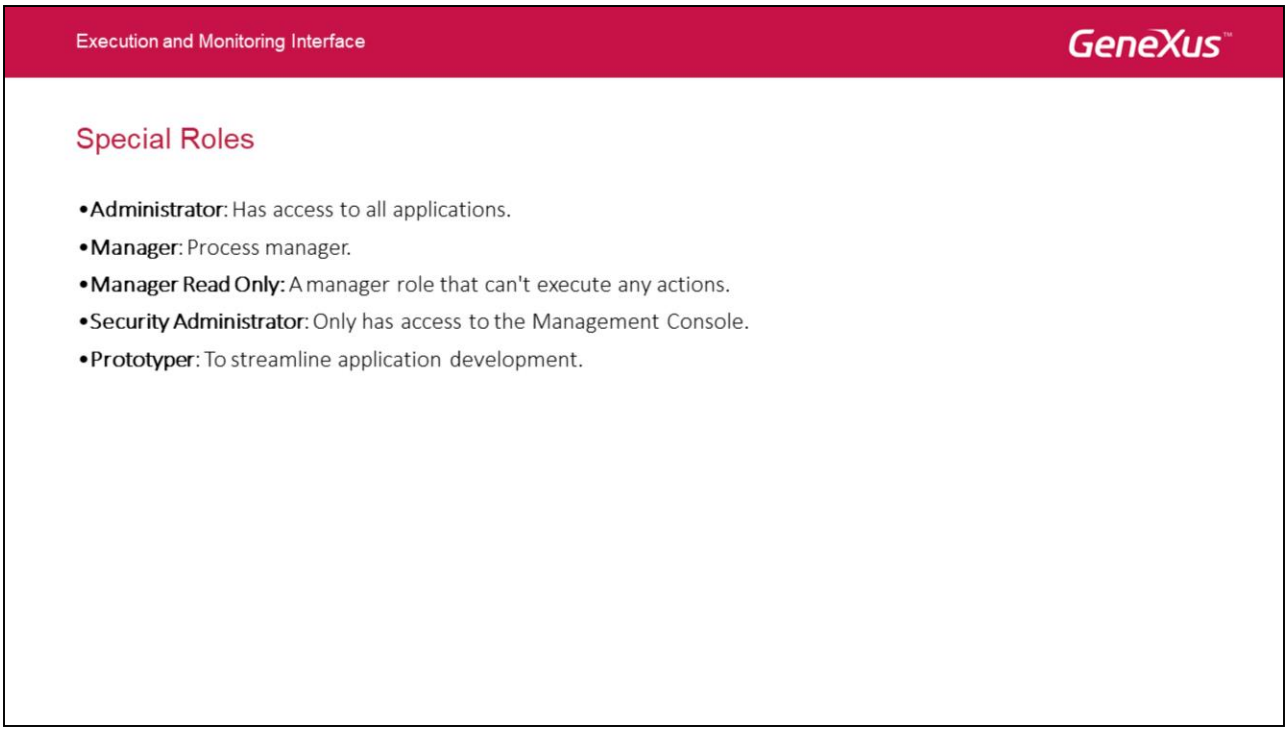
Mediante esta aplicación se manejan los usuarios, roles, unidades organizacionales y su definición. Esto nos permite restringir las tareas que ven los usuarios y puede procesar.



La aplicación de administración de usuarios permite dar de alta usuarios, asignarles roles, editar su información, mostrar su información, eliminarlos, agregarles unidades organizacionales y configurar que estarán fuera de la oficina.

Los usuarios se nominan automáticamente y si no están nominados no se les podrá asignar un rol.

Respecto al bloqueo, el motor procede a bloquear los usuarios si se intenta ingresar con una contraseña inválida más de tres veces (se puede modificar configurando la política de seguridad).



Además de los roles creados por el usuario para administrar la seguridad de las tareas, GXflow cuenta con roles especiales que pueden ser asignados a cualquier usuario:

Administrator: Cuando un usuario con este rol accede al cliente GXflow visualiza y puede realizar todas las acciones en todas las aplicaciones ya sea el escritorio, el administrador de procesos, el visor de eventos, la consola de administración y las preferencias del servidor.

Manager: Cuando un usuario con este rol accede al cliente GXflow visualiza solamente la aplicación administración de procesos y puede realizar todas las acciones relacionadas con las instancias de procesos.

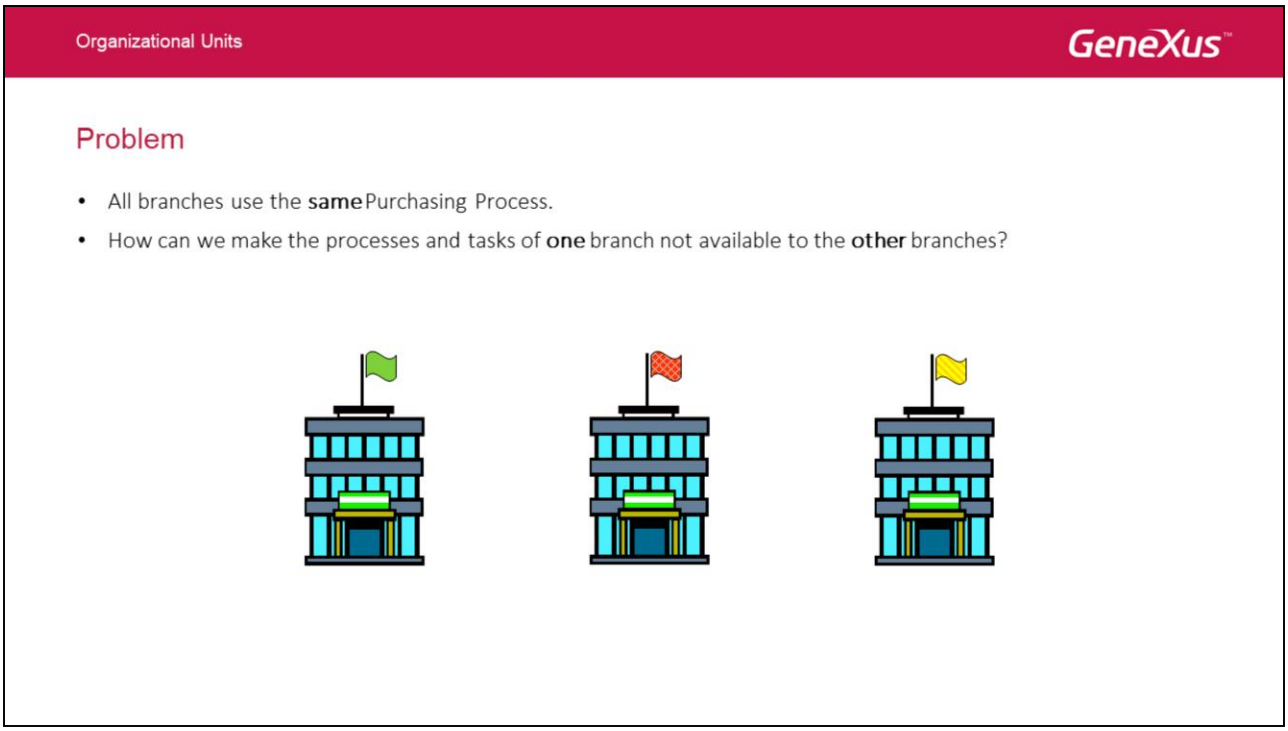
Manager Read Only: Al igual que el manager el usuario puede ver solamente la aplicación de administración de procesos pero a diferencia del mismo no puede realizar acciones.

Security Administrator: Cuando un usuario con este rol accede al cliente GXflow, este puede ver solamente la consola de administración.

Prototyper: Este rol permite ejecutar todas las tareas sin importar el rol que tengan asignado. Esto permite facilitar el desarrollo.

Module 4

Organizational Units



Muchas veces dentro de las empresas nos encontramos con que existen ciertas restricciones por la cual no todas las personas que trabajan en la empresa pueden ver las instancias de los mismos procesos.

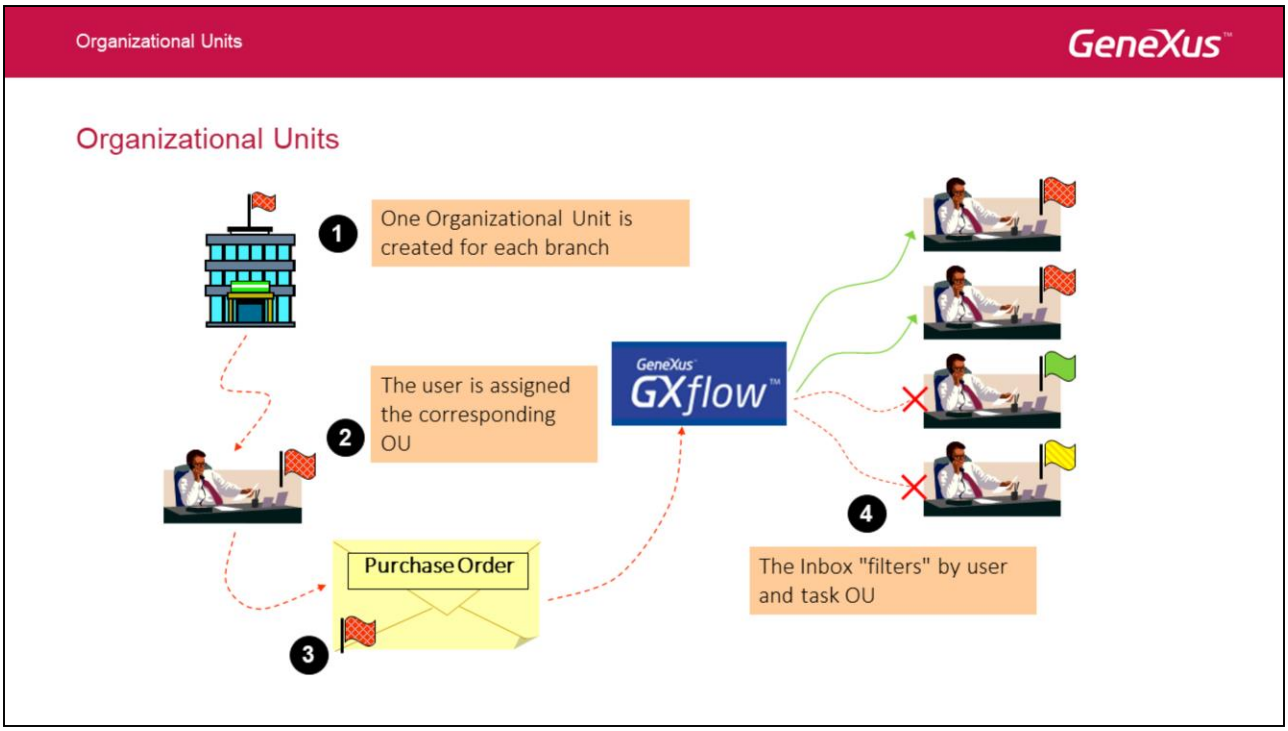
Por ejemplo, se tiene una Empresa con varias Sucursales y todas las Sucursales utilizan el mismo proceso de Compras.

¿Cómo hacemos para que los procesos y tareas de **UNA** Sucursal **NO** estén disponibles para las demás sucursales ?

Para solucionar esto se cuenta con el concepto de “Unidades Organizacionales”.

Las Unidades Organizacionales (UO) permiten definir grupos dentro de una organización, como ser sucursales, departamentos, equipos. Una vez definidas las unidades se crea una instancia de cada una de ellas y dentro de cada instancia se le asocian los miembros de la misma.

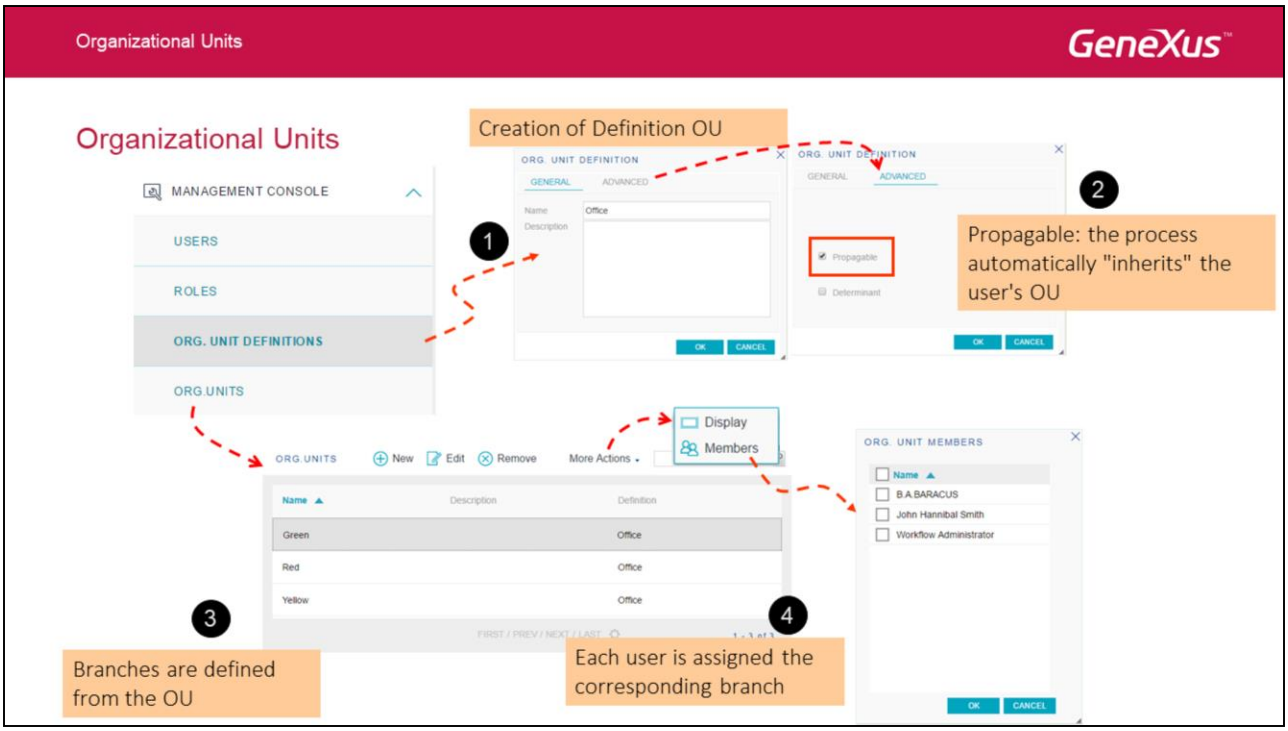
En nuestro caso la empresa cuenta con tres sucursales, la sucursal *Green*, la sucursal *Red* y la sucursal *Yellow*. Dentro de la sucursal *Green*, se cuentan con 2 personas para el área de compras, en la sucursal *Red* se cuenta con 5 personas y la sucursal *Yellow* cuenta con 4 personas.



¿Cuáles son los pasos que deberíamos seguir para poder implementar este caso ?

En primer luegar establecemos que la definición de la unidad organizacional será *Offie*.

1. Luego se define una UO por cada sucursal de la empresa, en este caso *Red, Green y Yellow.*
2. Una vez definidas las UO, se asigna a cada usuario la UO correspondiente.
3. A partir de ese momento el proceso va a heredar la UO del usuario que lo inicia.
4. Entonces en la bandeja de entrada de cada usuario no solo se va a filtrar por la tarea sino que también por la UO.



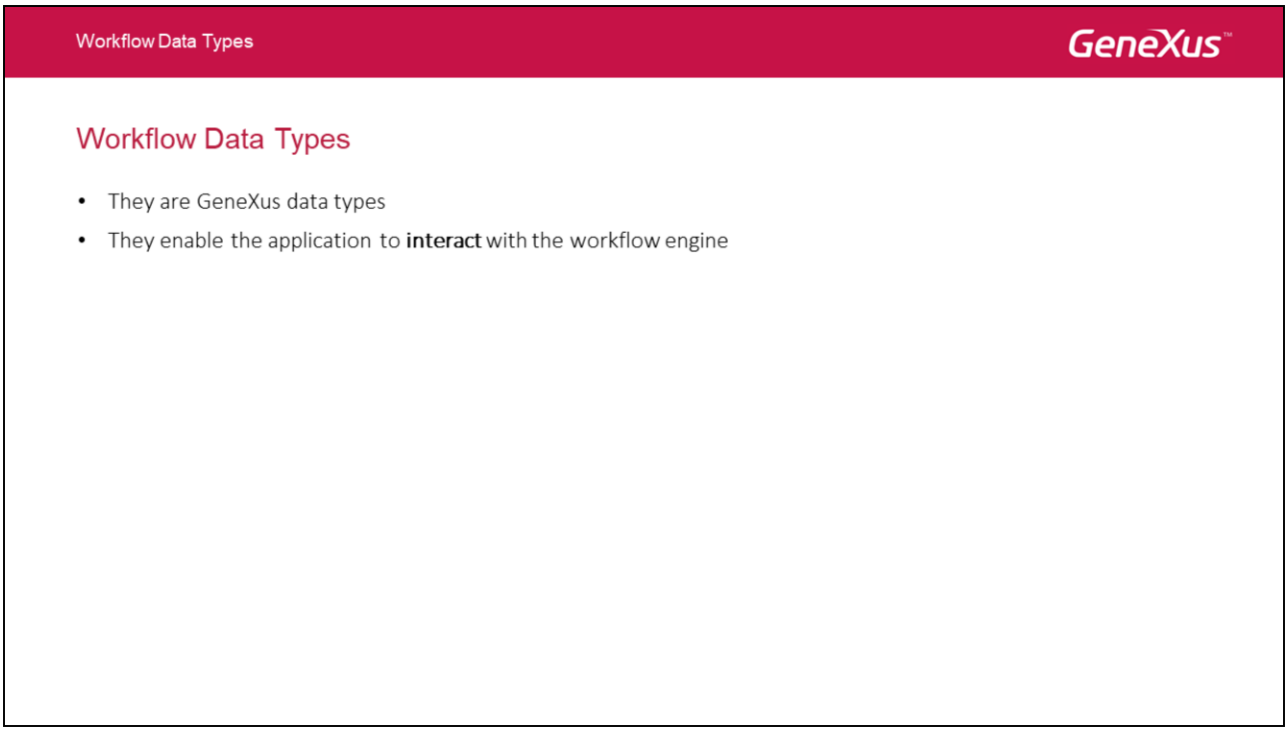
Para implementar esto, debemos ir al cliente de GXflow y entrar con el Usuario Administrador o con cualquier usuario que tenga permisos para esto. Luego, hay que ir a **Management Console - Organizational Model** y hacer lo siguinete:

1. En **Organizational Unit Definitions** crear una nueva llamada *Office*.
2. Se indica si el **Propagable**, para que el proceso herede de forma automática la UO del usuario.
3. Desde la opción **Organizational Units** se definen las diferentes sucursales con las que cuenta la Empresa, en este caso *Red*, *Green* y *Yellow*.
4. A cada sucursal se le asignan los usuarios correspondientes.

A partir de este momento ya queda definido que cada sucursal va a ver solamente los procesos de compra que pertenecen a la misma.

Module 5

Workflow Data Types

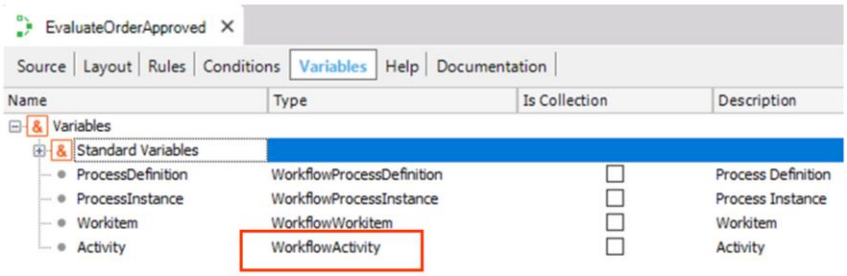


Los tipos de datos Workflow son un conjunto de tipos de datos GeneXus que permiten interactuar con el motor de workflow de GXflow.

GeneXus

Workflow Data Types

- Workflow Prefix
- Defined as **variables** in GeneXus objects.



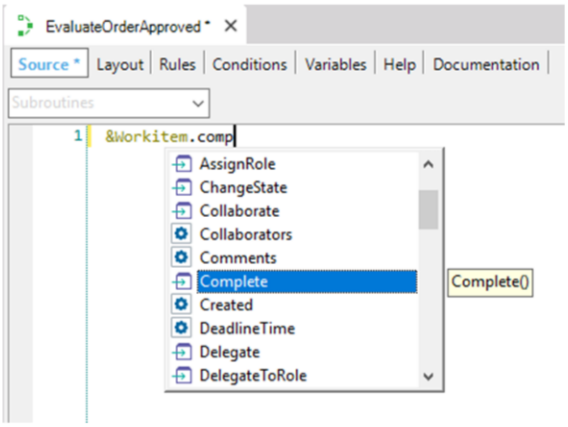
Todos los tipos de datos Workflow tienen este mismo prefijo para diferenciarlos del resto de los tipos de datos que soporta GeneXus.

Para poder usar un determinado tipo de datos Workflow, se debe definir primeramente una variable basada en dicho tipo de datos (mismo procedimiento que con cualquier tipo de dato GeneXus).

GeneXus

Workflow Data Types

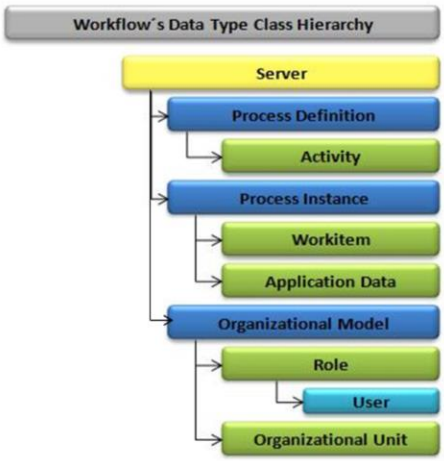
- Intellitips that facilitate use



Al igual que con el resto de los tipos de datos de GeneXus, se cuenta con información contextual (intellitips) que ayudan a la utilización de los mismos.

GeneXus

Hierarchy of Workflow Data Types



Los tipos de datos Workflow forman una jerarquía de tipos que nos permite navegar por la misma partiendo desde el punta de entrada principal que es el tipo de datos Server.

A partir del tipo de datos Server y utilizando métodos y propiedades podemos ir accediendo a otros tipos y así sucesivamente.

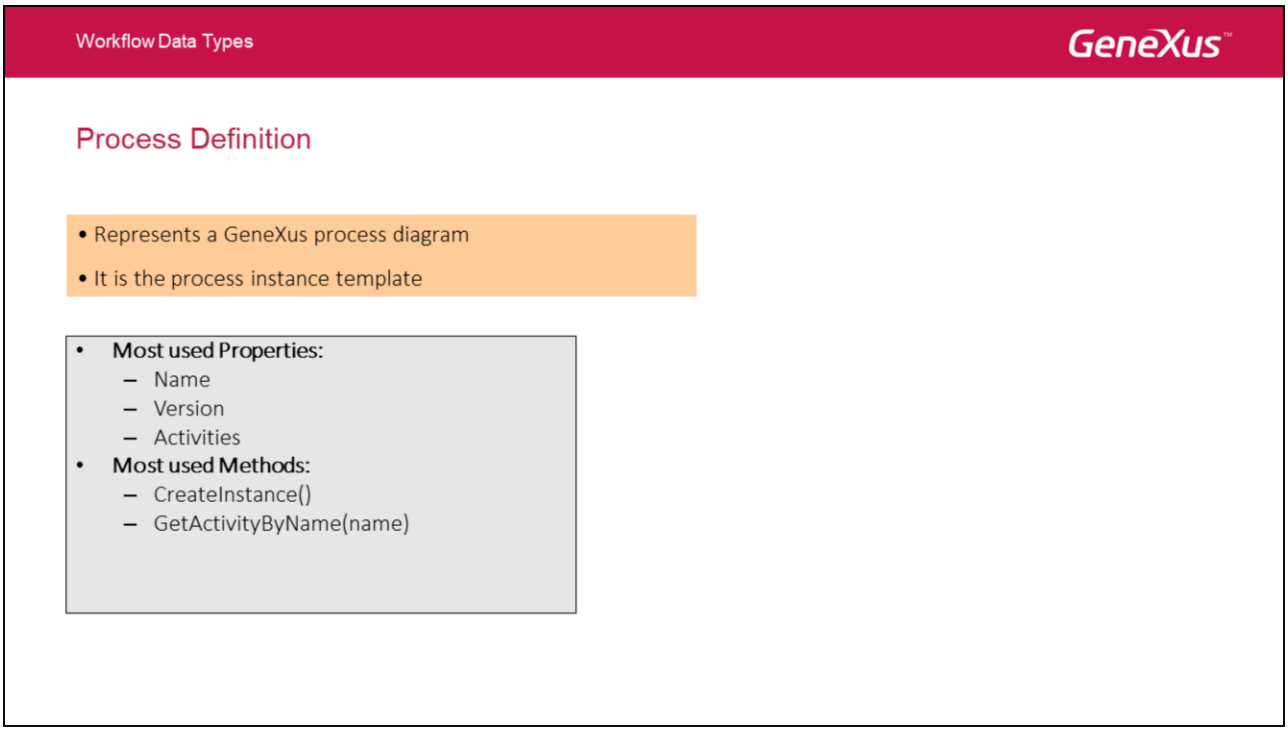
GeneXus

Basic Types

- **ProcessDefinition** – Data type: WorkflowProcessDefinition
- **ProcessInstance** – Data type: WorkflowProcessInstance
- **Workitem** – Data type: WorkflowWorkitem
- **Context** – Data type: WorkflowContext
- **Application Data** – Data type: WorkflowApplicationData
- **Error** – Data type: WorkflowError

Los tipos de datos básicos de workflow, en el sentido que son los más utilizados, son los siguientes.

Por un tema de comodidad, vamos a omitir el prefijo “Workflow” común a todos los tipos que vamos a ver.



El tipo Process Definition se corresponde con un diagrama de procesos publicado.

En rigor, se corresponde una determinada versión de un diagrama, ya que cada vez que se publica un diagrama, se crea un nuevo Process Definition con el número de versión incrementado en uno.

Algunas de las propiedades más importantes son las siguientes:

Name

Permite obtener el nombre del proceso

Version

Permite acceder al número de versión

Activities

Permite obtener una colección con las actividades que componen el proceso.

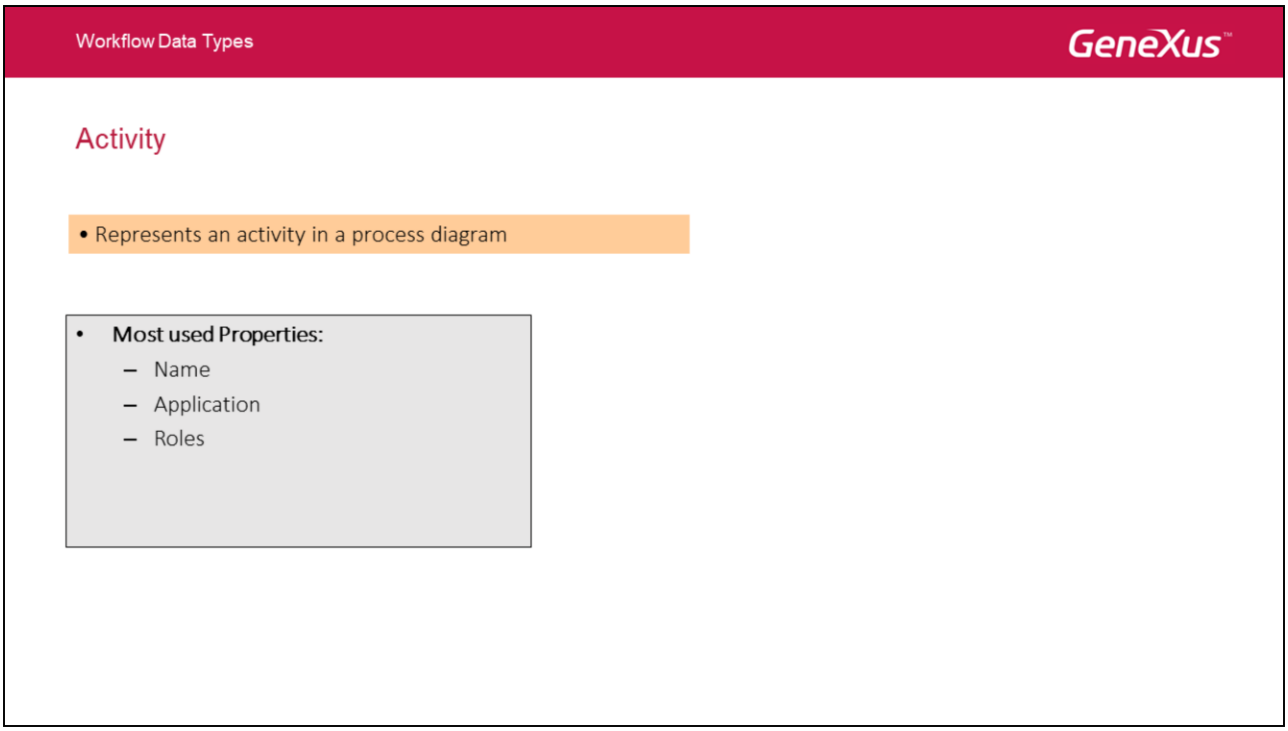
Dentro de los métodos más importantes tenemos:

CreateInstance

Permite crear una nueva instancia de proceso basada en esta definición.

GetActivityByName

Permite, recuperar una actividad a partir de su nombre.



El tipo Activity representa una actividad de un diagrama de proceso. Esto es, lo que un diagrama se representa con un rectángulo de esquinas redondeadas y significa cierto trabajo a realizar en el proceso.

Algunas de las propiedades disponibles son:

Name

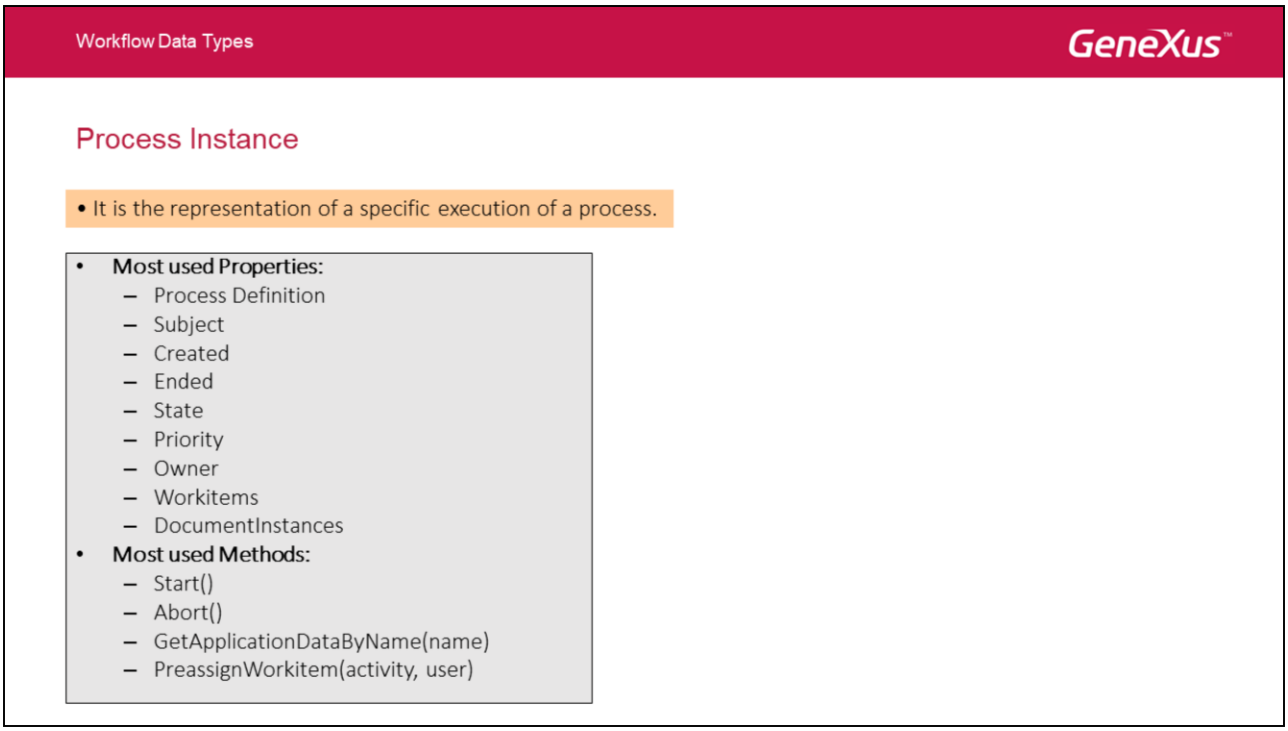
Devuelve el nombre de la actividad

Application

Devuelve el nombre de la aplicación asociada a la actividad.

Roles

Devuelve una colección con los roles que tienen permiso para ejecutar dicha actividad



El tipo Process Instance se utiliza para representar a las instancias de una definición de proceso.

Cada instancia de proceso representa una ejecución de una definición de proceso. Es decir, si tenemos por ejemplo un diagrama de procesos “Procesar Compra”, las diferentes instancias podrían ser “Procesar Compra Nro1”, “Procesar Compra Nro2”, etc.

Las propiedades más importantes son:

Process Definition

Devuelve la definición de proceso en la cual esta basada esta instancia

Subject

Permite acceder al asunto, una propiedad que se utiliza para describir a la instancia de proceso.

Esta propiedad es importante porque esta descripción debe ser generada por la aplicación.

Created

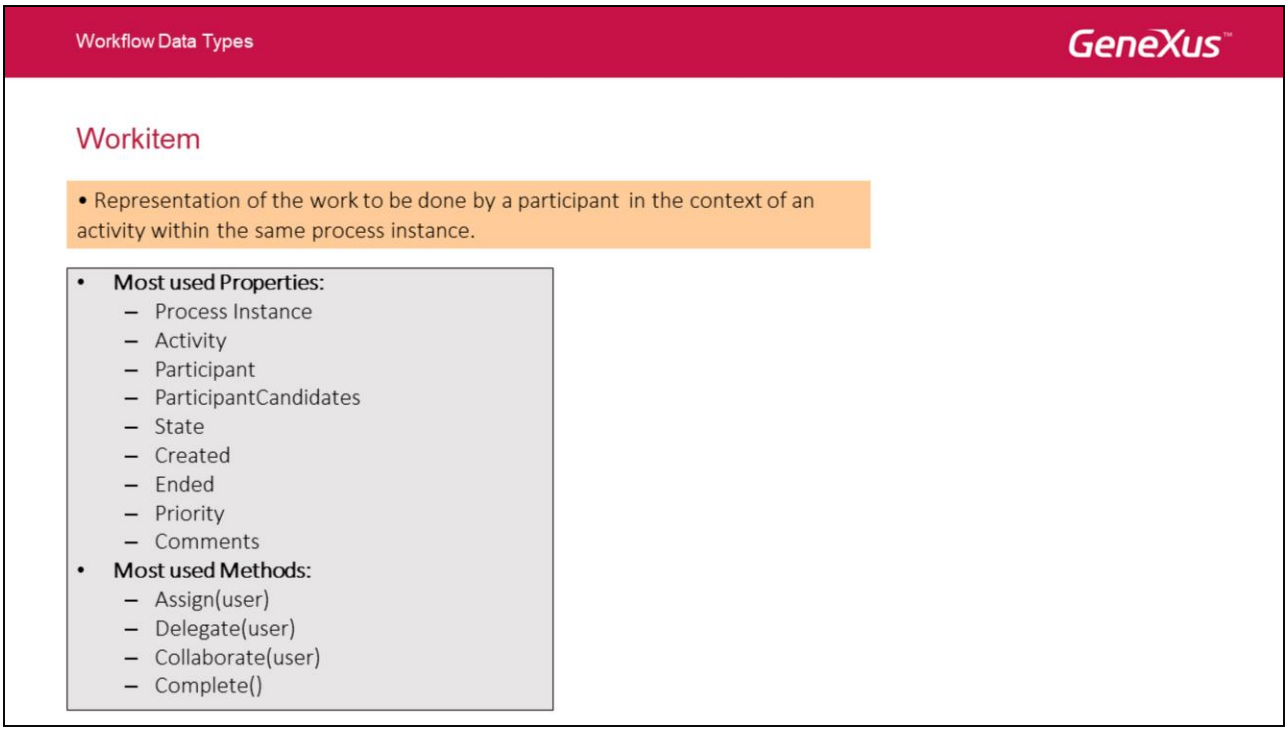
Devuelve la fecha de creación.

Ended

Devuelve la fecha de finalización.

State

Permite conocer el estado de la instancia de proceso (ver Enumerado WorkflowProcessInstanceState)



Un workitem representa el trabajo a ser realizado por un participante en el contexto de una actividad dentro de una instancia de proceso.

Dentro de las propiedades más importantes tenemos:

ProcessInstance

Es la instancia de proceso a la que pertenece.

Activity

Devuelve la actividad que representa el workitem.

Participant

Usuario que tiene el workitem asignado (puede no existir).

ParticipantCandidates

Devuelve una colección con los posibles candidatos para ejecutarlo (los candidatos los determina el motor en base a roles y unidades organizacionales que tienen asignados los usuarios).

State

Devuelve el estado del workitem (ver enumerado WorkflowWorkitemState)

Created

Devuelve la fecha de creación del workitem.

GeneXus

Workflow Context

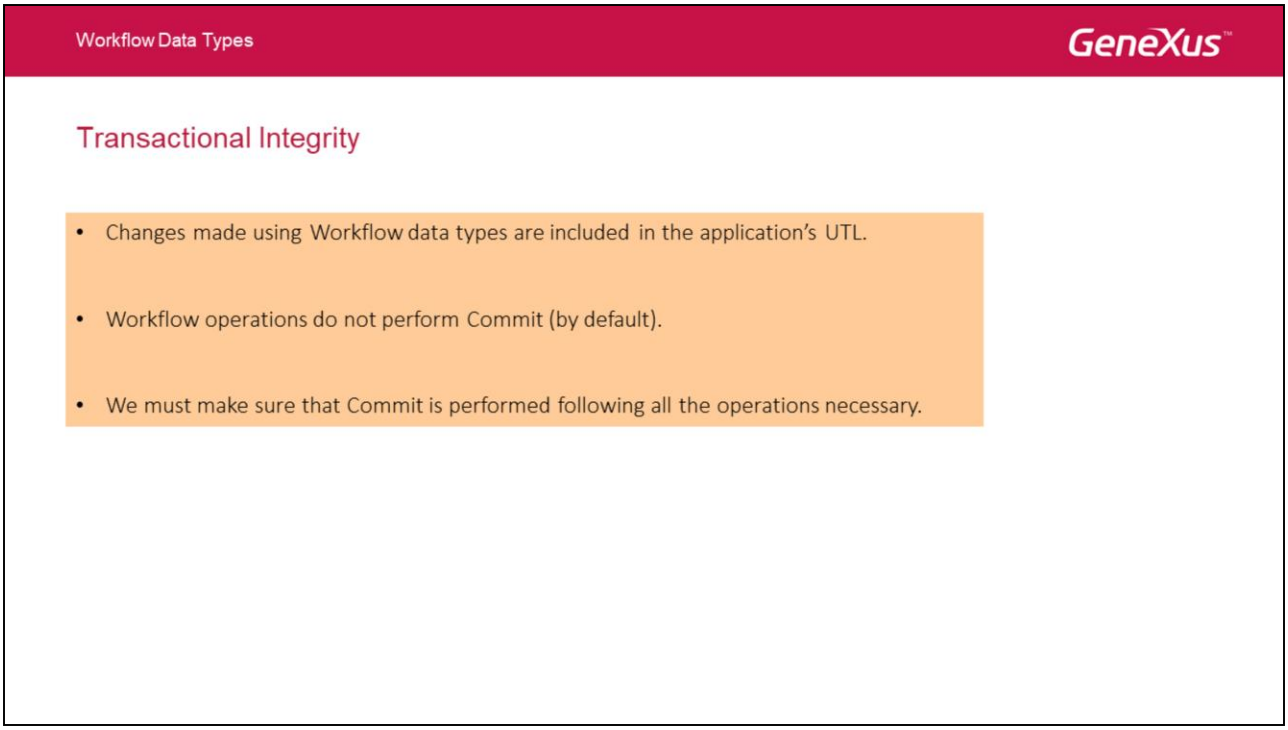
- Allows access to context information in executing an application associated with an activity
- It is automatically instanced
- Properties are:
  - ProcessDefinition
  - ProcessInstance
  - Workitem

El tipo de datos Context permite, valga la redundancia, obtener la información de contexto desde una aplicación asociada a una actividad.

La información de contexto esta formada por el workitem, instancia de proceso y definición de proceso en donde esta ejecutando la aplicación.

Estos pueden ser accedidos mediante las propiedades correspondientes (Workitem, ProcessInstance, ProcessDefinition).

Cabe aclarar que el contexto se instancia automáticamente siempre y cuando la aplicación asociada a la actividad (ya sea de tipo User o Script) sea un objeto GeneXus y que forme parte de la misma KB donde se encuentre el diagrama de procesos.

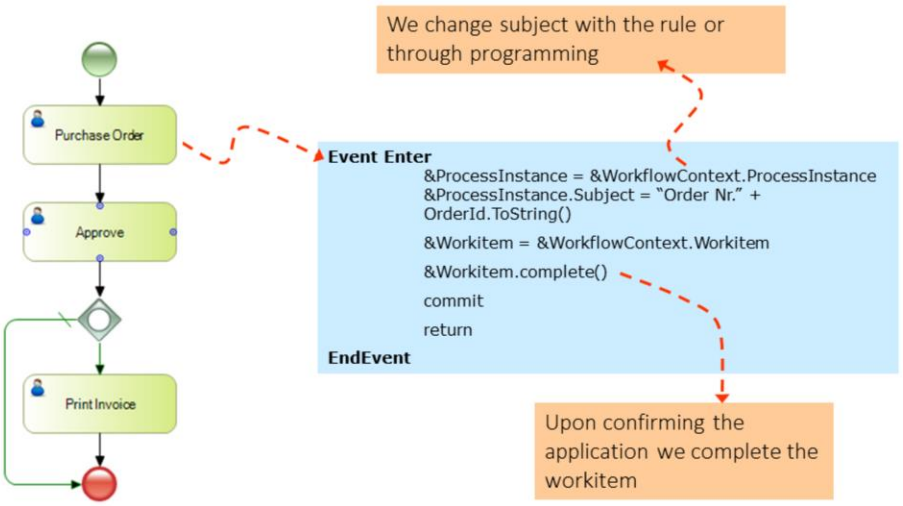


Los cambios realizados mediante el uso de los tipos de datos workflow están comprendidos dentro de la Unidad de Trabajo Lógica de la aplicación.

Esto significa que dichos cambios serán "comiteados" cuando lo haga la aplicación (ya sea que lo haga GeneXus o que lo indique explícitamente el usuario mediante el comando commit).

Las operaciones de workflow no hacen commit por lo tanto hay que asegurarse siempre al utilizar los tipos de datos workflow de definir bien la UTL y "commitear" al final.

Example



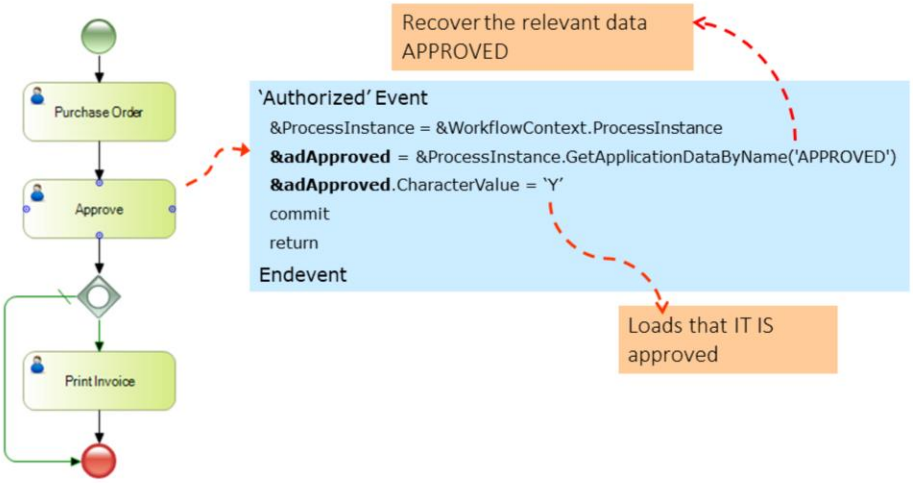
Vamos a pasar a ver ahora algunos ejemplos concretos de casos de uso de los tipos de datos Workflow.

Supongamos que tenemos un diagrama que modela un proceso de compras.

El proceso comienza con una actividad de usuario llamada “Ingresar solicitud de crédito” donde se genera la información de la solicitud. Supongamos que queremos, que una vez se confirme la solicitud, modificar el asunto de la instancia de proceso con el número de la orden. Para ello, recuperamos en primer lugar la instancia de proceso utilizando el contexto, y luego utilizamos la propiedad Subject para modificar el asunto.

De la misma forma, si quisiéramos que una vez se confirme la solicitud dar por finalizado el workitem recuperar el tipo utilizando el contexto y luego invocamos al método Complete.

Example



Continuando con el ejemplo anterior.

Supongamos que luego de la actividad “Solicitud de crédito” tenemos una actividad de usuario llamada “Aprobar solictud de crédito” en la cual un supervisor tendrá la posibilidad de autorizar o rechazar esa solicitud. Y que además, hemos definido una variable llamada “APPROVED” para dejar registrado el resultado de esa decisión.

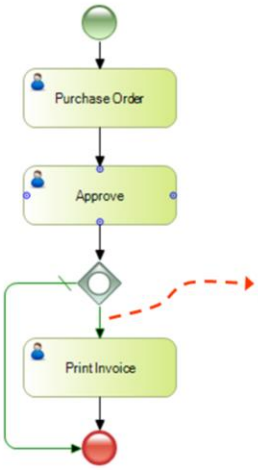
En la aplicación asociada a esta actividad, para poder modificar esta variable, debemos seguir los siguiente pasos:

En primer lugar debemos recuperar la instancia de proceso utilizando el contexto. Luego, sobre la instancia, utilizamos el método GetApplicationDataByName pasándole como parámetro el nombre de la variable que queremos recuperar (“APPROVED”). Una vez recuperada la variable utilizamos la propiedad CharacterValue para asignar el resultado de la acción (si autorizo o no).

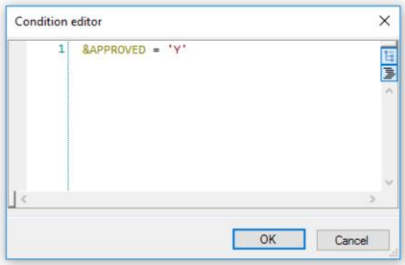
En este caso la propiedad para modificar el valor es la CharacterValue porque definimos la variable “APPROVED” del mismo tipo. Si la hubieramos definido de tipo númerico entonces para modificar o recuperar el valor deberíamos utilizar la propiedad NumericValue.

GeneXus

Example

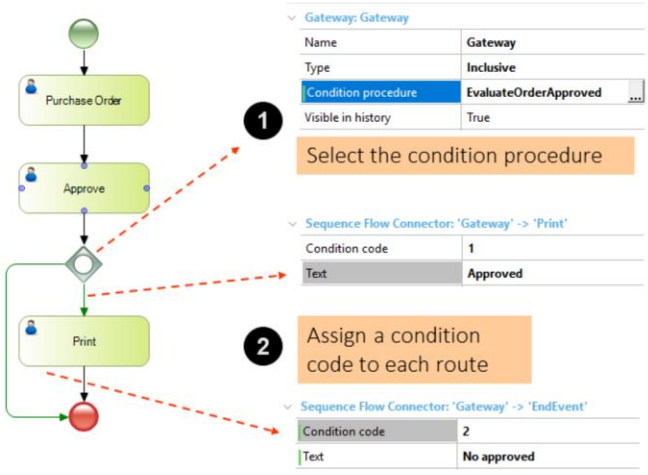


Recover the relevant data APPROVED using a gateway with a condition rule



Continuando con el ejemplo anterior…

Using a condition assessment procedure



Siguiendo con el ejemplo. Supongamos que luego de la actividad de aprobación existe una compuerta con la cual se quiere modelar que si se aprueba la solicitud se pase a una actividad donde se debe emitir el crédito y en caso contrario se da por finalizado el proceso.

Podríamos configurar la compuerta utilizando directamente una regla de condición que compare el valor de la variable “APPROVED” para saber por que camino tomar (como vimos anteriormente en modelado de procesos). Pero en este caso vamos a optar por asociar un procedimiento de condición.

Utilizar un procedimiento para evaluar una condición nos da mayor potencia ya que no estamos limitados a utilizar variable y/o atributos de transacciones asociadas al diagrama y en cambio tenemos la posibilidad de navegar a cualquier atributo de la KB y de modelar una condición mucho más compleja que lo que permite el editor de reglas.

El procedimiento para utilizar un procedimiento de evaluación de condición es el siguiente:

1. En las propiedades de la compuerta seleccionamos el procedimiento de condición.
2. En las propiedades de cada una de las aristas que parten desde la compuerta, asignamos un código de condición (es un número) y opcionalmente un texto que lo describa.

GeneXus

Condition assessment procedure

- parm rule (not to be modified!):

  parm(in: ProcessDefinition, in: ProcessInstance, in: Workitem, out: ConditionalCode)

- Type of ConditionalCode: Numeric(4)

- The value assigned to it must match the route to be followed. Example:
  1 – Approved
  2 – Not approved

Ahora veamos como se debe definir el procedimiento de condición.

Para comenzar, el procedimiento tiene que tener una regla “parm” específica.

Los parámetros que debe recibir el procedimiento son los siguientes:

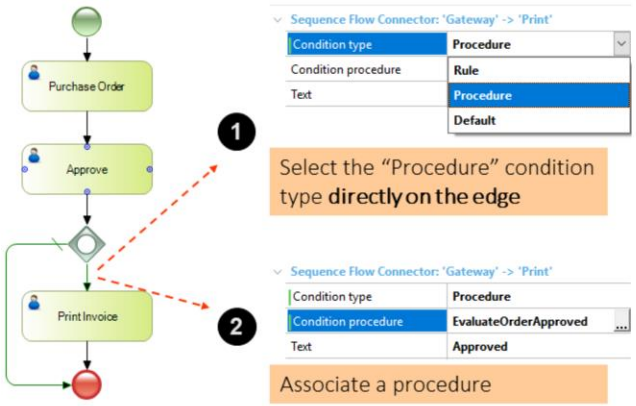
1. Una variable de tipo `Workflow`ProcessDefinition (entrada)
2. Una variable de tipo `Workflow`ProcessInstance (de entrada)
3. Una variable de topo `Workflow`Workitem (de entrada también)
4. Finalmente y la más importante, una variable de tipo número de cuatro de salida, que es donde se devuelve el resultado de la condición.

Las primeras variables son las de contexto y que pueden ser utilizadas por el procedimiento para recuperar información relevante del proceso.

La variable de salida es el código que vamos a devolver y es el que el motor de wf va a utilizar para comparar con los valores asociados a los códigos de las aristas que parten de la compuerta para decidir por que actividad continuar.

GeneXus

Using a condition assessment procedure Alternative Option



Una alternativa al utilizar los procedimientos de condición, es asignar el procedimiento directamente en la arista y no en la compuerta.

Los pasos serían los siguientes:

1. En las propiedades de la arista, en el tipo de condición seleccionamos "Procedure"
2. En la propiedad "Conditional procedure" seleccionamos el procedimiento.

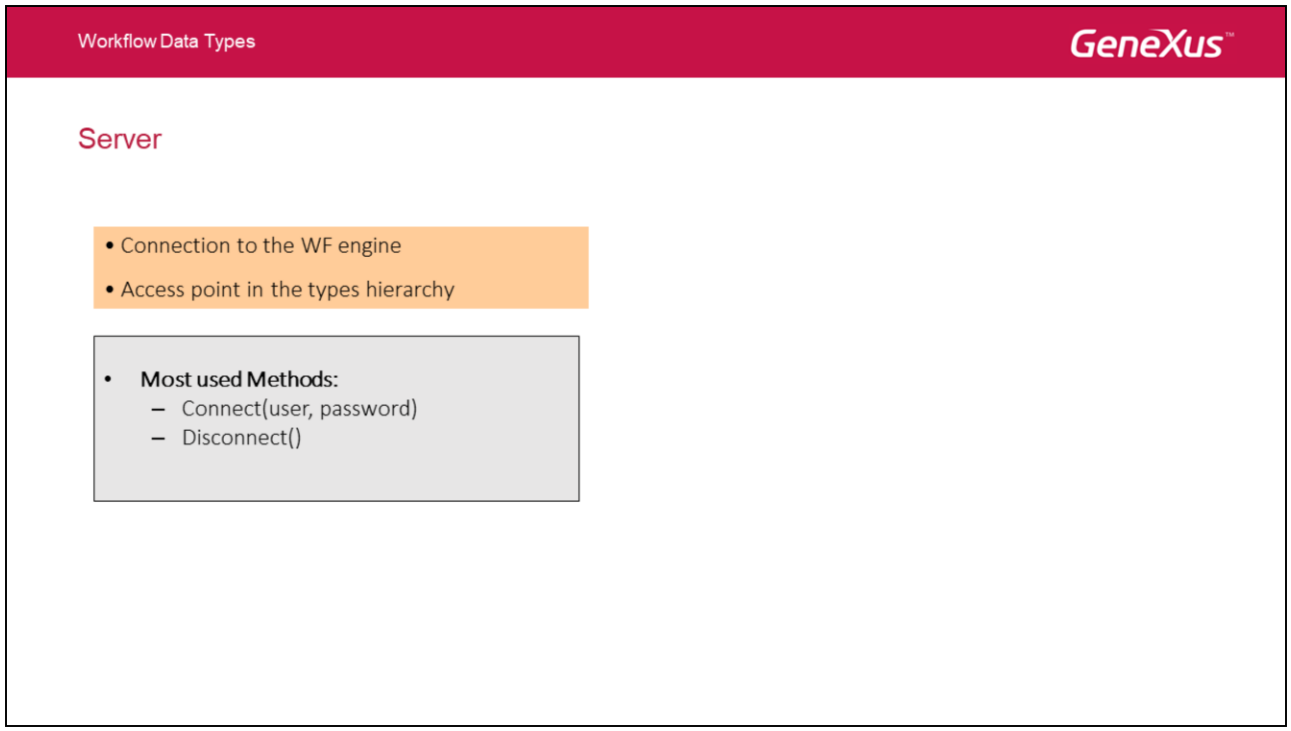
El procedimiento de condición debe respetar la misma regla parm del caso anterior, con la salvedad que el código de condición a retornar esta limitado a los valores uno y cero. Si se devuelve uno quiere decir que se debe continuar por dicha arista. Si se devuelve cero significa que no se debe seguir por ese camino.

GeneXus

Advanced Types

- **Server** – Data type: WorkflowServer
- **OrganizationalModel** – Data type: WorkflowOrganizationalModel

Pasemos a ver ahora tipos de datos workflow más avanzados. En particular vamos a ver los tipos de datos Server y OrganizationalModel.



El tipo de datos Server es el punto de entrada en la jerarquía de los tipos de datos Workflow, esto es, aquel que nos permite acceder a cualquier otro tipo de datos workflow.

El primer paso para poder utilizarlo es invocando al método connect, que es el que nos va a permitir establecer una conexión con el motor de Workflow de GXflow.

Una vez establecida la conexión puedo acceder al resto de sus métodos y demás propiedades.

Una vez finalizada la operativa podemos opcionalmente invocar al método Disconnect para liberar la conexión.

El tipo de datos Server dispone en general de dos tipos de operaciones o métodos.

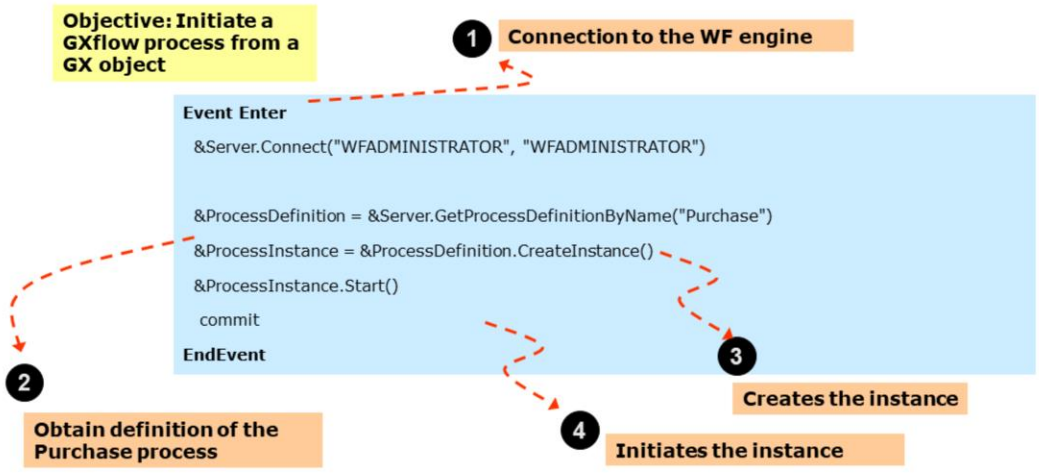
Algunas de ellas tienen el prefijo “Get” y nos permiten obtener un tipo en particular a partir de un identificador o un nombre.

El otro tipo de operaciones son las que tienen el prefijo “List” y nos permiten obtener una colección de variables de determinado tipo que cumplen con determinados criterios de búsqueda.

Los criterios de búsqueda se especifican con un parámetro de tipo Filter.

También se tiene la posibilidad de devolver la colección en un determinado orden. Para ello se dispone de un método replica del original con el literal OrderBy. En este caso el método además del filtro recibe una variable (ver enumerado WorkflowOrder) con el orden en el que se quiere obtener el resultado.

Example: Creating a process 'outside' the Inbox



Veamos un ejemplo. El objetivo es poder crear e iniciar la ejecución de un proceso de workflow desde un objeto GeneXus.

Para ello debemos seguir la siguiente secuencia de pasos:

1. Establecer una conexión con el motor de workflow con el método Connect
2. Obtener, a partir del nombre, la definición de proceso que queremos instanciar utilizando el método GetProcessDefinitionByName
3. Crear una nueva instancia de esta definición utilizado el método CreateInstance
4. Iniciar la ejecución de la nueva instancia creada (y que es retornada en el método anterior sino hay error) utilizando el método Start

Organizational Model

- Organizational Model with users, roles, etc.

- Most used Methods:
  - ListUsers
  - ListRoles
  - ListOrganizationalUnitDefinitions
  - ListOrganizationalUnits
  - GetUserByName
  - AddUser
  - AddRole
  - AddOrganizationalUnitDefinition
  - AddOrganizationalUnit

El tipo de datos OrganizationalModel permite acceder a la información de la estructura organizacional de la empresa, esto es, las unidades organizacionales, los roles funcionales, los usuarios y sus relaciones.

Similarmente al tipo de datos Server, existen métodos con prefijo Get que sirven recuperar un tipo por su identificador o nombre y métodos con prefijo List que nos permiten obtener una colección de Usuarios, Roles, UnidadesOrganizacionales, etc

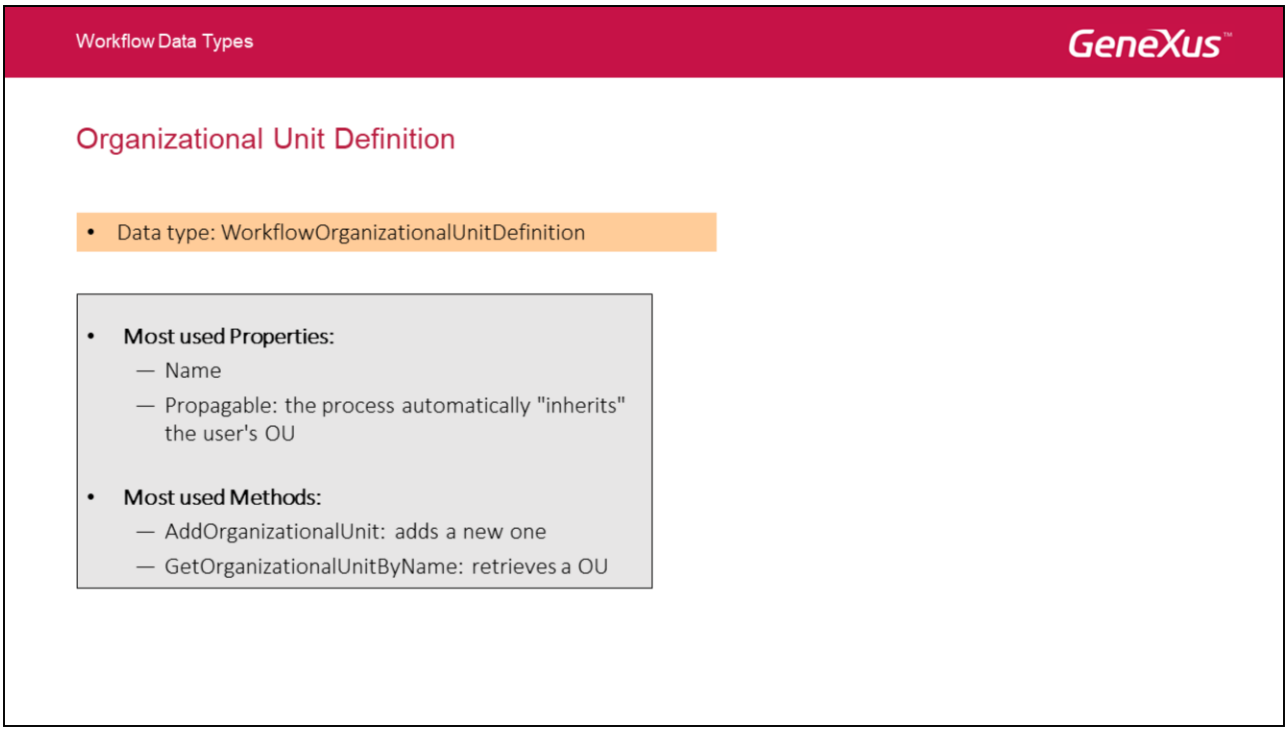
También existen métodos con prefijos Add y Remove que permiten agregar o eliminar entidades.

GeneXus

Organizational Unit

- Data type: WorkflowOrganizationalUnit

- **Most used Properties:**
  - Name
  - OrganizationalUnitDefinition: definition
- **Most used Methods:**
  - AddUser/RemoveUser: user management



Siguiendo con el ejemplo planteado de las Sucursales, realicemos la creación de las Unidades Organizacionales utilizando los tipos de datos de Workflow:

1 – Hay que conectarse al server con un usuario y password que tenga permisos, en este caso usaremos el administrador

&WorkflowServer.Connect('WFADMINISTRATOR','WFADMINISTRATOR')

2 – Se obtiene el modelo organizacional, para eso se utiliza el método **GetOrganizationalModel**.

&WorkflowOrganizationalModel = &WorkflowServer.GetOrganizationalModel()

3 - Se agrega como Definición de Unidad Organizacional *Office* para eso se utiliza el método **AddOrganizationalUnitDefinition**.

&WorkflowOrganizationalModel.AddOrganizationalUnitDefinition('Office', 'Offices', True)

4 - Se agregan las unidades organizacionales *Red, Green y Yellow*. Para eso se utiliza el método **AddOrganizationalUnit**.

&WorkflowOrganizationalModel.AddOrganizationalUnit('Office','Red', 'Office Red')

&WorkflowOrganizationalModel.AddOrganizationalUnit('Office','Green', 'Office Green')

&WorkflowOrganizationalModel.AddOrganizationalUnit('Office','Yellow', 'Office Yellow')

Nota: en todos estos casos se utilizaron variables que están basadas en el tipo de datos que poseen el mismo nombre que ellas.

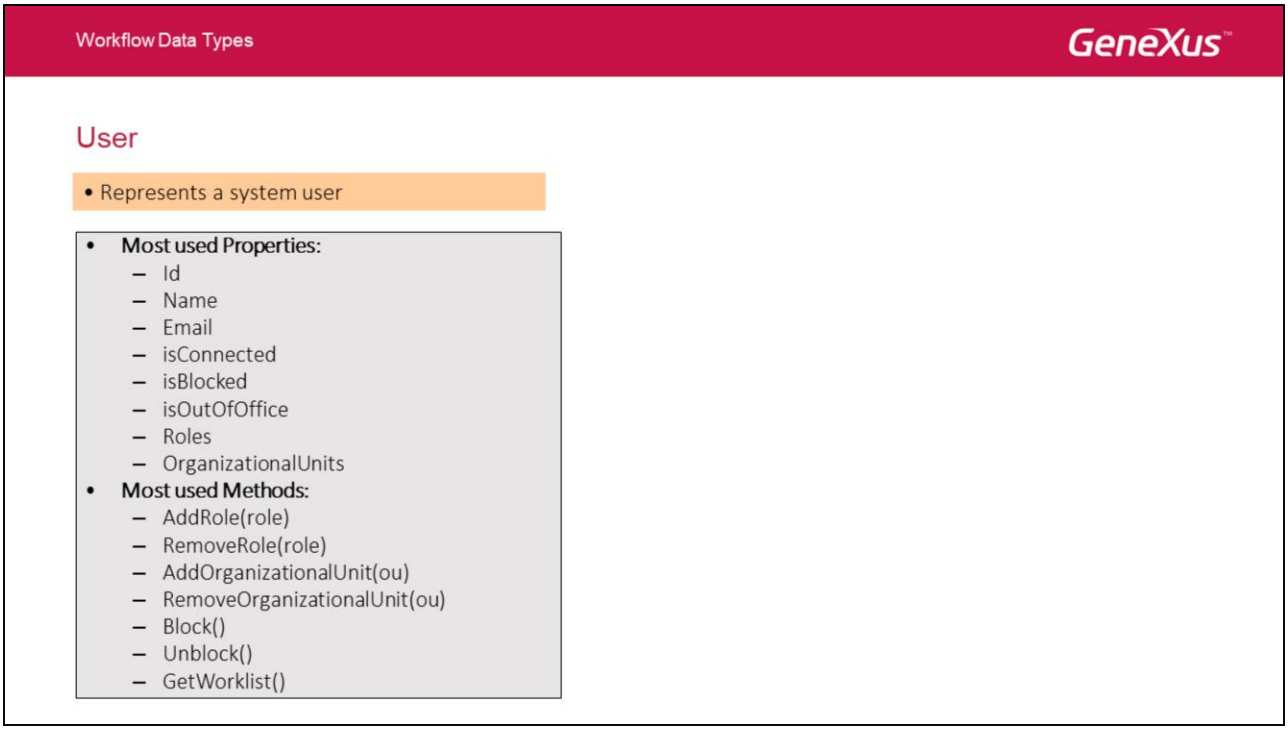
Role

- Represents a functional role

- **Most used Properties:**
  - Id
  - Name
  - Parent
  - Children
  - Users
- **Most used Methods:**
  - AddUser(user)
  - RemoveUser(user)

El tipo de datos Role representa un rol funcional. Los roles funcionales son los que asociamos a las actividades para definir permisos sobre las mismas.

Mediante el tipo de datos podemos acceder a las propiedades de un rol (identificador, nombre, quién es el rol padre, los roles hijos, los usuarios asociados, etc) y también disponemos de métodos para agregar y quitar usuarios.



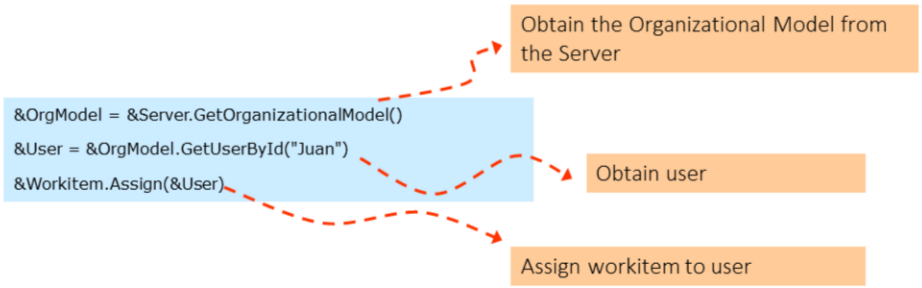
El tipo de datos User nos permite acceder a la información de los usuarios del sistema de workflow.

Disponemos de propiedades para saber: el identificador, el nombre, la dirección de correo electrónico, si esta conectado, si esta bloqueado, si esta de licencia, etc.

Además disponemos de métodos para asignar o desasignar roles o unidades organizacionales, bloquearlo, etc.

También tenemos un método GetWorklist que nos permite obtener una colección con todos los workitems activos y asignados o en el que el usuario es candidato (es decir, todos los workitems que el usuario podría ver en su bandeja de entrada).

Example: Assigning a workitem to a user

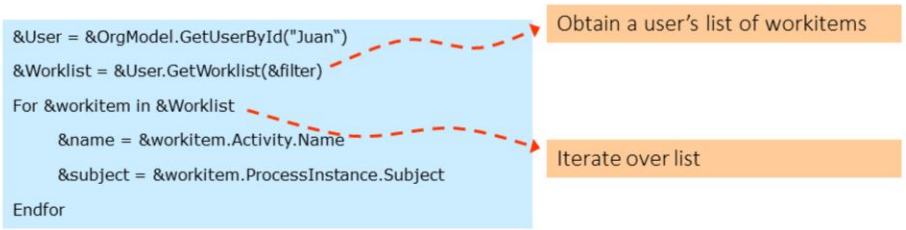


Veamos un ejemplo. Queremos asignar un workitem a un usuario en particular.

Para ello debemos seguir los siguientes pasos:

1. En primer lugar obtenemos el modelo organizacional a partir del tipo Server utilizando el método GetOrganizationalModel
2. Utilizando el método GetUserById recuperamos el usuario cuyo identificador coincide con el deseado (Juan)
3. Utilizamos el método Assign sobre el workitem que queremos asignar (y que recuperamos previamente con algunos de los mecanismos ya vistos) pasando como parámetro el usuario del paso anterior.

Example: Go over a user's pending task list



Siguiente ejemplo. Queremos recorrer la lista de tareas pendientes del usuario (esto puede servir para hacer algún listado, crear una bandeja de entrada customizada, etc).

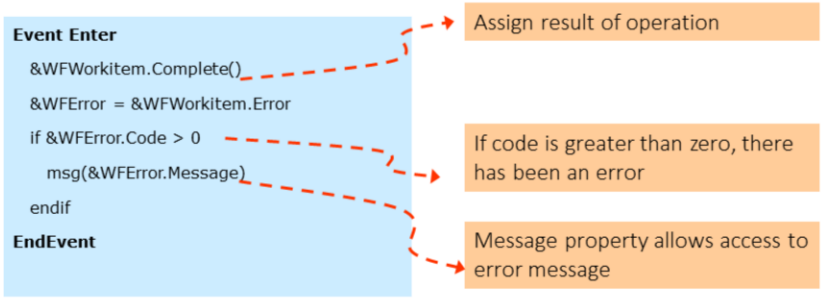
En primer lugar obtenemos el usuario de la misma forma que en el caso anterior.

Posteriormente utilizamos el método GetWorklist() que como habíamos dicho anteriormente devuelve una colección con todos los workitems activos y asignados al usuario (o en el que el usuario es candidato).

Una vez que obtenemos la colección procedemos de la misma forma que con cualquier colección de GeneXus. Por ejemplo, podemos iterar sobre la misma utilizando el constructor For &var in &collection.

Dentro de la iteración podemos obtener por ejemplo el nombre de la actividad asociada al workitem, el asunto de la instancia de proceso a la que pertenece, etc.

Error Management



Un tipo de datos que hemos evitado hasta el momento por un tema de simplificación pero que es muy importante, es el tipo de datos Error.

Este tipo de datos es utilizado para comunicar los diferentes errores que pudieran ocurrir cuando se esta interactuando con el motor de workflow.

Algunos tipos de datos tienen una propiedad error que permite que cuando se realiza una operación (es decir, se invoca un método del tipo), la propiedad error se inicializa con el resultado de la operación.

El tipo de datos Error tiene dos propiedades:

Code: que devuelve el código de error

Message: que devuelve el mensaje de error

Veamos un ejemplo de cómo se utiliza este tipo de datos para recuperar los diferentes errores que pueden ocurrir.

Supongamos que deseamos completar un workitem. Como vimos anteriormente, para ello debemos invocar al método Complete sobre el workitem correspondiente.

Inmediatamente despues que invocamos el método, utilizamos la propiedad Error sobre el tipo Workitem para recuperar el error.

Luego preguntamos por el valor de la propiedad Code sobre el tipo Error, la cual nos va a indicar si hubo error o no. Si la propiedad devuelve cero, entonces la operación se realizó con éxito. Si en cambio el código devuelve un valor mayor que cero, entonces hubo un error y consultando la propiedad Message podemos obtener el mensaje de error correspondiente.

Module 6

Events of Timer and Calendar Type

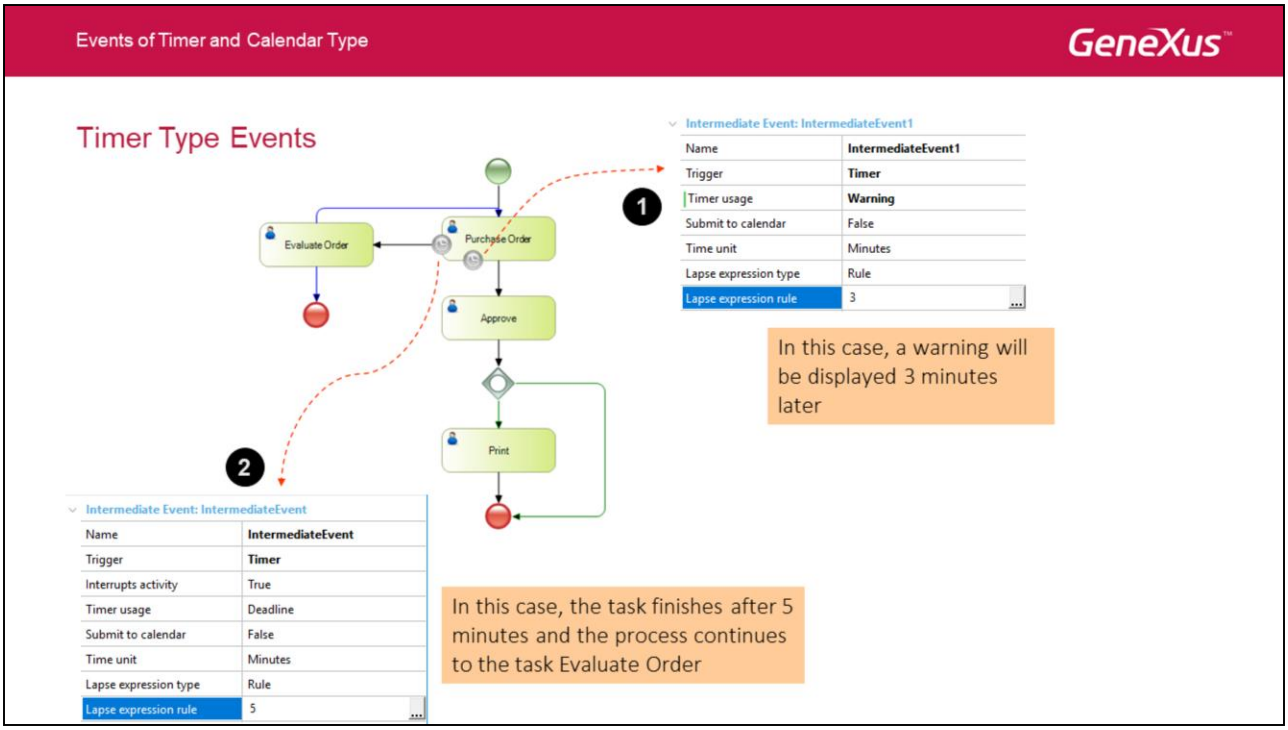
GeneXus

Timer Type Events

- Timeframe to execute a task or process.
- Options:
  - Warning – upcoming deadline notification.
  - Deadline – due date.

En algunos casos queremos indicar que una tarea o un proceso tiene un tiempo límite de ejecución, para indicar eso se utilizan los eventos intermedios de tipo Timer.

Dentro de un evento intermedio es de tipo Deadline o de tipo Warning. En caso de que sea de tipo Deadline se indica cual es la tarea que se debe seguir en ese caso.



Veamos un ejemplo de cómo definir un Warning y un Deadline.

Por ejemplo para la tarea Ingresar solicitud de crédito tiene que ser finalizada antes de los 5 minutos, pero a los 3 minutos le tiene que indicar un warning.

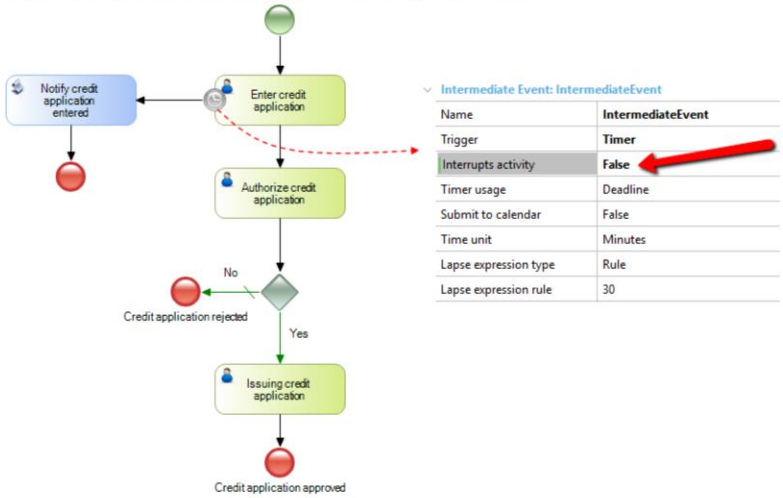
El warning se indica en la bandeja de entrada en la columna que tiene el calendario.

Cuando sea cumpla el deadline la tarea Ingresar solicitud de crédito finaliza y se ejecuta la tarea Evaluar solicitud.

GeneXus

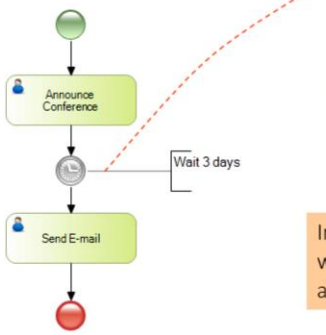
Time Monitoring

- It is also possible for the attached timer event to not interrupt the task.



GeneXus

Delays



Intermediate Event: IntermediateEvent

Name	**IntermediateEvent**
Trigger	**Timer**
Submit to calendar	False
Time unit	**Days**
Lapse expression type	Rule
Lapse expression rule	3

In this case, the task ends and we wait 3 days before sending an email

¿Cómo se define un delay en un diagrama?

Un delay se modela insertando un evento de tipo timer entre actividades.

Pueden representar una fecha y una hora especificas (ejemplo: esperar hasta el 15 de abril a las 5 p.m.), o una fecha relativa (esperar 6 días) o una fecha relativa repetitiva (ejemplo: esperar hasta el próximo lunes a las 8 a.m.).

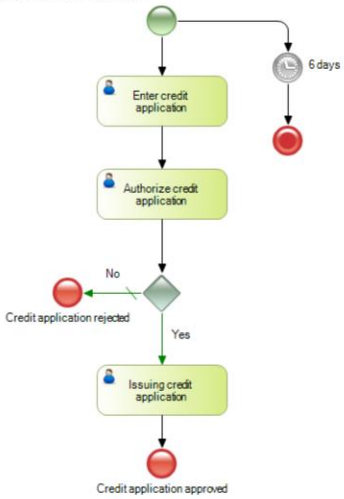
Cuando el timer genera la alerta el proceso continua su flujo.

El elemento previo al delay puede ser una actividad, un gateway u otro evento intermedio.

GeneXus

Process Deadline

- The total duration of a process must be demarcated.



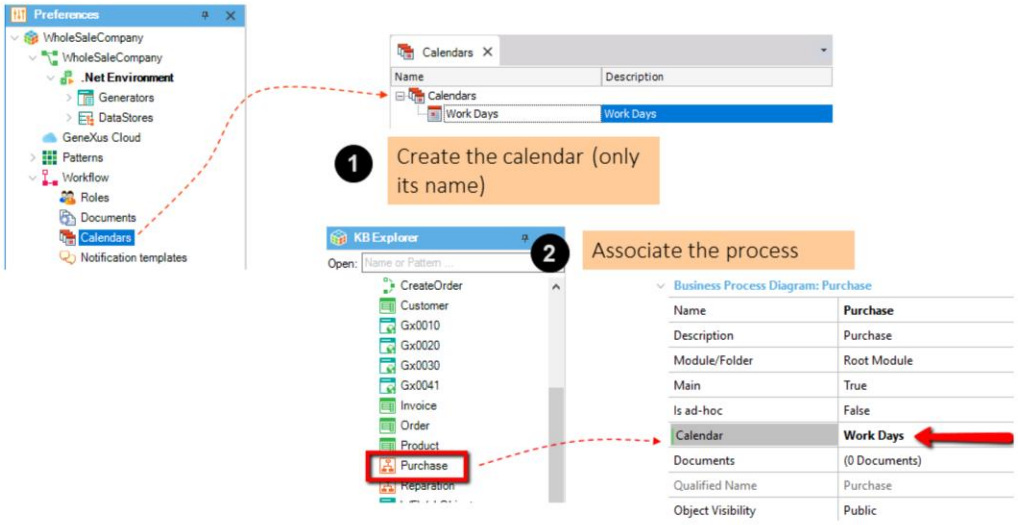
GeneXus

Calendar

- It contains the working days and times.
- Timer type events can be adjusted to the calendar.
- Available in tasks and processes.
- Options:
  - No calendar: the actual time is used.
  - Calendar: it is adjusted to the selected calendar

GeneXus

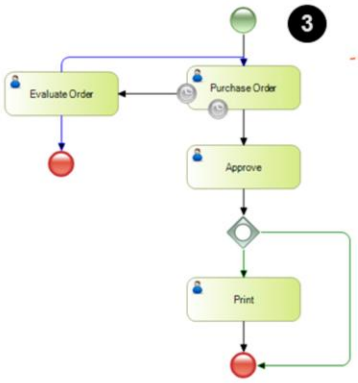
Calendar Configuration (I)



GeneXus

Calendar Configuration (II)

In activities we can configure the process calendar or use another one



Intermediate Event: IntermediateEvent

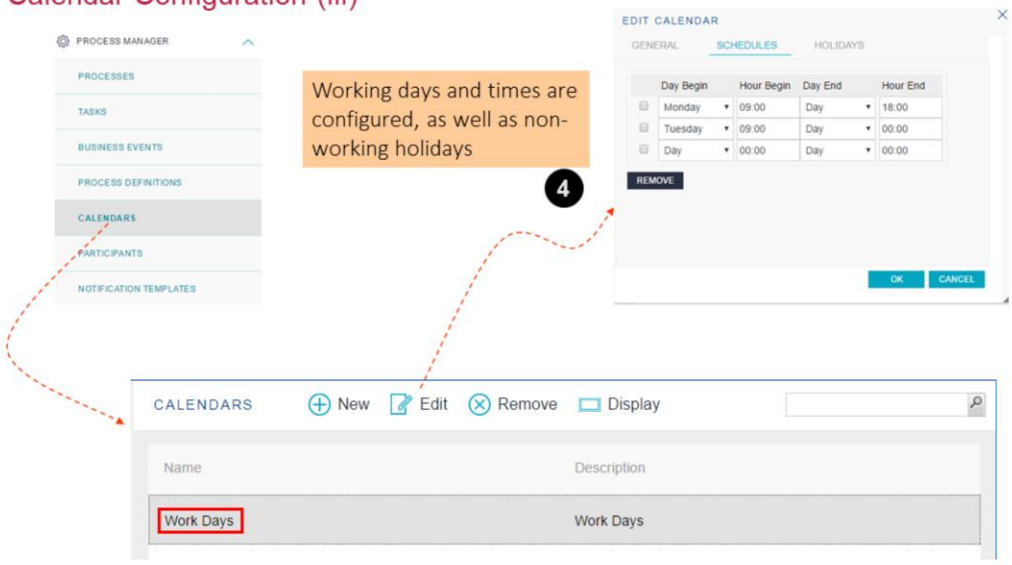
Name	**IntermediateEvent**
Trigger	**Timer**
Interrupts activity	True
Timer usage	Deadline
Submit to calendar	**True**
Calendar	Work Days
Time unit	Minutes
Lapse expression type	Rule
Lapse expression rule	5

Time unit used to calculate the deadline: days, hours, minutes o seconds

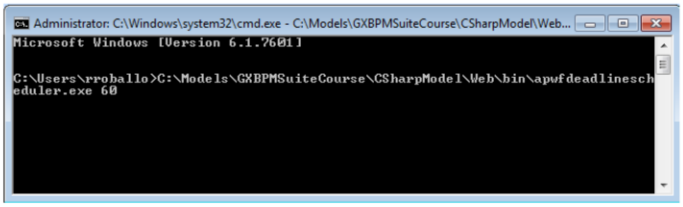
Adjust deadlines to the calendar (True) or use actual time (False)

GeneXus

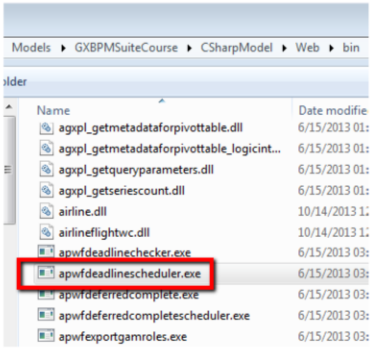
Calendar Configuration (III)



Calendar Configuration (IV)



- Utilities to check deadlines.
- Executed via command line.
  - **WFDeadlineScheduler**:
    - Executed on the background
    - The parameter is the elapsed time, in seconds, between each execution.
    - <WebApp>\web\bin\apwfdeadlinescheduler.exe 60
  - **WFDeadlineChecker**:
    - It is executed once and finished.



GeneXus

Calendar related Workflow Data Types

- WorkflowProcessInstance and WorkflowWorkitem
  - **Most used Properties:**
    - WarningTime – Warning date and time.
    - DeadlineTime – Deadline date and time.

  Both properties are read-write

- WorkflowCalendar
  - **Most used Properties:**
    - Name
  - **Most used Methods:**
    - IsWorkday – indicates if a date corresponds to a working day.
    - FirstWorkDay – returns the date of the first working day of the month.
    - LastWorkdayInMonth – returns the last working day of the month.

Los tipos de datos WorkflowProcessInstance y el WorkflowWorkitem manejan el warningtime y el deadlinetime través de propiedades. Por más información:

http://wiki.genexus.com/commwiki/servlet/wiki?17771 y
http://wiki.genexus.com/commwiki/servlet/wiki?11731

Por mas información sobre el tipo de datos Workflow Calendar
http://wiki.genexus.com/commwiki/servlet/wiki?11609

Module 7

Documents

GeneXus

Documents

- Ability of the workflow engine to manage files containing user data.
- The engine provides the mechanism to store (blob), control and secure access to the files.

Documents

Types of Documents:

- Microsoft Word, Excel, Power Point
- PDF
- Open Office: Text, Spreadsheet, Presentation
- StarOffice: StarWriter, StarCalc, StarImpress
- Images: JPG, BMP, GIF
- Text
- XML
- ZIP
- Other

Es posible definir de qué tipo serán los documentos que formarán parte del proceso, de esta forma se puede restringir al usuario a subir archivos de ese tipo. En caso de no especificar el tipo quedará como Other permitiendo subir archivos de cualquier tipo.

Templates

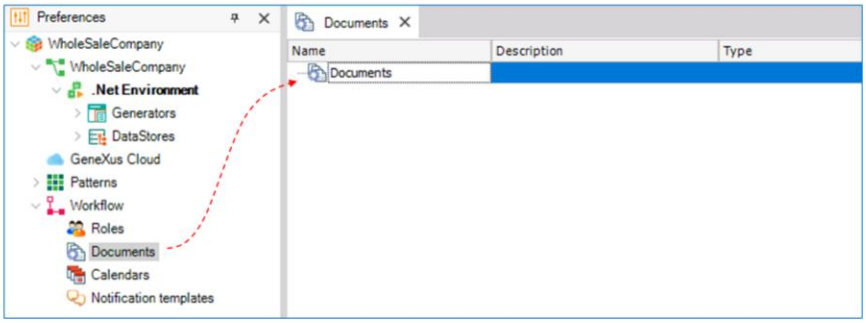
Document template:

- Documents can be based on templates that define their basic structure.
- Template Property:
  - Defined by the user (associated file).
  - Empty.

Los documentos pueden estar basados en plantillas, las plantillas definen la estructura básica del mismo. Para ello se utiliza la propiedad Template, la cual indica el nombre de la plantilla que utiliza ese documento.

Se deben crear las plantillas como los demás archivos asociados a la KB, para ello se debe ir a la opción Folder View - Files, los documentos creados en esta opción quedarán como disponibles para ser asociados como plantillas de los documentos manejados por el Gxflow.

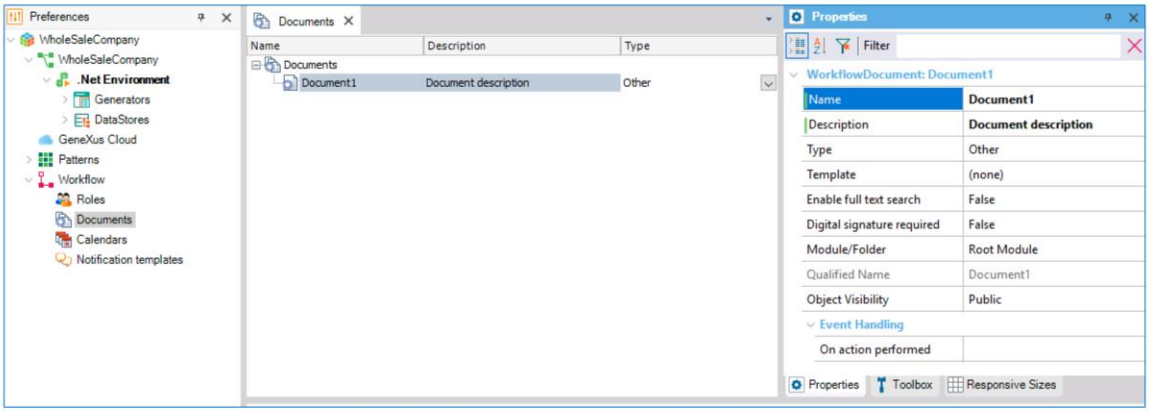
Definition in GeneXus



Para utilizar documentos en diagramas de procesos, primero hay que **definirlos a nivel de la base de conocimiento** (KB). Para esto hay que ir a **Preferences – Workflow –Documents** y presionar la tecla **Enter** para agregar nuevos documentos.

Al momento de definir el documento hay que indicar el nombre, una descripción y opcionalmente el tipo de documento.

Definition in GeneXus



Además se pueden configurar las siguientes propiedades:

Name: Nombre del documento

Description: Descripción del documento

Type: Tipo del documento

Template: Permite asociar una plantilla.

Enable Full Text Search: Permite habilitar la búsquedas sobre el contenido del documento.

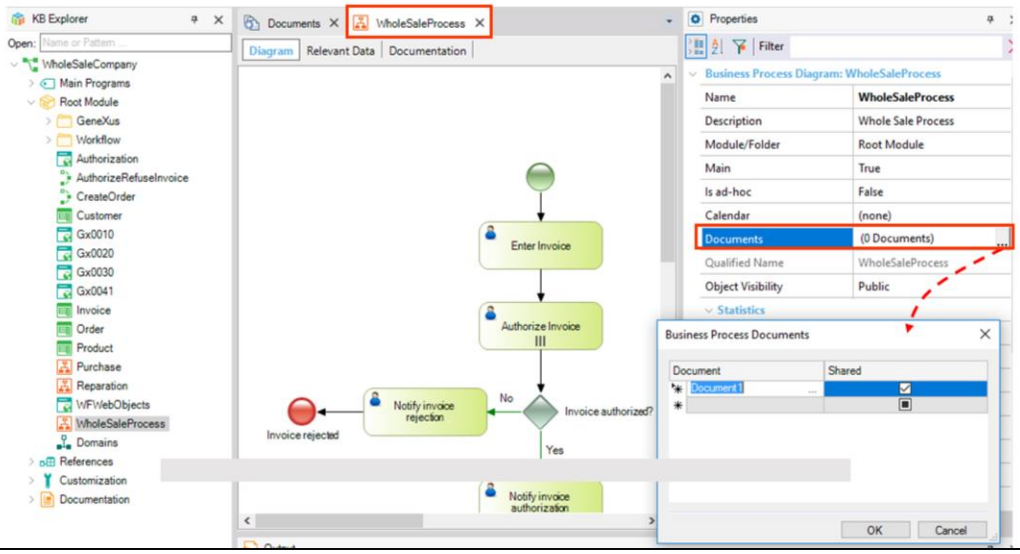
Digital Signature Require: Permite indicar que el documento debe ser firmado digitalmente.

Module/Folder: Donde será almacenado el documento

Object Visibility: De acceso público, privado o interno de la empresa.

Event Handling - On action performed: Permite asociar un manejador de eventos (un procedimiento) para que se ejecute cuando se realiza una acción sobre el documento.

Documents and Diagrams



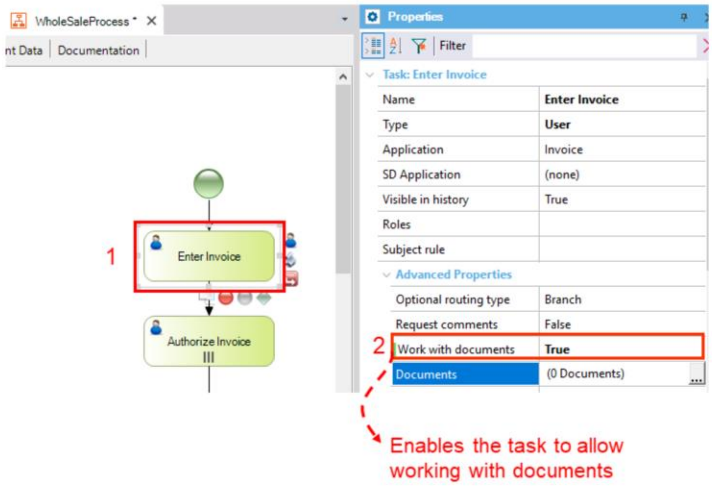
Los documentos se pueden asociar a nivel del diagrama.

Al asociarse a ese nivel, los documentos van a estar disponibles para todas las tareas del proceso.

Se puede indicar luego el rol que puede trabajar con ese documento, por lo que los usuarios que tengan ese rol podrán trabajar sobre los mismos.

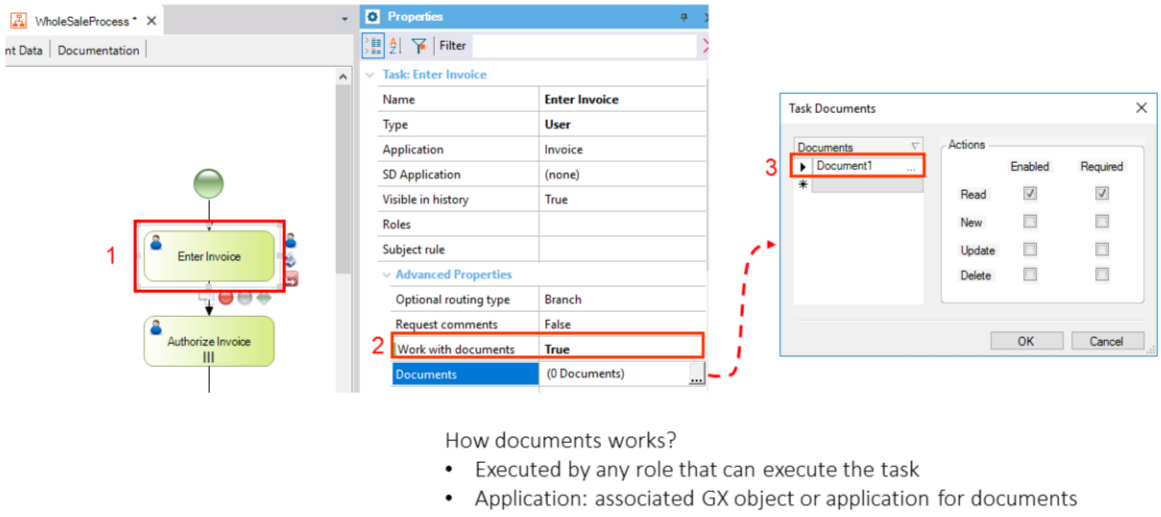
GeneXus

Documents and Tasks



Los documentos se pueden asociar a cada tarea, para ello hay que setear la propiedad Work with Documents en True, una vez hecho esto, se habilita la propiedad Documents donde se indican cuales son los documentos asociados a la tarea.

Documents and Tasks



How documents works?

- Executed by any role that can execute the task
- Application: associated GX object or application for documents

Al setear la propiedad Work with Documents en True, se habilita la propiedad Documents donde se seleccionan los diferentes documentos asociados a la tarea.

Además de indicar cuales son los documentos hay que indicar cuales son las acciones que se pueden realizar sobre los mismos.

Las acciones que se pueden realizar sobre los documentos son:

- Read -> Se permite la lectura del documento
- New -> Se permite la creación de un nuevo documento
- Update -> Se permite la modificación del documento
- Delete -> Se permite la eliminación del documento

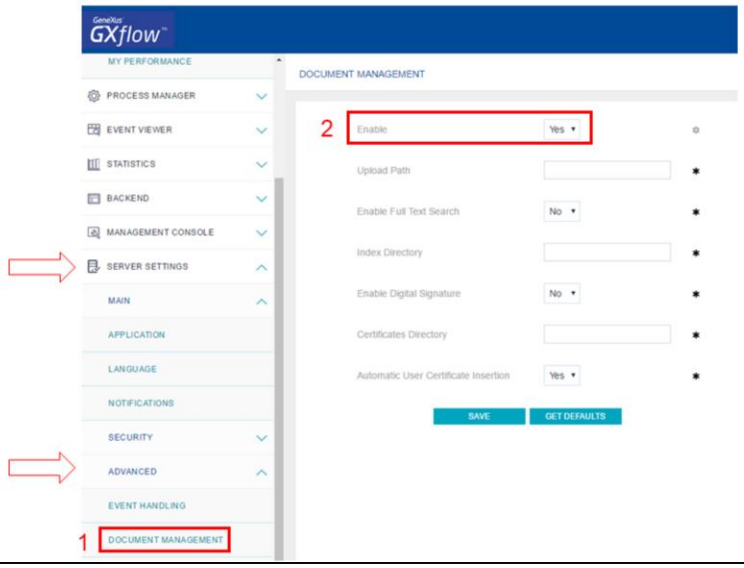
Además de indicar que acción se puede realizar sobre el documento hay que indicar si las mismas son Obligatorias (Required) o Disponibles (Enabled)

¿ Cómo funcionan el manejo de los documentos en GXflow?

El documento asociado a una tarea puede ser manejado por cualquiera de los roles que puedan ejecutar la tarea.

Si la tarea no tiene un programa GeneXus asociado, entonces se ejecuta una aplicación para trabajar con documentos.

Desktop Documents



¿Cuáles son las configuraciones que hay que hacer para poder trabajar con documentos?

Configuración en la KB:

Propiedades del generador -> Client server information -> Blob local storage directory: directorio donde se almacenan temporalmente los documentos blob durante la extracción o inserción en la base de datos.

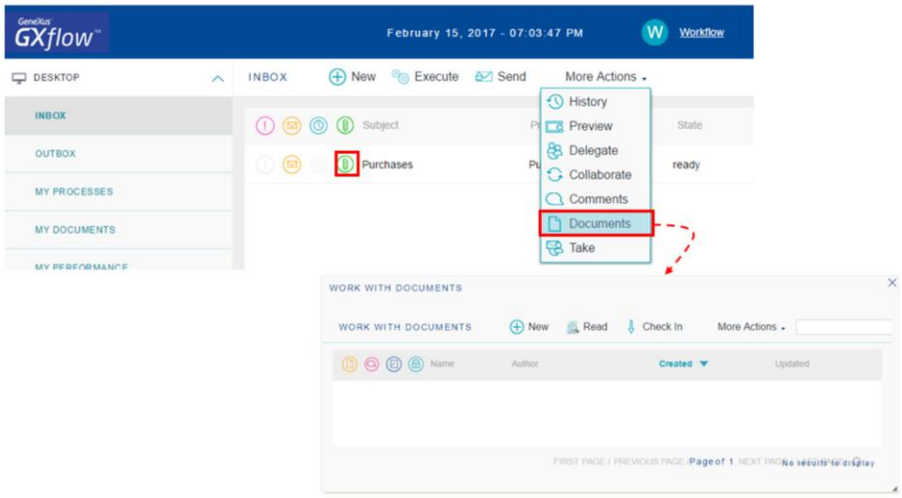
Configuración en el cliente GXflow:

En **Settings -> Server –> Advanced -> Document Management**.:

Enable: Habilita o no el manejo de documentos.

Upload path: directorio para subir los documentos en el servidor web

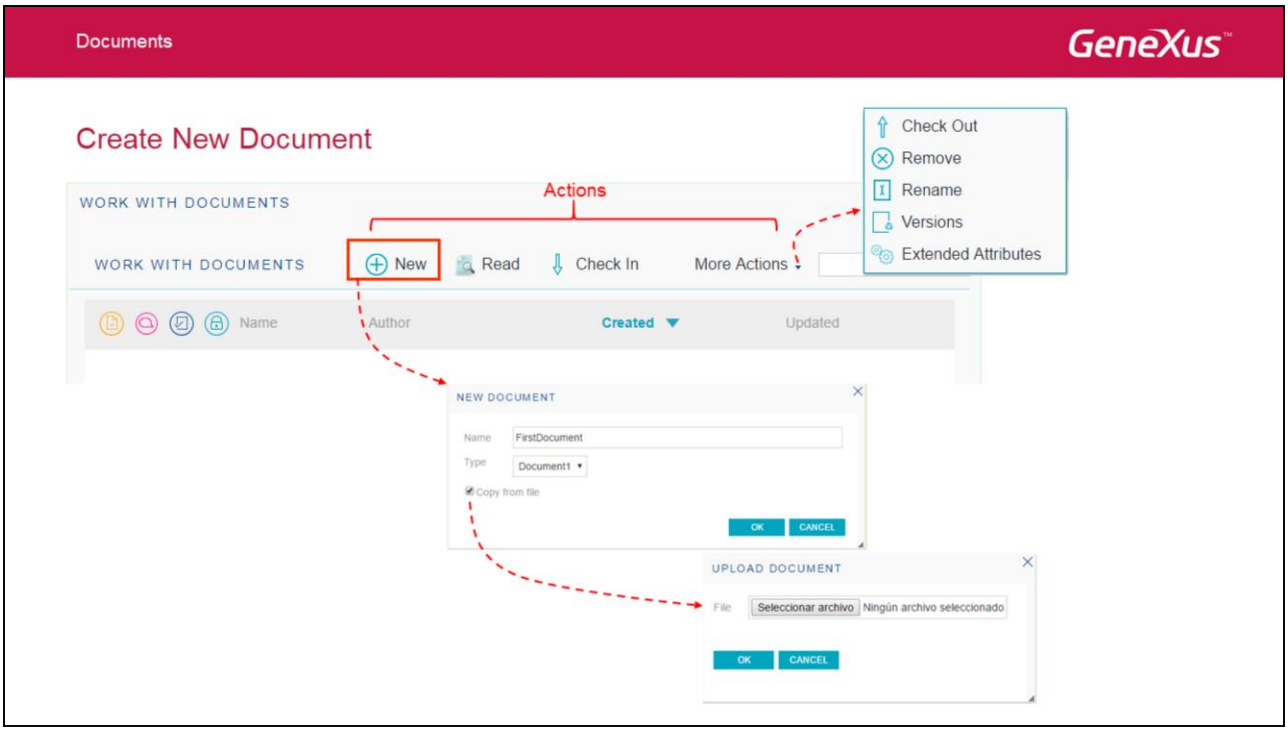
Desktop Documents



Para indicar que una tarea tiene documentos asociados se indica con la imagen del clip.

En el caso de que la tarea este compuesta de una aplicación y de documentos para acceder a los mismos, hay que ir a la opción **More Actions - Documents** donde se abre el Work with Documents.

En caso de que la tarea solo tenga asociado documentos, se abre automáticamente el Work with Documents.



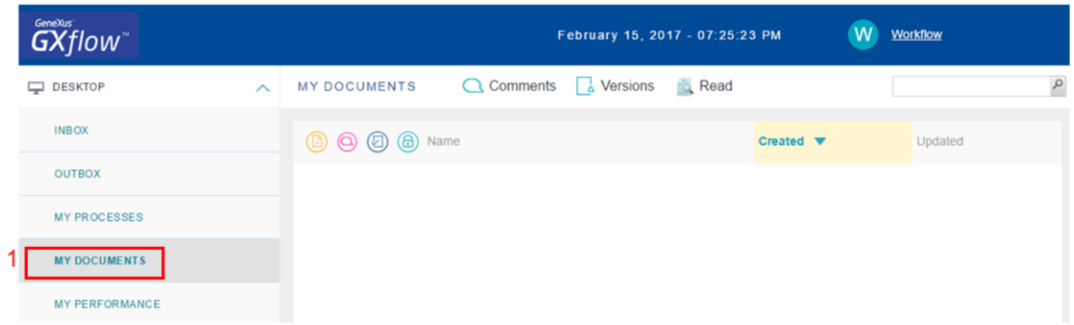
Dependiendo de los permisos que tenga el usuario sobre los documentos son las acciones que va a tener disponibles.

Las opciones que se ven en pantalla son New, Read, Check in y More Actions.

En el caso de New, se abre una nueva ventana en la cual hay que ingresar el nombre del documento y el tipo.

Al momento de crear el documento se lo puede crear copiando otro documento existente en el PC.

My Documents



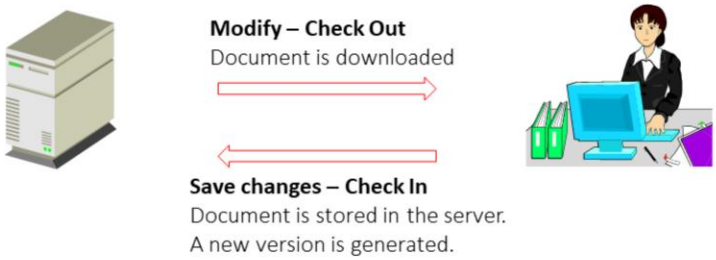
Bajo la opción **Desktop - My Documents** se pueden ver todo los documentos con los cuales el usuario ha trabajado.

Seleccionando un documento en especial se puede agregar comentarios al mismo, leer lo cual muestra la última versión del mismo o ir a versiones donde se pueden ver cada una de las versiones del documento.

GeneXus

Desktop Documents

- Document repository: The engine maintains a document repository.
- How it works:



El motor de GXflow tiene un repositorio de documentos, el cual funciona de la siguiente manera:

- Para modificar un documento hay que realizar un **Check Out**, eso hace que se descargue el documento.

- El usuario trabaja con el documento y cuando ya finalizo los cambios tiene que hacer un **Check In**, con los cual se sube el documento al servidor

- Al realizar un **Check In** del documento se crea una nueva versión del mismo.

GeneXus

Desktop Documents

Document repository:

- Only the first user who performs the Check Out operation can perform the Check In.
- Other users can read the document (Check Out) but they cannot perform the Check In operation.
- The administrator can cancel the Check Out.
- A new version is created with every Check In.
- The changes made between Check Out and Check In are NOT recorded by the Workflow engine.

Cuando un mismo documento es compartido por varios usuarios solo el primer usuario que hace Check Out puede hacer Check In.

Durante ese proceso el documento está bloqueado, pero los usuarios podrán leer el documento (Read) o podrán bajar el documento (Check Out)

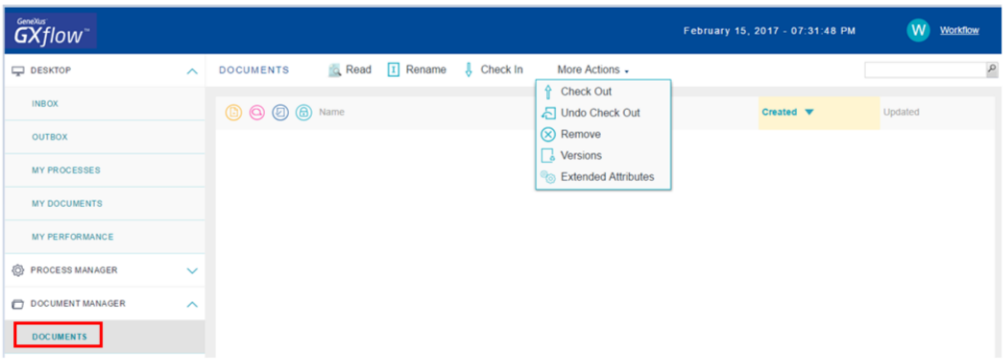
Solo el usuario administrador tiene los permisos suficientes para cancelar el Check Out de un documento.

Cada vez que se hace un Check In se crea una nueva versión del documento.

Los cambios realizados entre el Check Out y el Check In NO son registrados por el motor de Workflow.

GeneXus

Document Administrator

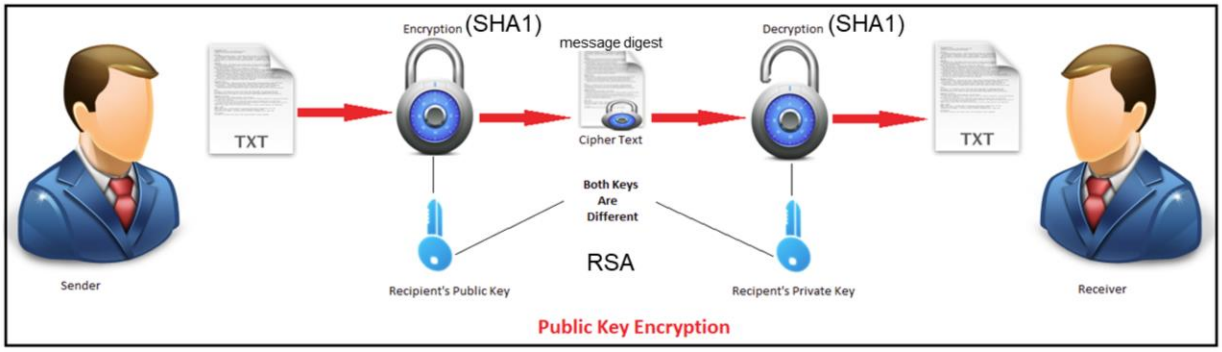


El administrador para poder realizar las tareas de administración de los documentos tiene que ir a la opción: **Document Manager - Documents**.

Ahí se vera la lista de documentos y el administrador podrá realizar varias operaciones como ser: deshacer un check out, renombrar un archivo o eliminarlo.

GeneXus

Document Digital Signature

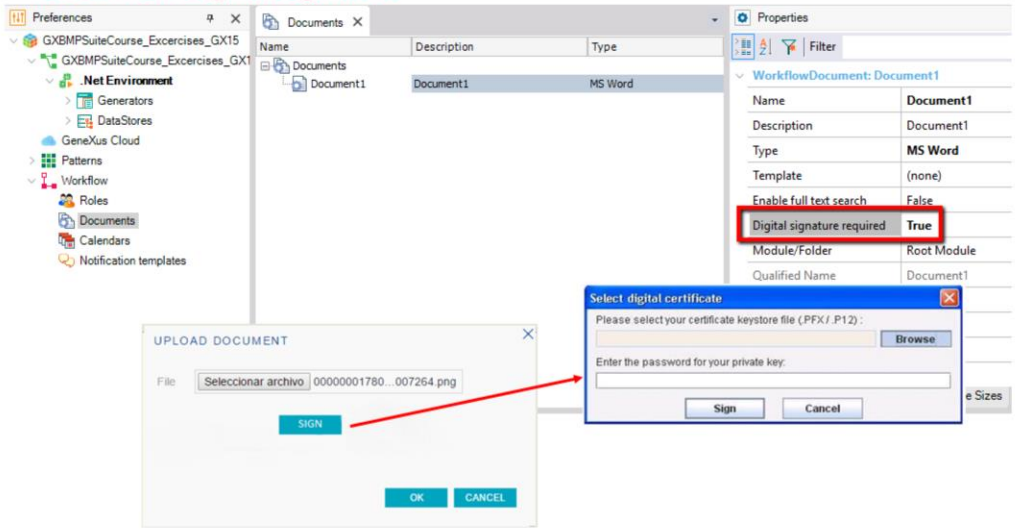


Los documentos pueden ser firmados antes de subirlos al servidor.

- La firma permite:
  - Verificar la integridad del documento (que no fue alterado).
  - Asegurar el no repudio del documento (comprobar que un usuario es quien efectivamente realizó la firma).
- Tecnología utilizada:
  - Algoritmo RSA para las firmas.
  - SHA1 (Secure Hash Algorithm 1) – Para calcular la firma.

GeneXus

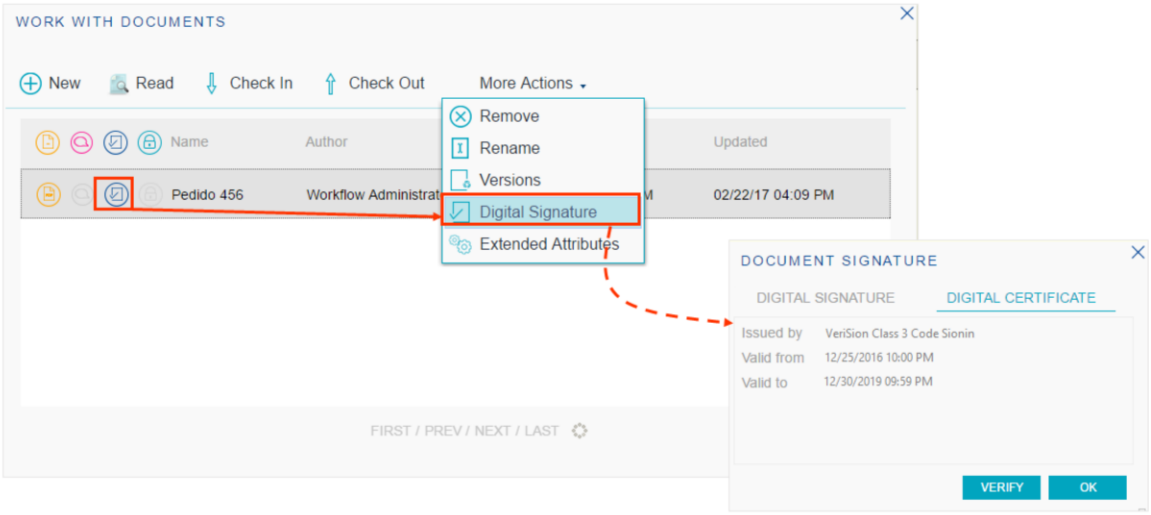
Document Digital Signature



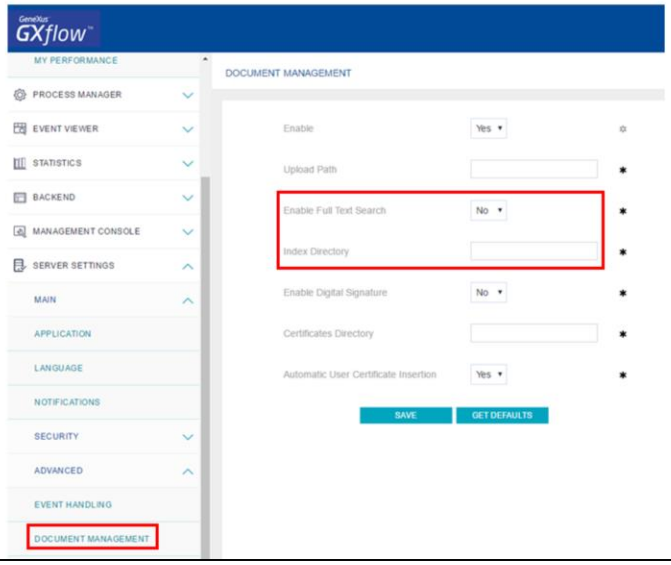
Antes de guardarse el documento en la Base de Datos se verifica:

- Que la firma sea válida (el documento no fue alterado).
- El certificado del usuario sea válido (no expiró y no revocado)
- El certificado es de confianza (emitido por alguna autoridad certificadora de confianza).

Document Digital Signature



Document Searches



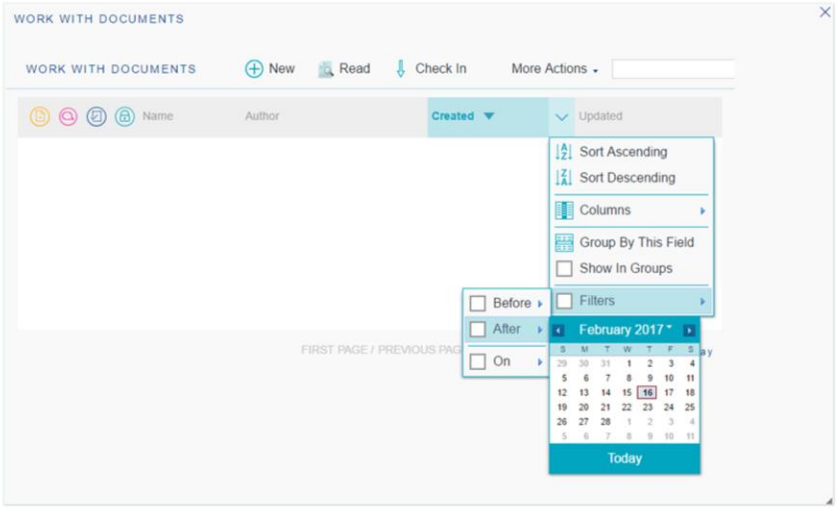
Para configurar las búsquedas sobre el contenido de los documentos hay que ir a: **Settings – Server – Advanced - Document Management**

Ahí hay que configurar las propiedades:

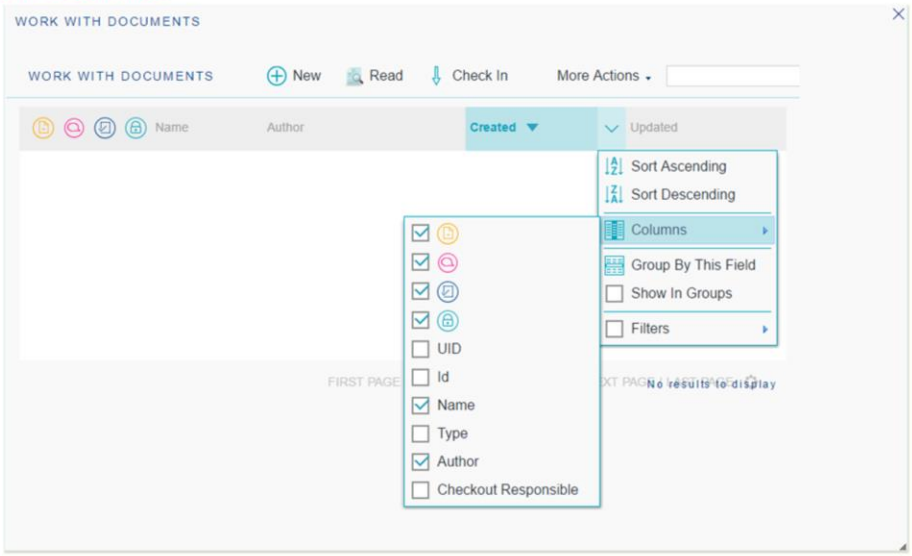
- **Full Text Search**: Permite habilitar el mecanismo de búsqueda.
- **Index Directory**: Permite indicar el directorio donde se almacenará el índice de palabras claves.

GeneXus

Document Searches



Document Searches



Filtros en las aplicaciones de Documentos:

- Tipo de documento.
- Autor – Usuario que creó el documento.
- Contenido – palabras que se encuentran dentro del documento.
- Filtros de Fecha.
- Responsable del check out.
- Firmado digitalmente.
- Comentarios.
- Si el documento está en uso o no.

GeneXus

Workflow Data Types

- WorkflowDocumentRepository: it represents the document repository

- **Most used Methods:**
  - ListDocumentDefinitions (filter): returns document definitions.
  - ListDocumentDefinitions (filter): returns document instances.

GeneXus

Workflow Data Types

- WorkflowDocumentDefinition: represents a document definition

- **Most used Properties:**
  - Id
  - Name
- **Most used Methods:**
  - CreateInstance(workitem, name) – Creates a new document instance associated with the work item
  - CreateInstanceFromFile(workitem, name, file) – Uses a file to create the document instance

GeneXus

Workflow Data Types

- WorkflowDocumentInstance: represents a document instance

- **Most used Properties:**
  - Name
  - Version – Document version
  - CheckedOutBy – User who downloaded the document
- **Most used Methods:**
  - Read (user, workitem, targetDirectory) – Opens the document to read it; it is temporarily downloaded to the directory.
  - CheckIn (user, workitem, file) – Adds the document to the repository and enables it again to be downloaded.
  - CheckOut (workitem, targetDirectory) – Downloads and blocks the repository document, and allows making changes to it.
  - UndoCheckOut (user) – Undo the blocking

GeneXus 15

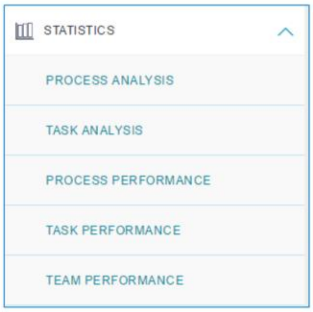
Module 8

Statistics

En este módulo podremos analizar las diferentes estadísticas incorporadas al Cliente de GXflow.

Statistics

- Real-time monitoring of system statistics within the GXflow Client.



El usuario administrador tiene la posibilidad de monitorear en tiempo real diferentes estadísticas generadas en el sistema.

Las estadísticas son:

- Análisis de Procesos.
- Análisis de tareas.
- Performance de Procesos.
- Performance de Tareas.
- Performance de Equipos.

La sección Análisis de procesos incluye las siguientes gráficas:

- Actividad del proceso:

Se subdivide en 2 gráficos:

1. Actividad de procesos: Permite ver el estado actual del proceso, la cantidad de instancias creadas, completadas y abortadas en un periodo seleccionado.

2. Tendencia de Instancia Activa: Permite ver la tendencia de la instancia activa para un proceso y período determinado.

- Ciclo del proceso

1. Estado del Proceso: Muestra el porcentaje de las intancias de procesos cerradas que han terminado en tiempo y aquellas que han demorado más de lo previsto.

2. Duración del proceso: Permite analizar la duración de los procesos mostrándo la duración promedio vs. la duración estimada de las instancias completas para el proceso y el período seleccionados.

- Ranking de procesos

Compara el número de instancias de proceso creadas para un proceso y periodo determinado.

GeneXus

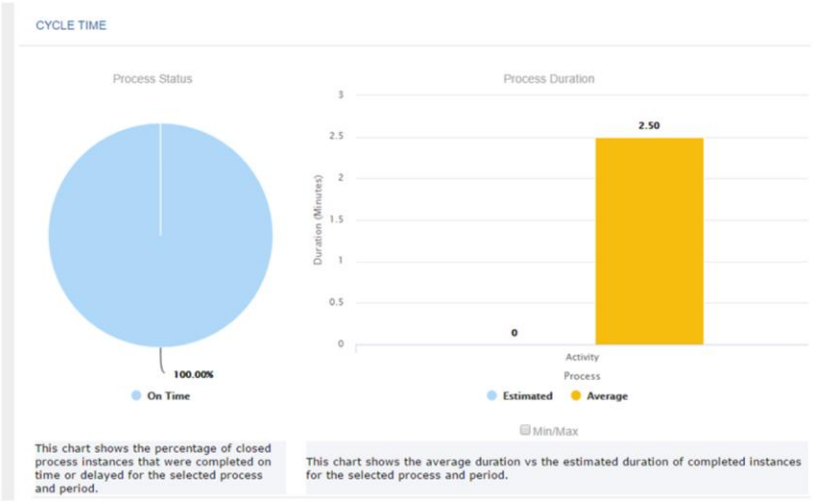
Process Analysis

- Process Activity

GeneXus

Process Analysis

- Cycle time



GeneXus

Process Analysis

- Process Ranking

This chart compares the number of created instances for each process in the selected period.

Al igual que en el caso de los procesos, se tiene un indicador para ver el tiempo de ciclo pero esta vez a nivel de tareas, los que nos permite un nivel de granularidad mas fino para hacer un análisis.

- Ciclo de la tarea:

Se puede analizar en dos gráficos,

1. Estado de la tarea: Muestra el porcentaje de tareas cerradas, completadas en tiempo y demoradas según el proceso, la tarea y el período seleccionado.
2. Duración de la tarea: Muestra la duración promedio frente a la duración estimada de las instancias de tareas completadas según un periodo seleccionado.

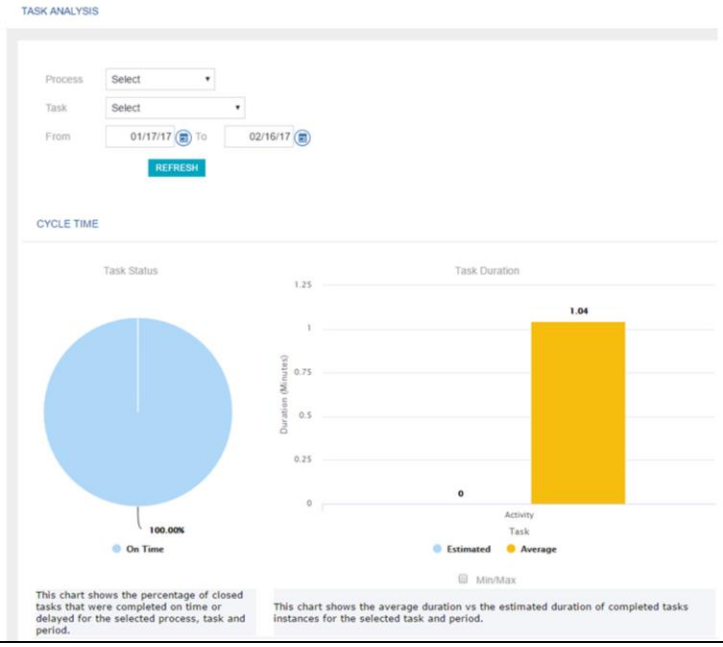
- Ranking de tareas:

Compara la cantidad de instancias creadas para cada tarea para el proceso y periodo seleccionado.

GeneXus

Task Analysis

- Cycle time



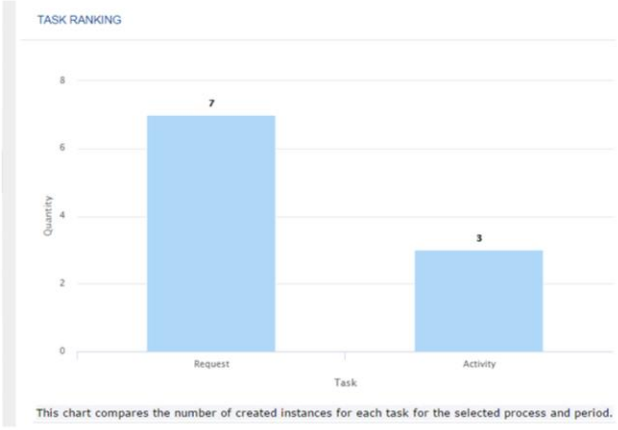
Se puede realizar el análisis de tareas en base a los siguientes filtros:

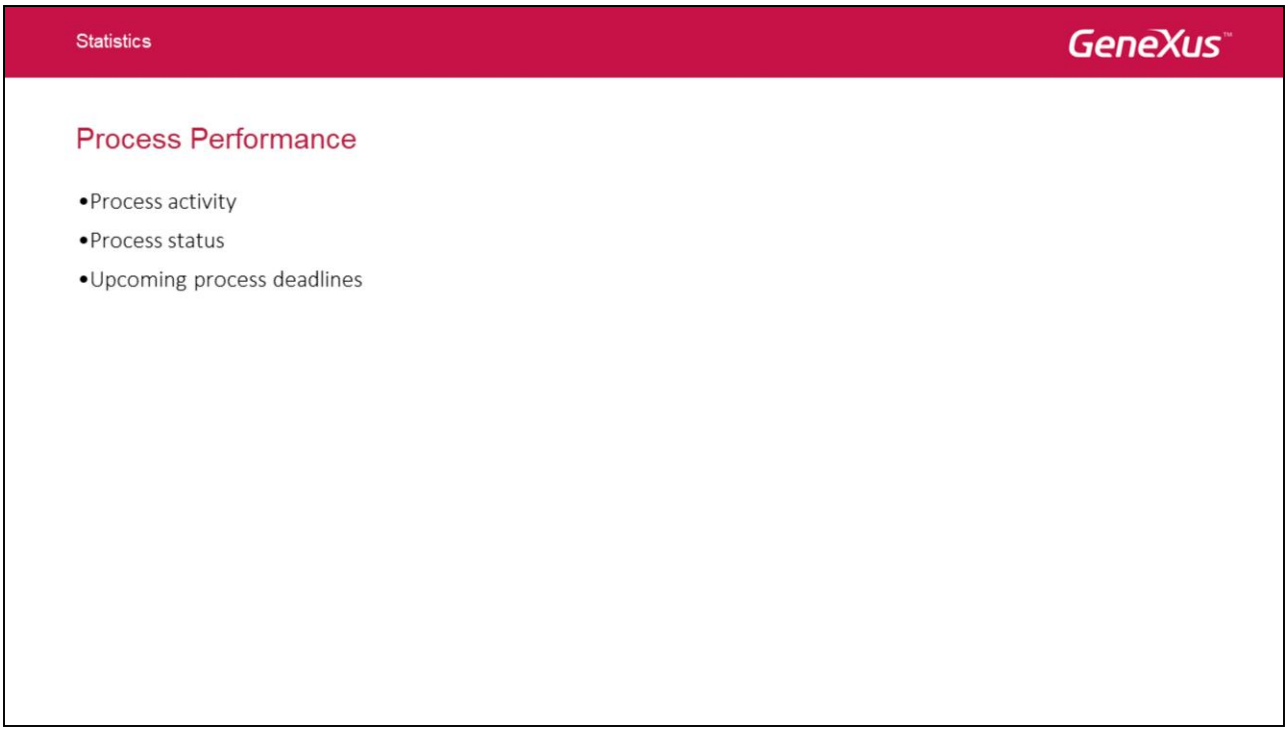
- Proceso
- Tarea
- Rango de fechas

GeneXus

Task Analysis

- Task ranking





Esta sección nos permite ver el estado actual de los diferentes procesos en ejecución, donde podemos identificar si están en tiempo y forma o si están demorados.

Las gráficas son:

- Actividad del proceso:

Este gráfico compara la tendencia de la instancia activa en los últimos días para el proceso seleccionado.

- Estado del proceso:

Cuenta a su vez con dos gráficos.

1. Estado del Proceso General: Muestra el porcentaje de instancias de procesos activos que están a tiempo, en situación de riesgo o retrasadas.
2. Detalle del Estado del Proceso: Muestra la cantidad de instancias de procesos activos que están a tiempo, en situación de riesgo o retrasadas para cada proceso.

- Próximos vencimientos del proceso:

Esta gráfica muestra cuando las instancias de procesos están próximas a vencer.

GeneXus

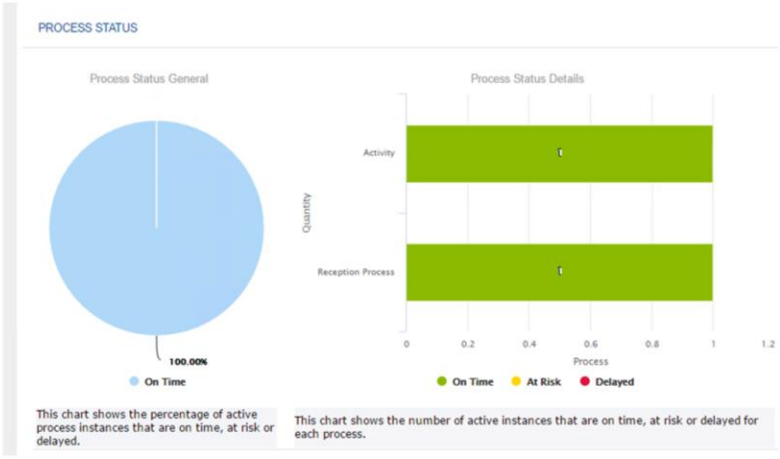
Process Performance

- Process Activity

GeneXus

Process Performance

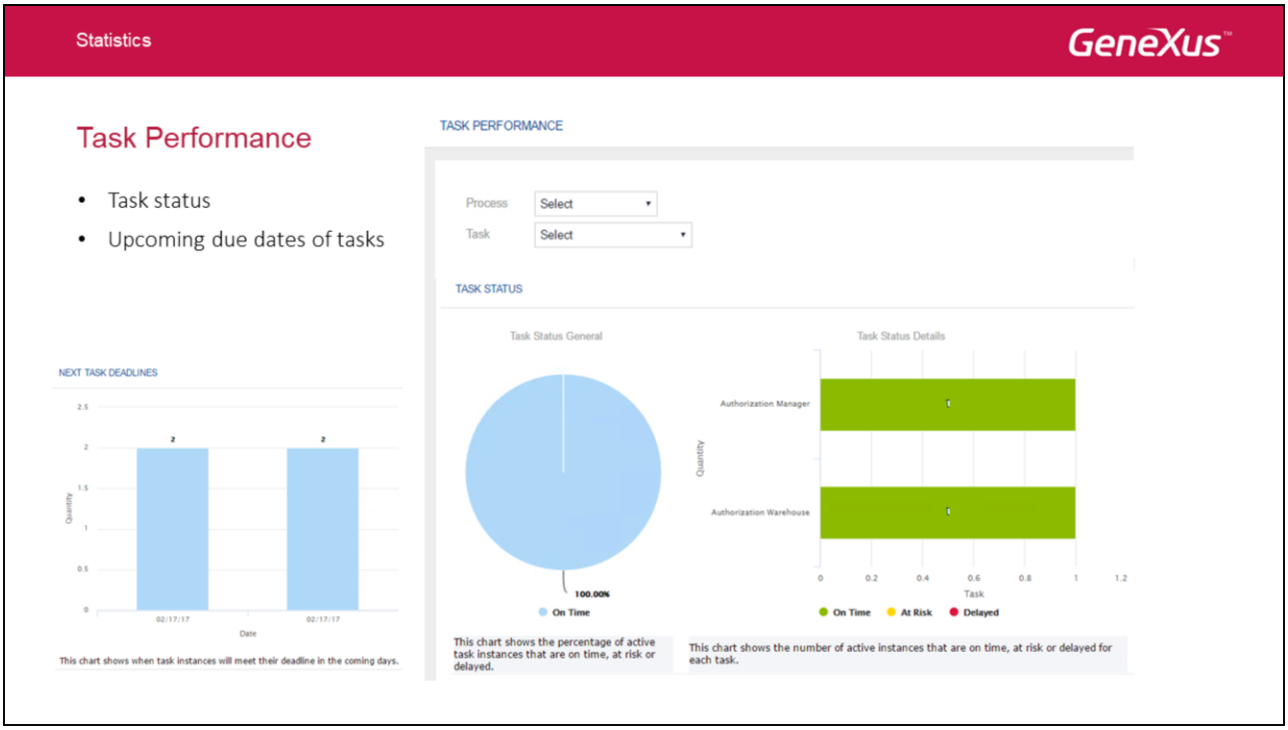
- Process status



GeneXus

Process Performance

- Next process deadlines



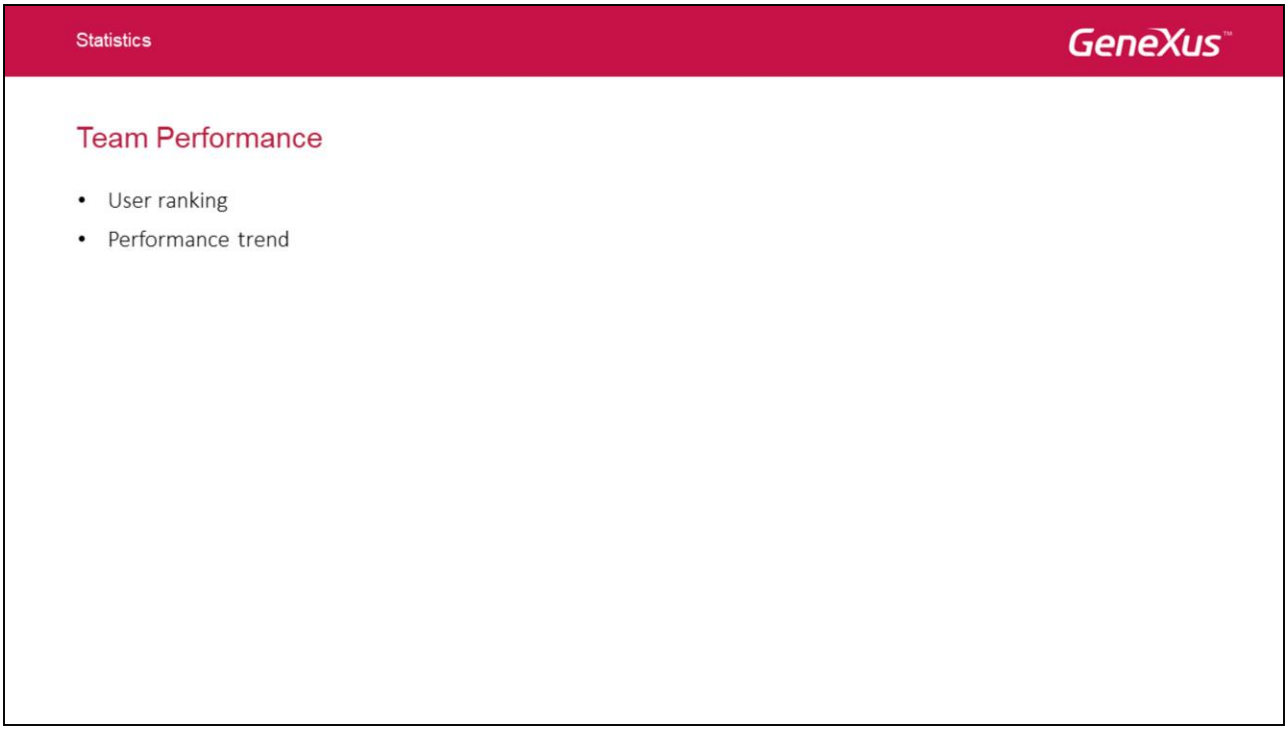
Así como la sección de performance de procesos se puede saber el estado en que se encuentran las tareas.

- Estado de la tarea:

1. Estado de la tarea general: Muestra el porcentaje de tareas activas que están a tiempo, en situación de riesgo o retrasadas.
2. Detalle del estado de la tarea: Cantidad de tareas activas que están a tiempo, en situación de riesgo o retrasadas para cada tarea

- Próximos vencimientos de la tarea:

Esta gráfica muestra cuando las tareas alcanzarán el vencimiento en los próximos días



Permite analizar la productividad de los diferentes usuarios de la organización, teniendo la posibilidad de comparar en base a un rol, un proceso o una tarea determinada.

- Ranking de usuarios:

Compara la cantidad de tareas completas a lo largo de los últimos días por cada usuario y proceso seleccionado. Sólo se muestran los usuarios que tienen al menos una tarea completa en el período seleccionado.

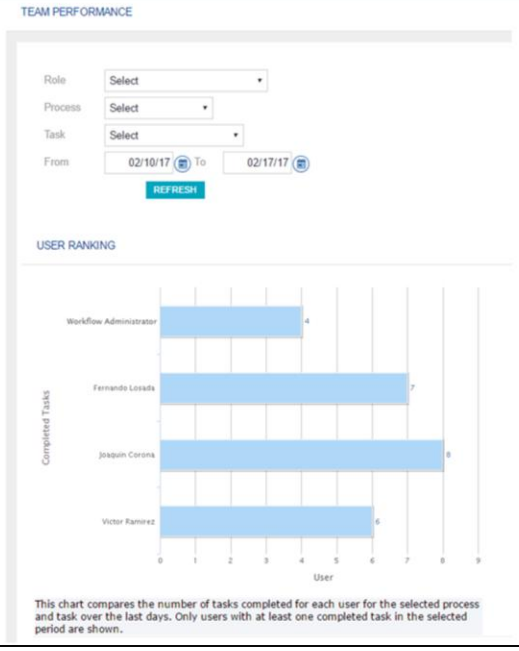
- Tendencia del rendimiento:

Esta gráfica compara la tendencia de rendimiento a lo largo de los últimos días entre los miembros de un rol seleccionado, para el proceso y tarea seleccionados.

GeneXus

Team Performance

- User ranking



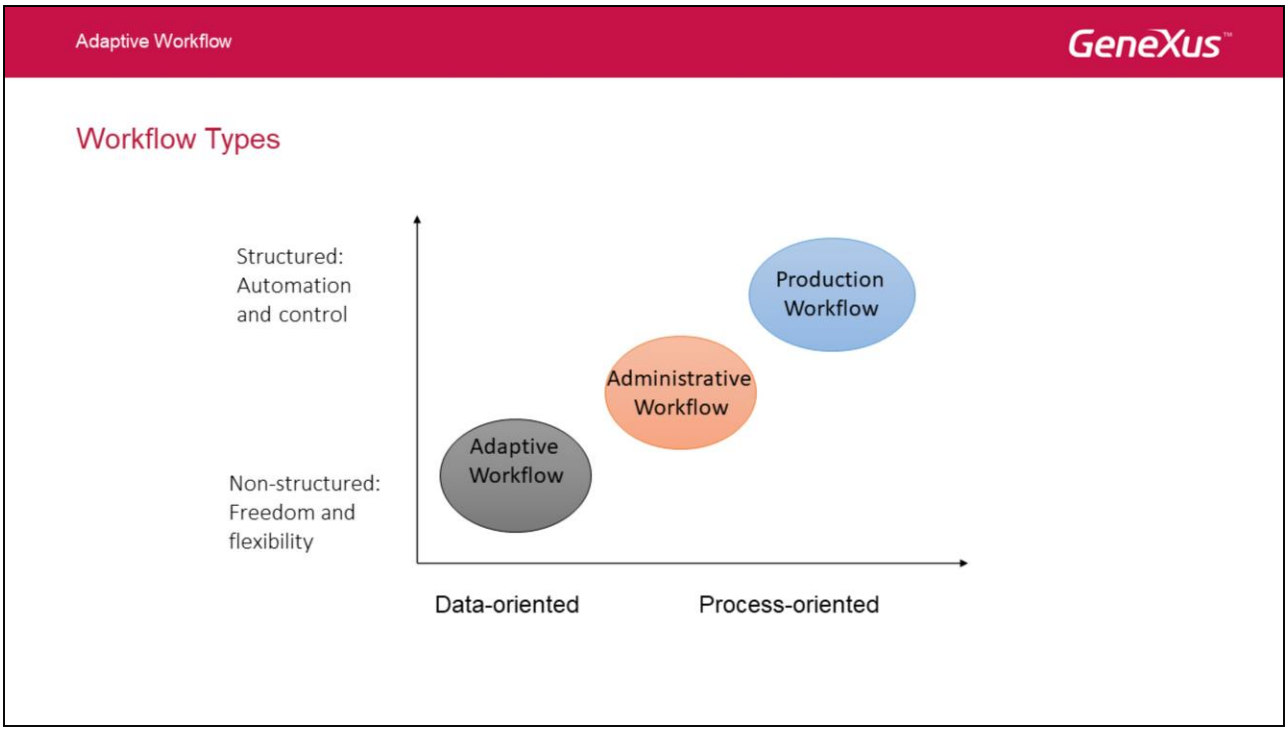
GeneXus

Team Performance

- Performance trend

Module 9

Adaptive Workflow



Tipos de Workflow

¿Qué tipo de Workflow deberíamos usar? Esto depende de lo que queramos hacer. En compañías muy grandes usan diferentes productos de Workflow de distintos proveedores. Para ayudar a entender mejor el mercado se ha propuesto una segmentación de los distintos tipos de Workflow.

Workflow de producción

El objetivo de los Workflows de Producción es administrar grandes números de tareas similares y optimizar la productividad. Es se consigue automatizando tantas actividades como se pueda hasta el punto donde la interacción humana es requerida solamente para manejar excepciones.

Los eventos que requieren la contribución de las personas son minimizados. Este tipo de Workflows son optimizados para atender altos niveles de calidad y certeza ejecutando tareas altamente repetitivas. Puede manejar procesos muy complejos, y puede estar estrechamente integrado a sistemas existentes. La tendencia es embeber el componente de Workflow en grandes aplicaciones donde su rol es actuar como una motor de reglas.

Workflow Administrativo

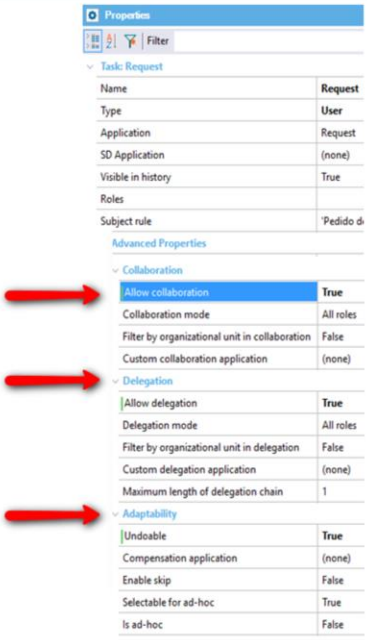
La funcionalidad más importante de un sistema de Workflow administrativo es la facilidad para definir el proceso. Habitualmente hay muchas instancias ejecutándose donde se involucra a un gran número de empleados. La definición de procesos generalmente se hace usando formularios y si la definición es demasiado compleja para el formulario entonces simplemente

se tiene que usar otro producto. Flexibilidad es más importante que la productividad y estos sistemas manejan uno o dos ordenes menos de magnitud en el número de instancias por hora comparándolos con los sistemas de Workflows de Producción.

GeneXus

Adaptability with GXflow

- Collaboration
- Delegation
- Undo a task
- Skip a task
- Ad-Hoc Processes



Como vimos en módulos anteriores GXflow permite modelar procesos de negocio bien estructurados y orientados a procesos mediante el uso de tareas batch y además permite un híbrido entre procesos estructurados y no estructurados y orientados a procesos o datos que se realiza mediante tareas interactivas y batch.

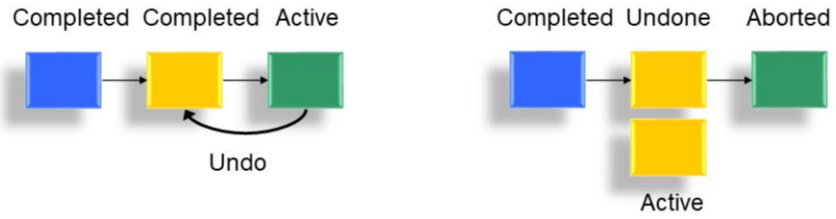
A continuación se describen algunas de las funcionalidades que nos permiten modelar de forma adaptativa y flexible los procesos de negocio con GXflow.

Undo a Task

- Undoability: ability to undo a task
- Compensation application: procedure executed upon aborting the process instance or undoing the task.

Adaptability

Undoable	True
Compensation application	(none)
Enable skip	False
Selectable for ad-hoc	True
Is ad-hoc	False



Este patrón de Workflow nos permite en tiempo de ejecución volver atrás algo que ya se hizo y de esta manera volver a comenzar el proceso de nuevo desde el punto donde se vuelve atrás.

Para las tareas del tipo “User Task” o “Script Task” se puede configurar la propiedad Undoable con el valor True. Esto permite que desde el “Manejador de procesos” un usuario con el rol de Manager o Administrator desde la opción Tareas pueda deshacer la ejecución de aquellas tareas que se completaron y cuyas sucesoras no se completaron aún. Para esto es necesario activar esta propiedad para aquellas tareas que queremos que se permita hacer la acción de Deshacer.

Cuando se configura esta propiedad con el valor True se habilita la propiedad “Compensation application” en la cual se puede asociar un procedimiento GeneXus el cual se ejecutará en el momento que se haga la acción de Deshacer la tarea. Eventualmente este procedimiento podría llegar a recomponer los datos que fueron modificados cuando se ejecutó dicha tarea.

Dicho procedimiento debe tener los siguientes parámetros:

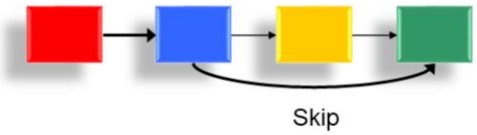
parm(&WorkflowProcessDefinition, &WorkflowProcessInstance, &WorkflowWorkItem);

Cada uno de esos parámetros están definidos del mismo tipo cuyo nombre es igual al nombre de las variables.

Skip a Task

Enable Skip

- It allows **skipping** the activity without executing it and continuing to the next one.

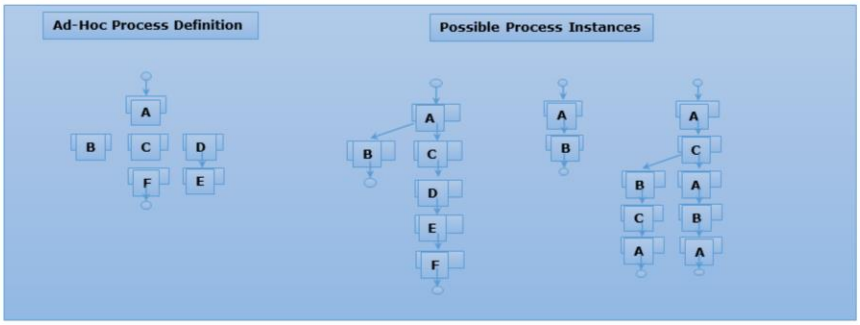


La propiedad Enable Skip está habilitada solamente para las tareas del tipo User, es decir, las tareas interactivas.

El usuario con rol Manager o Administrator podría llegar a evitar la ejecución de la tarea. Para esto desde la opción “Manejador de procesos → Tareas” se tiene que seleccionar la tarea que se quiere saltear y presionar la opción “Saltear”. Esto deja la tarea en estado “salteado” y continua el flujo como se lo haya definido a continuación de la tarea salteada.

Ad-Hoc Processes

- Non-structured process whose Process Definition can be **adapted at runtime.**
- The user selects the next step

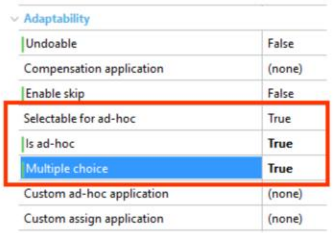


Los procesos Ad-Hoc son aquellos en los cuales hay determinados flujos que no pueden determinarse en la etapa de modelado del mismo, sino que el flujo puede variar dependiendo del contexto en cada ejecución y es el usuario el que decide qué camino tomar y a quién selecciona para realizar la tarea.

Por lo tanto tendremos una definición de las posibles tareas a realizar en el proceso y donde en algunos casos podremos determinar hasta determinados flujos pero en otros casos no. En el momento que se van generando instancias la propia ejecución de dicho proceso va definiendo nuevos flujos dependiendo de las decisiones que se toman en cada paso. Para el caso más extremo podríamos llegar a tener tantas definiciones del proceso como instancias se hayan ejecutado.

Ad-Hoc Task

- Is Ad-Hoc: Indicates that the following task will be selected from a list of possible successor tasks.
- Multiple Choice: More than one successor task can be selected.
- Selectable for Ad-Hoc: The task will be available in the list of successor tasks of Ad-Hoc tasks.

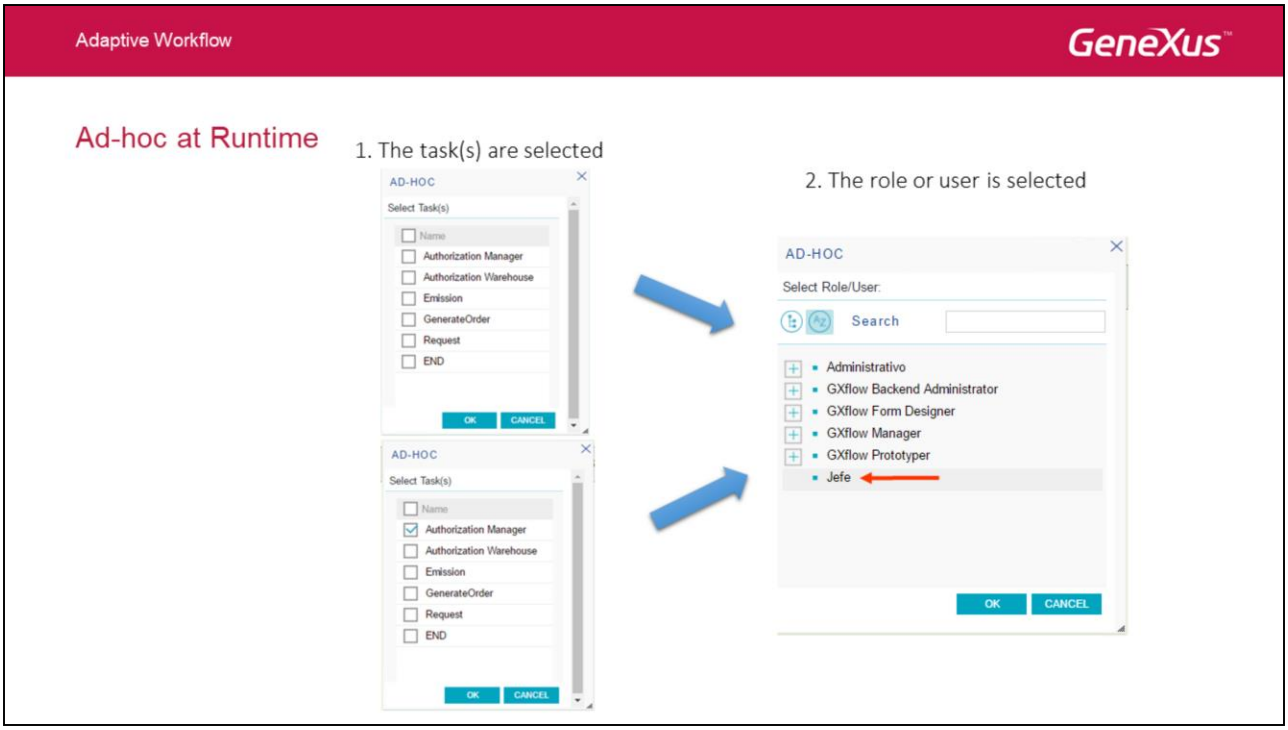


La propiedad “Is Ad-Hoc” esta disponible sólo para las tareas interactivas. Cuando se configura esta propiedad con el valor True estamos indicando que una vez que se de por finalizada dicha tarea el usuario deberá seleccionar la o las tareas sucesoras y además asignarlas al rol o usuario que desee. Esto permite salirse de la estructura del flujo que previamente se definió cuando se modelo el proceso.

Se puede seleccionar que tareas dentro del proceso queremos habilitar para esta selección Ad-hoc con la propiedad “Selectable for Ad-Hoc”, de esta manera se puede evitar que el usuario pueda saltar de un paso a otro cuando no tiene sentido.

A nivel de las propiedades del procesos existe una propiedad Ad-Hoc que indica que todas las tareas son Ad-Hoc, independientemente de la propiedad de tarea.

At the process properties level there is an Ad-Hoc property that indicates that all tasks are Ad-Hoc, regardless of the task property.

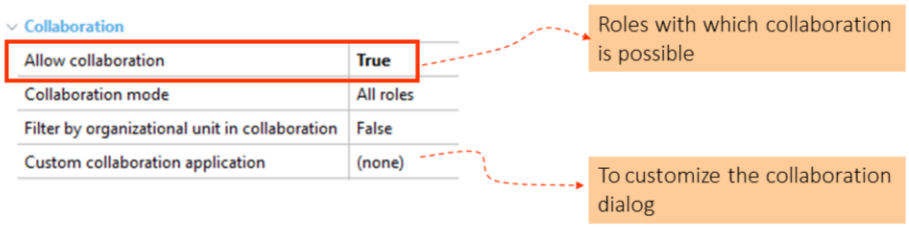


Una vez que se presiona la opción Enviar sobre una tarea que tiene marcada la propiedad Is Ad-Hoc con el valor True se muestra el dialogo para seleccionar cual o cuales serán las tareas siguientes. Este diálogo permite seleccionar una sola tarea ó más de una dependiendo de si la propiedad Multiple Choice esta configurada con el valor True o no.

Estos diálogos pueden personalizarse utilizando las propiedades “Custom Ad-hoc application” y “ Custom Assign application” de esta manera el desarrollador GeneXus cuenta con la posibilidad de personalizar el dialogo de selección de tareas y roles/usuario en cuyo caso a través del uso del tipo de datos de Workflow tendrá que realizar la creación y asignación de los Workitems que correspondan.

Collaboration

- Allow several users to work in parallel on the same task
- Task coordinator
  - Defines the list of collaborators
  - Is the only one who has the necessary permissions to complete the task



Hay casos en los cuales se necesita que varios usuarios colaboren para realizar determinada tarea, por ejemplo cuando es necesario realizar un informe técnico donde varios colaboradores aportan sus ideas.

Para habilitar la posibilidad de que un usuario que tiene asignada una tarea pueda pedir colaboración a otros usuarios es necesario configurar la propiedad Allow colaboration en True.

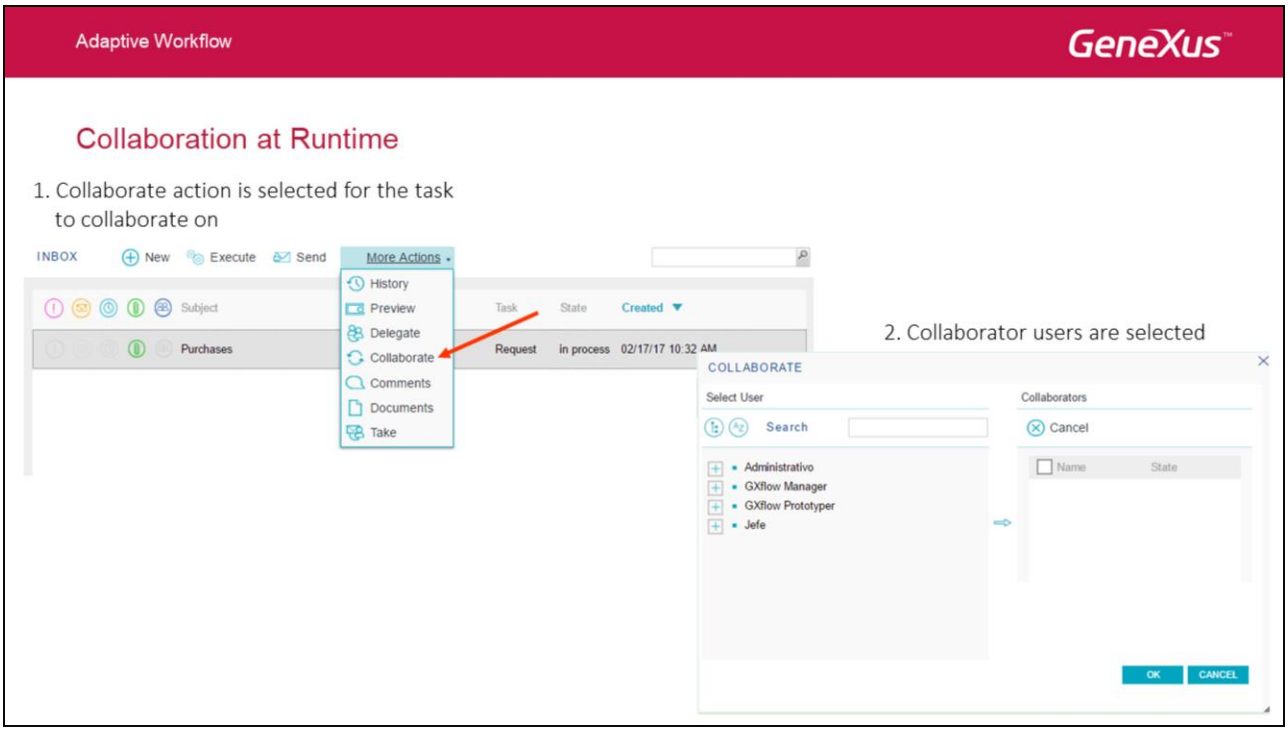
La propiedad Collaboration mode permite seleccionar quienes podrán colaborar si los usuarios de cualquier rol (All Roles), los usuarios que tienen los roles asignados a la tarea (Task Roles) ó los usuarios de determinados roles (Selected Roles).

El usuario entonces tiene la posibilidad de elegir en ejecución quienes van a colaborar en dicha tarea y en cualquier momento puede cancelar la colaboración de estos. El usuario que selecciona quienes van a colaborar se convierte en el coordinador de la tarea y sólo cuando este de por terminada la tarea el flujo seguirá adelante.

La propiedad “Filter by Organizational Unit in collaboration” permite indicar si los usuarios con los cuales se quiere colaborar deben estar en la misma Unidad Organizacional que el usuario que está seleccionando a los colaboradores o no. El valor True indica que se quiere aplicar este filtro.

La propiedad “Custom collaboration application” permite seleccionar un webpanel que será invocado cuando el usuario seleccione la acción “Colaborar” sobre la tarea. Esto da la posibilidad de personalizar la pantalla de colaboración y el desarrollador debe usar el Tipo de

datos de Workflow para realizar la asignación correspondiente.



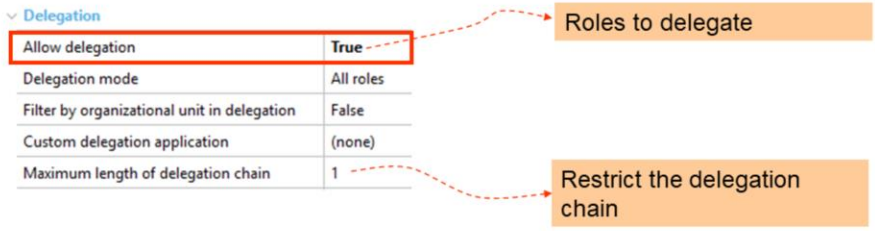
Cuando una tarea tiene habilitada la opción de Colaborar entonces el usuario que tiene asignada la tarea puede seleccionar la acción Colaborar luego de lo cual se le presenta la pantalla donde puede seleccionar los colaboradores.

Esta pantalla muestra a la izquierda un árbol con los roles y usuarios, al hacer doble clic sobre un usuario se lo va agregando a la lista de colaboradores que se muestra a la derecha. En caso de hacer doble clic sobre un rol se agregan todos los usuarios del rol seleccionado a la lista de colaboradores. Para cada uno de los colaboradores se genera una instancia de la misma tarea. El control de concurrencia sobre los datos de la aplicación que se estén manipulando en dicha tarea debe ser controlado por la propia aplicación.

Desde la lista de colaboradores se puede seleccionar uno o más usuarios y presionar el botón Cancelar lo cual permite cancelar la colaboración de determinados usuarios.

Delegation

- Delegate
  - Send the task to another user
- Delegate with return
  - When the task is completed, it returns to the original user



La delegación es una herramienta útil cuando queremos delegar en otro usuario la autoridad para hacer una tarea que nos fue encomendada.

Para habilitar la posibilidad de que un usuario que tiene asignada una tarea pueda delegar a otros usuarios es necesario configurar la propiedad Allow delegation en True.

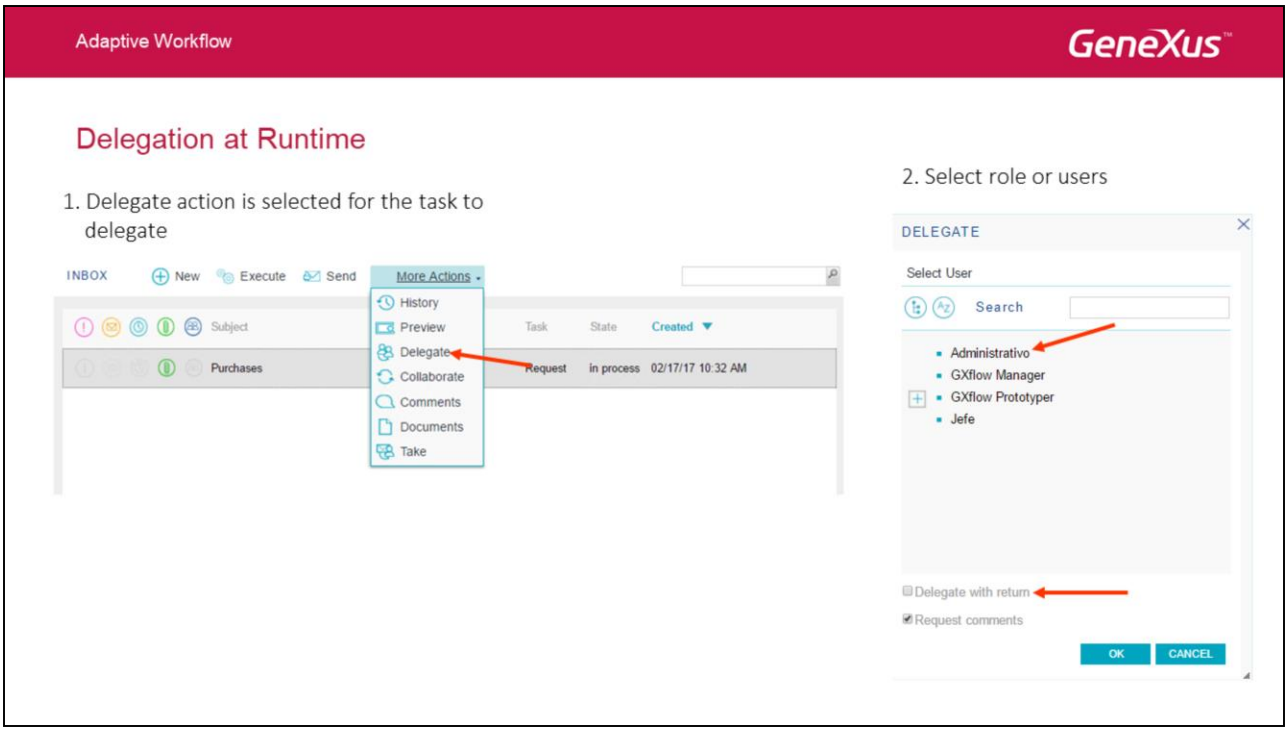
La propiedad Delegation mode permite seleccionar a quienes se podrá delegar si a los usuarios de cualquier rol (All Roles), a los usuarios que tienen los roles asignados a la tarea (Task Roles) ó a los usuarios de determinados roles (Selected Roles).

El usuario entonces tiene la posibilidad de elegir en ejecución a quien delegar dicha tarea y además puede especificar si quiere que la tarea le regrese una vez que el usuario al cual delego termina dicha tarea.

La propiedad “Filter by Organizational Unit in delegation” permite indicar si a los usuarios a los cuales puede delegar deben estar en la misma Unidad Organizacional que el usuario que está seleccionando a quien va a delegar. El valor True indica que se quiere aplicar este filtro.

La propiedad “Custom delegation application” permite personalizar el diálogo que será invocado cuando el usuario seleccione la acción “Delegar”. El desarrollador GeneXus permitirá la delegación correspondiente utilizando los tipos de datos de Workflow .

La propiedad “Maximum length of delegation chain” indica cuantas veces se dejará delegar la tarea.



Cuando una tarea tiene habilitada la opción de Delegar entonces el usuario que tiene asignada la tarea puede seleccionar la acción Delegar luego de lo cual se le presenta la pantalla donde puede seleccionar a quienes quiere delegar la tarea.

Esta pantalla muestra un árbol con los roles y usuarios. Al hacer clic sobre un rol entonces se está delegando la tarea al todos los usuarios dentro de ese rol, es decir, a todos ellos les aparecerá la tarea delegada y será el primero que la tome quien la podrá ejecutar. Al hacer clic sobre un usuario entonces se está seleccionando a ese usuario en particular para delegarle la tarea.

La opción “Delegar con retorno” le asegura al usuario que está delegando la tarea que luego de que se termine la tarea por parte del usuario al cual le delegó esta volverá a él.

La opción “Requerir comentarios” permite que al presionar OK en esta pantalla se pase a otra donde el usuario puede agregar comentarios que ayuden a entender el porqué de la delegación o que aporte otra información referida a la tarea.

Module 10

Monitoring Events

GeneXus

Events

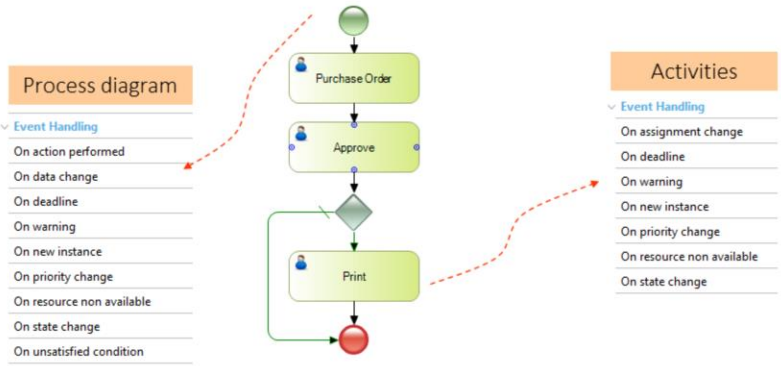
- During the operation of the Workflow engine, the following **internal events** take place:
  - Creation of a new work item
  - Creation of a process instance, etc.
- The event handling mechanism allows **running** a user-defined GeneXus procedure when the event takes place.

No se deben confundir estos eventos con los eventos que se declaran en los diagramas (los “business events”).

GeneXus

Events

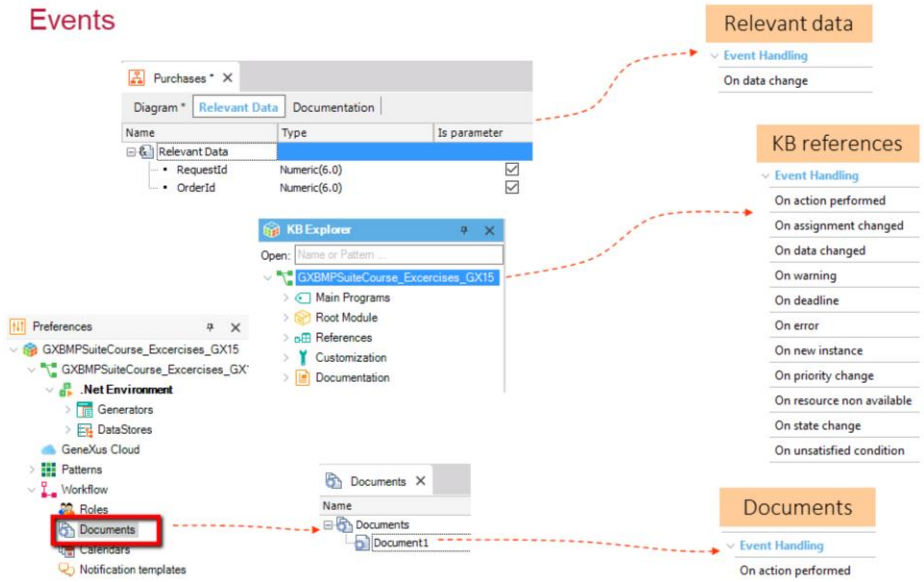
- Event handlers can be configured from different places:



Un procedimiento manejador de eventos puede declararse en distintos niveles:

-Nivel de proceso: se declaran en las propiedades del business process diagram. En este nivel el manejador se invocará siempre que el evento esté relacionado con el proceso en cuestión. Por ejemplo, si el manejador se asocia al evento “On new instance”, el mismo será invocado cada vez que se cree una nueva instancia del proceso y también cada vez que se cree un nuevo workitem de cualquier tarea del proceso.

-Nivel de actividad: se declaran en las propiedades de las actividades. En este nivel el manejador se invocará siempre que el evento esté relacionado con la actividad en cuestión. Por ejemplo si el manejador se asocia al evento “On new instance”, el mismo será invocado solamente cuando se cree un workitem de la actividad a la cual está asociado.



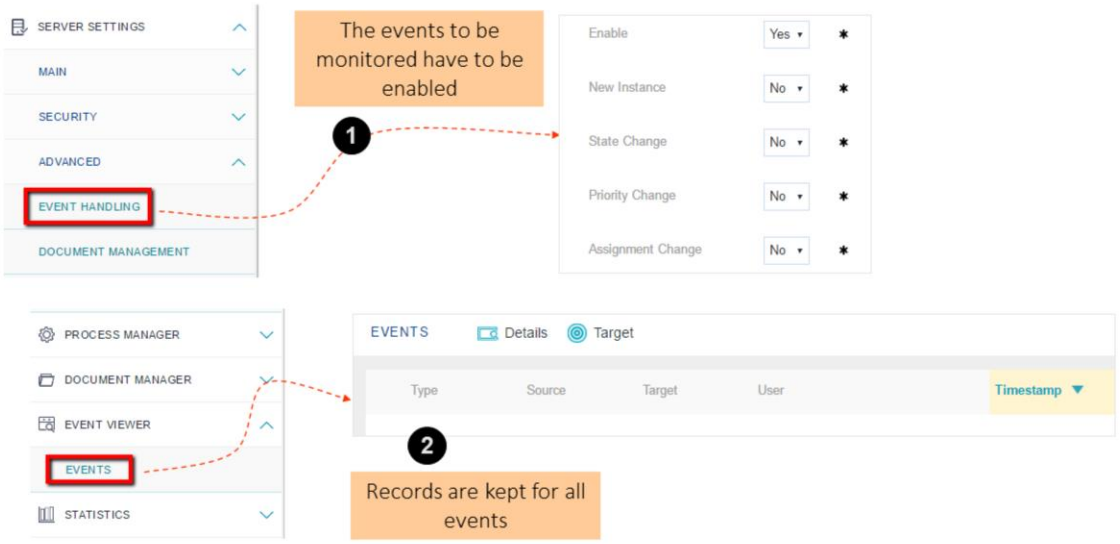
-Nivel de dato relevante: se declaran en las propiedades del dato relevante. En este nivel solo se tiene el evento “On data change” que ocurre cada vez que se modifica el valor de un dato relevante

-Nivel de documento: se declaran en las propiedades del documento. En este nivel solamente se tiene el evento “On action performed” que ocurre cada vez que se realiza una acción sobre el documento (creación, lectura, actualización o eliminación).

-Nivel environment: el manejador se invocará cada vez que ocurra el evento al cual está asociado sin importar si se originó en una entidad particular (instancia de proceso, workitem, etc.).

GeneXus

Events



GeneXus

Events

Event	Occurrence	Level
NewInstanceEvent	A new instance has been created	Process or task
DataChangeEvent	The value of a relevant data item has been changed	Relevant data item
StateChangeEvent	An instance status has been changed	Process or task
PriorityChangeEvent	An instance priority has been changed	Process or task
AssignmentChangeEvent	The assignment of an instance has been changed	Task

GeneXus

Events

Event	Occurrence	Level
WarningEvent	Instance warning	Process or task
DeadlineEvent	Instance deadline	Process or task
ErrorEvent	An error has occurred	All
ActionPerformedEvent	An action has been performed over a process document	Documents

Workflow Data Types

- **EventRepository**: Keeps a history log of all events occurred.
  - ListEvents(filter, events, error): Lists the events in the repository.

- **Event**: Represents a generic event.
  - TimeStamp: Creation date and time.
  - Type: Type of event.
  - User: User who caused the event.

GeneXus

Workflow Data Types

- **StateChangeEvent**: Change of status.
  - OldState: Previous status.
  - NewState: New status.
- **PriorityChangeEvent**: Change of priority.
  - OldPriority: Previous priority.
  - NewPriority: New priority.
- **DataChangeEvent**: Change of relevant data item.
  - DataName: Relevant data.
  - OldValue: Previous value.
  - NewValue: New value.
- **ActionPerformedEvent**: Execution of an action.
  - ActionPerformed: Performed action.
  - Context: Associated work item.

El tipo de datos base para trabajar con eventos es el WorkflowEvent, los demás tipos extienden a este tipo base agregando métodos y propiedades específicas para el evento. Por ejemplo, cuando se quiere trabajar con un evento de cambio de valor de un dato relevante, el tipo WorkflowDataChangeEvent agrega propiedades que permiten saber cuál era el valor del dato relevante antes del evento (OldValue) y cuál es el nuevo valor (NewValue).

Programming an event handler

- The procedure must have a parm rule of this type:

  Parm(in:&Event);

- &Event will be of WorkflowEvent type or one of its extensions, as applicable.

GeneXus

Programming an event handler

Example: the creation of instances for a certain process is to be monitored

- The procedure will receive a parameter of WorkflowEvent type.
- The procedure code would be as follows:

```
Do Case
    Case &Event.TargetType = WorkflowObjectType.PROCESS_INSTANCE
          &ProcessInstance = &Event.Target.ToProcessInstance()
          //...

    Case &Event.TargetType = WorkflowObjectType.WORKITEM
         &Workitem = &Event.Target.ToWorkitem()
         //...
EndCase
```

Cuando se escribe un manejador de procesos es necesario considerar los distintos casos donde se invocará este manejador. En particular, para el caso del ejemplo, si al manejador lo asociamos a un proceso, el mismo se invocará cada vez que:

- Se creen instancias del proceso.
- Se creen workitems de las tareas del proceso.

Para distinguir estos casos el tipo WorkflowEvent provee la propiedad, que indica cuál es el tipo de la entidad de workflow sobre la que ocurrió el evento. Los valores de esta propiedad están basados en el dominio WorkflowObjectType y en el ejemplo solamente se utilizan los valores PROCESS_INSTANCE y WORKITEM ya que son las dos posibilidades que se tienen para el tipo de evento que se está manejando.

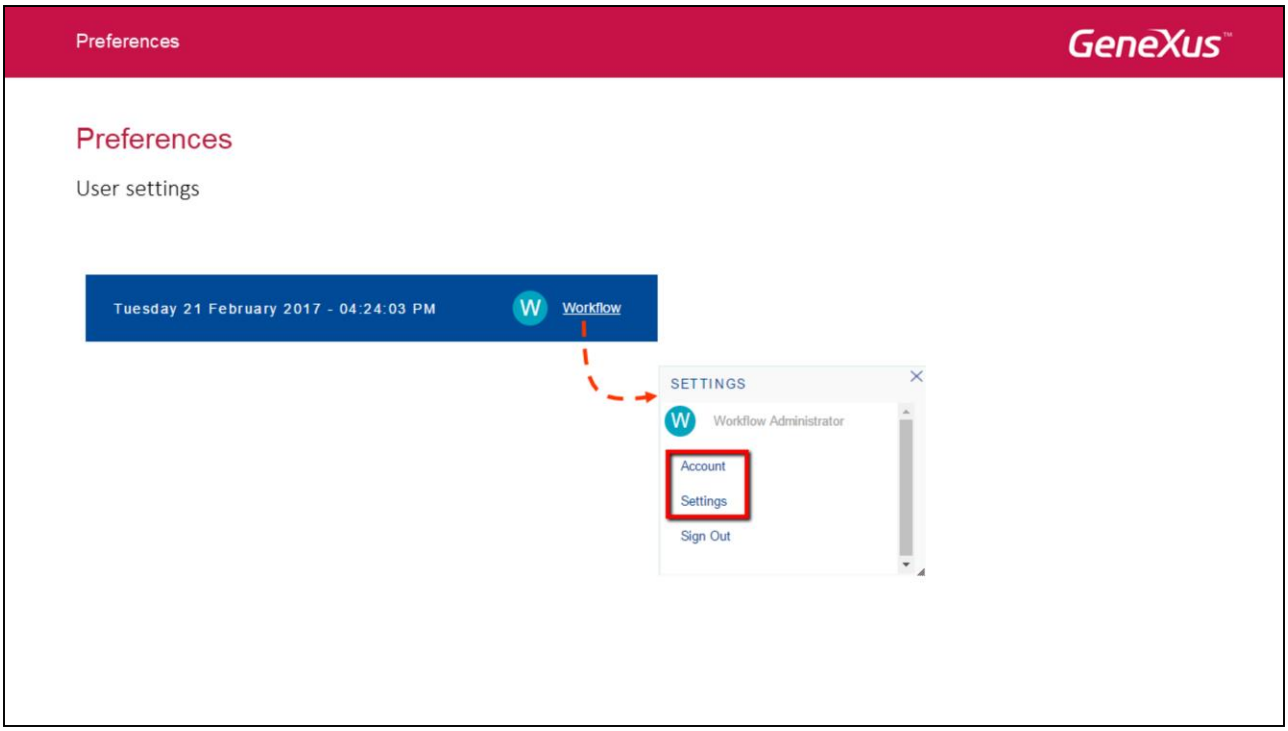
Otra propiedad importante del tipo de datos WorkflowEvent es la propiedad Target, la cual representa genéricamente a la entidad de workflow sobre la que ocurre el evento. Una vez que se sabe cuál es el tipo concreto de la entidad es posible convertir el valor de la propiedad Target a su valor concreto. Por ejemplo, si la entidad es una instancia de proceso:

&ProcessInstance = &Event.Target.ToProcessInstance() // &ProcessInstance es de tipo WorkflowProcessInstance

Luego se pueden utilizar todos los métodos que provee el tipo WorkflowProcessInstance para operar sobre la instancia de proceso.

Module 11

Preferences



Preferencias de Usuario

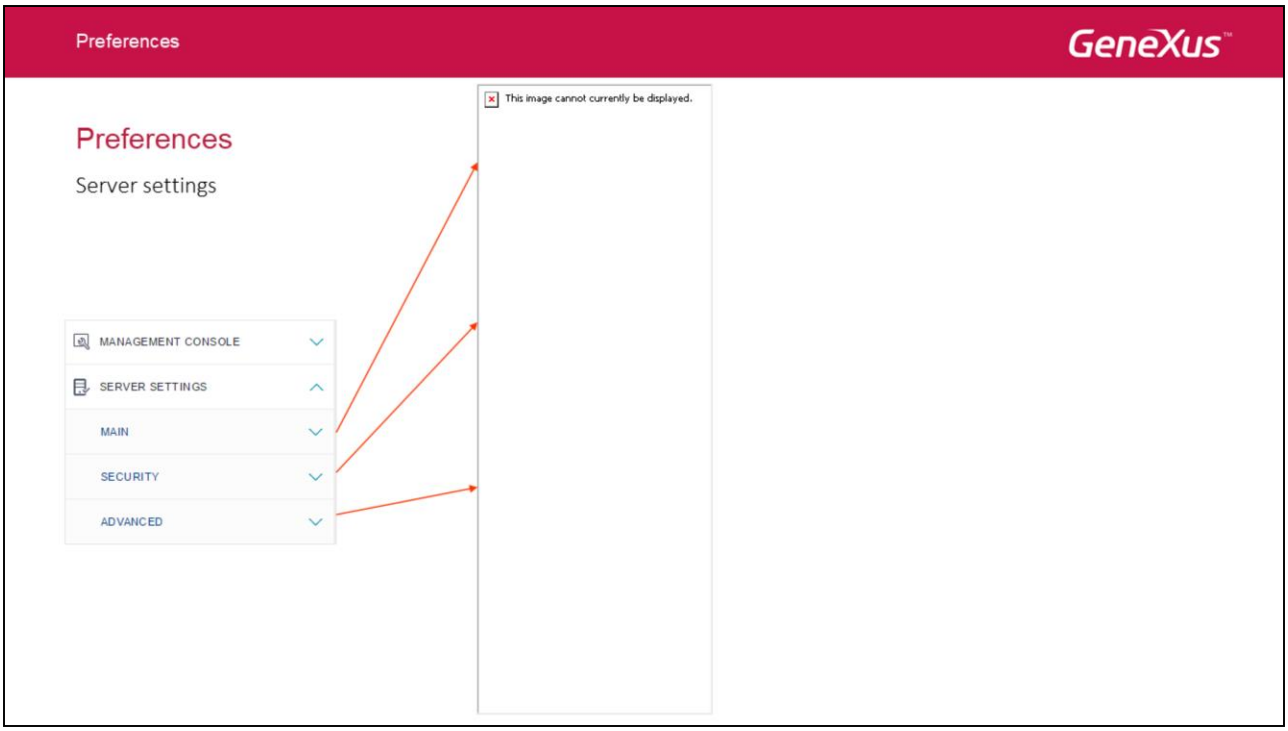
Son preferencias que puede ver cada uno de los usuarios del sistema

Cuenta

Permite al usuario configurar información de su cuenta (nombre, dirección de correo, si esta de licencia)

Configuración

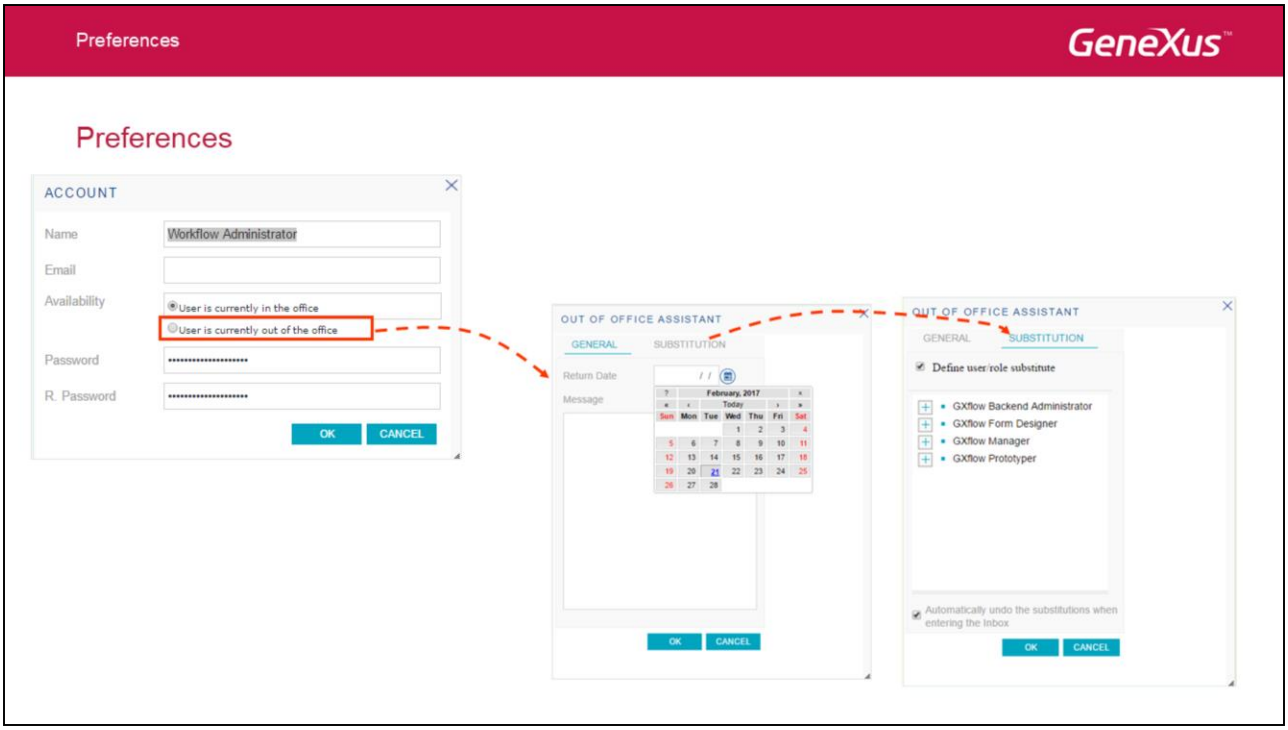
Permite configurar aspectos de la interfaz gráfica (cantidad de filas por página, número de botones, habilitar/deshabilitar doble clic, habilitar/deshabilitar la recepción de notificaciones, refrescar y ejecutar automáticamente la bandeja).



Preferencias del Servidor

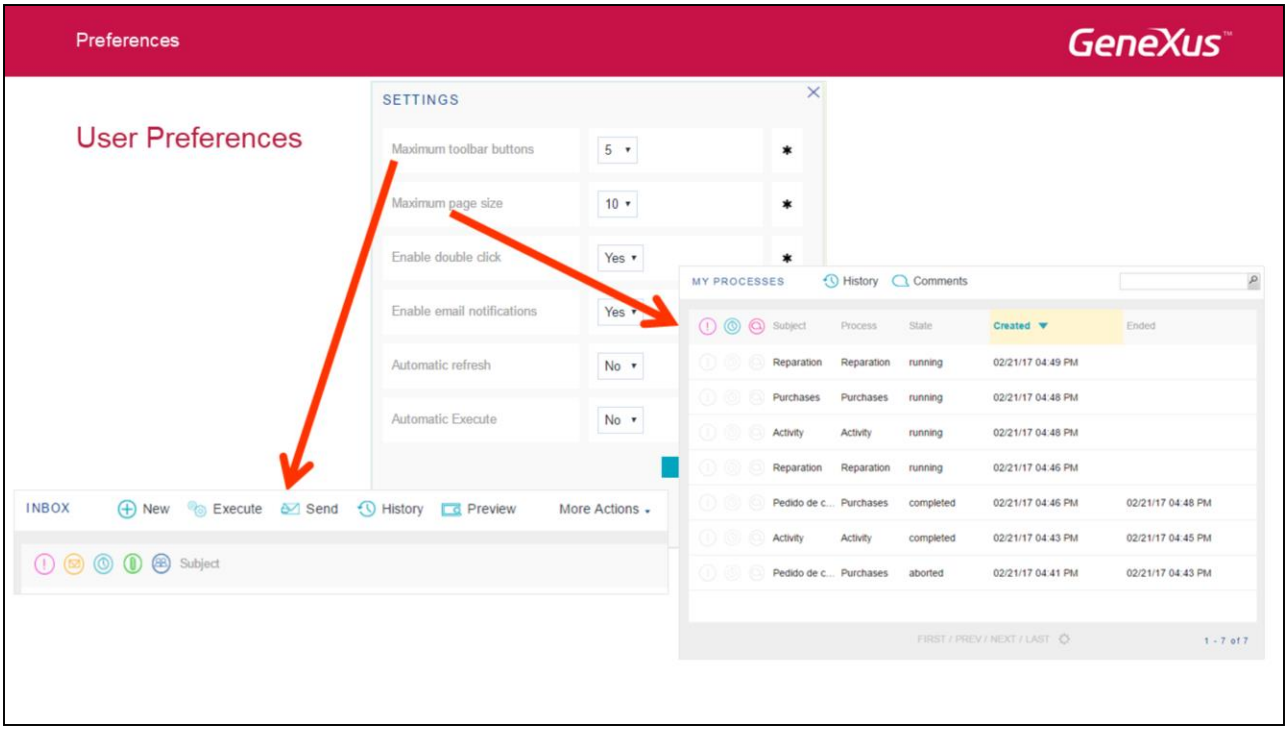
Son preferencias globales del servidor de wf. Solo pueden ser vistas/modificados por un usuario con el rol de administrador.

- **Aplicación:** Conjunto de preferencias que se aplican en tiempo de ejecución (ejecutar las tareas en una nueva ventana, ajustar automáticamente el tamaño de los formularios o definirles un tamaño estándar, encriptar parámetros de la url).
- **Lenguaje:** Permite configurar el idioma predeterminado del aplicativo Cliente (aunque después cada usuario puede elegir su idioma en forma independientemente).
- **Notificaciones:** Permite configurar la cuenta de correo para el envío de las notificaciones. Se requiere de un servidor SMTP.
- **Autenticación:** Esquema que se va utilizar para autenticar los usuarios, sus valores son: Gxflow, Windows, Externo y GAM, el valor default es Gxflow.
- **Manejo de Sesiones:** Permite configurar la duración de las sesiones de los usuarios.
- **Política de contraseñas:** Permite configurar vencimiento de las contraseñas y reglas de Gxflow (duración, caracteres mínimos, uso de mayúsculas y minúsculas, caracteres números, etc.) o customizar sus propias reglas.
- **Manejo de Eventos:** Permite configurar el nivel de granularidad en la generación de eventos de auditoría. Se pueden habilitar/deshabilitar a nivel general o para cada tipo de evento.
- **Manejo de Documentos:** Permite habilitar y configurar el manejo de documentos. Se puede configurar la ruta de subida, habilitar/deshabilitar la búsqueda full-text y el directorio donde se almacenará el índice, habilitar el uso de la firma digital, etc.



Si elegimos que el usuario está fuera de la oficina y presionamos OK, se abre el asistente del Out of Office.

Podemos elegir la fecha de regreso, el mensaje que se desea que aparezca y qué usuario será el sustituto.



Las preferencias Generales por usuario son:

Cantidad máxima de botones en la barra de herramientas: Define la cantidad de botones de la barra de herramientas a mostrar. Los valores son de 1 a 10 incluyendo el botón Más acciones.

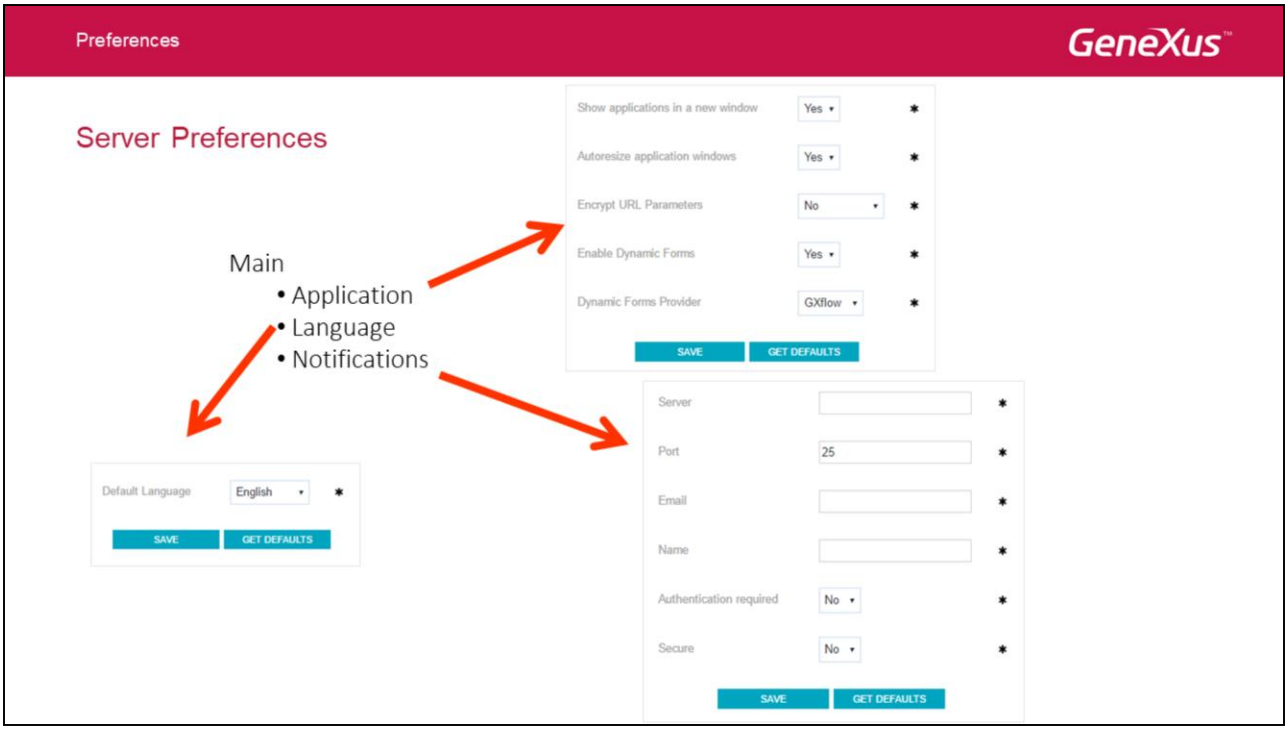
Filas por página: Indica el número de renglones que se mostrarán por página.

Habilitar doble clic: Permite habilitar o no la ejecución de una tarea mediante doble clic sobre su descripción en el Inbox. Sus valores son:
True: habilita la ejecución de la tarea (valor default) o False: no habilita la ejecución de la tarea.

Habilitar notificaciones email: Permite configurar la cuenta de correo para el envío de las notificaciones. Se requiere de un servidor SMTP

Refrescar automáticamente: Permite que el inbox se refresque automáticamente. Al activar la propiedad se puede elegir el tiempo de espera para refrescar. Los valores posibles son: 30, 60 o 90 segundos y 2, 3 o 5 minutos.

Ejecutar automáticamente: Cuando un usuario inicia una nueva instancia del proceso, la tarea inicial es asignada y ejecutada para dicho usuario en forma automática en caso de tener habilitada ésta propiedad.



Preferencias Main:

1. Application

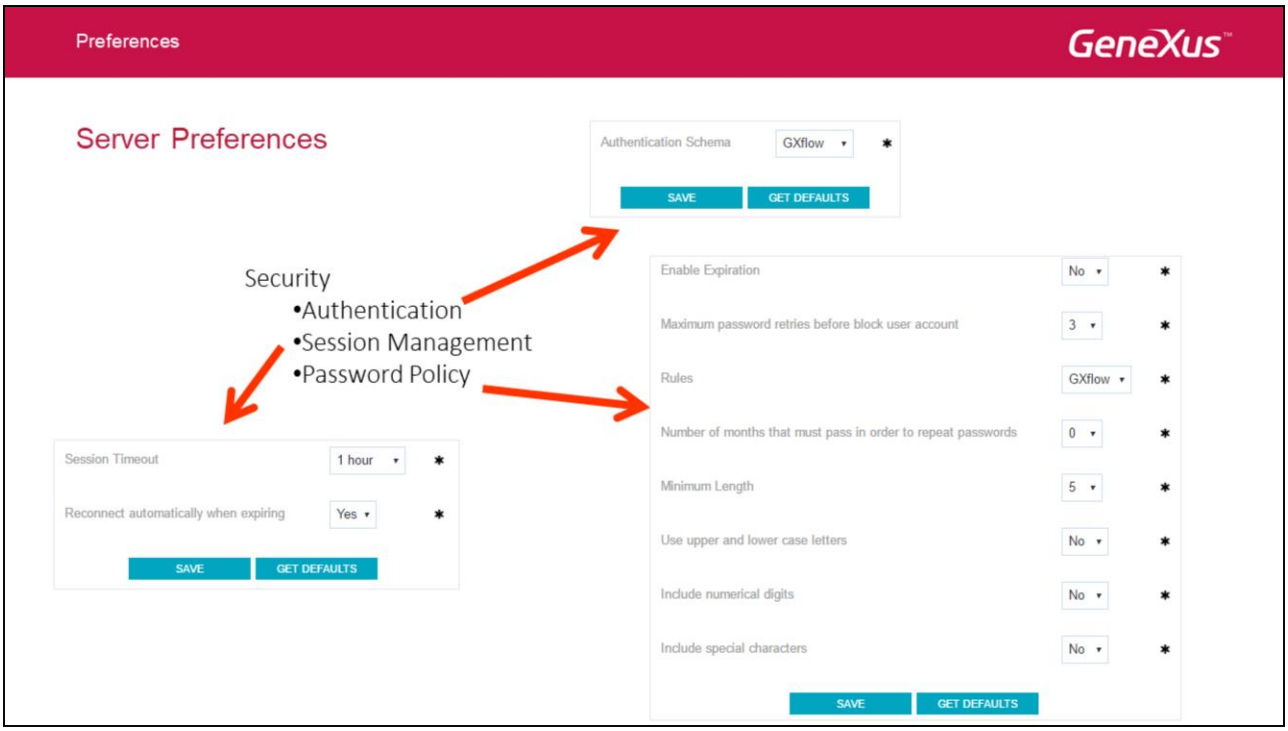
Se tiene la posibilidad de configurar que la tarea se ejecute en una nueva ventana, ajustar automáticamente el tamaño de la misma y poder encriptar los parámetros de la URL.

2. Language

Permite configurar el idioma predeterminado del aplicativo Cliente.

3. Notifications

Se configuran los datos del servidor SMTP que permita el envío de mails.



Preferencias del Servidor

1. Authentication

Los esquemas para autenticar a los usuarios son: GXflow, Windows y Externo, el valor default es GXflow.

GXflow: Esquema estándar de autenticación. La verificación se realiza en base a los usuarios definidos en el cliente.

Windows: Se utiliza el usuario del Sistema Operativo Windows.

Externo: Se crea una aplicación personalizada para verificar los usuarios.

2. Session Management

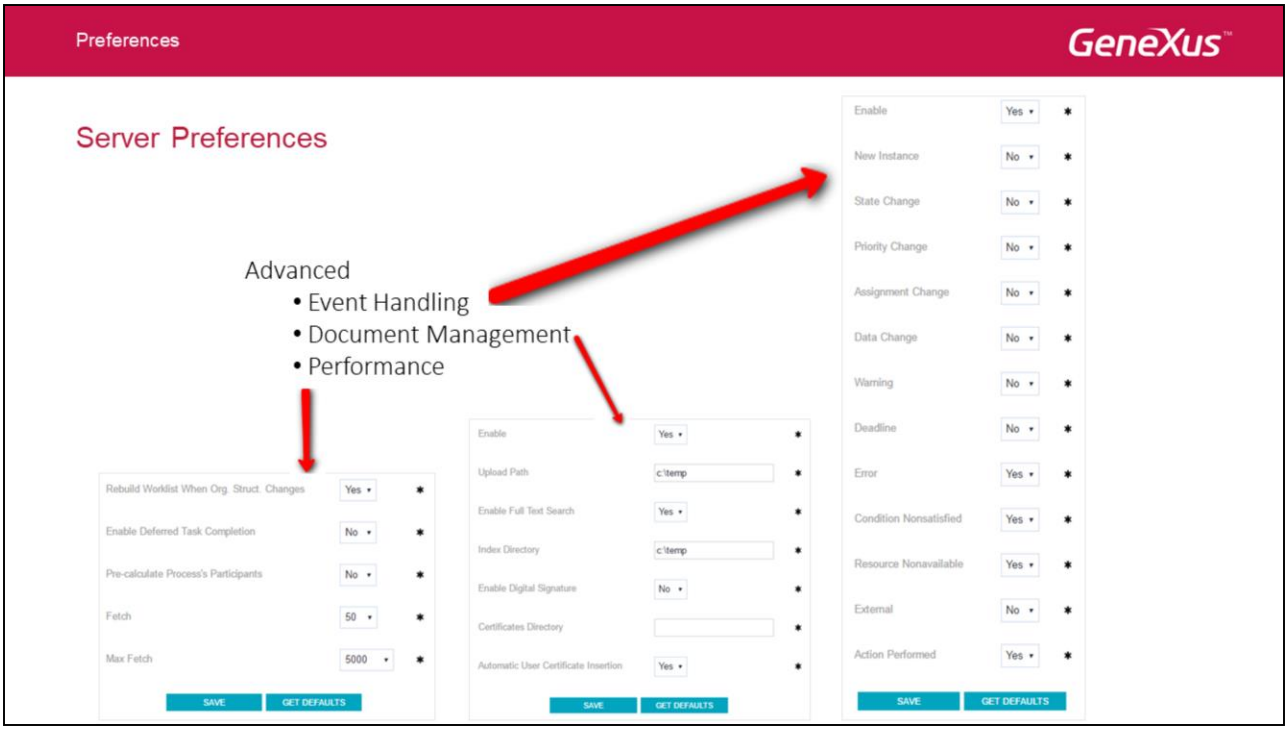
Permite configurar la duración de las sesiones de los usuarios. Los valores a configurar son:

Tiempo de espera de la sesión: Permite especificar la duración de la sesión de un usuario conectado. Puede ser 1,2,8 ó 24 horas.

Volver a conectar automáticamente cuando expira: Permite especificar que cuando la sesión del usuario registrado expira, se conecte automáticamente para evitar una nueva autenticación.

3. Password Policy

Permite configurar vencimiento de las contraseñas y reglas de Gxflow (duración, caracteres mínimos, uso de mayúsculas y minúsculas, caracteres númericos, etc.) o customizar sus propias reglas.



Avanzado:

Event Handling

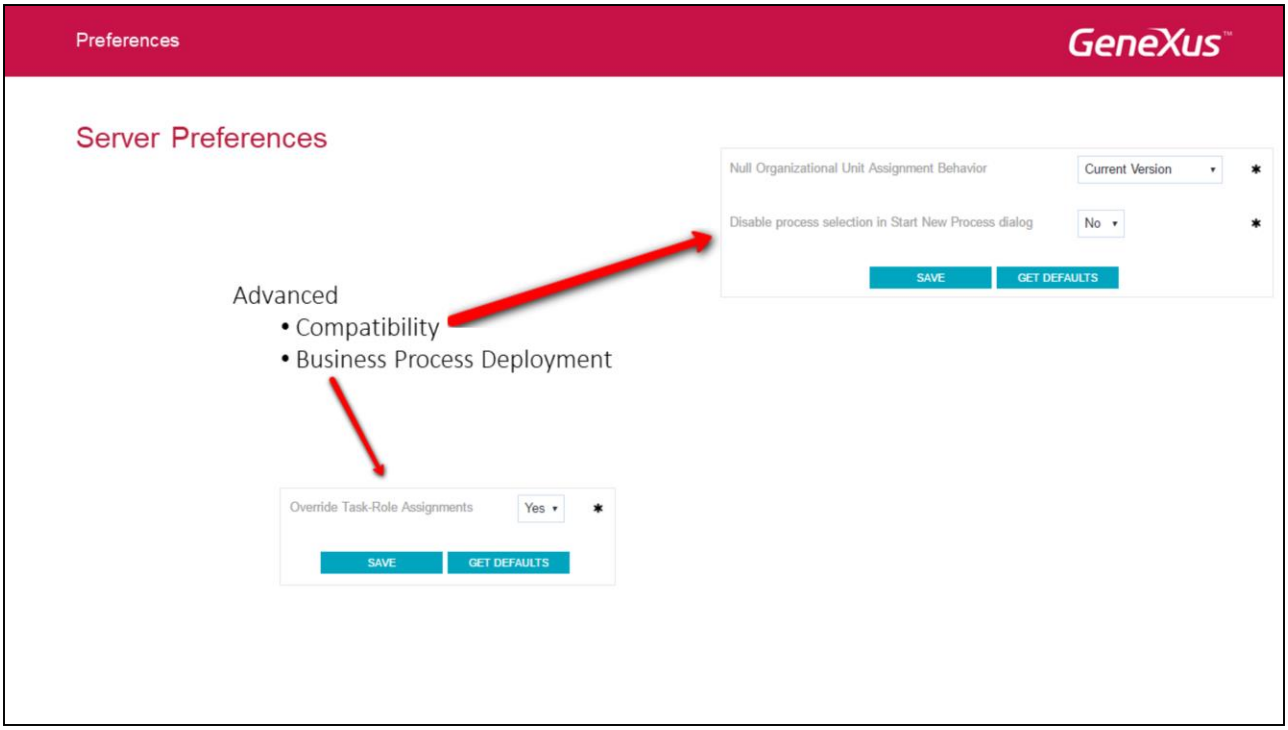
Permite configurar el nivel de granularidad en la generación de eventos de auditoría. Se pueden habilitar/deshabilitar a nivel general o para cada tipo de evento.

Manejo de Documentos

Permite habilitar y configurar el manejo de documentos. Se puede configurar la ruta de subida, habilitar/deshabilitar la búsqueda full-text y el directorio donde se almacenará el índice, habilitar el uso de la firma digital, etc.

Performance

Son un conjunto de preferencias que permiten modificar el momento en el que el motor de workflow efectúa determinadas acciones que afectan la performance de la aplicación. La configuración predeterminada es la que consideramos mejor tomando como referencia el cociente Usabilidad / Performance. Sin embargo, en determinados escenarios puede ser necesaria su reconfiguración. Por ejemplo, si en el escenario real la relación entre usuarios, roles y unidades organizacionales es muy dinámica (cambia permanentemente) probablemente haya que cambiar para que no se reconstruya la worklist ante estos cambios y agendar una reconstrucción general cada x tiempo.



Compatiblidad

Permite mantener la compatibilidad con bases de conocimiento migradas de GXflow 9.0 U1.

Valores:

Current Version: El usuario **no** puede ver las instancias de procesos con unidades organizacionales que él no tiene asignadas.

Version 9.0 and prior: El usuario puede ver las instancias de procesos con unidades organizacionales que no tenga asignadas.

El valor por defecto es **Current Version**.

Module 12

Deployment

El despliegue de una aplicación basada en Workflow se realiza desde el IDE, como cualquier otra aplicación GeneXus, exportando el conjunto de diagramas de proceso.

Deployment

The complete process consists of deployment of three components:

- The binary files of the business process diagrams and their dependencys
- The database files and metadata of process definitions
- The protection

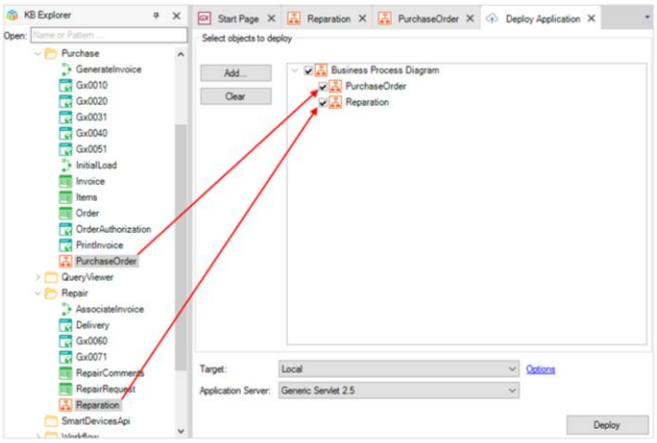
El proceso completo consiste de tres components: los archivos binaries, la base de datos / metadata y la protección.

GeneXus

Deployment of binaries application files

1) Build > Deploy application
2) Drag every Business Process Diagram from KBExplorer. Dependencys are automatically included.
3) Set your preferred options and press Deploy

More info:
http://wiki.genexus.com/commwiki/servlet/wiki?32092



En esta etapa se distribuyen los archivos binarios de los diagramas de procesos y de los objetos GeneXus asociados a los mismos.

GeneXus prove un mecanismo muy simple para empaquetar los archivos generados y desplegar la aplicación.

Se siguen los siguientes pasos:

1) Build > Deploy application. Esto desplegará una ventana contextual en el IDE
2) Arrastrar cada objeto Business Process Diagram desde el KBExplorer y soltarlo en la aventana de “objetos a desplegar”. No es necesario incluir cada dependencia del diagram (como WebPanels o procedimientos), ya que GeneXus los incluirá automáticamente sin mostrarlo en la pantalla.
3) Setear las opciones que se prefieran y presionar Deploy

GeneXus

Deployment of database and metadata files

1) Creation of business process deploy file (file with extension .bpd)
2) Use of Business Process Deployer tool, to impact database and update metadata

More info:
http://wiki.genexus.com/commwiki/servlet/wiki?11607

En esta etapa se disponibilizan las definiciones de la base de datos y la metadata de los procesos que se modelan en GeneXus para que puedan ser ejecutados por el motor de workflow.

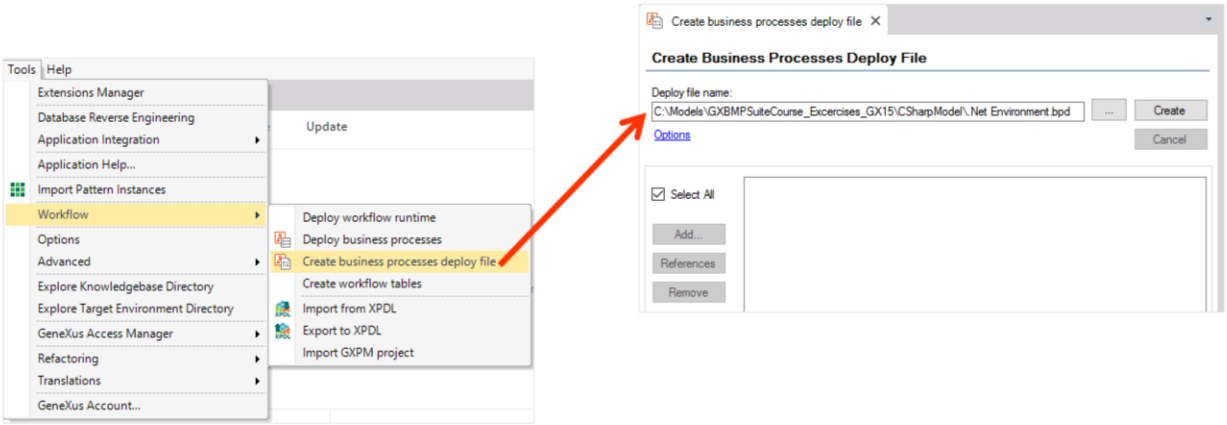
Es importante aclarar que esta operación puede requerir realizar reorganizaciones en la base de datos. La primera vez que se haga un deploy de procesos se crearán las tablas de workflow. En siguientes deploys también puede ser necesario ejecutar reorganizaciones de upgrade cuando se va a utilizar un nuevo upgrade o versión de GXflow. Todas estas reorganizaciones se ejecutarán automáticamente solicitando previamente la autorización del usuario.

La herramienta Business Process Deployer hace posible el impacto de la base de datos de producción incluyendo a los nuevos procesos de negocios o modificando los existentes.

Los procesos que están corriendo mientras se hace el deploy no serán afectados, solo los procesos creados luego del impacto reflejarán los cambios.

Creation of Business Process Deployment file

1) Tools Menu > Workflow > Create business process deploy file (file with extension .bpd)



Lo primero que se debe hacer es crear el archivo de deploy de procesos desde GeneXus mediante la opción del menú "Tools->Workflow-> Create business processes deploy file".

El resultado de esta operación es un archivo que contiene lo siguiente:

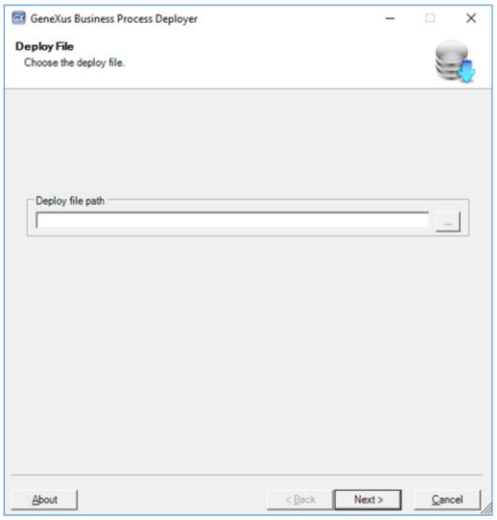
- La definición de los procesos modelados en GeneXus.
- Todas las reorganizaciones que podrían requerirse antes de realizar el deploy de procesos (creación de tablas y reorganizaciones de upgrade). Estas reorganizaciones ya están generadas y compiladas para la plataforma con la que se está trabajando en GeneXus. Esto hace que el archivo de deploy sea de la plataforma con la que se está trabajando. Si por ejemplo este archivo se crea para para el ambiente .NET SQLServer no podrá ser utilizado para realizar el deploy en una base de datos MySQL.

Use of Business Process Deployer to impact database and update metadata

2) Open the BPDeployer and import the generated file.

Business Process Deployer:

- Tool to perform process deployment.
- Uses the deployment file created in GeneXus as input.
- No licenses are required.



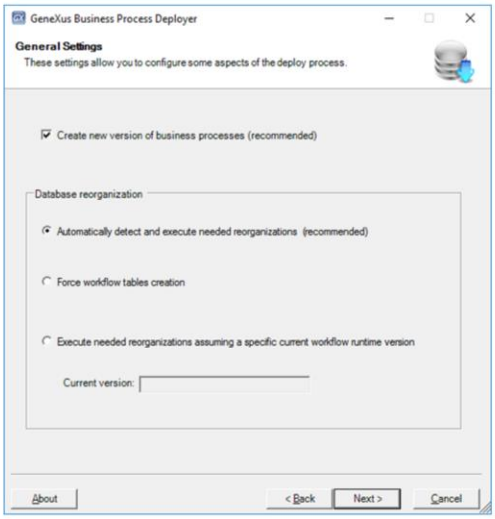
Una vez que se tiene creado el archivo de deploy de procesos, para ejecutar el deploy en la base de datos de producción se utiliza el GeneXus Business Process Deployer. Esta herramienta se incluye en el paquete de instalación del ambiente de producción de GXflow y no requiere licencias. La misma puede ejecutarse en modo wizard (por defecto) o por línea de comandos.

El ejecutable se encuentra en la carpeta de instalación de GeneXus bajo:

- \Packages\Gxpm\Tools\BPDeployer\Net (para .Net)
- \Packages\Gxpm\Tools\BPDeployer\Java (para Java)

Use of Business Process Deployer to impact database and update metadata

- Create new version of business processes
- Automatically detect and execute necessary reorganizations



Una vez seleccionado el archivo de deploy se deben configurar las siguientes opciones:

Create new version of business processes (recommended)

Esta opción permite establecer que los procesos incluidos en el archivo de deploy serán tomados en cuenta a partir de nuevas instancias de proceso. Las instancias de proceso que están actualmente en ejecución seguirán basándose en la definición de proceso a partir de la cual fueron creados.

Automatically detect and execute needed reorganizations (recommended)

El archivo de deploy generado en GeneXus contiene la versión con la cual fueron creados esos procesos y la base de datos de producción tiene su propia versión. Esta opción permite especificar que automáticamente se realicen las reorganizaciones necesarias para migrar la base de datos de producción a la versión indicada en los diagramas.

Force workflow tables creation

Esta opción permite forzar la creación de las tablas de workflow.

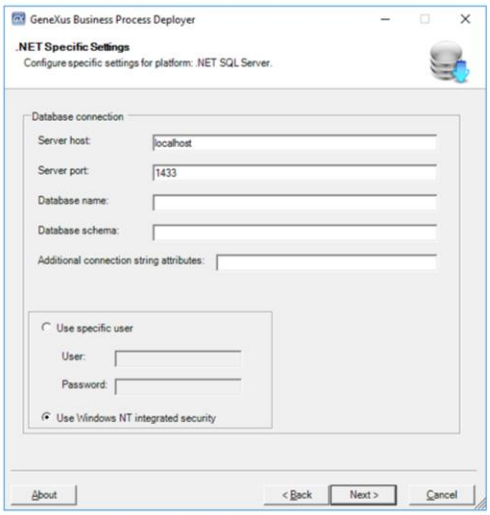
Execute needed reorganizations assuming a specific current workflow runtime version

Esta opción permite especificar una versión específica, de esta forma se ignora la versión de la base de datos de producción.

GeneXus

Use of Business Process Deployer to impact database and update metadata

- Configuring the production database

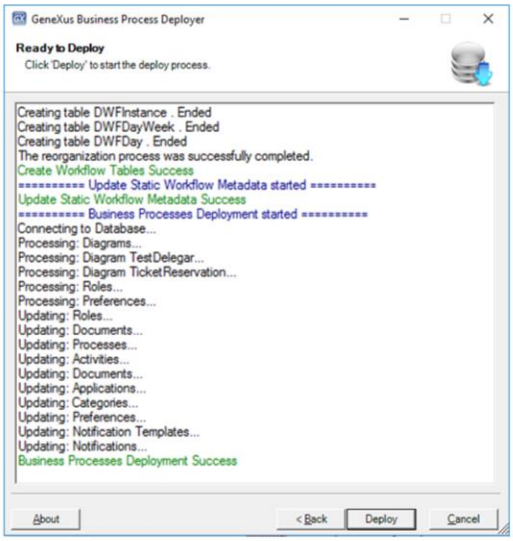


El siguiente paso es configurar los datos de la base de datos de producción.

GeneXus

Use of Business Process Deployer to impact database and update metadata

- Reorganizing the database
- Process Deployment



Por último hacemos el deploy de los procesos y la primera vez se crearán las tablas de workflow. Luego a medida que nuestros procesos sean actualizados a nuevas versiones de GeneXus se realizarán las reorganizaciones necesarias para actualizar las tablas de workflow en la base de datos de producción.

Protection settings (for Windows platform)

1. Install **GXflow Production Environment**.
2. Go to Start > All Programs > GXflow GeneXus 15 > License Manager and install licenses.
3. Copy the file "protect.ini" file that resides in C:/Program Files (x86)/GeneXus/GXflow/GXflowGx15/LicMgr to:
   - NET: application´s bin directory (replace the existing one).
   - Java: Tomcat´s lib directory (by default in C:/Program Files/Apache Software Foundation/Tomcat X.Y/lib)
   - Java: JBoss's bin directory or the folder associated with the startup process.
4. Install GeneXus Protection Server.
5. (Only for Java) Copy and extract this file to the Tomcat´s lib directory (by default in C:/Program Files/Apache Software Foundation/Tomcat X.Y/lib).
6. Go to Start > All Programs > GeneXus 15 Business Process Deployer and deploy the processes.

For other platforms, go to :
http://wiki.genexus.com/commwiki/servlet/wiki?19848

More info about this subject:
http://wiki.genexus.com/commwiki/servlet/wiki?11607

Cuando la aplicación está pronta para ser desplegada en el servidor, la protección debe ser actualizada con ella.

En esta sección se dan los pasos cuando la plataforma es Windows. Si usan otra plataforma como Linux o IBM iSeries (AS400), sigan los links en pantalla.

Deployment

GeneXus

GXflow Production Environment

It consists of installing the following:

- GXflow License Manager
- Business Process Diagram Deployer

Steps:

1. Download the setup file from Download Center
2. Execute *Setup.exe* file at production server.
3. Choose the target path for the application and click "Install" to initialize the process.
4. Once the application is installed, close the dialog.
   It is recommendable to restart your computer once installation process has finished

More info:
http://wiki.genexus.com/commwiki/servlet/wiki?19849

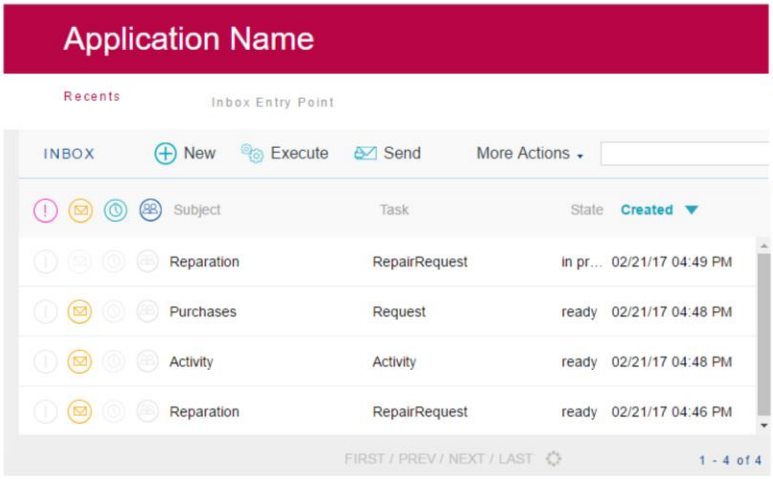
El ambiente de producción de GXflow comprende todo lo necesario para poder ejecutar una aplicación que utilice procesos de negocio.

Esencialmente está compuesto por el License manager de GXflow que permite administrar las licencias de los usuarios que se conectan o acceden al cliente GXflow y el Business Process Deployer que permite impactar la base de datos con los cambios en los procesos.

Module 13

GXflow Client Integration & Customization

GXflow Entry Point

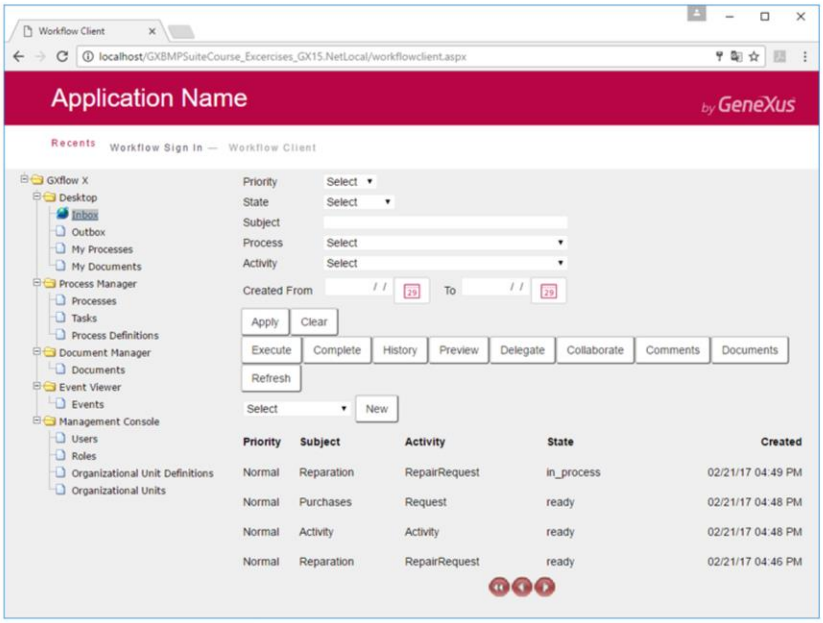


More info: http://wiki.genexus.com/commwiki/servlet/wiki?10721

GXflow Entry Point es un **user control** que se distribuye con GeneXus (<directorio de GeneXus>/Packages/Gxpm/Extra/GXflowEntryPoint) y se instala al igual que el resto de los user controls de GeneXus.

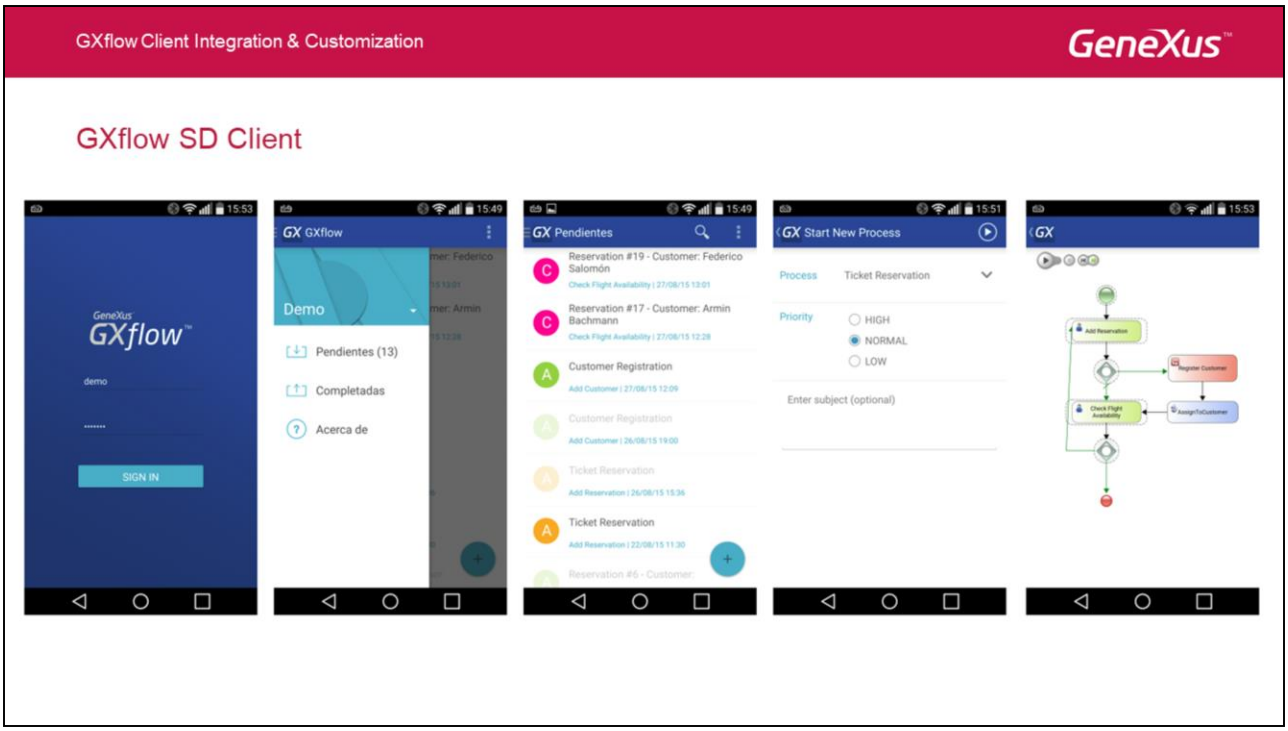
Permite utilizar los distintos componentes del cliente GXflow (Bandeja de Entrada, Bandeja de Salida, Procesos, etc.) en como aplicativos separados, por ejemplo en un web panel como se muestra en la imagen.

GXflow Custom Client



GXflow Custom Client es un conjunto de objetos (xpz) que utilizan el API e implementan el cliente de GXflow. Este xpz es abierto, es decir que el usuario puede consolidarlo y modificar el cliente, tanto desde el punto de vista estético como desde el punto de vista del comportamiento.

Se distribuye con GeneXus (<directorio de GeneXus>\Packages\Gxpm\Extra\CustomClient.xpz).



El **cliente GXflow Smart Devices** al igual que el GXflow Custom Client es un conjunto de objetos, construidos utilizando la API de GXflow que el usuario puede consolidar en su base de conocimiento. Esto permite incorporar a la aplicación la posibilidad de ejecutar procesos y tareas a través de un dispositivo móvil.

Existen dos formas de incorporar estos objetos, dependiendo de si ya tenemos una KB creada o no.

Si no hemos creado aún una KB, creamos una KB desde GeneXus Server, eligiendo la KB **GXflowClientSDDemo** del servidor : **http://samples.genexusserver.com/xev3** (hoy disponible desde un GXServer en Evolution 3. en breve estará disponible en un servidor en GeneXus 15). Luego agregamos los objetos diagramas de proceso y los objetos Smart Devices que vamos a asociar a las tareas del diagrama.

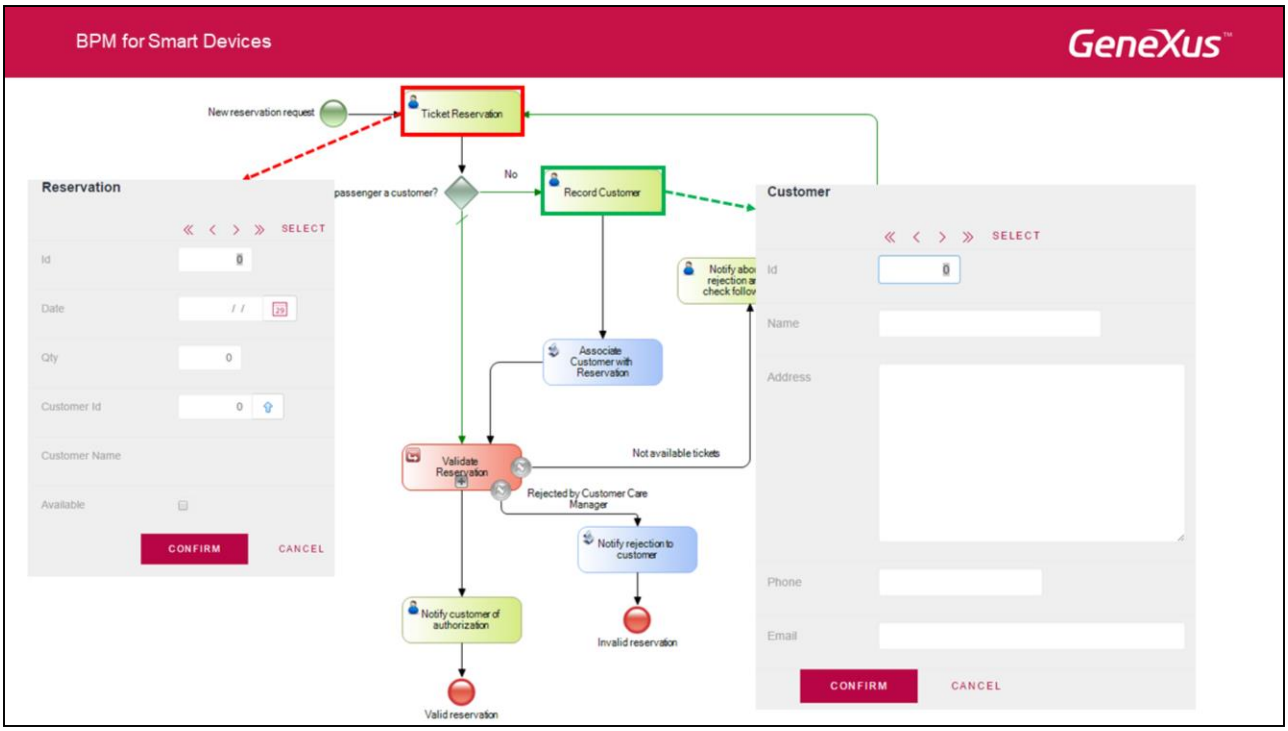
Si ya tenemos una KB creada, hoy deberíamos crear la KB del server como en el caso anterior y luego exportar los objetos de la carpeta WorkflowClientSD a nuestra KB, pero en breve estará disponible un XPZ para facilitar el proceso. Cabe aclarar, que antes de importar los objetos, debemos tener el GAM aplicado a nuestra KB y ya importada la API de Workflow, es decir que ya hagamos ejecutado un diagrama de procesos al menos una vez.

Para ejecutar la aplicación, o bien compilamos nuestra KB o también podemos descargar el cliente Gxflow desde el Apple Store o desde Google Play y cambiar la Service URL para apuntar a nuestra instalación local.

Por más información: http://wiki.genexus.com/commwiki/servlet/wiki?29037

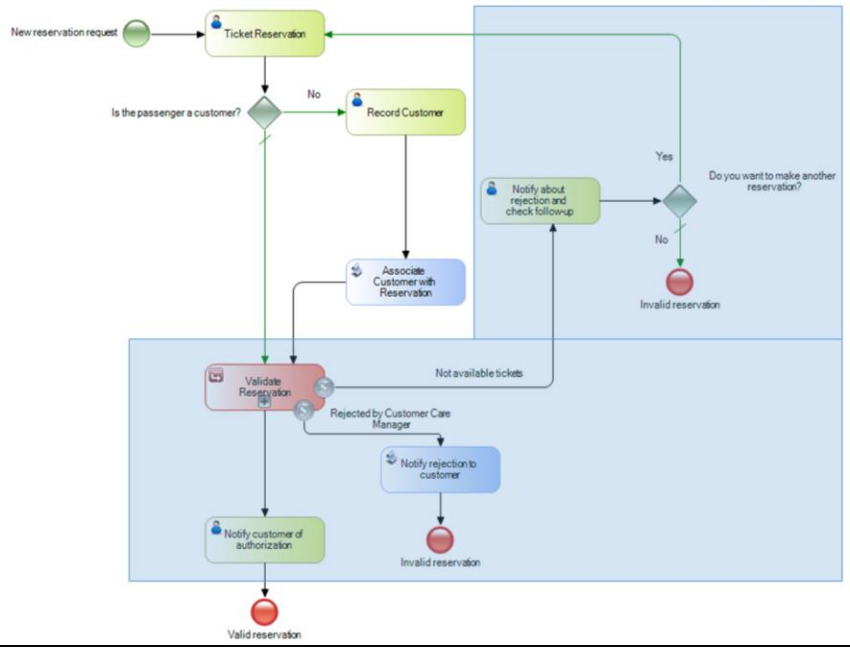
Module 14

BPM for Smart Devices



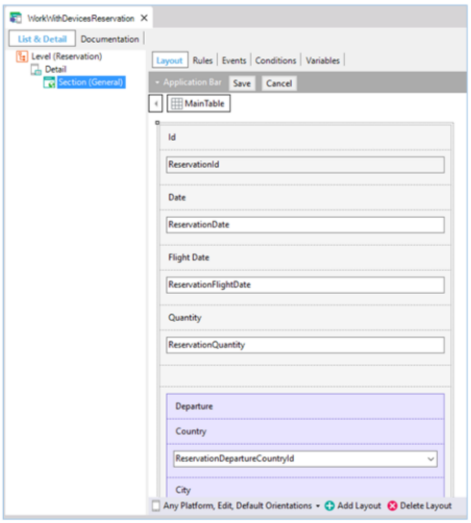
Hasta ahora hemos asociado objetos GeneXus que se ejecutaban en una página web, a las tareas interactivas de los diagramas de procesos.

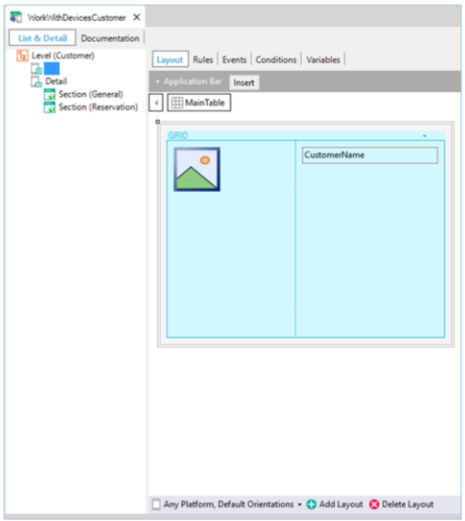
Pero también es posible asociar a las tareas de usuario, objetos GeneXus que se ejecuten en un Smart Device. Por ejemplo, en el caso del proceso de ingresar una reserva en la agencia de viajes, es posible ejecutarlo desde un dispositivo móvil.



Vamos a simplificar el proceso de reserva de pasajes que habíamos diseñado. Quitamos el subproceso de validación y algunos símbolos que no usaremos. Para especificar la disponibilidad de la reserva utilizaremos la transacción Reservation y su atributo ReservationAvailable.

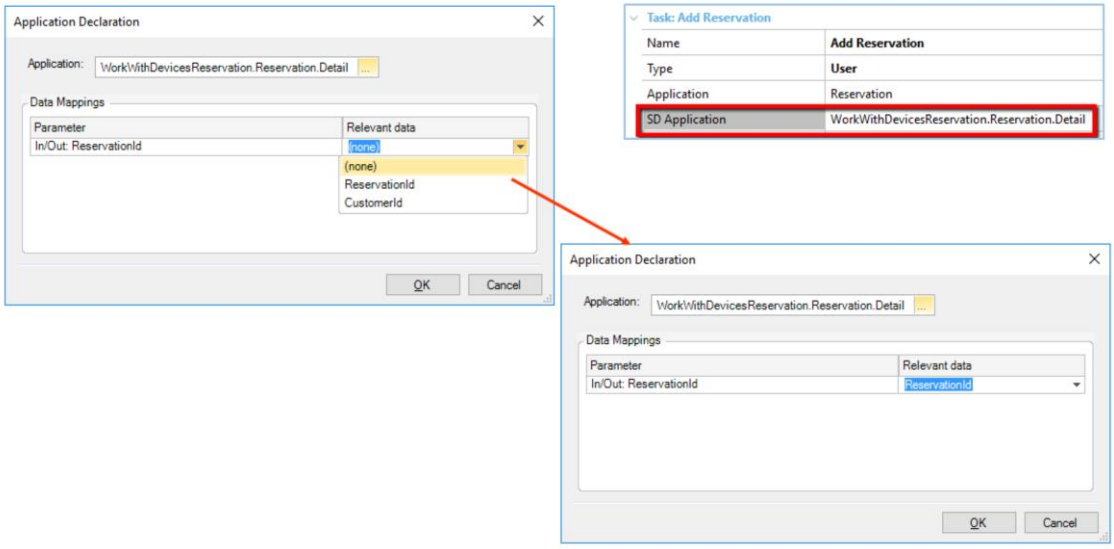
Association of objects WW for SD





Lo primero que vamos a hacer ahora es aplicar el patrón Work With Smart Devices a las transacciones Reservation y Customer, que son las involucradas en el proceso. Para eso, seleccionamos ambas transacciones, damos botón derecho, elegimos Apply Pattern y Work With for Smart Devices.

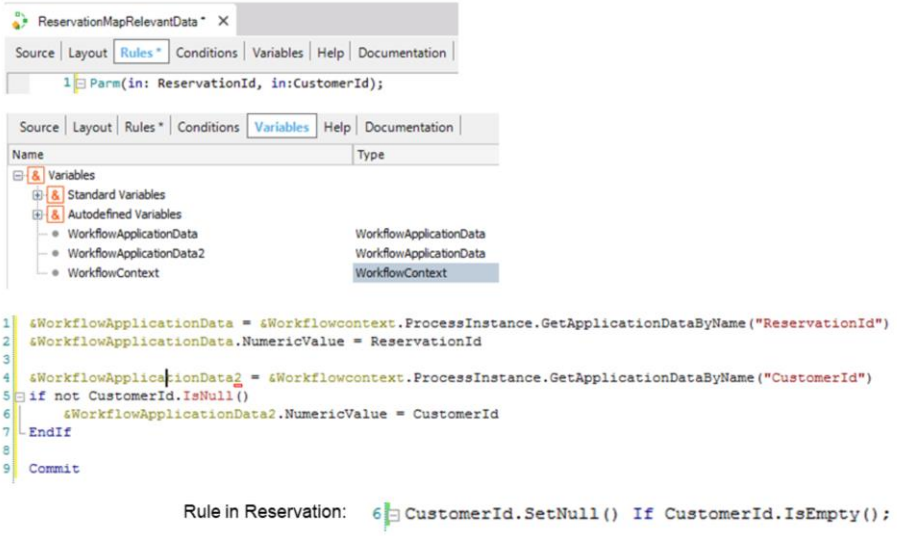
Association of object WorkWithDevicesReservation to the Add Reservation task



Ahora asociaremos a la tarea TicketReservation, la aplicación SD que generó el patrón.

En las propiedades de la tarea, en la propiedad SD Application presionamos el botón y elegimos WorkWithDevicesReservation.
Presionamos tabulador y asociamos el dato relevante ReservationId.

Creation of procedure to associate relevant data of Reservation

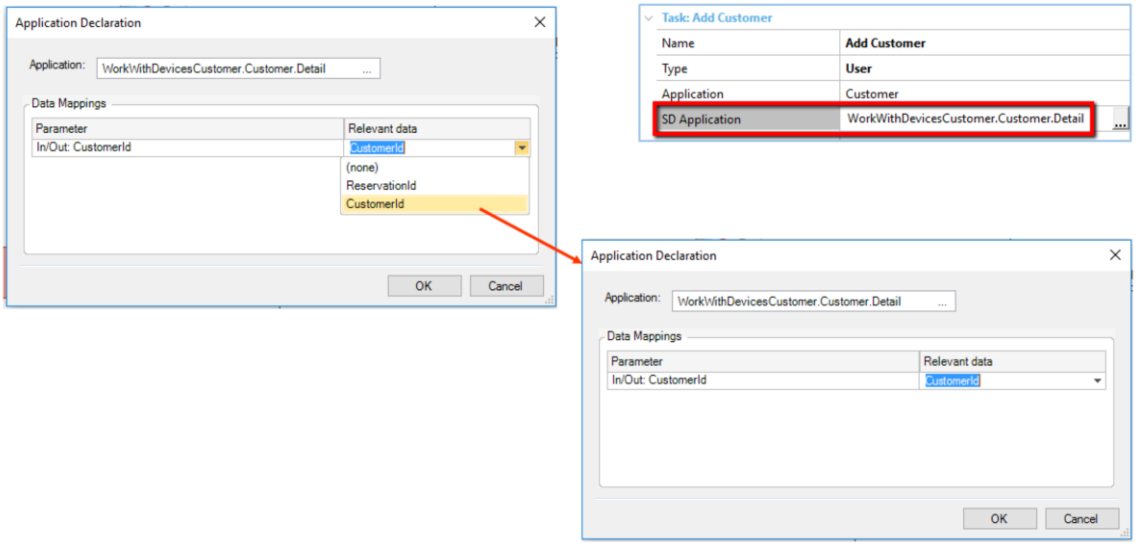


Los datos relevantes no se mapean automáticamente entre el diagrama y el objeto SD. Necesitamos crear un objeto procedimiento *ReservationMapRelevantData*, para realizar el mapeo, usando la API de Workflow.

No debemos olvidarnos de incluir el Commit, cuando trabajamos con tipos de datos Workflow.

Como preguntamos por el valor isNull() de CustomerId, debemos agregar la regla en la transacción Reservation.

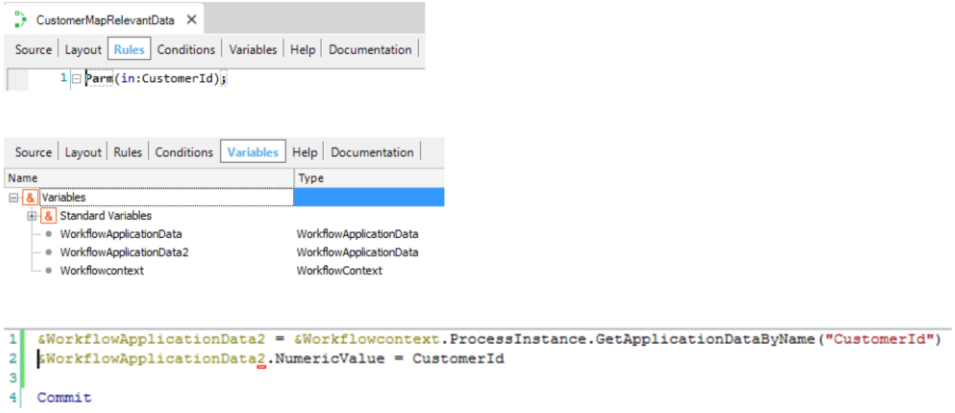
Association of object WorkWithDevicesCustomer to the Add Customer task



En las propiedades de la tarea, en la propiedad SD Application presionamos el botón y elegimos WorkWithDevicesCustomer.

Presionamos tabulador y asociamos el dato relevante CustomerId.

Creation of procedure to associate relevant data of Customer



Análogamente, necesitamos crear un objeto procedimiento *CustomerMapRelevantData*.

No debemos olvidarnos de incluir el Commit, cuando trabajamos con tipos de datos Workflow.

Inclusion of invocations to mapping procedures in Save events of WW for SD

In WorkWithDevicesReservation:

```
Event 'Save'
    Composite
        SDActions.Save()
        ReservationMapRelevantData.Call(ReservationId,CustomerId)
        return
    EndComposite
EndEvent
```

In WorkWithDevicesCustomer:

```
Event 'Save'
    Composite
        SDActions.Save()
        CustomerMapRelevantData.Call(CustomerId)
        return
    EndComposite
EndEvent
```

Tasks prior to executing the app

1) Import and set up **GXflow para Smart Devices client**

http://wiki.genexus.com/commwiki/servlet/hwiki?HowTo%3A+Configuring+GXflow+Client+For+Smart+Devices

2) Invoke each SD object used in our process diagram

We do it on the WorflowSDClient dashboard:

```
Event 'DummyCalls'
        WorkWithDevicesReservation.Reservation.Detail(1)
        WorkWithDevicesReservation.Reservation.List()
        WorkWithDevicesCustomer.Customer.Detail(1)
        WorkWithDevicesCustomer.Customer.List()
EndEvent
```

Para poder ejecutar el diagrama de procesos que construimos, debemos importar y configurar el **cliente GXflow para Smart Devices.**

Una vez importado y configurado el cliente GXflow para SD, es necesario que tengamos una invocación a cada objeto SD usado en nuestro diagrama de procesos.
Lo hacemos agregando código en el dashboard WorkflowSDClient.

Execution on a mobile device

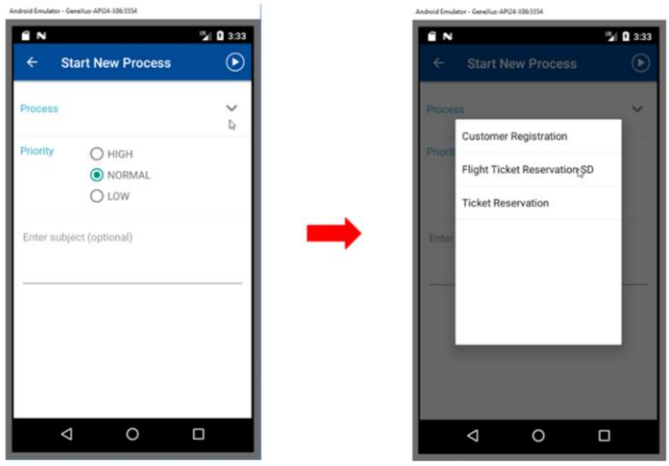
Presionamos F5 y vemos que el emulador de Android se ejecuta automáticamente mostrando la pantalla de login.

Para nuestro Login usaremos el usuario administrador de workflow. Ingresamos al usuario WFADMINISTRATOR y lo usamos también como contraseña.

Execution on a mobile device

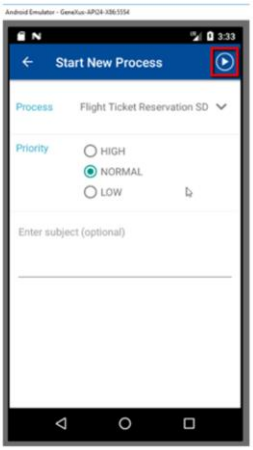
Seleccionamos la bandeja de entrada y presionamos el botón de “+” para instanciar un proceso.

Execution on a mobile device



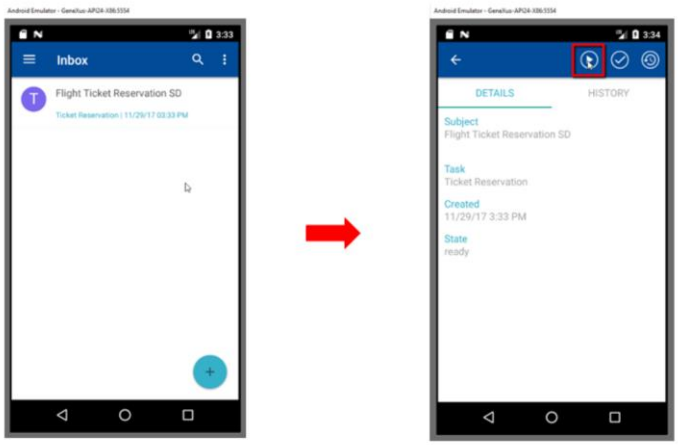
Ahora seleccionamos el proceso FlightTicketReservationSD.

Execution on a mobile device



Vemos que se abre la ventana del proceso. Presionamos el botón de Start para iniciar el mismo.

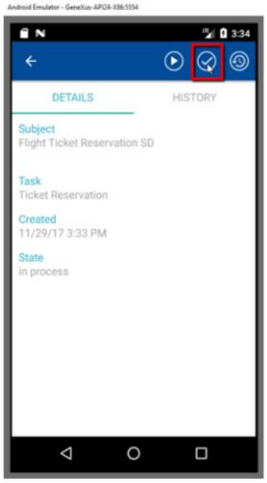
Execution on a mobile device



Ahora vemos que el proceso se ha iniciado y que tenemos la tarea TicketReservation pendiente de ejecutar.

Hacemos clic sobre la tarea y presionamos el botón de la flecha para iniciarla.

Execution on a mobile device

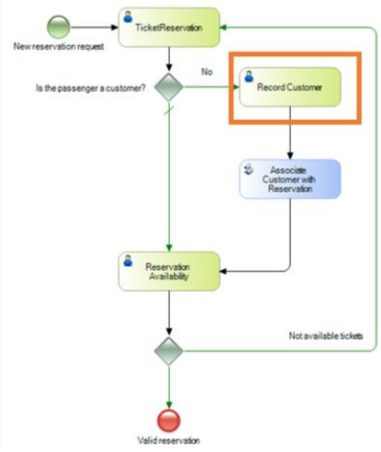


Vemos que se abre el objeto SD para trabajar con reservas, para que ingresemos una reserva. Vamos a ingresar una reserva nueva, dejando el Id sin especificar ya que es autonumerado y dejamos el identificador del cliente vacío.

Ahora presionamos el botón de Confirmar.

Para finalizar la tarea, elegimos Completar.

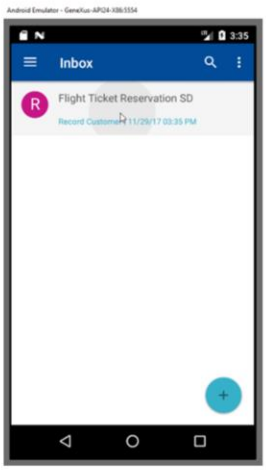
Execution on a mobile device

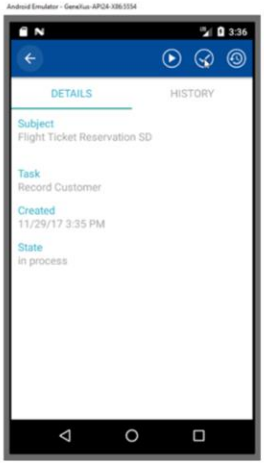


Vemos que se abre la bandeja de entrada y ahora la tarea pendiente de ejecución es RecordCustomer.

Como dejamos el cliente sin ingresar, el motor de workflow evaluó la condición del exclusive Gateway “Is the passenger a customer?” y determinó que la siguiente tarea será RecordCustomer, la cual invocará al objeto SD Trabajar con Clientes, para que ingresemos al cliente.

Execution on a mobile device

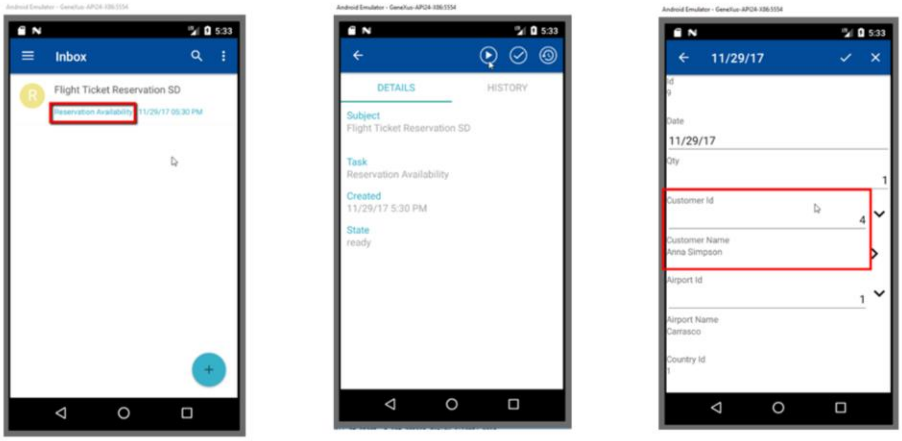




Ejecutamos la tarea Record Customer…Ingresamos los datos del cliente y presionamos Confirmar.

Para finalizar, completamos la tarea Record Customer.

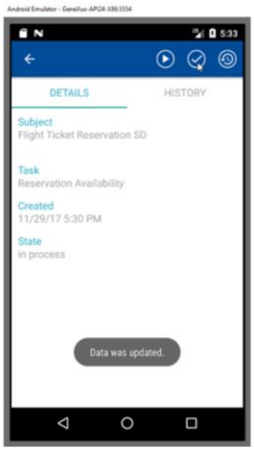
Execution on a mobile device



Ahora la próxima tarea que aparece como pendiente es ReservationAvailabilityDamos click sobre la misma para abrirla y a continuación la ejecutamos.

Observemos que el cliente fue exitosamente asignado a la reserva, dado que se ejecutó correctamente el procedimiento asociado a la tarea batch AssociateCustomerToReservation.

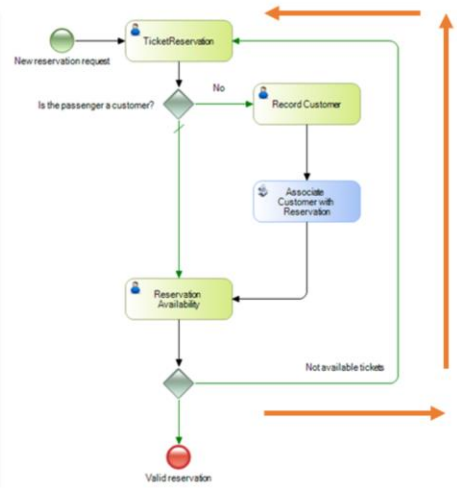
Execution on a mobile device



Vamos a marcar que la reserva está disponible y presionamos Confirmar.

Para finalizar, completamos la tarea ReservationAvailability.

Execution on a mobile device



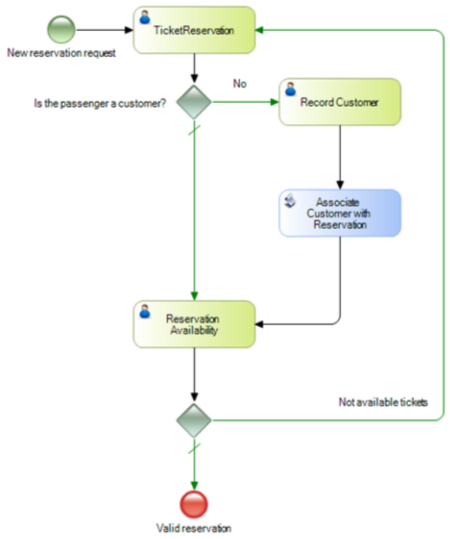
Ahora la bandeja de entrada no muestra más tareas pendientes, lo que implica que finalizó la ejecución del proceso de reserva de la agencia de viajes.

Si hubiéramos marcado que la reserva no estaba disponible, se hubiera ejecutado nuevamente el objeto SD WorkWithDevicesReservation, para que ingresáramos una reserva nueva.

BPM for Smart Devices

GeneXus

Smart Devices: https://training.genexus.com/smart-devices-en/mobile-applications-with-genexus-15-course-en?en



BPM Suite: https://wiki.genexus.com/commwiki/servlet/wiki?24723

Hemos visto así cómo ejecutar un diagrama de proceso de negocios, en un dispositivo móvil.

En particular, hemos asociado a las tareas, los objetos generados por el patrón Work With for Smart Devices, pero también podríamos haber asociado un objeto SD creado por nosotros, como por ejemplo un panel para SD.

En este caso hemos generado la aplicación para Smart Devices en Android y ejecutamos la misma utilizando un emulador, pero es posible prototipar sobre un dispositivo físico y generar aplicaciones para otras plataformas como por ejemplo dispositivos iOS como Ipad o Iphone.

Para saber más sobre aplicaciones Smart Devices, visite el link que se muestra en pantalla.

Y para conocer más posibilidades de la suite BPM de GeneXus, visite el siguiente link del wiki.

GeneXus 15

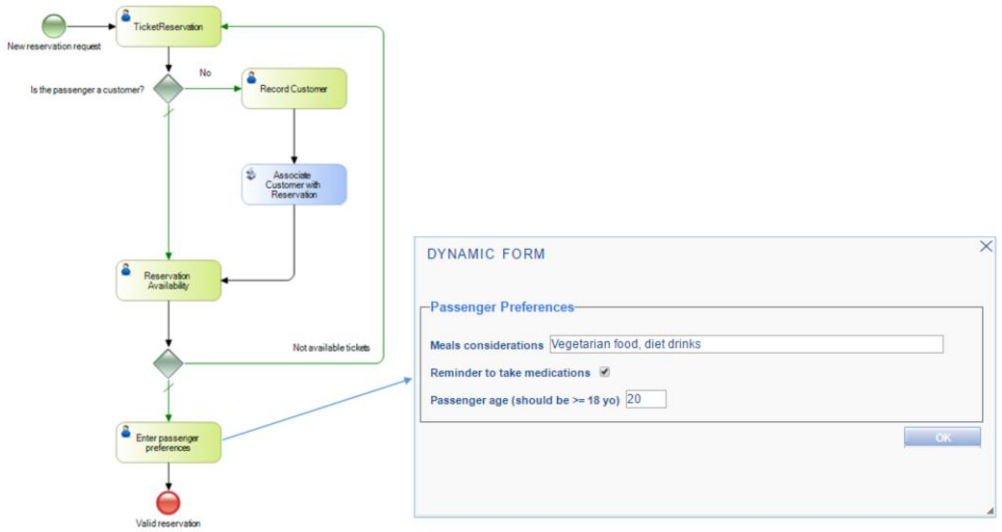
Module 15

Dynamic Forms

En general, para ejecutar una tarea interactiva en un diagrama de proceso de negocios, tenemos que tener en nuestra KB un objeto transacción o web panel para ser asociado a la tarea.

Los formularios dinámicos permiten crear pantallas de entrada de datos directamente desde el cliente workflow y asociarlas con las tareas interactivas, sin la necesidad de escribir una sola línea de código,

Dynamic Forms

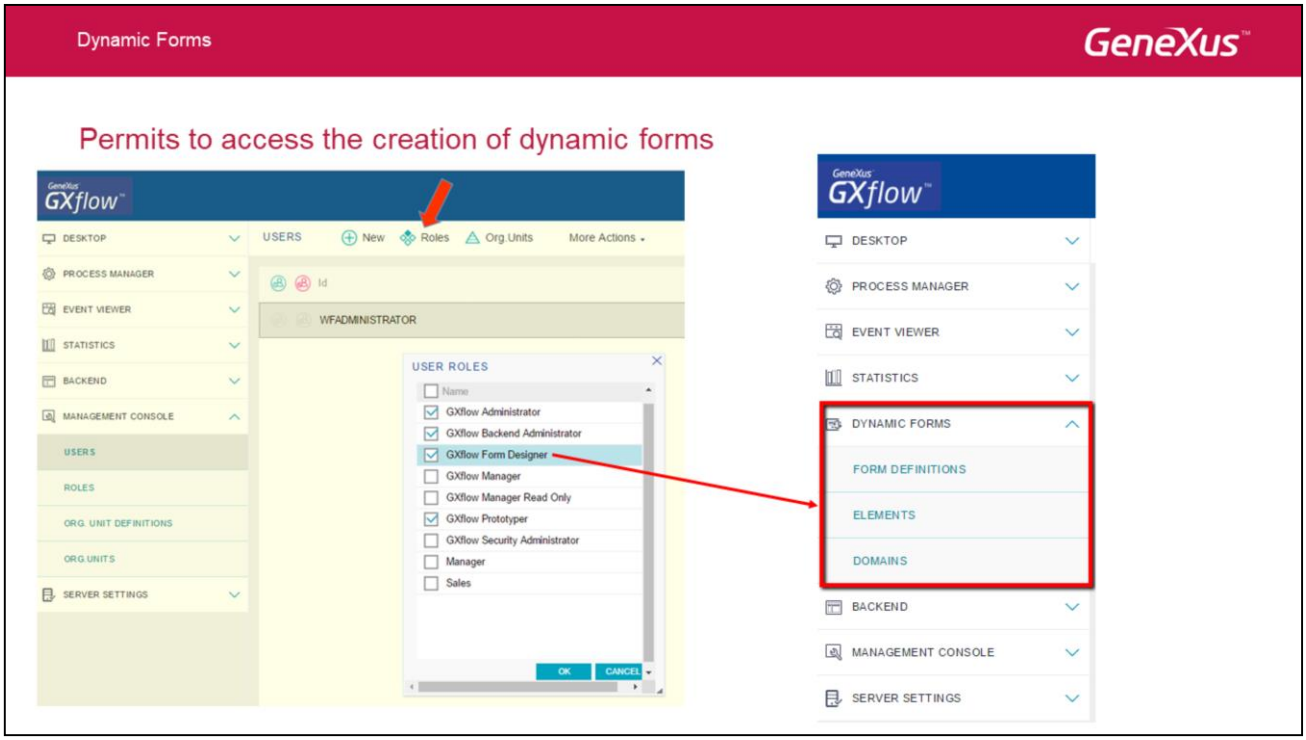


Un escenario típico del uso de estos formularios es cuando recibimos un requerimiento nuevo de ingreso de datos y el sistema ya está en producción. En lugar de rediseñar la aplicación, podemos crear formularios dinámicos y adaptarlos en tiempo de ejecución para almacenar información, sin tener que modificar la base de datos de la aplicación.

Otro caso podría ser cuando se desea guardar datos de un cierto proceso que no teníamos previsto en el workflow original, en el cual debemos incluir información que sólo es relevante a cada instancia del proceso. Los formularios dinámicos nos permiten almacenar esta información en las tablas de workflow sin tener que contar con estructuras en la base de datos para este tipo de información.

En nuestro ejemplo, la agencia de viajes desea registrar algunas preferencias, de los clientes que han hecho una reserva de pasajes.

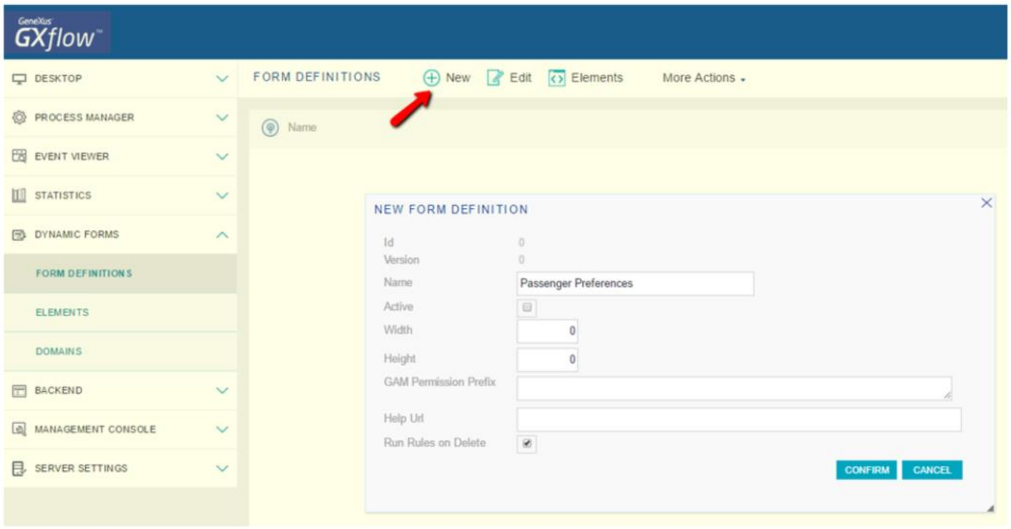
Estos datos no los queremos almacenar en la base de datos del sistema, porque pueden depender de las preferencias del pasajero para cada vuelo en particular, por lo que solicitaremos esos datos mediante una tarea asociada a un formulario dinámico, en el mismo proceso de reserva, cuando se confirmó la misma.



Para tener acceso a la creación y modicación de formularios dinámicos, debemos tener los permisos adecuados. Para eso, en el cliente de GXflow vamos a Management Console, seleccionamos Users y en la ventana de Usuarios presionamos el botón Roles y marcamos Gxflow Form Designer.

Nos volvemos a loguear y vemos que en el Navegador, bajo estadísticas ahora hay un grupo llamado Dynamic Forms que tiene 3 componentes: las **definiciones de los formularios** dinámicos que hagamos, los **elementos** de los formularios definidos y los **dominios** que se utilicen en la definición de los elementos de los formularios.

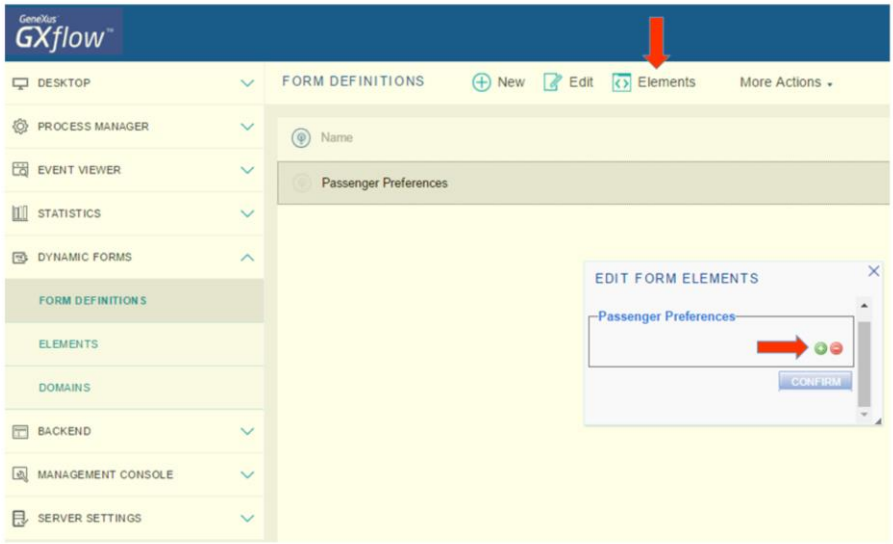
Creation of dynamic forms



Para crear un formulario dinámico, damos clic en Form Definitions y presionamos el botón de New.

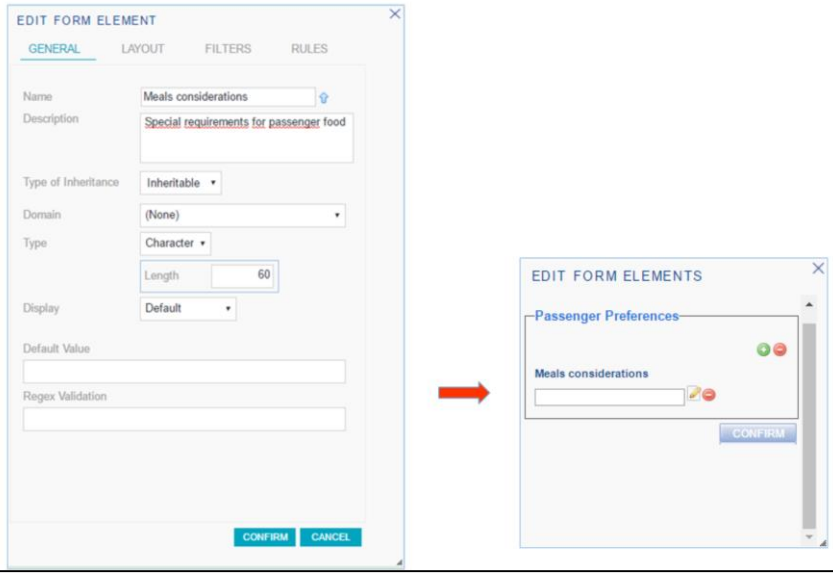
Le damos el nombre **Passenger Preferences** y presionamos Confirmar.

Creation of dynamic form elements



Para definir los elementos que incluiremos dentro del formulario, seleccionamos el formulario Passenger Preferences y presionamos el botón Elements. Vemos que se abre la ventana del formulario, movemos el mouse dentro del marco y presionamos el botón de +.

Creation of a text element



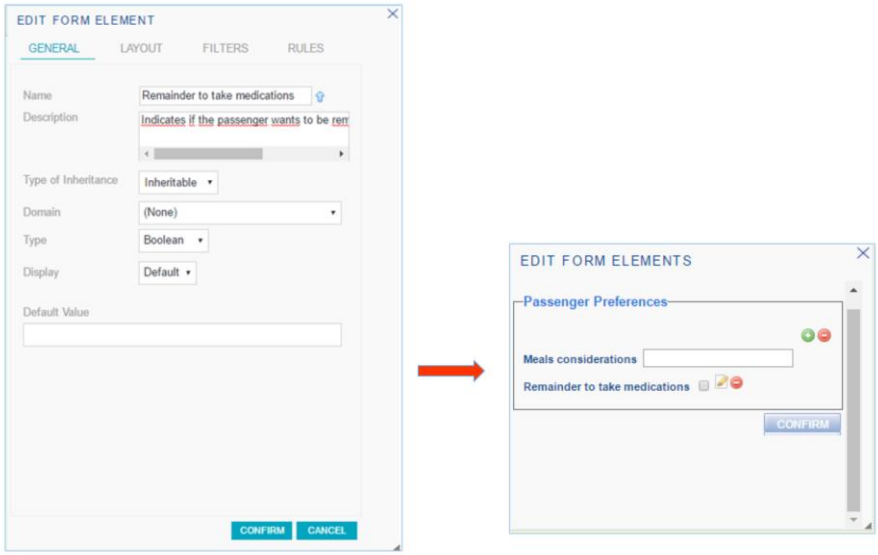
Se abre un cuadro de diálogo para ingresar el element.

En este elemento almacenaremos los requerimientos del cliente acerca de sus comidas, por lo que ingresamos el nombre “Meals considerations”, y en la descripción ingresamos: “Special requirements for passenger food”

Modificamos el largo del tipo Character a 60 caracteres

Dejamos el resto de los campos con su valor por defecto, y presionamos Confirmar.

Creation of a Boolean element

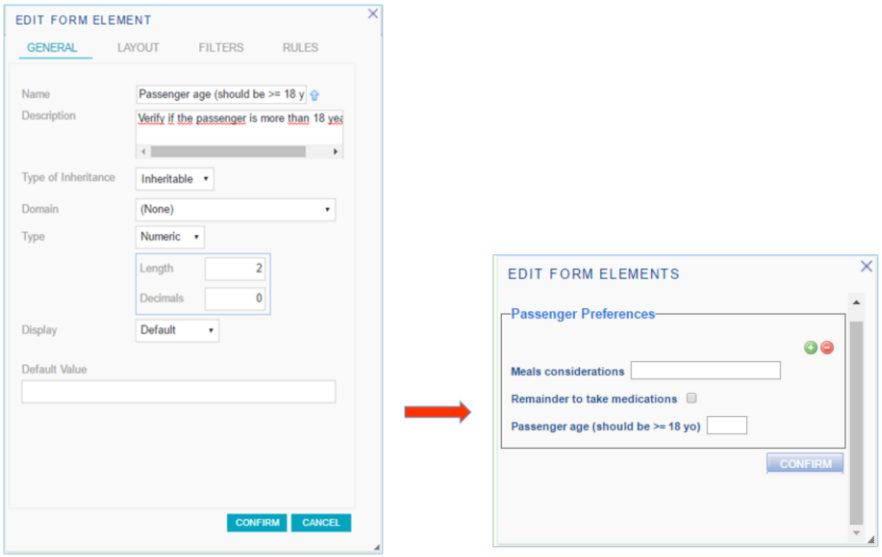


Presionamos de nuevo el signo "+" e ingresamos un nuevo elemento para registrar que el pasajero desea que se le recuerde la toma de medicamentos.

Para el nombre escribimos: " Reminder to take medications " y para la descripción: Indicates if the passenger wants to be reminded to take medicines."

Dejamos el valor del tipo en Booleano y presionamos Confirmar.

Creation of a numeric element



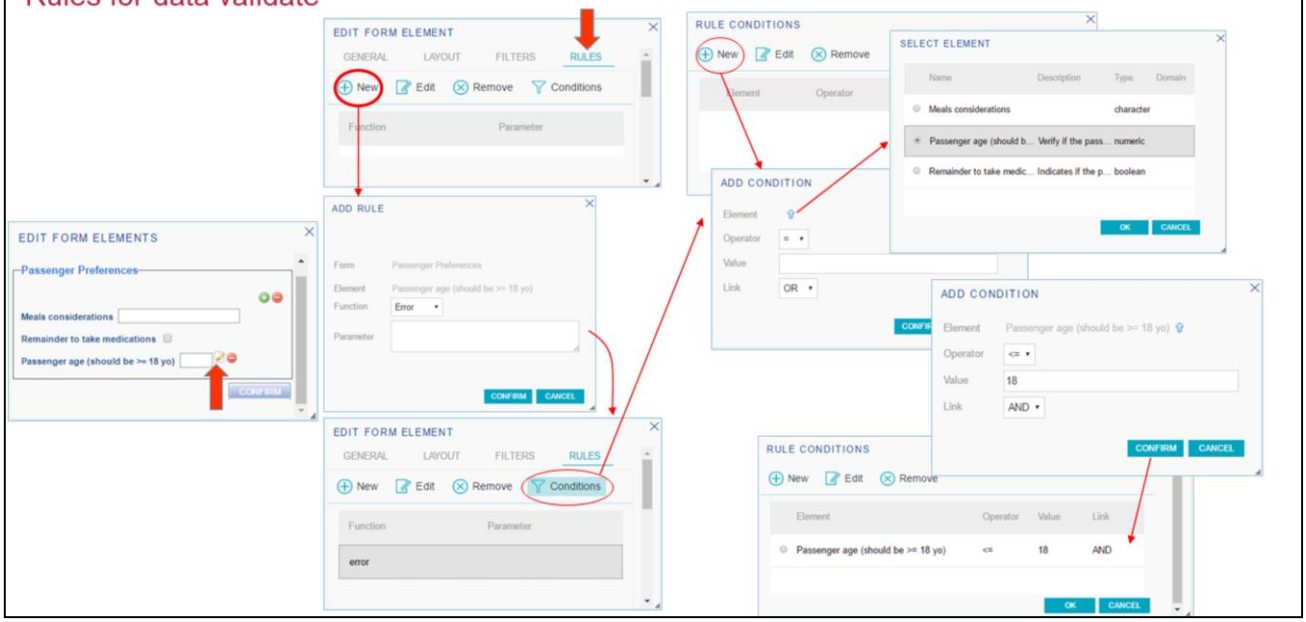
Por último creamos un elemento más para almacenar la edad del pasajero.

Al nombre le ponemos “Passenger age (should >= 18 years old)”, y en la descripción: “Verify if passenger is over 18 years old”.

Seleccionamos el tipo Numeric, asignamos 2 caracteres de largo, sin decimales.

Presionamos Confirm.

Rules for data validate

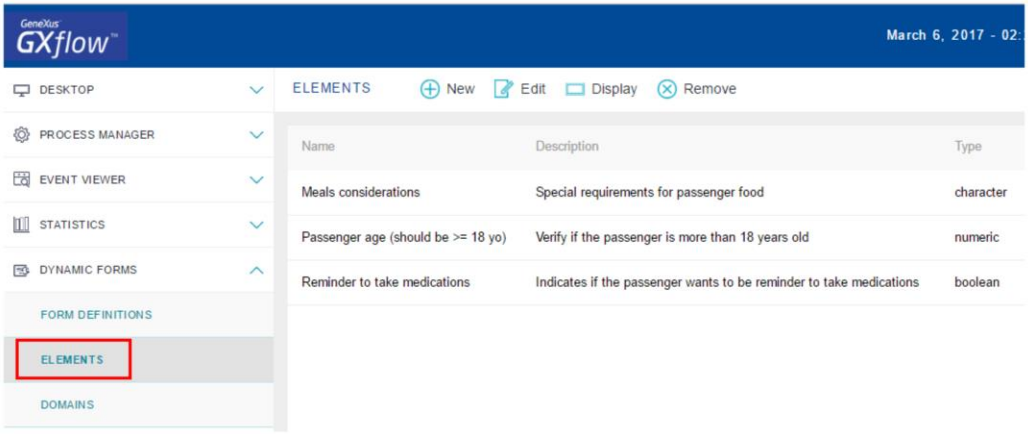


Vamos a agregar una regla para que se controle automáticamente que no pueda ingresarse un pasajero con edad inferior a 18 años.

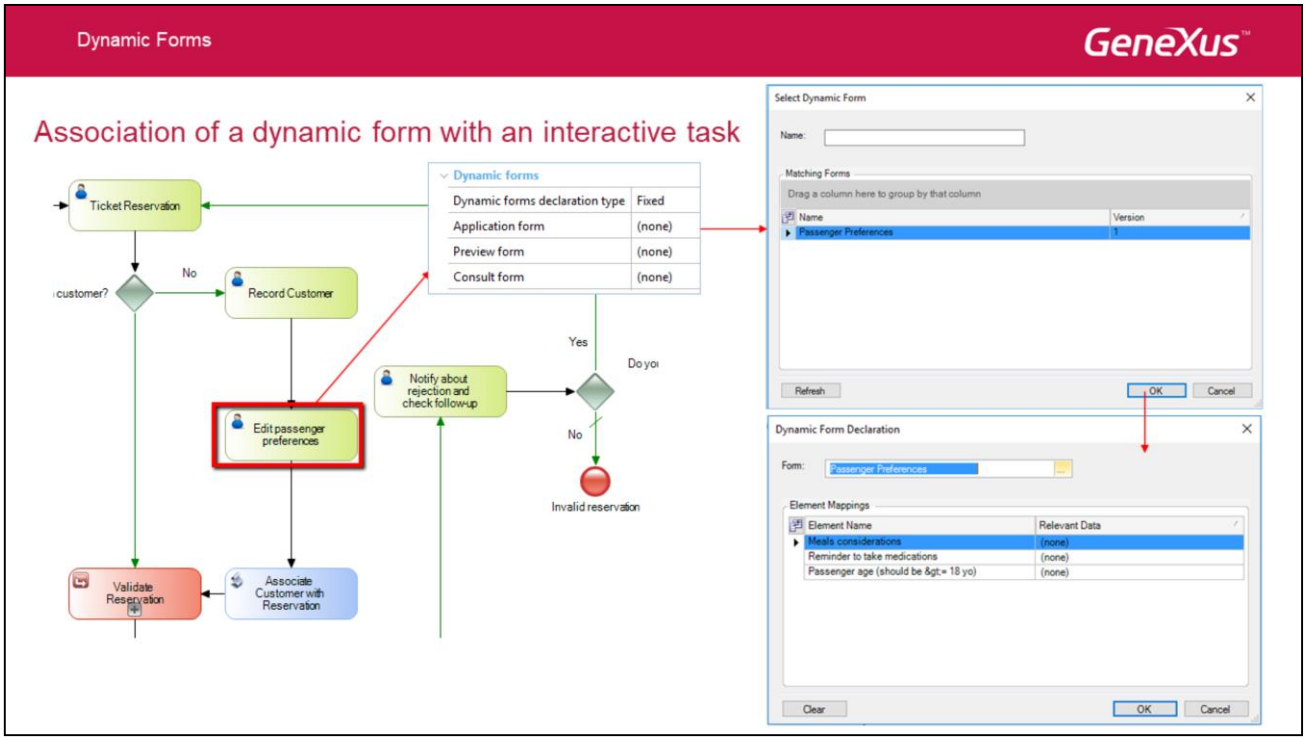
Para eso, pasamos el mouse sobre el elemento de la edad y elegimos Editar. En la solapa Rules, presionamos New, luego elegimos una regla del tipo Error.

Luego seleccionamos la regla y le agregamos una condición, en base al elemento de la edad del pasajero, con el operador <=, el valor 18, el link AND y confirmamos.

Verifying form elements

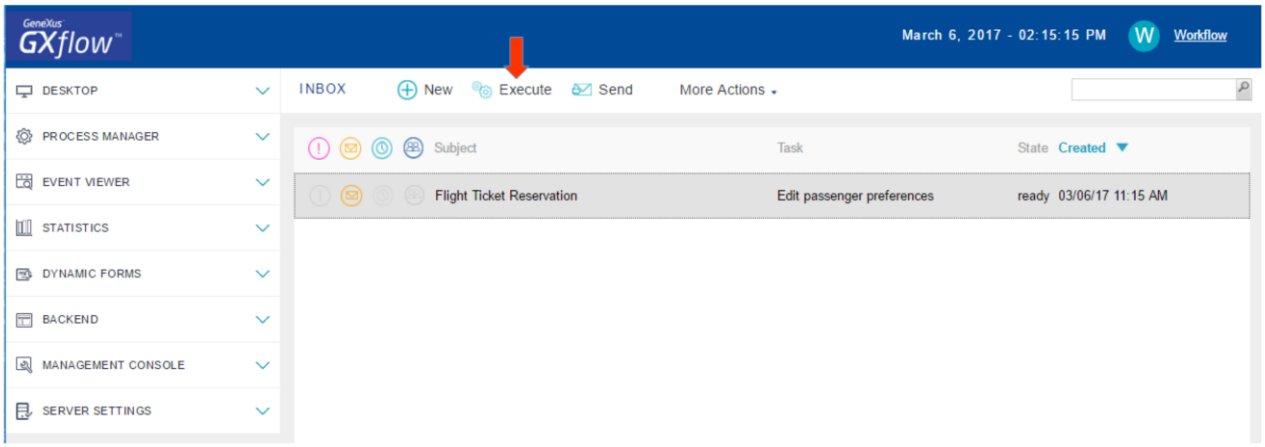


En la sección Elements podemos ver las definiciones que acabamos de hacer.



En las propiedades de la tarea, vamos al grupo **Dynamic Forms** y presionamos el botón de la propiedad **Application form**. Ahora, presionamos ahora el botón en Forms, seleccionamos al formulario Passenger Preferences y presionamos OK.

Execution of a dynamic form

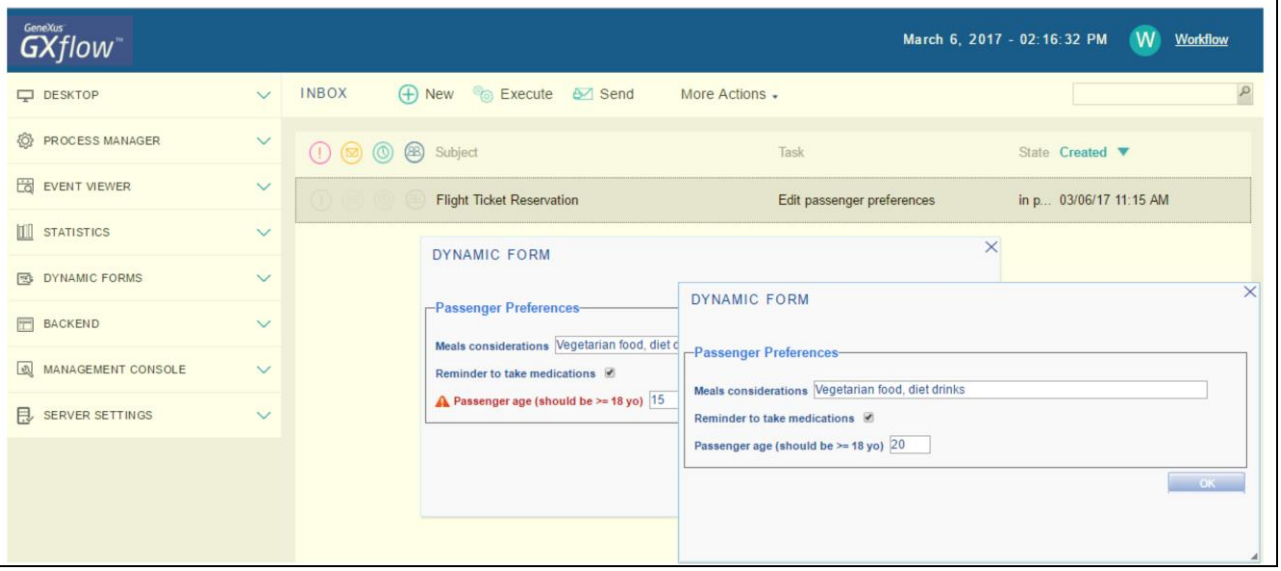


Para ejecutar el diagrama, hacemos click derecho y seleccionamos Run.

Cuando el Gxflow client se abre, podemos ver que la tarea TicketReservation del proceso FlightTicketReservationDynForm aparece como pendiente. Entonces, seleccionamos la tarea y ejecutamos. Para continuar con la ejecución, ingresamos una reservación seleccionado un cliente…y ahora marcamos la reservación como disponible.

Vemos que la siguiente tarea es **Enter passenger preferences**, entonces la ejecutamos.

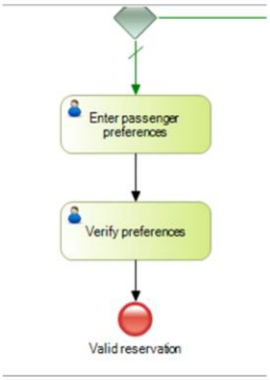
Execution of a dynamic form



En Meals considerations ingresamos: “Vegetarian food, diet drinks” marcamos el recordatorio para tomar medicamentos e ingresamos una edad de 15. Presionamos Confirmar.

Vemos que la regla se dispara, dándonos una alerta que el valor ingresado para la edad no es válido, ya que la misma debe ser mayor que 18. Si ingresamos un valor mayor a 18 (p.ej.20) al presionar confirmar el formulario se cierra.

Verification of stored data



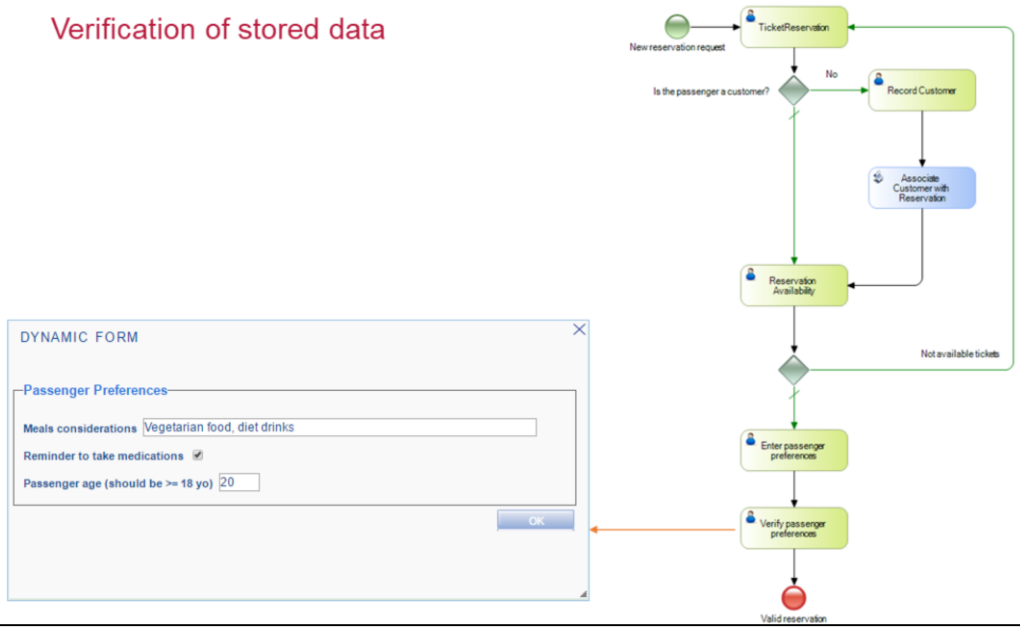
Task: Verify preferences	
Name	**Verify preferences**
Type	**User**
Application	(none)
SD Application	(none)
Visible in history	True
Roles	
Subject rule	
Advanced Properties	
Optional routing type	Branch
Request comments	False
Work with documents	False
Send e-mail	False
Strong synchronization	False
Skip predecessors	True
Enable deferred completion	True
Consult application	(none)
Consult SD application	(none)
Preview application	(none)
Preview SD application	(none)
Metadata	
Dynamic forms	
Dynamic forms declaration type	Fixed
Application form	(none)
Preview form	(none)
Consult form	(none)

Los datos que hemos almacenado, quedaron guardados en las tablas de workflow y estarán disponibles dentro de la instancia del proceso.

En nuestro ejemplo, el proceso justo termina después de ingresar estos datos, pero si el proceso continuara y hubiera otra tarea asociada al mismo formulario, el mismo se abriría con los datos previamente ingresados.

Para probar esto, modificamos el diagrama de proceso y agregamos una tarea nueva entre la tarea **Enter passenger preferences** y el None End Event, con el nombre **Verify preferences**. Asociamos la tarea que creamos recién al formulario dinámico **Passenger preferences** y ejecutamos el diagrama de procesos.

Verification of stored data



Cuando llegamos a la ejecución de la tarea **Verify passenger preferences**, al abrirse el formulario asociado, todos los campos contienen los datos que ingresamos previamente cuando se ejecutó la tarea para ingresar las preferencias del usuario.

Los datos ingresados en el formulario, quedaron almacenados en la base de datos de Workflow y estarán disponibles en toda la instancia del proceso.

More information about dynamic forms

http://wiki.genexus.com/commwiki/servlet/hwikibypageid?25809

Module 16

Independent DataStore for Workflow tables

GeneXus

Independent DataStore for Workflow tables

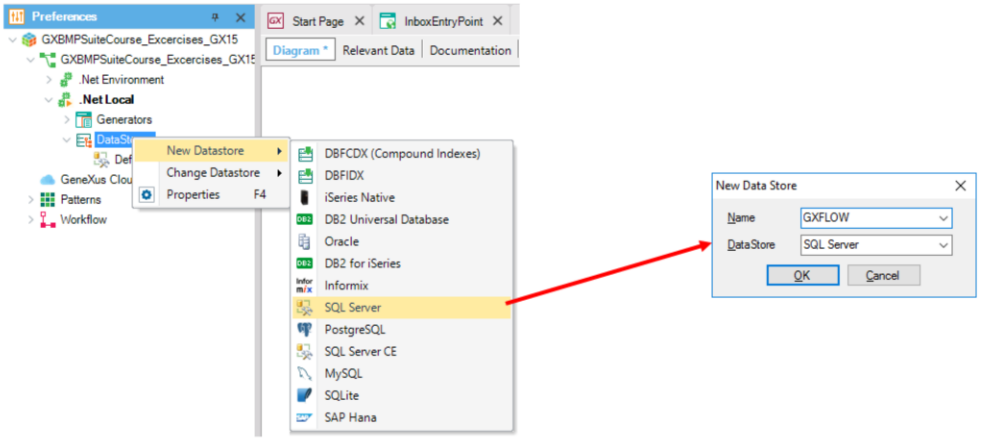
Workflow Data Store

Por defecto, las tablas de workflow se crean en el mismo data store donde se almacenan las tablas de la aplicación.

Pero es posible almacenar esas tablas en un data store independiente. Esto mejora la seguridad de los datos de la aplicación y de los datos de workflow, ya que es posible fijar restricciones de acceso en forma independiente para cada tipo de tabla.

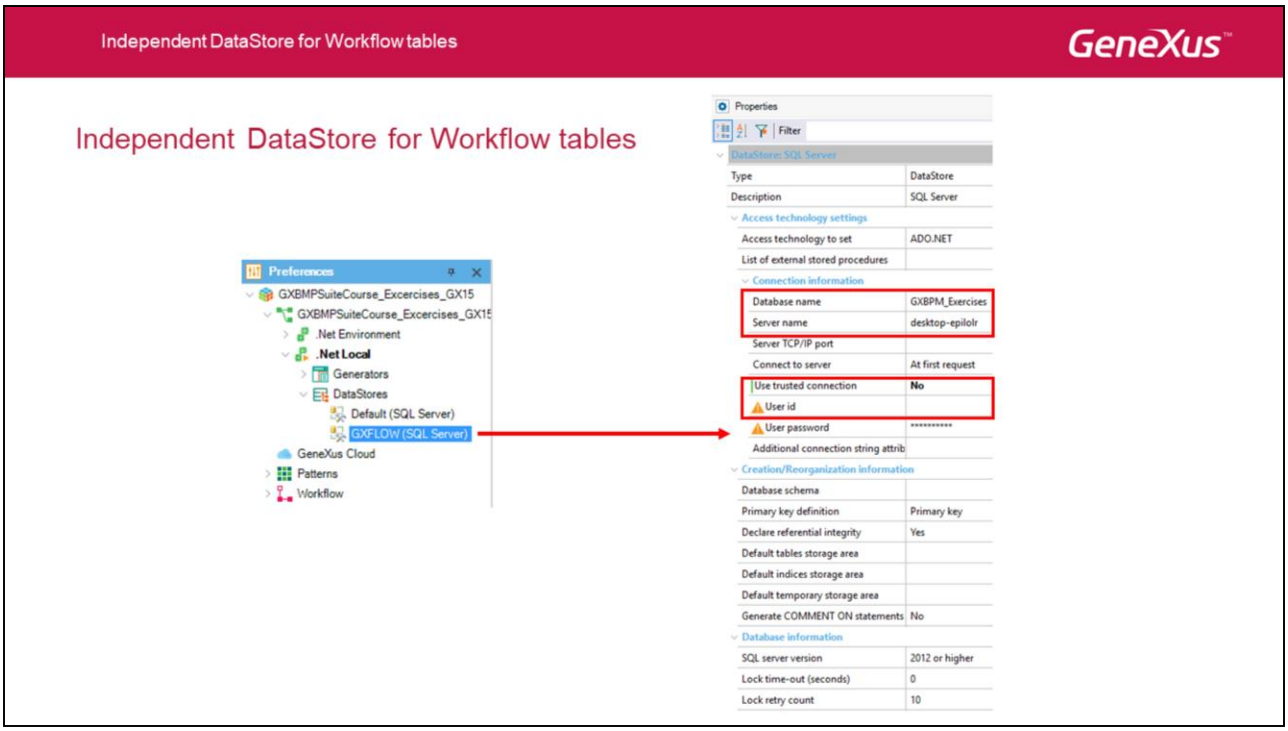
GeneXus

Independent DataStore for Workflow tables



Lo primero que hay que hacer es crear un data store nuevo y el mismo debe llamarse GXFLOW.

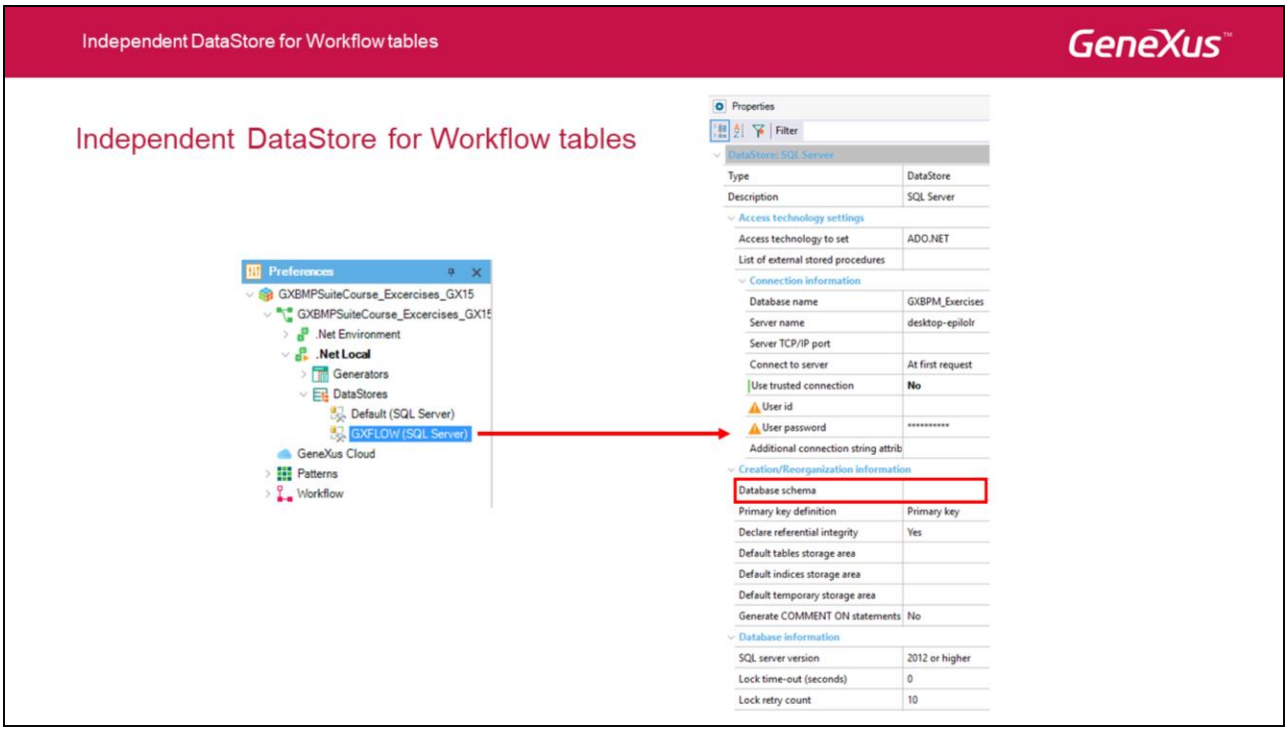
Es importante respetar este nombre, de otra forma GeneXus no reconocerá éste como el data store donde debe guardar las tablas de workflow y las continuará guardando en el default data store.



Una vez creado, es posible cambiar el valor de las propiedades Database name, Server name, User ID y User Password.

Esto permite que la base de datos donde se almacenan las tablas de workflow sea totalmente independiente de la base de datos de la aplicación.

En este caso las operaciones sobre cada base de datos son independientes y no es posible integrar operaciones sobre distintas bases de datos en una misma unidad de trabajo lógica (UTL)



Otra posibilidad es que la base de datos sea la misma, pero las tablas de workflow estén asignadas a un **esquema** diferente.

En este caso no se cambian los valores por defecto de las propiedades de conexión (nombre de la base, servidor, usuario y password) y se asigna un valor diferente al esquema.

Para asignar las tablas de workflow a un esquema diferente, se utiliza la propiedad **Database schema**.

Como resultado, el motor de GxFlow realiza todas las operaciones en el default data store y estas operaciones integran las mismas unidades de trabajo lógicas que el resto de la aplicación.

GeneXus 15

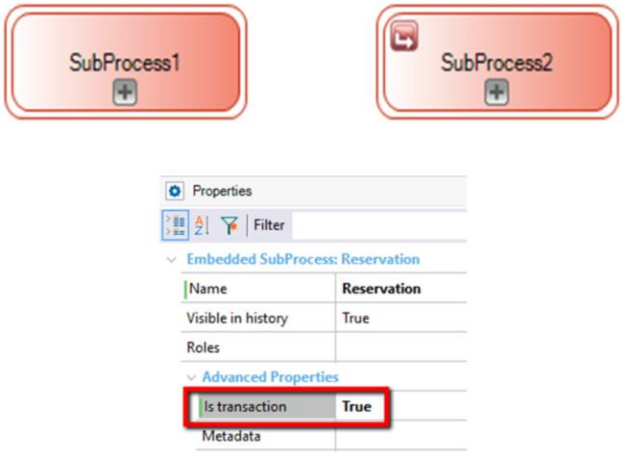
Module 17

Transactional Subprocesses

Hay procesos en los cuales es necesario coordinar varias actividades que necesitan cumplirse exitosamente todas ellas para que el flujo del proceso pueda seguir y en caso de que alguna no se cumpla satisfactoriamente, es necesario regresar a todas ellas a su estado inicial.

Estas actividades integran lo que denominamos una unidad de trabajo lógica (UTL), es decir una unidad atómica que debe ejecutarse en forma indivisible. Una UTL en un proceso de negocios podría durar días o incluso semanas.

Transactional Subprocesses



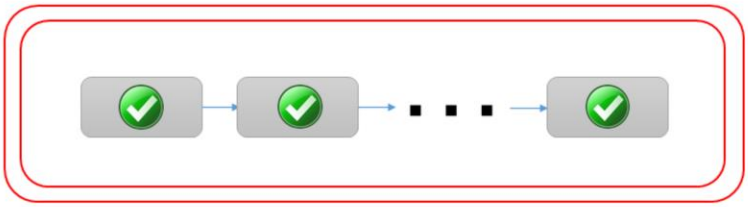
Para modelar estos escenarios de negocio con transacciones, donde cada transacción está compuesta de varias operaciones que deben realizarse y terminarse correctamente todas ellas, o bien ninguna de ellas, utilizamos a los subprocesos transaccionales.

En BPMN se representan con un borde doble.

En GeneXus para definir un subproceso como transaccional, asignamos la propiedad **Is transaction** con el valor True. Al hacerlo, veremos que cambia el borde del símbolo de subproceso y se representa con un borde doble.

Possible results in a transaction

1) All operations conclude satisfactorily



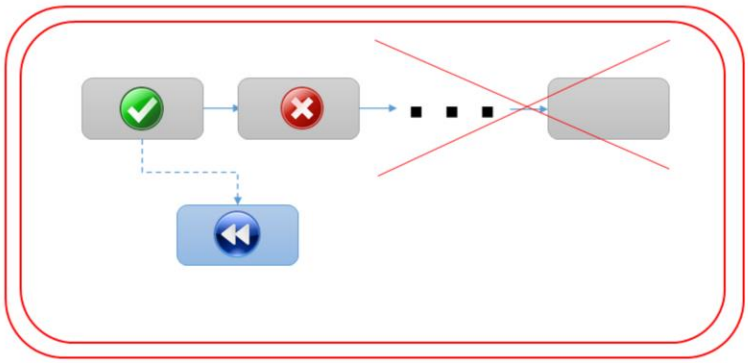
Un subproceso transaccional finaliza satisfactoriamente cuando los cambios de los datos en la base de datos terminan en un estado consistente. En el caso de que se produzca un fallo en la ejecución de una transacción, es necesario iniciar acciones que deshagan los cambios, llevando los datos al estado que tenían antes de estos cambios.

Las transacciones en un modelo de procesos de negocios pueden tener tres resultados posibles:

- Que todas las operaciones **finalicen satisfactoriamente**, con lo cual el proceso continúa por el flujo normal

Possible results in a transaction

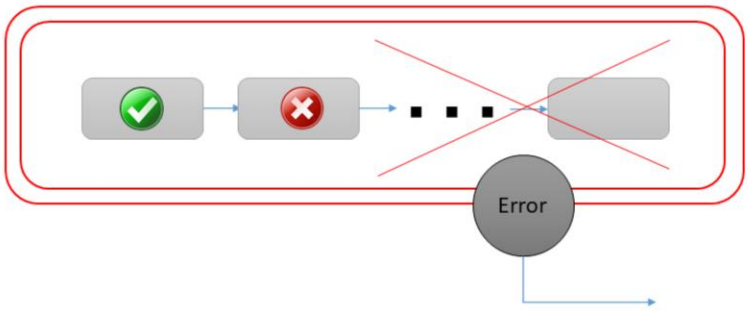
2) A failure occurs: we must revert tasks already concluded



- Que **ocurra una falla** y es necesario revertir las actividades que ya fueron completadas dentro del subproceso. En este caso deben ejecutarse ciertas tareas de compensación que se encargan de dejar al proceso en el estado inicial existente antes de iniciarse el subproceso transaccional.

Possible results in a transaction

3) An unexpected error occurs: no compensation takes place

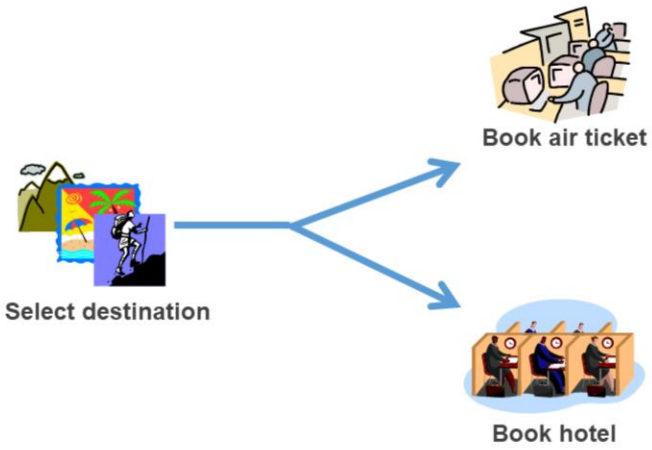


- Que **suceda un error inesperado**, en cuyo caso las actividades del subproceso son interrumpidas y no se ejecuta ninguna actividad compensatoria. En este caso el proceso continúa con la ejecución de un evento intermedio de error.

Por lo tanto, para modelar un subproceso transaccional, es necesario capturar eventos que respondan a estas situaciones, es decir eventos de error y de cancelación de procesos debido a una falla.

Veremos a continuación los conceptos mencionados hasta aquí, en base a un ejemplo.

Example: Coordinating a trip



En la Agencia de Viajes en la cual hemos venido trabajando, la coordinación de un viaje podría ser un ejemplo de un subproceso transaccional, ya que para poder completarse correctamente hay que poder completar exitosamente varias actividades, como podrían ser la reserva del pasaje aéreo, la reserva del hotel, el alquiler de coche y la entrada a una atracción turística, entre otras.

Consideremos un proceso de reserva que consiste de dos actividades: la reserva de un vuelo y la reserva de una habitación de hotel.

Consideremos que ambas actividades se ejecutan en una única transacción, es decir que si no se consigue el pasaje aéreo y sí se consiguió la reserva del hotel, es necesario deshacer la reserva del hotel, y viceversa, si se consiguió el pasaje pero no el hotel, debe cancelarse la reserva del pasaje.

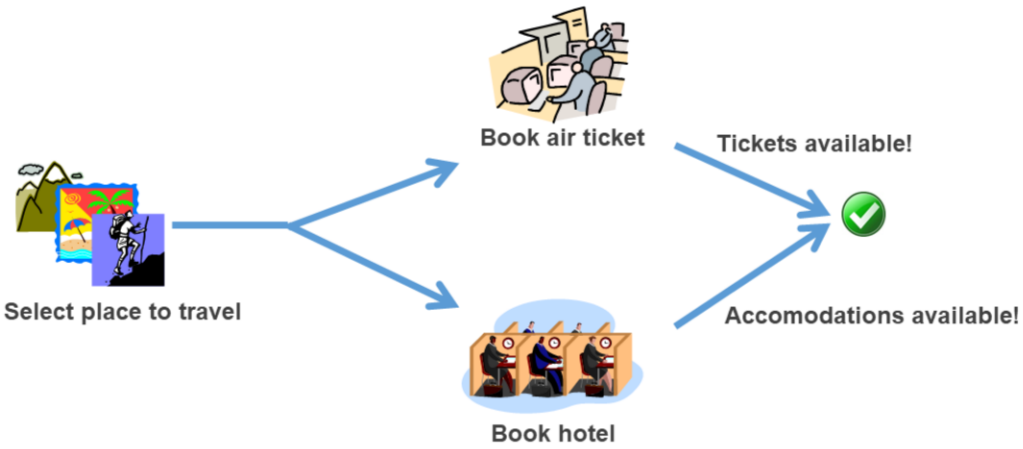
Esto implica que debemos considerar los tres casos antes vistos:

Que consigamos ambas reservas y el proceso finalice normalmente.

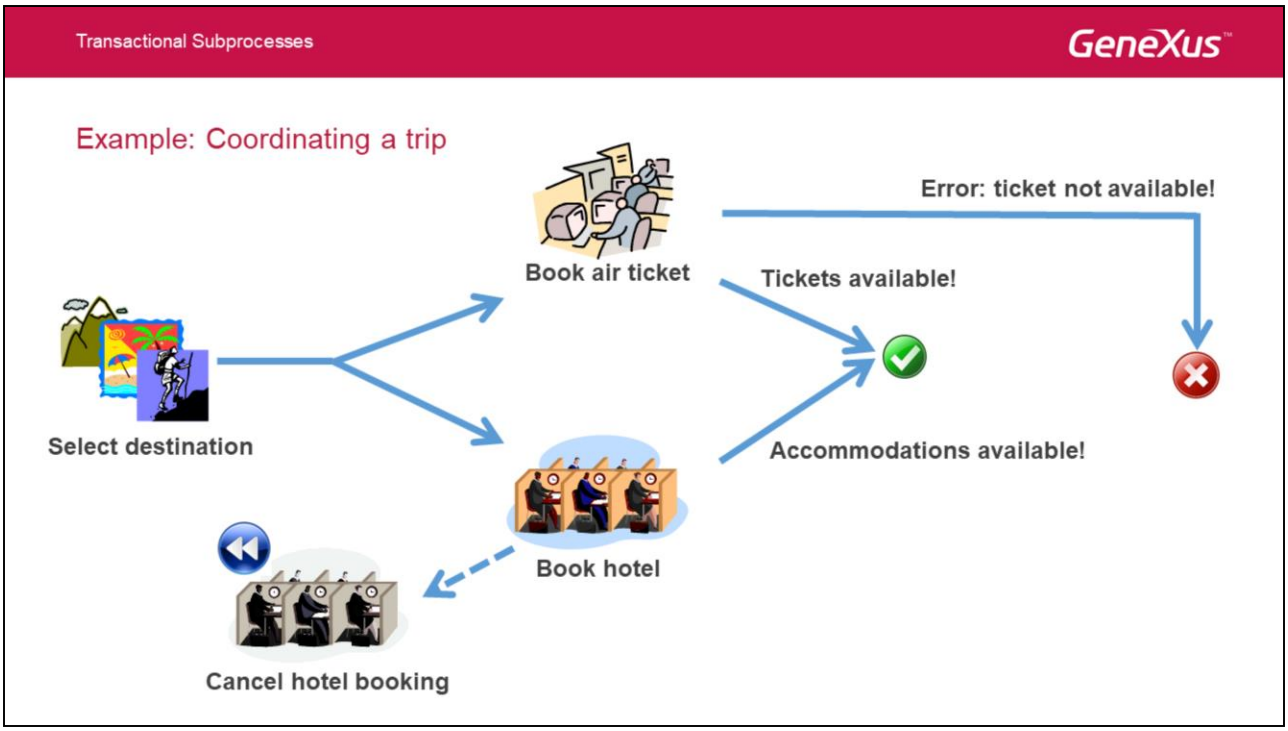
Que falle alguna de las reservas y el proceso se cancele, por lo que deberá deshacerse la reserva que se haya obtenido, ya que deben obtenerse ambas o ninguna.

Que suceda un error inesperado y el proceso termine.

Example: Coordinating a trip

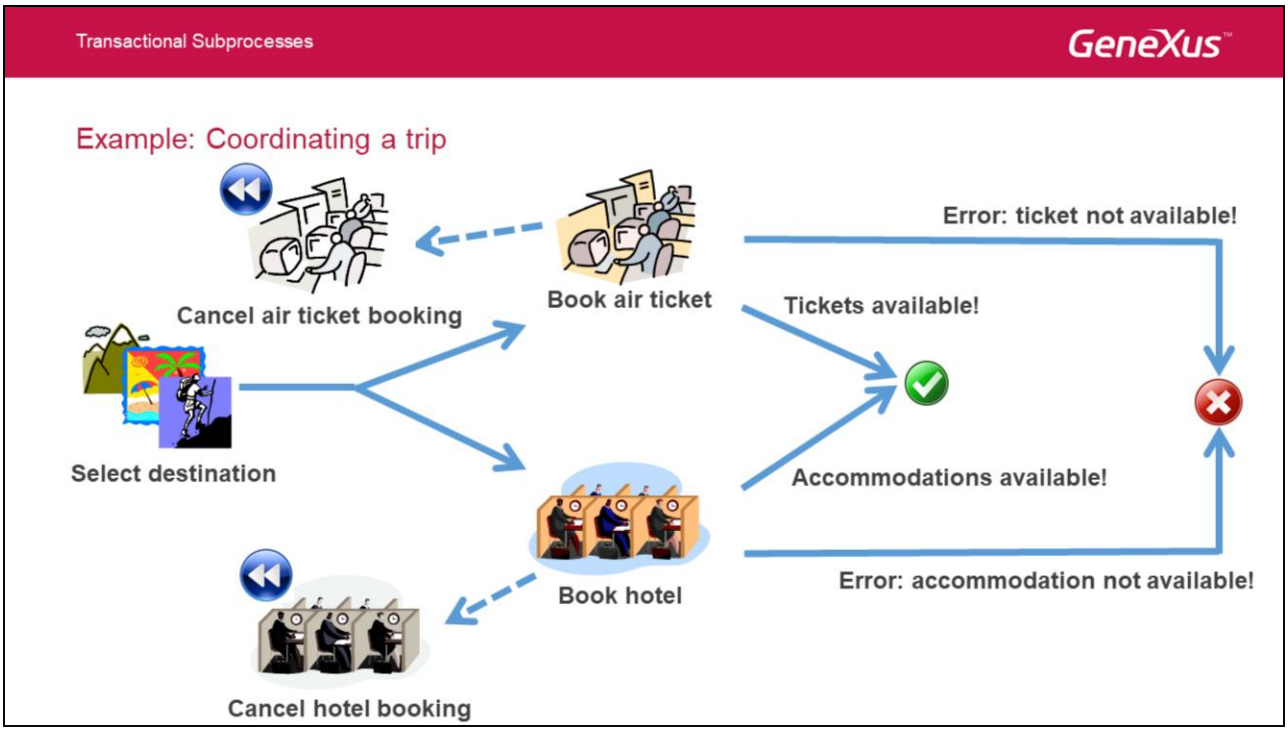


En el caso de que ambas reservas se obtengan satisfactoriamente, el proceso finaliza.



Pero si falla la obtención del pasaje aéreo y se logró finalizar la reserva del hotel, el proceso no puede darse por completado porque una de las actividades de la transacción no se cumplió, por lo que es necesario deshacer las actividades que llegaron a cumplirse.

En este caso sería necesario ejecutar un proceso compensatorio que deshaga la reserva del hotel, de modo que el proceso de coordinación de viaje queda completamente sin efectuar.



Si por el contrario, la reserva del ticket aéreo hubiera sido exitosa y no fue posible obtener lugar en el hotel, es necesario deshacer la reserva del pasaje.

Las tareas de cancelar las reservas de hotel o de los pasajes se conocen como actividades de compensación y muchas veces son realizadas por un sistema externo.

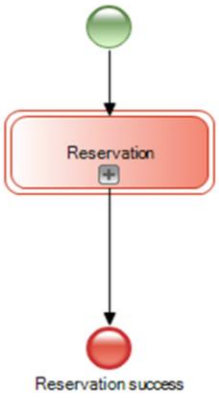
Hasta ahora hemos visto el caso de que el proceso transaccional finalice exitosamente porque todas las actividades que lo integran finalizaron correctamente, o el caso de que se produzca una falla en alguna de las actividades del proceso transaccional y esta falla produce una cancelación del proceso, siendo necesario deshacer las actividades que llegaron a completarse de modo de que el proceso se revierta hasta el estado anterior al comienzo del mismo.

El tercer caso tiene que ver cuando se produce un error que no puede ser manejado por el subproceso y evita que el mismo continúe, como por ejemplo que un servidor no responda o que se produzca una caída del sistema. En este caso las actividades del subproceso son interrumpidas sin compensación y la base de datos hace rollback hasta el último commit anterior al comienzo del subproceso.

En nuestro ejemplo, cuando el proceso de reserva no finaliza exitosamente, se detecta esa situación y se le comunica al cliente que no es posible realizar la reserva.

Veamos como modelamos este proceso con GeneXus BPM Suite.

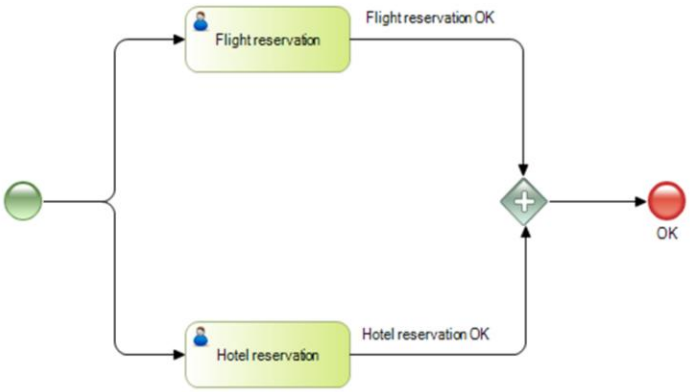
Modeling in GeneXus



En primer lugar ejecutamos el IDE de GeneXus o el GeneXus Business Process Modeler y en nuestra Knowledge Base, creamos un objeto Business Process Diagram.

Arrastramos desde la toolbar un None Start Event, un símbolo de subproceso al que conectaremos desde el Start Event y un None End Event al que conectamos desde el subproceso. En las propiedades del subproceso, asignaremos a la propiedad **Is transaction** el valor **True.** Esto hará que el símbolo del subproceso cambie y se vea con doble borde, que es la forma en que en BPMN se representa a un subprocesso transaccional.

Modeling in GeneXus

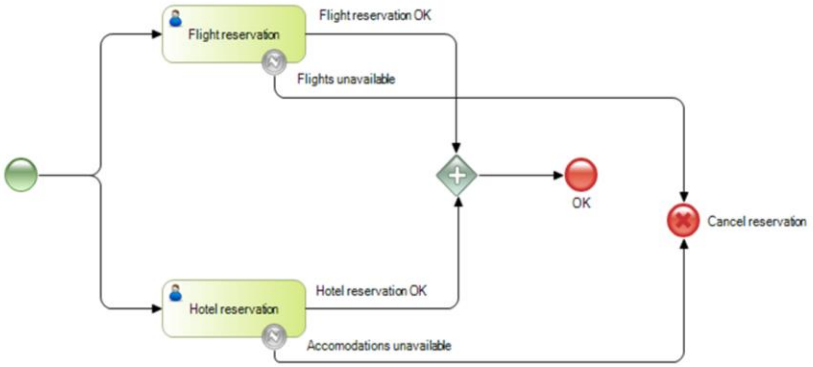


Ahora abrimos el subproceso, comenzamos agregando un None Start Event, luego insertamos 2 tareas interactivas, una para la reserva de vuelos y otra para la reserva del hotel y las conectamos desde el evento de Start. La salida de ambas las unimos en un Parallel Gateway, y la salida del mismo la conectamos a un None End Event.

Con esto estamos modelando el caso de que ambas tareas finalicen correctamente, en cuyo caso el Parallel Gateway podrá sincronizar ambos flujos que le llegan y el flujo saliente seguirá hasta el End event. Sin embargo es necesario tomar en cuenta los otros casos, ya que si uno de las dos tareas falla, el flujo jamás podrá avanzar debido al Parallel Gateway.

GeneXus

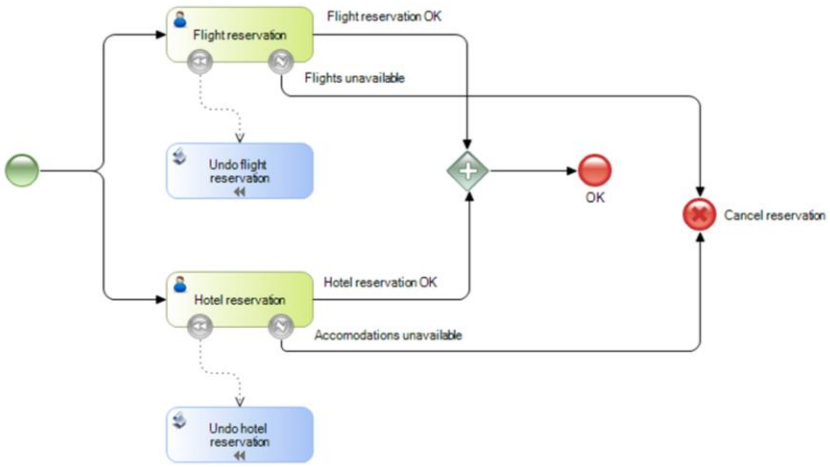
Modeling in GeneXus



Para tomar en cuenta las posibles fallas de las reservas, agregamos sendos Intermediate Error Event a cada tarea y las salida de los mismos las conectamos a un Cancel End Event.

Este evento Cancel se disparará cuando alguna de las dos tareas interactivas falle. El evento Cancel es un evento especial utilizado en los procesos transaccionales, que se dispara **durante** la ejecución del subproceso, en lugar de dispararse al finalizar el mismo.

Modeling in GeneXus



Cuando la transacción se cancela, antes de que pase el control al proceso padre, se disparan las **tareas de compensación**. Esto es que todas las actividades que terminaron exitosamente hasta ese momento, deben ser deshechas ejecutando las tareas de compensación definidas para cada una de ellas.

En nuestro caso, debemos definir tareas que deshagan la reserva del pasaje o la reserva del hotel, según el caso.

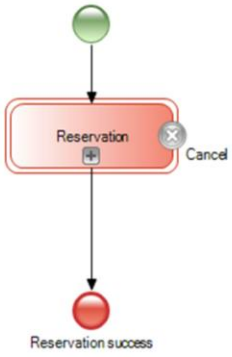
Para esto agregamos un Intermediate Cancel Event adjunto a la tarea de reserva de vuelo, insertamos una tarea script con el nombre Undo flight reservation y la unimos desde el Intermediate Cancel Event.

Repetimos lo mismo para la tarea de reserva de hotel, insertando otro Intermediate Cancel Event y una tarea no interactiva con el nombre Undo hotel reservation, que conectamos desde el Intermediate Cancel.

Observamos que desde el momento que conectamos las tareas script con los Intermediate Cancel Event, cambia el aspecto de la conexión ya que ahora el conector es del tipo Association y aparece sobre las tareas un símbolo “<<”. Este tipo de conector se representa con una línea punteada y permite establecer una relación diferente a la de los conectores de secuencia ya que las actividades asociadas no pueden ser parte de ninguna secuencia de flujo, es decir, no pueden tener conectores de secuencia de entrada o salida.

GeneXus

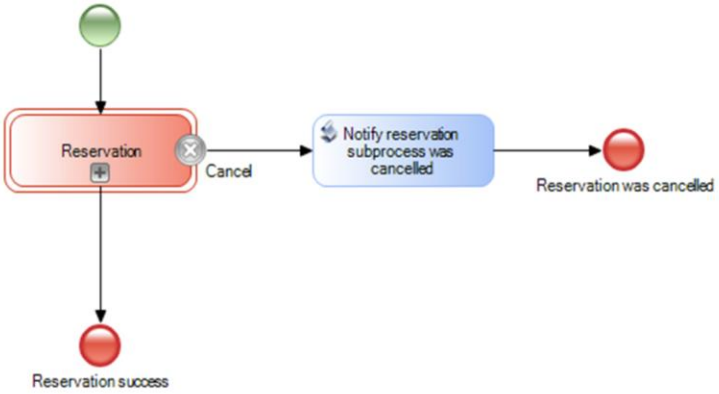
Modeling in GeneXus



Una vez que la compensación de la transacción se completa, se interrumpe la ejecución en el subproceso y se dispara un evento de cancelación desde el subproceso al proceso padre.

Para capturar este evento desde el proceso padre, debemos adjuntar un Cancel Intermediate Event al símbolo del subproceso.

Modeling in GeneXus



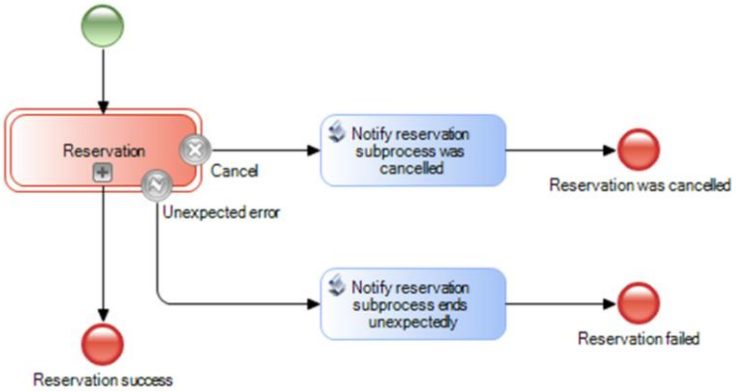
Al recibirse la cancelación en el proceso padre, comunicamos al usuario que el proceso de reserva fue cancelado y terminamos el proceso.

Al igual que vimos anteriormente con el evento Error, el Cancel utiliza un mecanismo de throw-catch desde el Cancel End Event del subproceso, hacia el Cancel Intermediate Event adjunto al subproceso, en el diagrama padre.

Este evento cancel dispara automáticamente las tareas de compensación que hayan sido definidas en el subproceso, asegurando la integridad transaccional del mismo.

Cualquier otro tipo de interrupción que sufra el subproceso como un evento de error, se aborta la ejecución del mismo sin ninguna compensación.

Modeling in GeneXus



http://wiki.genexus.com/commwiki/servlet/hwikibypageid?24922

A fin de proveer un manejo de este error, insertamos un Intermediate Error Event adjunto al símbolo del subproceso y mediante una tarea script enviamos un aviso de que el proceso de reserva finalizó en forma inesperada.

De esta forma hemos visto el funcionamiento de los subprocesos transaccionales y cómo los modelamos en GeneXus.

Puede encontrar más información, en el siguiente link:
http://wiki.genexus.com/commwiki/servlet/hwikibypageid?24922

Thank you!

Documentation:
genexus.com/gxflowdocs

GeneXus™ 15