Custom Client

En este vídeo analizaremos el Custom Client proporcionado por GeneXus

Custom Client es proporcionado por GeneXus y puede ser importado a la Knowledge Base para usarlo o modificarlo como parte de nuestro proyecto. Contiene todos los Webpanels y procedimientos necesarios para implementar muchas funcionalidades del GXflow Standard Client.

Utilizando el Custom Client, puedes ejecutar la bandeja de entrada del usuario, ver la bandeja de salida, administrar las definiciones de procesos, workitems, instancias e incluso monitorear la ejecución.

Si nos fijamos en el código fuente programado en estos Webpanels, podemos ver cómo se utilizan las APIS de GXflow, para conectarse al motor y obtener la información necesaria en cada caso, y ejecutar diferentes funciones a cada objeto.

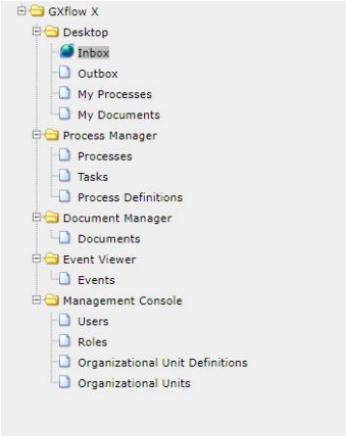
Contiene toda la funcionalidad del Custom Client. En este video vamos a ver algunos Web Panels para entender su código y cómo se pueden modificar.

Podemos personalizar el Custom Client por varias razones, por ejemplo podemos agregar u ocultar columnas en la bandeja de entrada o en cualquier otra pantalla. Podemos agregar acciones o personalizar algún código, cambiar la UX e integrar una parte o todo el Custom Client en nuestra aplicación.

Para ver más información sobre Custom Client podemos seguir el link que se muestra a continuación:

https://wiki.GeneXus.com/commwiki/servlet/wiki?11364,HowTo%3A+Customize+The+GXflow+Client

Custom Client contiene algunas carpetas, por ejemplo: Escritorio, Administración de Procesos, Administración de Documentos, Visor de eventos y una Consola de administración.



En el folder del escritorio contiene la bandeja de entrada, la bandeja de salida, mis procesos y mis documentos, brindan la misma información y acciones que el Standard Client.

El folder Administrador de procesos contiene las páginas del proceso, la tarea y las definiciones de proceso, que son las páginas más utilizadas de la sección del administrador de procesos.

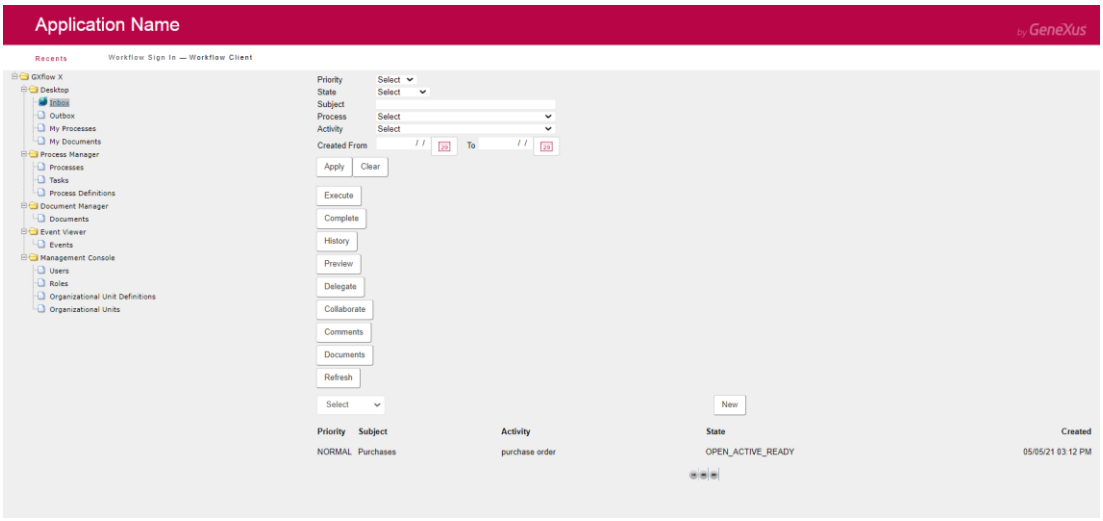
El folder administrador de documentos contiene administración de documentos al igual que el Standard Client.

El folder visor de eventos contiene la página del administrador de eventos al igual que el Standard Client.

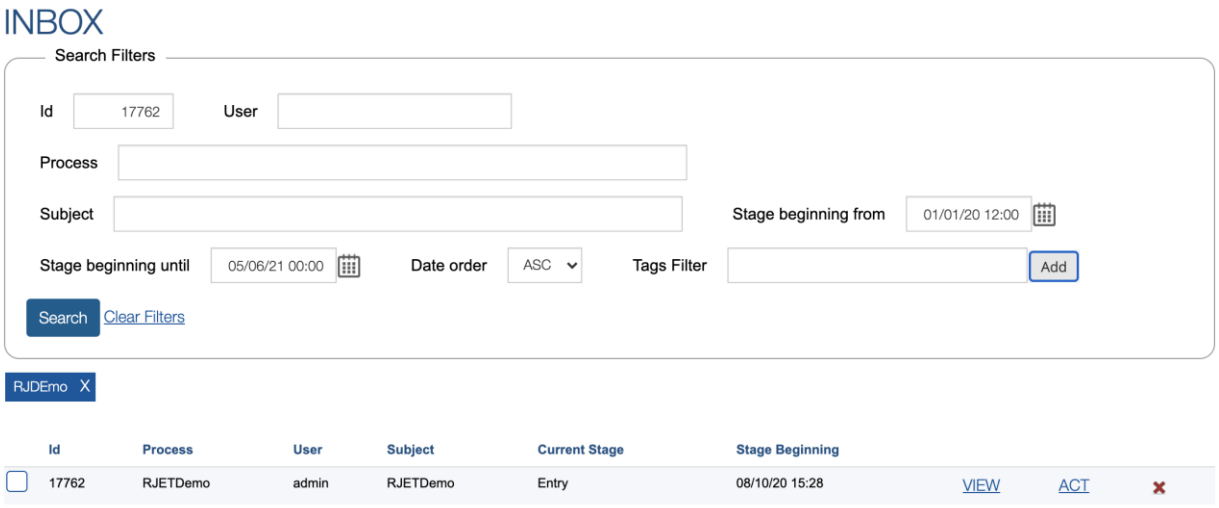
El folder consola de administración contiene la administración de usuarios, de roles, las definiciones de las unidades organizacionales y las unidades organizacionales, al igual que el Standard Client.

Las estadísticas, el backend, la configuración del servidor y el administrador de licencias no se implementan en el ejemplo de Custom Client.

El uso principal del Custom Client es incorporar los panels de GXflow a nuestros proyectos utilizando la misma UX. Por ejemplo, este panel presenta una bandeja de entrada personalizada con una apariencia y comportamiento que coincide con el resto de su aplicación. El usuario de nuestra aplicación no sentirá la diferencia con otras partes de la misma.



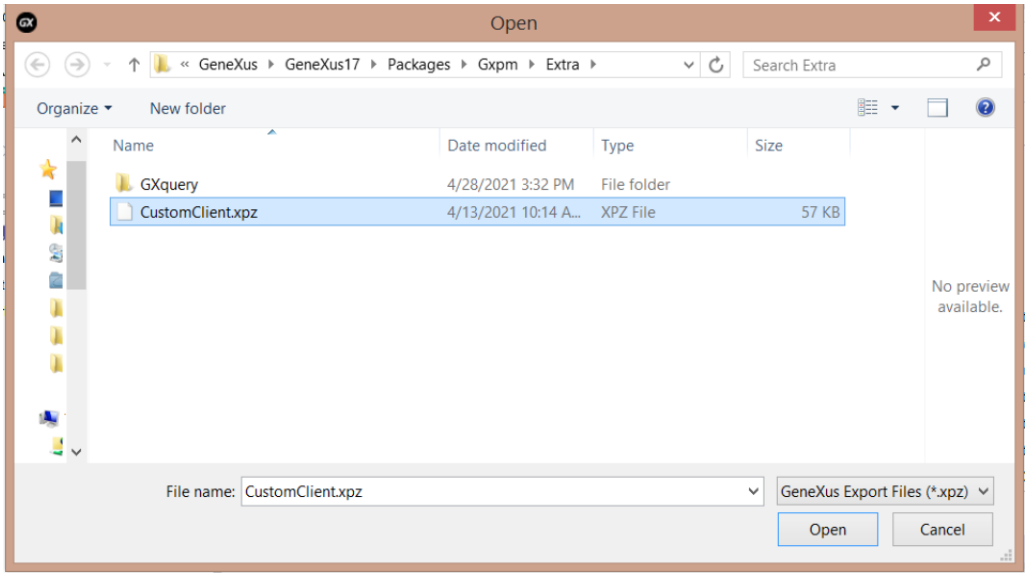
Custom Client Inbox



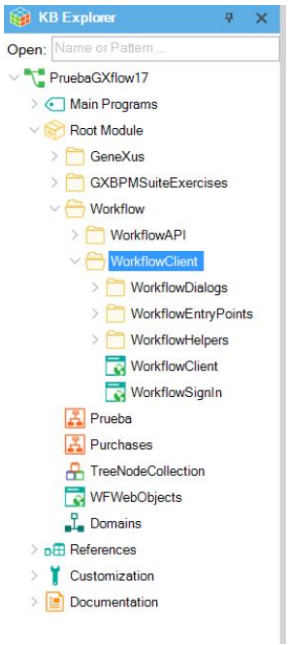
Customized Client Inbox

Existe un archivo llamado CustomClient.xpz y se distribuye con la instalación de GeneXus dentro de: <directorio de instalación de GeneXus> \ Packages \ GXPM \ Extra.

Antes de importarlo, compruebe que se haya creado al menos un Business Process Diagram en la Knowledge Base. Esto es necesario para asegurarse de que el modelo tenga los tipos de datos Workflow, de lo contrario, habrá un error durante la compilación del proyecto.



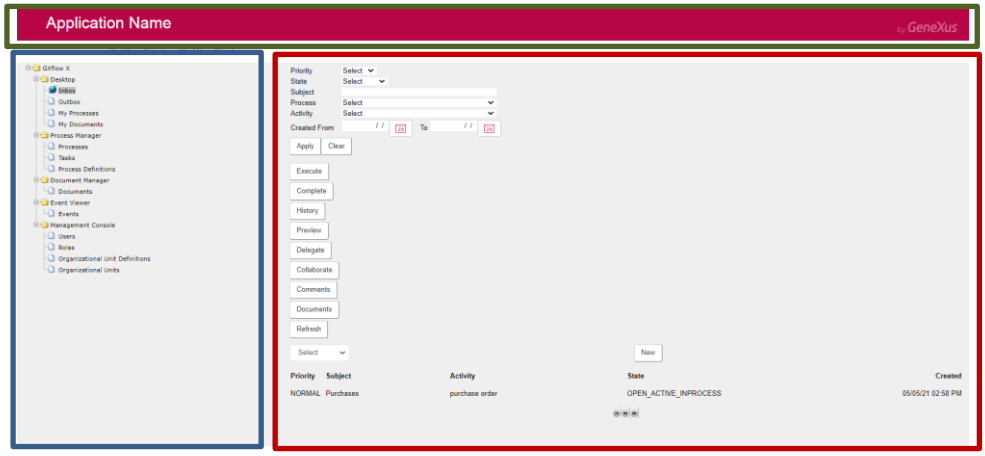
Al importar el xpz a nuestra Knowledge Base, podemos ver todos los Web panel, Procedimientos y SDTS que se van a incorporar a la base de conocimiento. Se encontrarán en la carpeta, Workflow Client dentro de la carpeta Workflow del Knowledge Base Explorer. Si se ejecuta el objeto principal se podrá ver el Custom Client en ejecución.



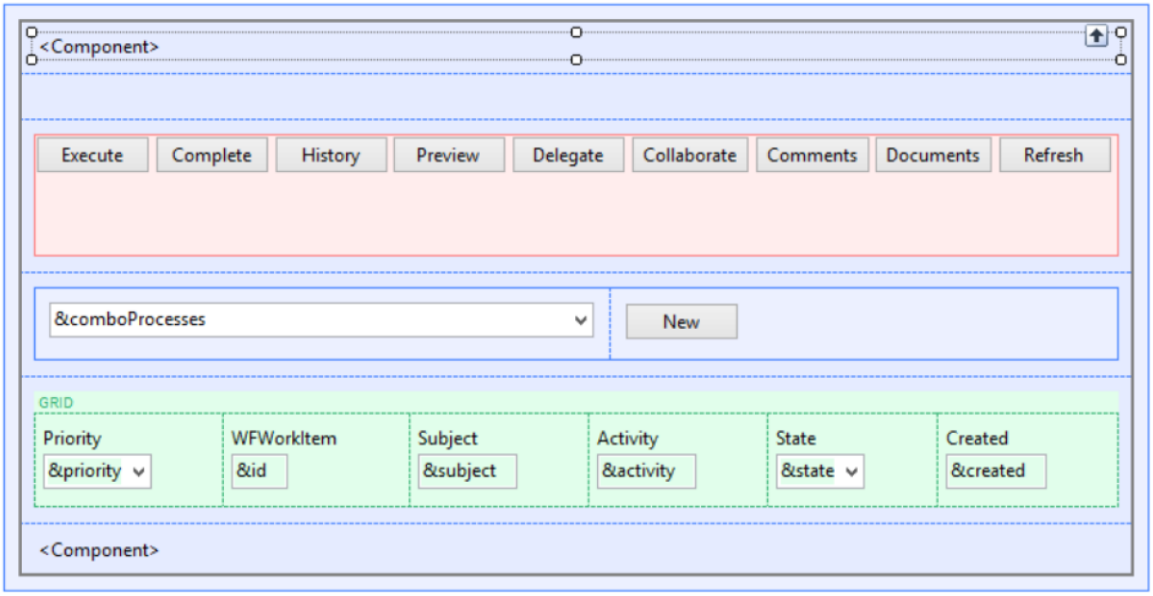
Primero tenemos que iniciar sesión, así que vamos a escribir el usuario de Workflow y luego su contraseña. Después de iniciar sesión, en el objeto principal del Workflow Client se dará clic en el enlace de la bandeja de entrada para ir a la misma, luego ahí se encuentran todos los filtros, las acciones y la grilla con todos los workitems que tenemos para ejecutar y completar. Comenzaré una nueva instancia seleccionando un proceso y luego clic en el botón nuevo. Aquí se crea el workitem y una instancia del proceso. Ahora ejecutaré este workitem el cual me solicita que cargue un documento, así que crearé un nuevo documento del tipo Doc y lo seleccionaré del sistema de archivos. Después de seleccionar mi archivo, he confirmado y se cargará aquí. Después de completar esta tarea, presiono el botón completar para crear los siguientes workitems en la definición del proceso. Puedo ver, por ejemplo, el historial del proceso que me muestra todos los elementos de trabajo que se han creado.

El objeto principal del Custom Client es el Webpanel llamado Workflow Client, como hemos visto durante la ejecución. Si hacemos clic en el link, la bandeja de entrada aparecerá en la parte derecha del Webpanel.

En esta página, el Workflow Client tiene tres partes: Primero el Header que se toma de la Master Page del Web panel. Luego un menú de árbol con todas las funciones que son provistas por el Cliente Personalizado, podemos hacer clic en esos enlaces de Webpanels y nos mostrará la página seleccionada en la parte derecha del panel. En este caso, he seleccionado la bandeja de entrada y mostrará el Webpanel de la bandeja de entrada del flujo de trabajo.

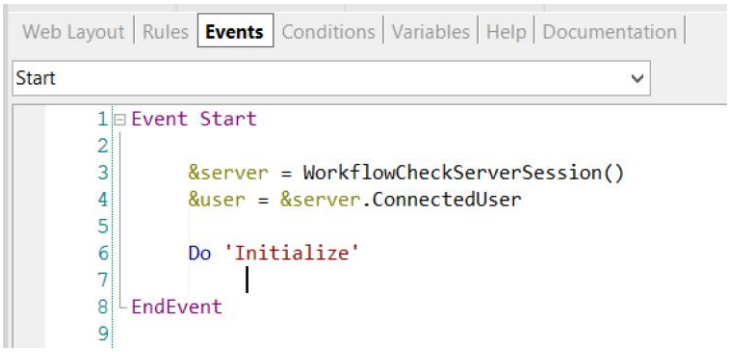


La bandeja de entrada tiene los filtros del flujo de trabajo, luego tiene una tabla con todas las acciones que se pueden ejecutar desde el grid, ejecutar, completar, historial, vista previa, delegar. Luego tiene un combo box que tiene todos los procesos que puede iniciar el usuario que inició sesión. Posteriormente se conforman los workitems que pertenecen a la bandeja de entrada del usuario. Después de eso, tiene un Webcomponent con el control de páginas del grid.



VALIDATE SESION

Si vemos el código de este panel, el primer evento es Start, aquí el Webpanel validará la sesión del usuario.



Cada objeto del Custom Client comprueba si hay una sesión válida y, para ello, utiliza un procedimiento llamado WorkflowCheckServerSession. Aquí tenemos dos variables, la variable &server, que es del tipo de dato WorfklowServer y la variable &user, que es del tipo de datos Workflow User.

Primero va a cargar la variable WorkflowServer desde la sesión web, usando el procedimiento WorkflowCheckServerSession, que veremos en la siguiente parte del curso.

```
Sub 'Initialize'

      Do 'Load Creatable Processes'

      WCFilters.Visible = &showFilters
      WCFilters.Object = WorkflowFilters.Create(WorkflowEntryPoint.INBOX)
      WCPaging.Object = WorkflowPaging.Create(WorkflowEntryPoint.INBOX)

EndSub
```

Después de eso, si tiene el WorkflowServer cargado, obtendrá el usuario conectado del WorkflowServer y luego inicializará la página web. En este caso estamos inicializando la bandeja de entrada del flujo de trabajo, por lo que se van a cargar todos los procesos creados del combo box, que hemos visto durante la ejecución.

```
314 Sub 'Load Creatable Processes'
315
316       &comboProcesses.Clear()
317       &comboProcesses.AddItem(0, GetMessageText('Select'))
318
319      &filter = new()
320      &processDefinitions = &user.ListCreatableProcessDefinitions(&filter)
321
322       If &processDefinitions.Count = 0
323            tableNewProcess.Visible = 0
324       Else
325            For &processDefinition in &processDefinitions
326                 &comboProcesses.AddItem(&processDefinition.Id, &processDefinition.Name)
327            Endfor
328       Endif
329
330 EndSub
```

Primero limpia los elementos del combo y se agrega el valor por defecto, luego utilizando los datos del usuario que se obtuvieron en el Evento Start se utiliza el método ListCreateTableProcessDefinition. Este método obtiene todas las definiciones de proceso que un usuario conectado puede iniciar.

El método ListCreateTableProcessDefinition tiene un parámetro de filtro del tipo de dato WorkflowFilter, pero en este caso lo dejamos vacío.

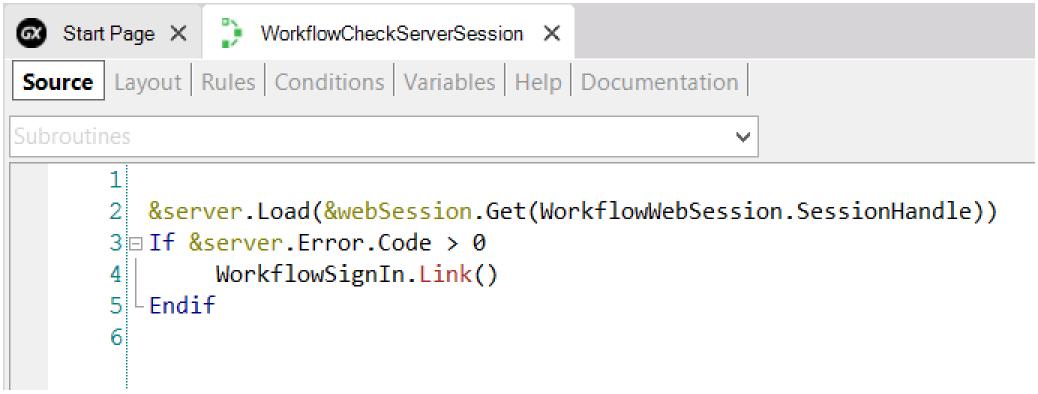
Si la variable &ProcessDefinition, que es una lista de definición de procesos, no está vacía, vamos a recorrerla para cargar todos los ID y el nombre de las definiciones del proceso en el combo box

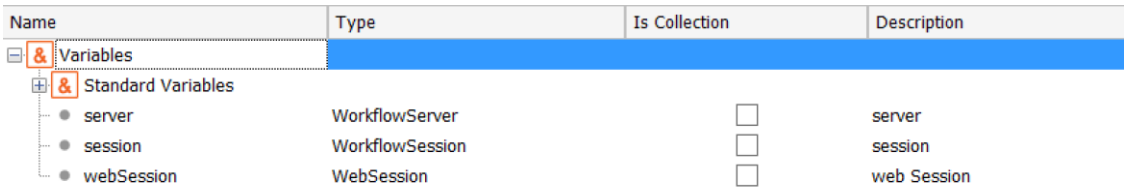
Después de cargar el combo box, crearemos dos Web Components, primero con un WorkflowFilter y el segundo con un WorkflowPaging. El primer Web Component sirve para filtros y el segundo para el paginado del grid.

Ahora veremos el procedimiento WorkflowCheckServerSession.

Este procedimiento obtienen el Id de la sesión desde la WebSession y carga esta conexión en la variable Server del tipo de dato WorkflowServer. Esta sesión se guardó en la Websession cuando se realizó log in. Si al cargar la sesión se detecta un error, se redirigirá al WorkflowSignIn.

Cuando la sesión está cargada, el usuario logueado es el que se utiliza para conectar con el WorkflowServer.





REFRESH & FILTERS

El siguiente método en la bandeja de entrada es el evento Refresh del Grid. Este ejecuta una subrutina llamada "Load Filters" que traerá todos los filtros de la WebSession. Se carga en la variable &filter, todos los filtros que se recuperaron de la sesión. Después de eso, obtenemos también de la WebSession, la página actual de la bandeja de entrada.

Los filtros están en un Web Component llamado WorkflowFilters Este guarda el SDT de filtros en la WebSession.

```
Event Grid.Refresh

    Do 'Load Filters'
    &workitems = &user.GetWorklistOrderBy(&filter, WorkflowOrder.CREATED_DESC)

EndEvent
```

```
Sub 'Load Filters'

    &filter.Load(&session.Get(WorkflowEntryPoint.INBOX + !'Filter'))
    &page = Val(&session.Get(WorkflowEntryPoint.INBOX + !'Page'))
    If &page > 1
        &filter.Start = (&page - 1) * WorkflowPaging.SIZE
    Else
        &filter.Start = 0
    Endif
    &filter.Limit = WorkflowPaging.SIZE
EndSub
```

Si observamos el Web panel WorkflowFilters, tiene una prioridad, estados, asunto, proceso, actividad, nombre, usuario y tiene dos subrutinas principales.

La primera se ejecuta cuando se hace la carga, recupera la sesión que había guardado previamente todos los filtros de tipo de dato, y asigna los valores a las variables del panel. En caso de que el panel se llame desde la bandeja de salida, asigna la fecha del día de a la variable Ended from.

```
Sub 'Load Filters'
      &filter.Load(&session.Get(&entryPoint + !'Filter'))
      &cmbPriority            = &filter.Priority
      &cmbState                    = &filter.State
      &subject                = &filter.Subject
      &name                        = &filter.Name
      &createdFrom            = &filter.CreatedFrom
      &createdTo                   = &filter.CreatedTo
      &endedFrom                   = &filter.EndedFrom
      &endedTo                = &filter.EndedTo
      &cmbEventType           = &filter.EventType
      &cmbProcessDefinition   = &filter.ProcessDefinition.Name
      &cmbActivity            = &filter.Activity.Name
      &cmbUser                = &filter.User.Id
      Do Case
            Case &entryPoint = WorkflowEntryPoint.OUTBOX
                  If &filter.EndedFrom.IsEmpty()
                        &endedFrom = &Today
                  Endif
                  If &filter.EndedTo.IsEmpty()
                        &endedTo= &Today
                  Endif
            Case &entryPoint = WorkflowEntryPoint.MY_PROCESSES Or &entryPoint = WorkflowEntryPoint.PROCESSES
                  Or &entryPoint = WorkflowEntryPoint.TASKS Or &entryPoint = WorkflowEntryPoint.DOCUMENTS
                  Or &entryPoint = WorkflowEntryPoint.MY_DOCUMENTS Or &entryPoint = WorkflowEntryPoint.EVENTS
                  If &filter.CreatedFrom.IsEmpty()
                        &createdFrom = &Today
                  Endif
                  If &filter.CreatedTo.IsEmpty()
                        &createdTo= &Today
                  Endif
      EndCase
EndSub
```

Si se llama desde My procesesses, task, documents, my documents y events, asigna la fecha del día en la variable Created from.

Después de que el usuario complete la información y haga clic en el botón Aplicar, ejecutará un evento desde donde llamaremos una subrutina para guardar filtros. En esta subrutina se asignara toda la información cargada por el usuario en las variables del filtro en un SDT.

Después de cargar toda la información con la variable de filtro, cargara esa información en la sesión web, para tenerla disponible

Una vez cargados los filtros en el evento Refresh, invocaremos el método “GetWorkListOrderBy” utilizando la variable de usuario que obtuvimos en el evento Start. Este método devuelve una colección de tareas, esta lista incluye todos los workítems que requieren ser completados por el usuario. Este método puede ser filtrado con los valores que el usuario selecciono en el Webcomponent workfilter. También recibe un parámetro que nos permite indicar el orden en el que queremos recibir la información, en este caso vamos a utilizar el dominio “WorkFlowOrder”, e indicaremos que ordene por fecha de creación en forma descendente. Esto nos devolverá la lista de workítems. Vamos a cargar esta lista en las variables correspondientes.

Posteriormente en el evento Grid.Load, iteramos sobre la lista y por cada elemento de la lista cargamos la información en las variables del grid y ejecutamos el método Gridl.load(). Esto hará que todos los workitems se carguen en el grid correspondiente.