El Debugger ABAP

INDICE

Debugger ABAP.

Es el proceso de identificar y corregir errores de programación. En inglés se conoce como *debugging*, porque se asemeja a la eliminación de *bichos* (*bugs*), manera en que se conoce informalmente a los errores de programación.(definición Wikipedia)
El Debugger es una herramienta imprescindible para los programadores de cualquier lenguaje. Con él es posible probar y depurar aplicaciones que se están desarrollando, visualizar y modificar variables en tiempo de ejecución o comprobar errores que aparecen en aplicaciones que fallan.

Para poder utilizar el debugger ABAP será necesario colocar un Breakpoint en el código que queramos revisar. Un breakpoint sirve para lanzar el debugger ABAP con lo que el programa aparecerá en un nuevo modo, para así poder analizar desde la línea de código donde hemos puesto dicho breakpoint.

Que es un breakpoint / Punto de interrupción.

Como su nombre lo dice es un punto que fijamos en una línea de código en la que queramos detener la ejecución del programa, esto nos permite analizar los datos en ese momento justo antes de ejecutar esa línea de código.
Se pueden colocar tantos como se necesiten, aunque normalmente se colocan en puntos críticos donde consideremos se están produciendo errores en el programa.

Tipos de breakpoint

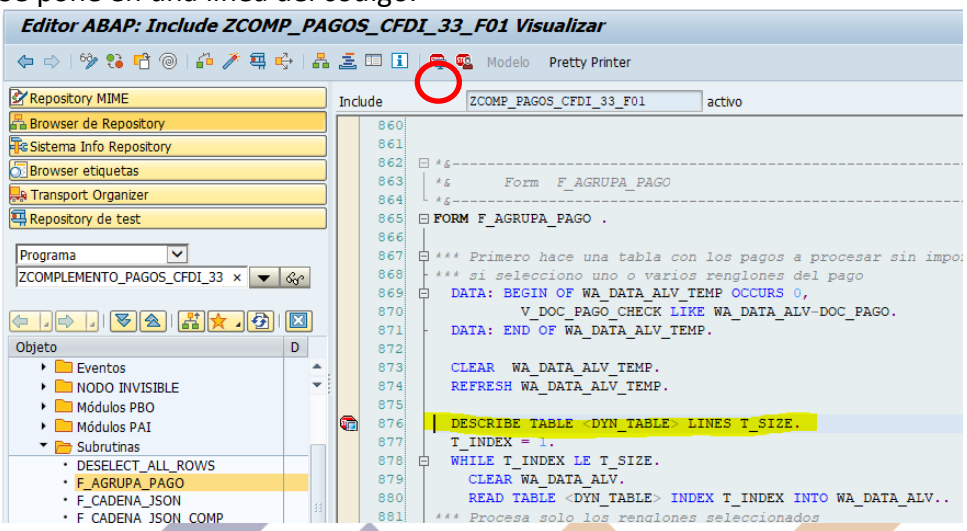
Breakpoint de sesión  : Estos puntos de interrupción solo son válidos para la sesión actual, para aplicaciones que se lancen dentro de la misma entrada al sistema. Adicional estos breakpoint son independientes de usuario.

Breakpoint externos: Breakpoint de usuario  : Estos puntos de interrupción son válidos para cualquier aplicación que se ejecute en el sistema, independientemente de donde se haya lanzado. Por ejemplo, en aplicaciones Web Dynpro o BSP, que se ejecutan desde un navegador web, solo podrían utilizarse este tipo de breakpoints. Estos suelen tener validez de 2 horas y se asocian a un usuario.

Ingresando en modo debugging

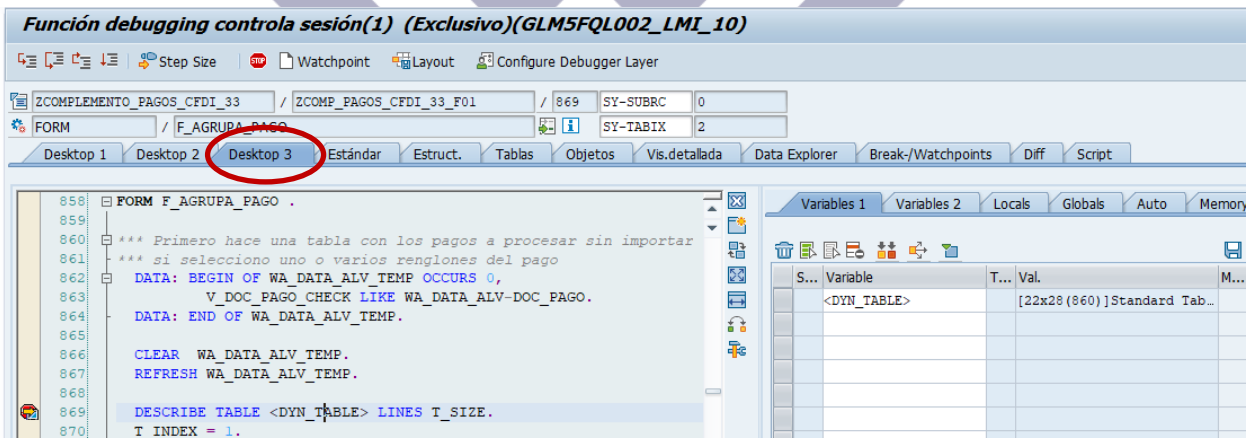
En ABAP hay varias formas de ingresar al modo de debugueo:
Por medio de un Break Point incluido en el código de programa. Como vemos en la imagen siguiente, se está marcado en uno de los cuadros rojos el icono del breakpoint, el cual una vez posicionado en el código, usaremos para indicar que en la línea de código (en este caso la línea 876) queremos realizar un Breakpoint, esto hará que de forma automática aparezca un icono de stop donde seleccionamos.

Se pone en una línea del código.

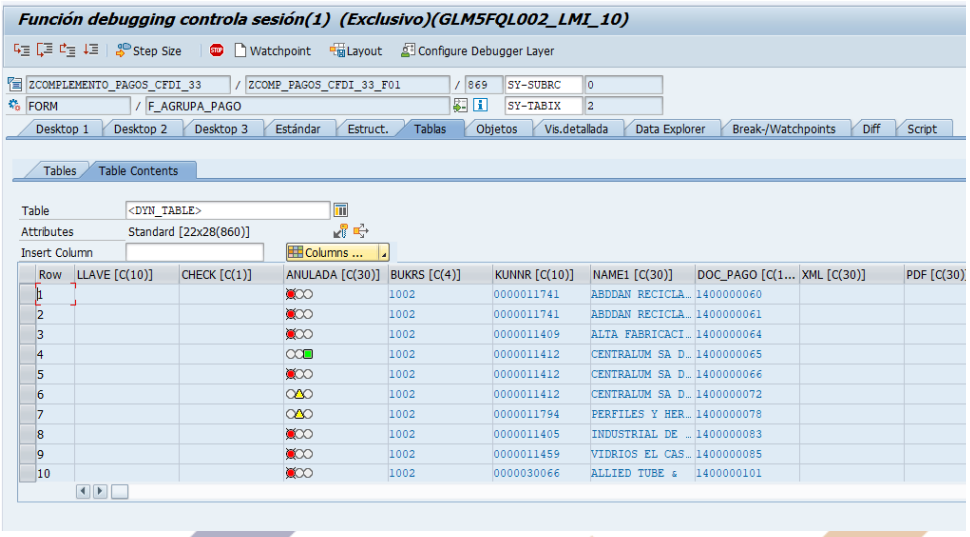


Y al ejecutar la transacción se detendrá en la línea en la que fijamos el break.

Si nos vamos a la pestaña de “Desktop 3”, a la izquierda nos aparecerá el código de la aplicación en ejecución y a la derecha una tabla con las variables del programa. En las dos primeras pestañas podremos poner las variables que queramos: en “Locals” nos aparecerán todas las variables locales del programa y en “Globals” las globales.



Para ver el contenido de las tablas o variables, damos doble clic sobre el objeto en este caso es la tabla <DYN_TABLE>, del lado derecho se muestra la tabla y en la columna Val, el contenido de esta, damos doble clic sobre <DYN_TABLE> y mostrará el contenido.



Para regresarnos a la ventana anterior clic en la pestaña Desktop 3, y podemos continuar con el debug.

Avanzar por el programa ABAP.

Por último, para poder avanzar por el programa, se pueden utilizar las siguientes opciones:

Paso a paso (F5):  paso a paso Se introducirá en la línea de código que estemos, es decir, si es una subrutina o una función navegará hasta ella para poder depurarla. Si es una sentencia simple (IF, MOVE, CLEAR…) pasará a la siguiente línea de código.

Ejecutar (F6):  ejecutar Avanzará a la siguiente línea de código, independiente de que sea una subrutina, función o sentencia simple, ejecutando la subrutina o función en cuestión.

Retornar (F7):  Retornar Si se está dentro de una subrutina o función, saldrá de la misma ejecutándola y parando justo en la línea siguiente a su llamada.

Continuar (F8):  Continuar: El programa se ejecutará hasta encontrar otro breakpoint. Si no existen más, el programa se ejecutará hasta terminar o pasar el control al usuario.

BIBLIOGRAFÍA.