

Capítulo 4

Archivando valores.

En casi todas las instalaciones es necesario almacenar valores dentro de la base del WinCC, para posteriormente representarlos en forma de gráfica o de tabla. En la nuestra en concreto necesitamos guardar la temperatura de las calderas del agua potable cada segundo durante un mes.

Los parámetros cada cuanto debe de guardarse los valores y por cuanto tiempo son dos factores fundamentales para que el proyecto de WinCC acabe bien, y uno pueda volver algún día por la instalación sin temor a oír la famosa frase: “¡Ahí está, a por él!” (que recuerdos me trae el tema...bien, continuemos).

Recordemos las máximas del buen programador:

- No hagas nada que no te pidan, te arrepentirás.
- No hagas lo que te pidan, si no aquello que sabes hacer.
- No digas que lo sabes hacer, y aparenta que es muy difícil de realizar.

Recordando estas máximas podemos entender que si aceptamos guardar valores cada poco tiempo, el disco duro rascará cosa mala, y te puedo asegurar que a nadie le importará que les expliques que te lo pidieron ellos: la culpa será tuya. Si aceptamos almacenar valores de manera ilimitada “hasta que se llene el disco duro”, lo que pasará es que cuando se intente abrir una gráfica, tendrá que manejar un archivo de cientos de miles de datos, con lo que el WinCC tardará minutos en visualizar la gráfica, y esta vez ni intentarás explicar porqué, ya que anteriormente ya viste lo que pasó. Resumiendo: no dejes que te digan cada cuanto hay que guardar, dí cada cuanto se puede tú.

Lo aconsejable sería un tambor de unos 3000 valores. Esto podría ser, p. Ej., guardar cada quince minutos, durante las veinticuatro horas durante un mes. Ten encuenta, que si aceptas almacenar esta misma cantidad, pero durante un año, son 35000 valores, que deberemos de leer para poder mostrar la gráfica una vez pase el año, y de ahí continuar con el suplicio.

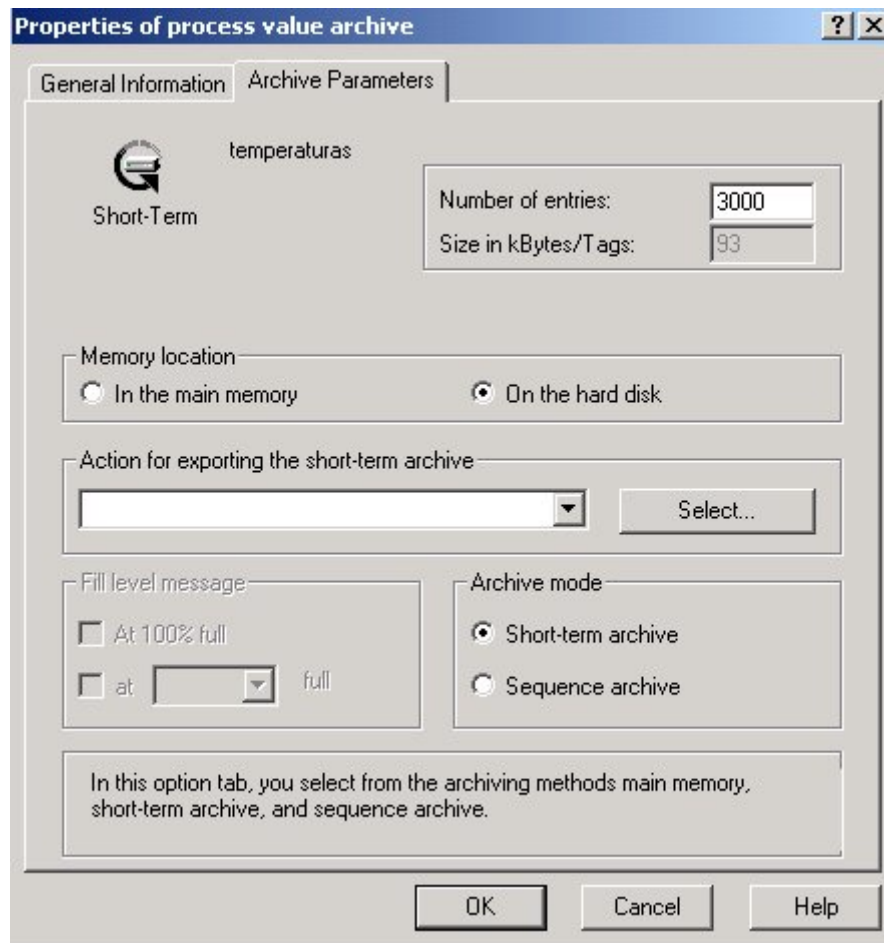
Generar archivos.

Como ya sabemos en el tag logging es donde generamos los archivos en los cuales almacenamos los valores en le disco duro. Como tendremos en nuestra instalación tres valores de temperatura, vamos a generar de momento un archivo, en el cual almacenaremos los tres valores indicados. Antes de nada, dentro del tag logging, presionando con el botón derecho sobre archives, seleccionamos pre-settings->analog tag y definimos como van a ser nuestra tendencia a alamacenar en el archivo “por defecto”.

Aunque antes dije lo que dije, en el ejemplo voy a tomar un default para tiempo de adquisición de un segundo y de archiving/display también de un segundo, ya que como es un ejemplo no vamos a estar 15 minutos esperando a que salga un valor por la pantalla. Por lo tanto seleccionamos un segundo de adquisición. Guardamos y de nuevo con el botón derecho sobre archives, seleccionamos archive-wizard, que nos guiará en la creación de nuestro archivo, que llamaremos temperaturas, y al que le asociamos tres

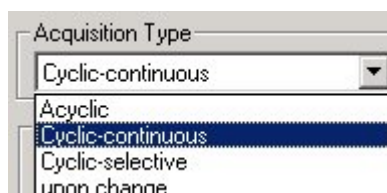
Un ejemplo para WinCC

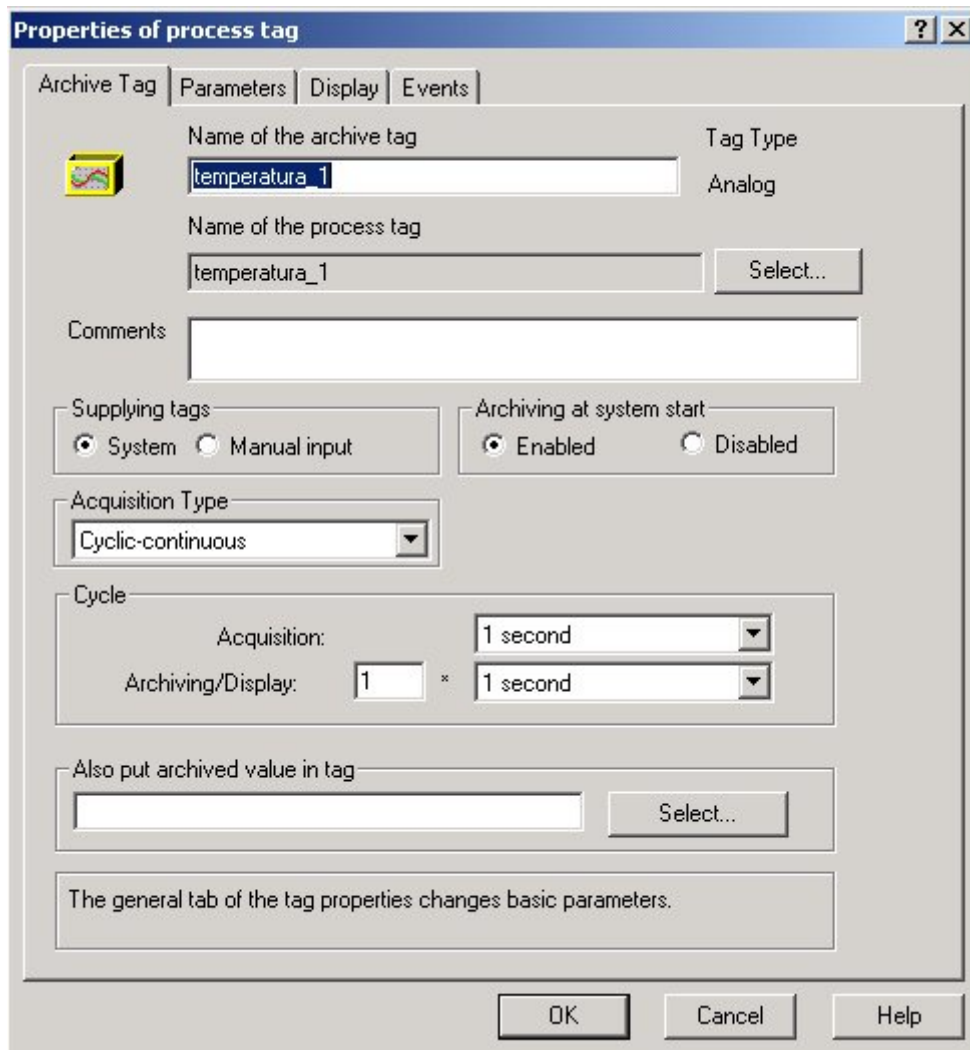
variables, temperatura_1, 2 y 3, todas de formato real. Una vez creado el archivo con las tres variables, presionando con el botón derecho sobre el archivo temperaturas, vamos a definir el tamaño del tambor, que será de 3000 variables.



El siguiente paso no sería necesario en nuestro ejemplo, ya que previamente hemos definido que las tendencias tendrían un tiempo de adquisición y de almacenamiento predeterminado para todas. Pero si deseamos cambiar dichos parámetros deberemos de presionar sobre las tendencias creadas, p. Ej. sobre temperatura_1, y modificar dichos valores.

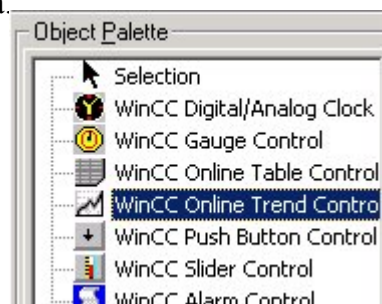
En cualquier caso, lo que sí que es importante es tener claro el tipo de tendencia que debemos de utilizar. Vamos a gastar la de cy clic- continuous, ya que deseamos almacenar cada cierto tiempo, pero existen más tipos, como muestra la figura: Mas adelante haremos diferentes ejercicios prácticos en nuestro ejemplo.





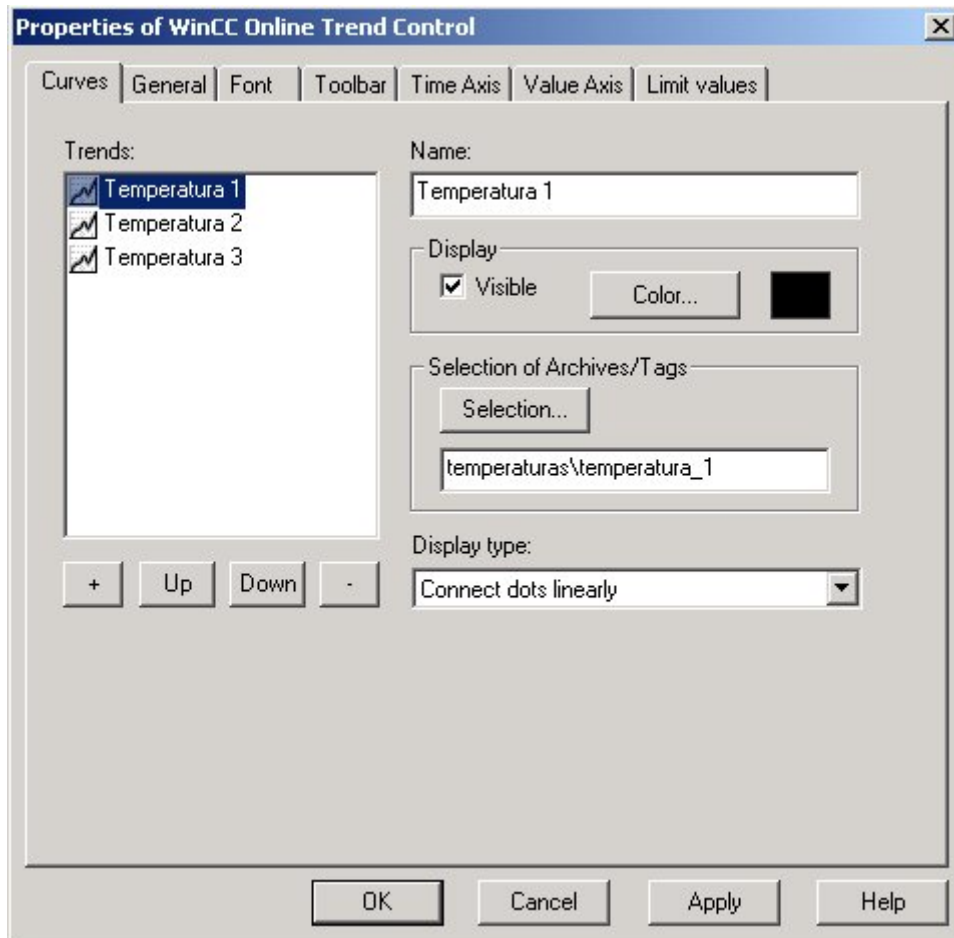
Pues si ya tenemos el tema acabado, guardamos y en control center, activamos la parte de runtime del tag logging. Ya tenemos casi acabado el tema. Nos queda lógicamente mostrar dichos valores en alguna picture. Para ello, creamos una que se llame graficas_temperatura.pdl, y ponemos en top.pdl un botón que acceda a la misma (recuerda, para crear graficas_temperatura.pdl debes de abrir plantilla.pdl y guardar dicha picture con un nuevo nombre, no te pongas a hacer una nueva picture, que te tienes que hacer otra vez todo el tema del top y bottom.

Estando en la nueva picture, colocamos un control de tipo wincc online trend control en el centro de la pantalla.



Un ejemplo para WinCC

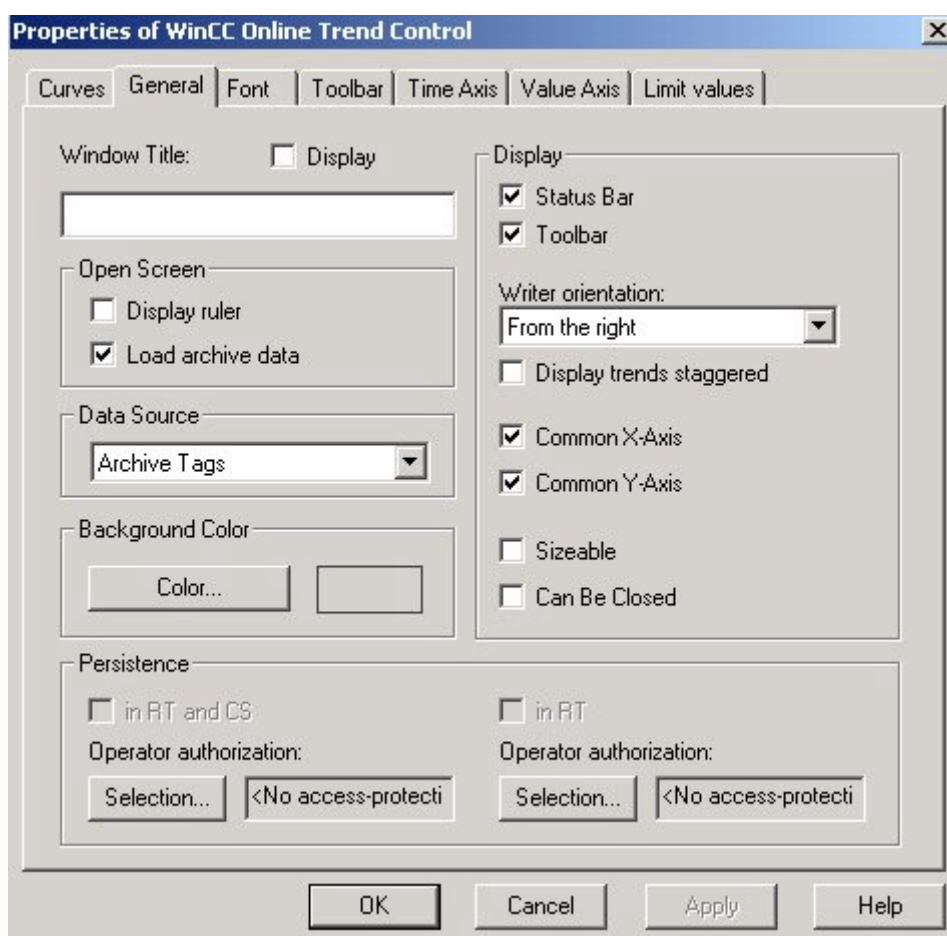
Una vez colocado el control en la picture, vamos a ver como ajustamos las propiedades del mismo (para sacarlas, pincha dos veces encima del objeto).



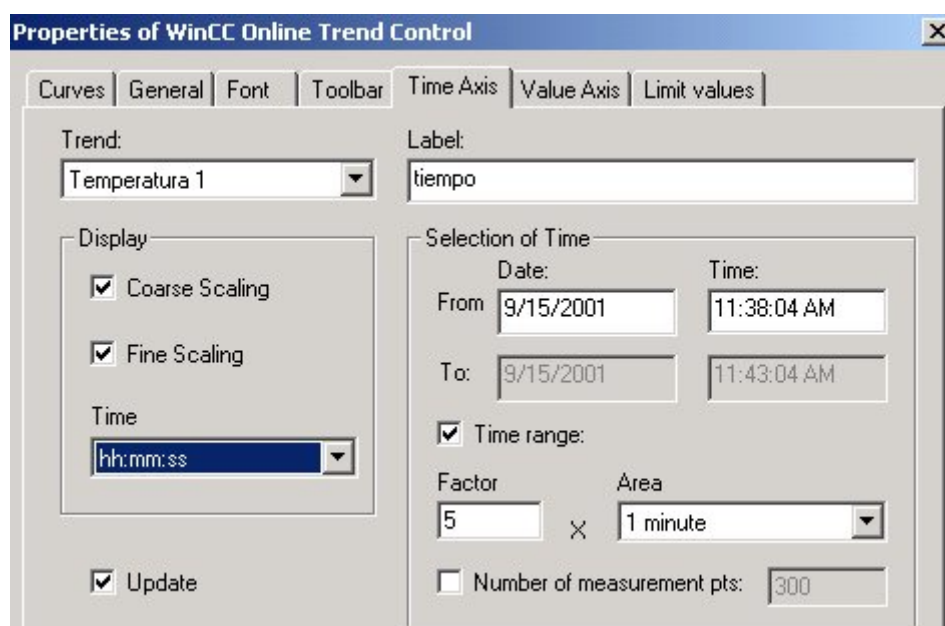
Para ir añadiendo las tendencias temperatura 1, 2, y 3 a la ventana presionaremos el botón +. En selection iremos asociando a cada tendencia generada su variable dentro del archivo generado previamente.

En la siguiente solapa, general, vamos a seleccionar:

- Display window title: no, porque si le pones título el usuario puede mover la ventana, que se encuentra flotante sobre la picture.
- Load archive data: si, para que cuando entres cargue de la base de datos la curva, en caso contrario no muestra nada la gráfica.
- Writer orientation: desde la derecha.
- Common x-axis: si, para que no muestre un eje x por cada tendencia, sino uno común.
- Common y-axis: si, por lo mismo de antes.
- Sizeable: no, que luego los operarios hacen una destroza.
- Display trends staggered: no. Esto separa una gráfica diferente para cada tendencia. Queremos que todas estén en la misma gráfica, para comparar unas con otras.



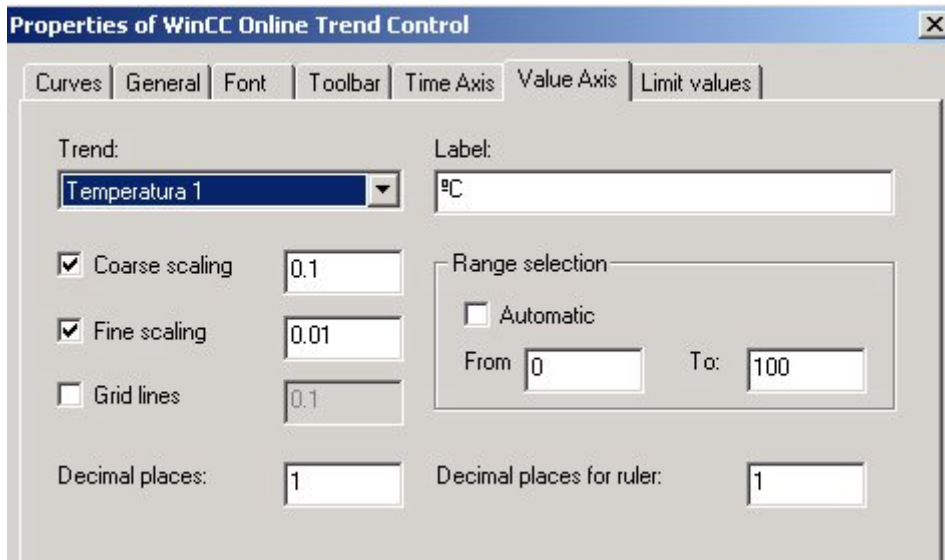
En time axis, lo más importante es definir cuánto tiempo se debe de mostrar en el eje x.



Un ejemplo para WinCC

Hay dos maneras de realizar esto:

- Seleccionar un time range, como hemos hecho, de 5 minutos.
- O seleccionar un número de puntos de medida. Como guardamos valores cada segundo, en 5 minutos son 300 valores.



En value axis vamos a definir el eje y. Cosas a tener en cuenta:

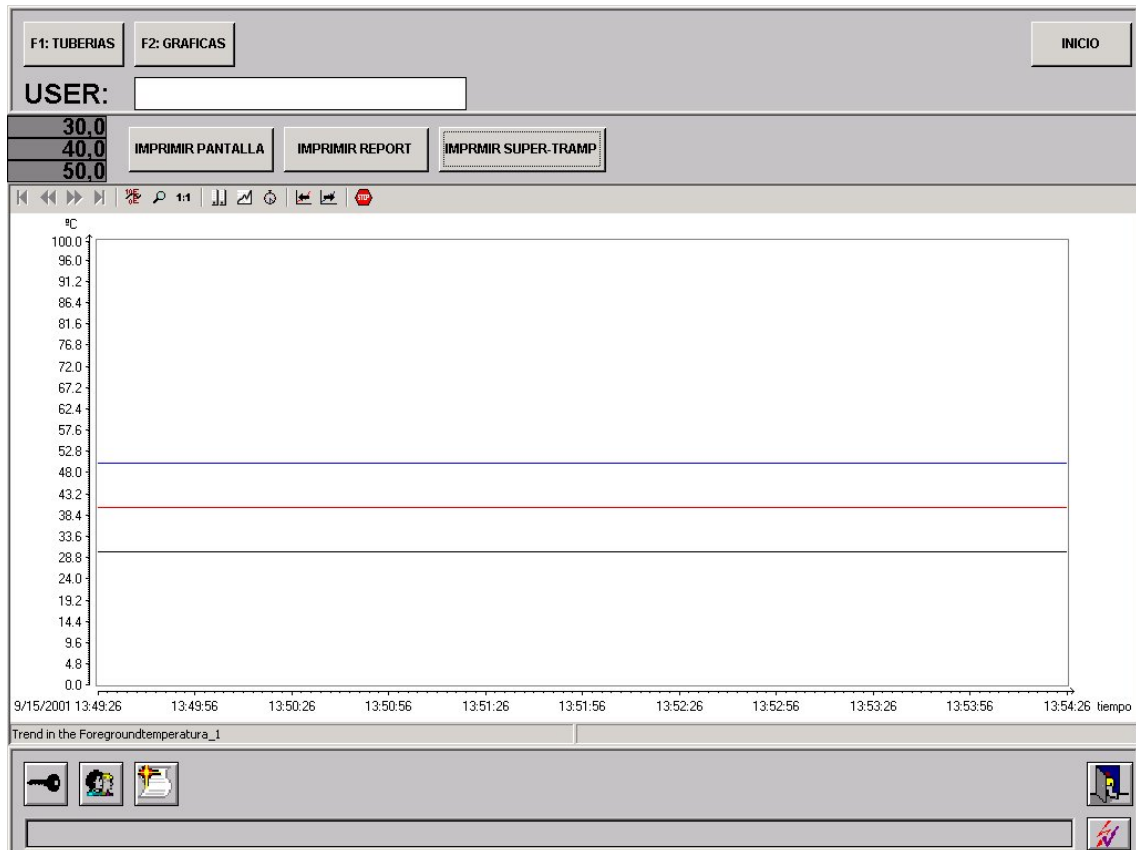
- Si dejamos la opción automatic en range selection el eje y variará en función del valor máximo y mínimo que actualmente se represente en pantalla. Esto lo tenemos que cambiar por un valor fijo, porque si no se moverá más que un flan en un terremoto. Atención: no basta con modificar este parámetro una vez. Como en nuestro caso tenemos tres tendencias, tendremos que cambiarlo en las tres, y no solo en temperatura 1.
- Podemos poner las grid lines, que son guías para orientarse en la gráfica, pero vale la pena ponerlas el usuario, y no que estén fijas en la pantalla todo el rato.
- Decimal place ruler: ponemos solo un decimal, que si no sale un churro de ceros cuando ponemos la regla para ver un dato en un momento determinado.

Ya hemos terminado de definir nuestra ventana de gráficas. Solo nos queda ponernos sobre ella (ver imagen inferior) de tres I/O relacionadas con nuestras tres variables, temperatura_1, temperatura_2 y temperatura_3. Ahora tendremos que imprimirla de alguna manera. Tenemos dos maneras, una cutre, otra salchichera y la super-trampa.

La cutre la seleccionamos en el botón imprimir pantalla con el código:

```
ProgramExecute("C:\\SIEMENS\\WINCC\\BIN\\PRTSCR.EXE");
```

El problema es que si imprimimos la pantalla tal cual está, no vamos a ganar para pagar tinta de impresora. Por eso es mejor, siempre que se pueda, utilizar la forma salchichera de imprimir. La forma salchichera, mas conocida a partir de ahora como imprimir report, consiste en hacerse un layout que posteriormente imprimiremos con el botón preparado especialmente para la ocasión.



El problema de esta manera de imprimir una gráfica es que no vamos a poder configurar el eje de tiempos en runtime, por lo que se va a mover menos que los ojos de espinete, y será muy poco práctica. Avisado estas, aunque si no me crees, te explico como se hace el tema. Ingredientes necesarios para realizar la receta son:

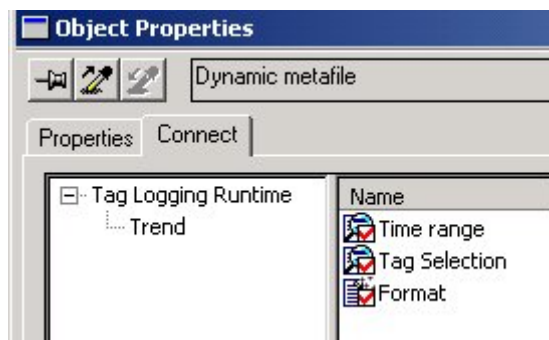
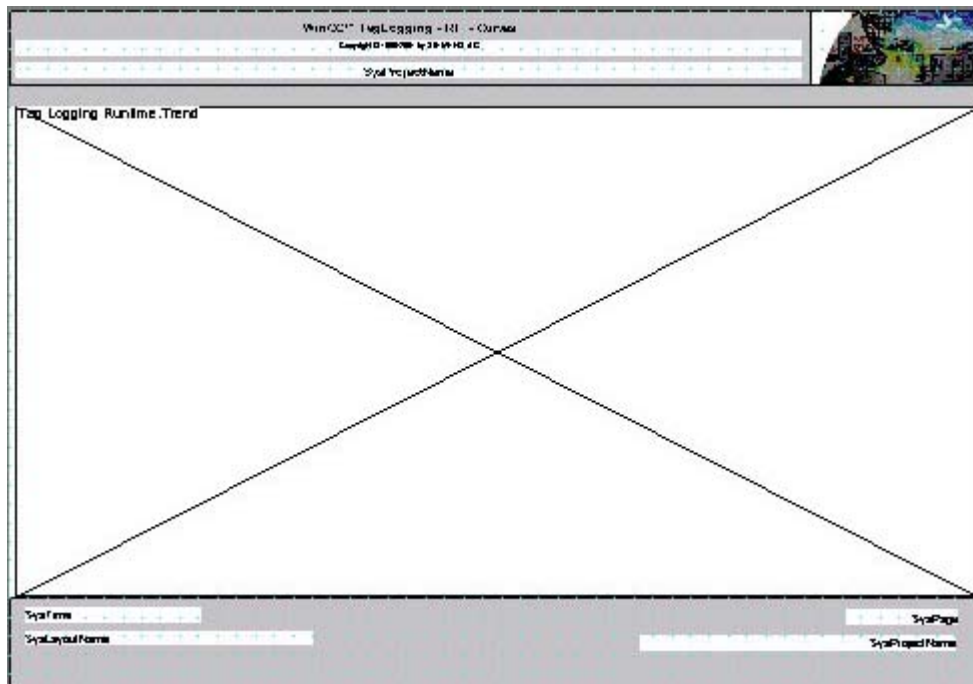
- Un layout
- Un job
- Un botón, a ser posible fresco, del día.
- Una gráfica bien troceada y en su punto.

Receta:

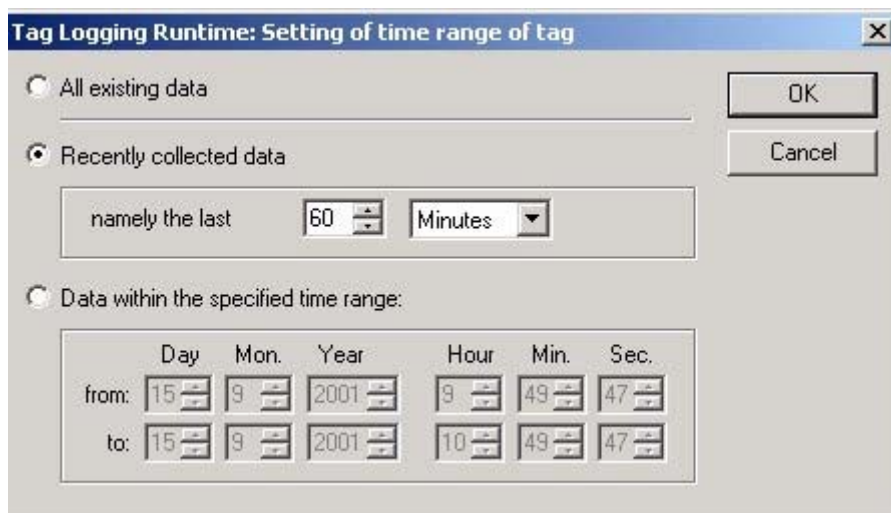
Se abre el layout @cctlgrtcurves.rpl y se guarda con otro nombre, en nuestro caso grafica_temperatura.rpl. Se edita dicho rpl, y editando las propiedades generales del layout se modifica en geometry orientation y se pone landscape para que la hoja aparezca apaisada. Además se modifica la estructura de objetos para que se queden como muestra la figura inferior.

Editando el tag logging runtime trend (esa x grande del centro), podemos acceder a su conexión con las variables de proceso.

Un ejemplo para WinCC



En time range podemos definir el rango de tiempos a mostrar.

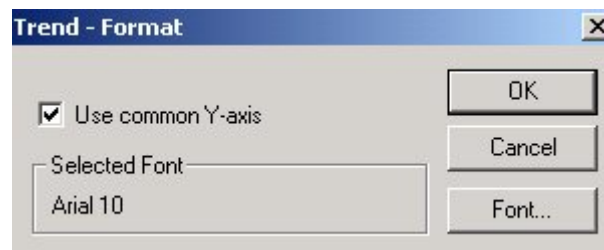


Todo el problema está aquí, ya que si no seleccionamos un periodo de tiempo de x minutos a partir de la fecha actual, el tema no va cara al aire. ¿Y si queremos mostrar el mes pasado?, deberemos de seleccionar la tercera opción de la ventana, data within.. pero el problema es que estos valores no son parámetros, por lo que no es un camino válido. Tal vez exista alguna manera de cambiar estos valores, pero la desconozco.

Por lo tanto este camino solo nos sirve para mostrar la última hora, o día, o algo así en la pantalla y/o impresora, y punto.



En la siguiente opción seleccionamos las tendencias que queremos mostrar en nuestro report.



Y por último podemos decirle en la última opción que utilice un eje y común para todas las tendencias que se muestran en el report, lo cual es lo lógico.

Ya tenemos nuestro report, guardamos y salimos, y nos creamos un nuevo job, llamado graficas_temperatura que lleva asociado el layout que hemos creado.

Y para finalizar, volvemos a nuestra picture graficas_temperaturas.pdl y nos colocamos un botón que hemos denominado “imprimir report”, y que lleva asociado el siguiente código:

```
RPTJobPreview("GRAFICAS_TEMPERATURAS"); //Return-Type :BOOL
```

Con este código realizamos un preview, desde el que se puede imprimir la gráfica. Si directamente queremos imprimir y punto, el código será:

```
RPTJobPrint("GRAFICAS_TEMPERATURAS"); //Return-Type :BOOL
```

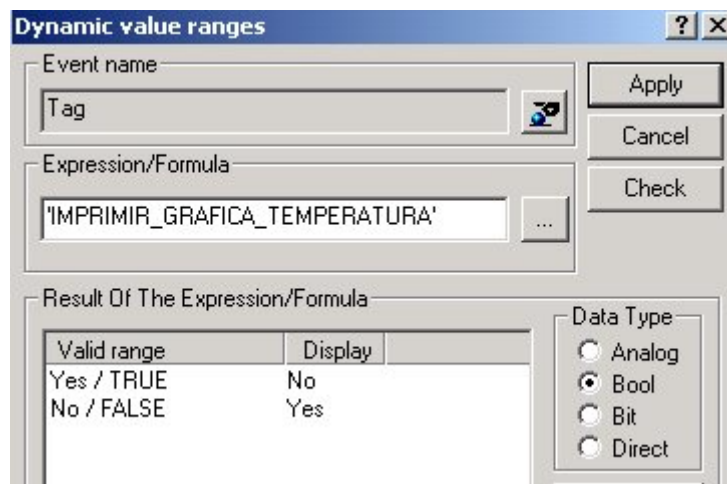
Como ves es bastante limitada esta manera, aunque muy sencilla de realizar los reports de las gráficas. Lástima que no sepa como poder modificar el eje x de tiempos de manera parametrizable.

Visto lo visto, el único camino es recordar a ese grandísimo grupo de música, supertramp. El truco está en imprimir la pantalla, pero cuando la vas a imprimir la modificas para que parezca un report.

Comenzaremos por ocultar top y bottom. ¿Cómo?. Primero nos creamos una variable interna de tipo bool, que llamaremos imprimir_grafica_temperatura. Cuando le demos al botón supertramp activaremos dicha variable.

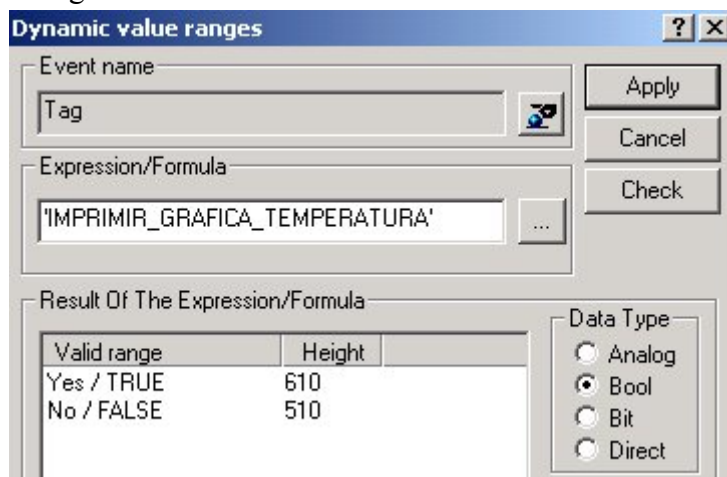
```
SetTagBit("IMPRIMIR_GRAFICA_TEMPERATURA",TRUE);
```

Ahora para ocultar top y bottom en la propiedad visible le asociamos el siguiente dynamic dialog:



Atención: para que funcione bien el tema en event name debes de asignarle upon change, no cada 2 segundos, que si no no chuta.

Continuamos. Ya que hemos ocultado bottom, tenemos que estirar la ventana de la grafica hasta abajo. En la propiedad height de la ventana de graficas asociamos el siguiente código:

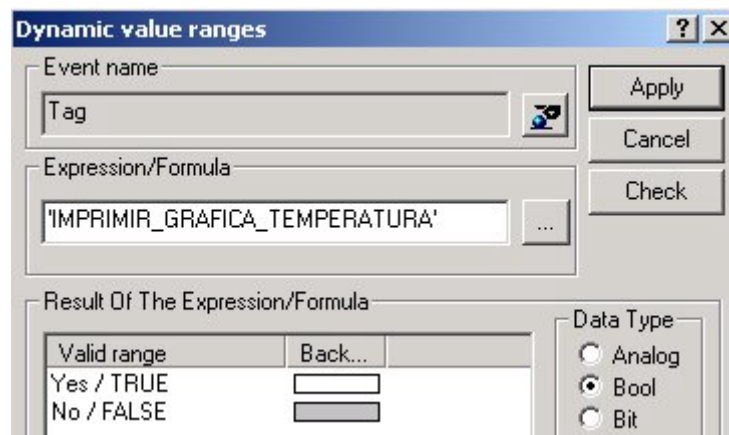


Mas cosas...Deberemos de ocultar la barra de toolbar y de statusbar de la ventana de graficas. Esto tiene un poco más de tela, y deberemos de acceder directamente a sus propiedades. Para ello, en el evento del botón supertramp, justo debajo de la activación de la variable que estamos gastando todo el rato, introducimos el siguiente código:

```
SetPropBOOL(lpszPictureName,"VENTANA_GRAFICA","Toolbar",FALSE);  
SetPropBOOL(lpszPictureName,"VENTANA_GRAFICA","Statusbar",FALSE);
```

También podemos poner un texto descriptivo, a modo de cabecera, en el lugar donde está top. Para que no nos moleste el solape, cambiamos top a la capa 2, luego la desactivamos y entonces ya podemos escribir en su lugar lo que queramos. No importa que sean textos visibles, porque volviendo a activar top tapaná a dichos textos. Cuando le demos al botón supertramp ocultará top, con lo que quedará al descubierto los textos o cosas que nos hacen de cabecera de la gráfica en la impresión.

Mas cosas...el fondo de la ventana debería de ser blanco, para no gastar tinta. Esto se consigue, en la propia picture, en la propiedad background color, asociándole el siguiente dynamic:



El código que hay que poner a continuación en el botón supertramp va a ser el siguiente:

```
ProgramExecute("C:\\SIEMENS\\WINCC\\BIN\\PRTSCR.EXE");
```

Y debajo de esta instrucción vamos a deshacer las pruebas, como los ladrones de guante blanco:

```
SetTagBit("IMPRIMIR_GRAFICA_TEMPERATURA",FALSE);  
SetPropBOOL(lpszPictureName,"VENTANA_GRAFICA","Toolbar",TRUE);  
SetPropBOOL(lpszPictureName,"VENTANA_GRAFICA","Statusbar",TRUE);
```

Si lo haces verás que acabas en un momento y no es tan complicado. ¿Qué menudo apaño?. Pues listo, a ver si se te ocurre a ti algo mejor...

El resultado sería el siguiente, si no deseamos quitar los botones (que yo recomiendo quitarlos y todo para que quede perfecto el truco:

GRAFICAS TEMPERATURAS

