ATTITUDES

Información sobre Power Systems, incluidos AS/400, iSeries y System i

Año 26 - Diciembree 2011-Enero 2012 Nº 277 Precio: 7 Euros

COLABORACIONES

Áreas de datos ajustables

Si usted ha trabajado en informática durante más de 15 minutos, sin duda, habrá oído la pregunta "¿Sería complicado...?". Las personas que dependen de los sistemas a los que damos soporte están contstantmente pensando en nuevas y mejores formas de acercar los sistemas de información a la realidad del negocio. Cada pequeña cosa que usted puede hacer para evitar el esfuerzo de modificar el software valdrá la pena. A continuación muestro dos formas muy sencillas para reducir el esfuerzo necesario para cambiar la longitud de un área de datos de caracteres.

Consideremos el área de datos, FacInfo (Facility info - Información de las instalaciones), el cúal contiene un elemento de información: Un ID de instalación de cuatro caracteres. Un programa de CL que utilice el área de datos podría incluir instrucciones fuente como los del siguiente trozo de código.

Así es como los programadores suelen definir y acceder a las áreas de datos de caracteres, y este código funciona bien en el día a día.

Un día, el habitual autodesignado autoridad superior o archipámpano, pregunta: "¿Qué tan difícil sería añadir la ubicación de la instalación a estos y aquellos informes?" Usted sabe que el tamaño del área de datos debe ser aumentado. La respuesta que le dé depende de cómo el programa CL y programas RPG definan el área de datos.

Si sus programas CL acceden a los datos como se mostraba en el código anterior, entonces alguién tendrá que cambiar el código fuente manualmente.

Sigue en página 5

Expresiones lógicas que acaban en ilógicas

Se supone que la programación de computadoras es una ciencia lógica. Uno podría pensar entonces, que los programadores de computadoras son pensadores lógicos. Pero no es así. Permítanme darles algunos ejemplos de pensamiento ilógico tomados de programas reales en producción. A continuación, mostraré una sencilla solución para evitar que expresiones lógicas no terminen en algo ilógico.

Vamos a empezar con este ejemplo tomado de las especificaciones de formato fijo de RPG. Le sugiero que encuentre la lógica mal por si mismo antes de leer mis comentarios.

Este código lleva a cabo la intención del programador. Sin embargo, el primer IF y su correspondiente ENDIF son innecesarias. Aquí hay otro.

El segundo GOTO nunca tendrá lugar. Cualquier cálculo no etiquetado que siga a una ramificación incondicional no se ejecutará y algunos compiladores de lenguaje indican este error. El compilador CL, sin embargo, no le importa en absoluto, y ni tan siquiera generará un mensaje de aviso. Aquí hay un tercer ejemplo, antes de pasar a algo más sustancioso.

Sigue en página 8

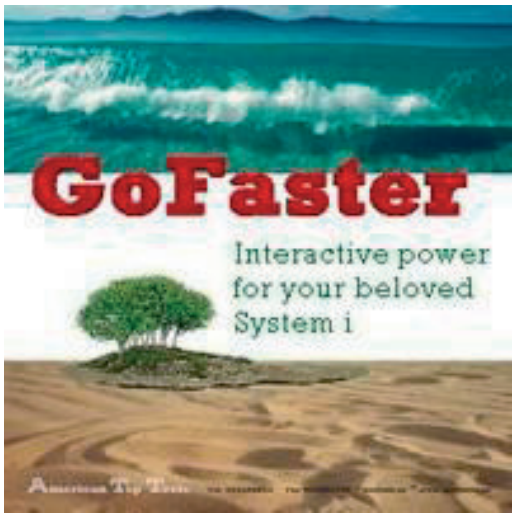
Leer una vez, ajustar varias

El uso de SQL en lugar del las E/S nativas para realizar consultas y manipular la base de datos, va más allá de una simple sustitución. Se requiere una forma de pensar diferente. Las caracteristicas equivalentes difieren y hoy mostraré las diferencias entre una actualización SQL y una actualización nativa.

Considere el siguiente escenario sobre un proyecto en el que estuve involucrado recientemente.

Tenía que leer una tabla de base de datos (archivo físico) de arriba a abajo. Después de recuperar cada fila (registro), llamaba a uno o dos de los programas de lenguaje de alto nivel, utilizando parámetros para pasar valores de datos de la tabla y para recibir otros valores de datos a cambio. Dependiendo de los valores retornados, tuve que actualizar algunas columnas (campos). En otras palabras, tuvieron que realizarse demasiadas actualizaciones diferentes.

Sigue en página 13



SUMARIO

Colaboraciones