AS/400

SC10-9424-00

COBOL/400 Guía del usuario

Versión 3 Release 1.0

AS/400

SC10-9424-00

COBOL/400 Guía del usuario

Versión 3 Release 1.0

¡Nota!

Antes de utilizar esta información y el producto al que le da soporte, asegúrese de leer la información general que aparece en el apartado “Avisos” en la página ix.

Primera Edición (septiembre 1994)

Esta publicación es la traducción del original en inglés *COBOL/400 User's Guide*, SC09-1812-00.

Esta edición se aplica al programa bajo licencia IBM ILE* COBOL/400* (Programa 5763-CB1), Versión 3 Release 1 Modificación 0, y a todas los releases posteriores y modificaciones hasta que se indique lo contrario en nuevas ediciones. Asegúrese de que está utilizando la edición adecuada para el nivel del producto.

Solicite sus publicaciones a través de su representante o sucursal IBM de su localidad. En la dirección que figura a continuación no se pueden adquirir publicaciones.

Al final de esta publicación se incluye un impreso para los comentarios del lector. Si falta dicho impreso puede enviar sus comentarios a:

IBM, S.A.
Centro de Traducción y Publicaciones
Avda. Diagonal, 571
08029 Barcelona
España

También puede enviar sus comentarios por fax:

Desde España: (93) 209 11 16
Desde otros países: 34 3 209 11 16

Si tiene acceso a Internet, puede enviar sus comentarios por correo electrónico a pubas400@vnet.ibm.com.

Al enviar información a IBM se cede a dicha firma el derecho no exclusivo de utilizar o distribuir dicha información de la manera que crea más conveniente, sin incurrir por ello en ninguna obligación hacia el lector.

Contenido

Avisos

Las referencias hechas en esta publicación a productos, programas o servicios IBM no implican que IBM tenga la intención de hacerlos disponibles en todos los países en los que opera. Cualquier referencia a un programa bajo licencia IBM en esta publicación no implica ni establece que sólo puedan utilizarse programas bajo licencia IBM. Cualquier producto, programa o servicio funcionalmente equivalente que no vulnere ninguno de los derechos de la propiedad intelectual de IBM puede ser utilizado en lugar de dicho producto, programa o servicio IBM. La evaluación y la verificación de la operativa conjunta con otros productos, a excepción de aquéllos expresamente designados por IBM, es responsabilidad del usuario.

IBM puede tener patentes o solicitudes de patentes pendientes que cubren el tema tratado en este documento. El hecho de que esté en posesión de este documento no le otorga ninguna licencia sobre dichas patentes. Puede enviar las solicitudes de licencia, por escrito, al IBM Director of Licensing, IBM Corporation, 208 Harbor Drive, Stamford, Connecticut, USA 06904-2501

Los cambios relevantes o las actualizaciones en el texto vienen indicadas mediante una línea horizontal (|) a la izquierda del cambio o adición en cuestión.

Esta publicación contiene ejemplos de datos e informes utilizados en operaciones de gestión ordinarias. Para ilustrarlos de la forma más completa posible, los ejemplos incluyen nombres de personas, empresas, marcas y productos. Todos estos nombres son ficticios y cualquier parecido con los nombres y direcciones que utilice una empresa de gestión real es pura coincidencia.

Información acerca de la Interfaz de Programación

La presente publicación tiene la intención de proporcionar ayuda al cliente para escribir programas COBOL/400.

En esta publicación también aparece información de asesoramiento asociado y de la interfaz de programación de uso general.

Las interfaces de programación de uso general permiten que el cliente escriba programas que reciban servicios del compilador COBOL/400.

La información de la interfaz de programación de uso general y el asesoramiento asociado se identifica de manera explícita allí donde se produce, ya sea mediante una instrucción introductoria a un capítulo o sección, o bien mediante el enmarcado siguiente:

Interfaz de Programación de Uso General

Información de asesoramiento asociado y de la interfaz de programación de uso general...

Fin de Interfaz de Programación de Uso General

Marcas Registradas y Marcas de Servicio

Los términos siguientes, marcados con un asterisco (*) en esta publicación, son marcas registradas de IBM Corporation en los Estados Unidos y/o en el resto de los países:

Application System/400	AS/400	CICS/400
COBOL/400	IBM	ILE
Operating System/2	Operating System/400	OS/2
OS/400	RPG/400	SAA
SQL/400	System/370	Systems Application Architecture
400		

Acerca de este manual

Este manual proporciona toda la información que un programador necesita para escribir, compilar, comprobar, depurar y ejecutar programas COBOL/400* en el sistema Application System/400* (AS/400*) IBM*.

Este manual hace referencia a otras publicaciones IBM. Estas publicaciones se listan en el apartado "Bibliografía" en la página 405 con el título completo y el número de orden de base. Cuando se hace referencia a ellas en el texto, se utiliza una versión abreviada del título.

A Quién Va Dirigido Este Manual

Este manual se ha concebido para programadores que tienen algunos conocimientos sobre el lenguaje de programación COBOL y para los operadores que ejecutan los programas. Es una guía completa que enseña a los usuarios del sistema AS/400 el lenguaje COBOL/400. El usuario debe tener un conocimiento base sobre los conceptos referentes al proceso de datos, sobre el lenguaje de programación COBOL y sobre el Operating System/400* (OS/400*) de IBM.

Cuando utilice este manual, el usuario podrá:

- Diseñar programas COBOL/400
- Codificar programas COBOL/400
- Introducir, compilar y ejecutar programas COBOL/400
- Comprobar y depurar programas COBOL/400
- Estudiar ejemplos codificados COBOL/400.

Nota: Es preciso efectuar una lectura profunda de los cuatro primeros capítulos antes de proceder a la lectura del resto del manual.

Utilice este manual junto con la publicación *COBOL/400 Reference*, SC09-1813, que describe cada componente y característica del lenguaje COBOL/400. La publicación *COBOL/400 User's Guide*, SC09-1812 y el manual *COBOL/400 Reference* describen pormenorizadamente el lenguaje y el compilador COBOL/400.

Para obtener información sobre toda la biblioteca de documentos AS/400, consulte el manual *Guía de Publicaciones, GC10-9237 (GC41-9678)*, que contiene una descripción resumida del contenido de cada publicación AS/400.

Antes de proceder a la lectura de este manual, es preciso que se familiarice con la siguiente información:

- El uso de los controles e indicadores de la pantalla de visualización, así como la utilización de todas las teclas del teclado, como por ejemplo:
  - Teclas de movimiento del cursor
  - Teclas de función
  - Teclas de salida de campo
  - Teclas de Inserción y Supresión
  - Teclas de Restauración de errores.

  Para obtener información acerca de la estación de pantalla, consulte:
  - *Guía para Nuevos Usuarios, SC10-8881 (SC41-8211)*.

- El funcionamiento de la estación de pantalla si está enlazada con el sistema AS/400 de IBM, así como la ejecución del software AS/400. Ello implica tener conocimientos sobre el uso del sistema operativo OS/400 y sobre el Lenguaje de Control (CL) para poder llevar a cabo las siguientes operaciones:
  - Iniciar y finalizar la sesión en la estación de pantalla
  - Interactuar con pantallas
  - Utilizar la Ayuda
  - Introducir mandatos CL
  - Utilizar las Herramientas para el Desarrollo de Aplicaciones
  - Responder a mensajes
  - Realizar gestión de archivos.
- La publicación *Programación: Lenguaje de Control Guía del Programador, SC10-8977 (SC41-8077)*, que contiene los conceptos fundamentales de las funciones CL de OS/400.

  Para ampliar el conocimiento sobre el sistema operativo y el lenguaje de control, consulte las siguientes publicaciones IBM:
  - *Programming: Control Language Reference, SC41-0030* (manual en tres tomos).
  - *Programación: Guía para la Gestión de Trabajos, SC10-8978 (SC41-8078)*
  - *Advanced Backup and Recovery Guide, SC41-8079*
- La publicación *Guía para la Gestión de Datos, SC10-9008 (SC41-9658)*, que proporciona información acerca de la utilización del soporte de gestión de datos que permite que una aplicación trabaje con archivos.

  Además, el manual incluye información sobre:
  - La estructura y los conceptos fundamentales del soporte de gestión de datos en el sistema
  - El soporte de gestión de datos para estaciones de pantalla, impresoras, cintas y disquetes, así como soporte de spool
  - La redirección de alteraciones temporales de archivos (eventualmente realizando cambios en los archivos cuando se ejecuta una aplicación)
  - La copia de archivos mediante mandatos del sistema para copiar datos de un lugar a otro
  - La adaptación de un sistema utilizando datos de doble byte.
- El uso de las siguientes Herramientas para el Desarrollo de Aplicaciones:
  - El Programa de Utilidad para la Ayuda del Diseño de Pantallas (SDA), que se utiliza para diseñar y codificar pantallas. En la publicación *Application Development Tools: Screen Design Aid User's Guide and Reference, SC09-1340* aparece más información referente a este producto.
  - El Programa de Utilidad para la Entrada del Fuente (SEU), que es un editor de pantalla completa que puede utilizarse para introducir y actualizar los miembros fuente. En la publicación *Programa de Utilidad para Entrada del Fuente (SEU) Guía del Usuario y Manual de Consulta, SC10-9018 (SC09-1338)* aparece más información referente a este producto.
- El Lenguaje de Consulta Estructurada (SQL), que permite insertar instrucciones SQL en los programas COBOL/400. En las publicaciones *SAA* Lenguaje de Consulta Estructurada SQL/400 Manual de Consulta, SC10-8997 (SC41-9608)*

y *SAA* Lenguaje de Consulta Estructurada SQL/400 Guía del Programador, SC10-8998 (SC41-9609)* aparece más información referente a este producto.

- El programa bajo licencia Customer Information Control System/400 (CICS/400*), que permite introducir transacciones en estaciones de trabajo remotas y procesarlas simultáneamente mediante programas de aplicaciones escritas por el usuario. El programa bajo licencia incluye funciones para crear, utilizar y efectuar un mantenimiento de bases de datos y para poder establecer comunicación con CICS con otros sistemas operativos.

  En la *Guía de Programación de Aplicaciones CICS/400*, SC33-0822 aparece más información sobre cómo utilizar este producto para la programación de aplicaciones.

Estándares Industriales Utilizados en el Diseño del Compilador

El compilador COBOL/400 se ha diseñado de acuerdo con los estándares industriales siguientes interpretados por IBM, como en septiembre de 1987:

- El subconjunto intermedio estándar del American National Standards Institute (ANSI X3.23-1985).
- La International Standards Organization (ISO) 1989-1985.
- El nivel intermedio de la Federal Information Processing Standards Publication (FIPS PUB 21-2) de marzo de 1986. Se ofrece soporte adicional para muchos dispositivos de nivel superior.

Algunos fragmentos de este manual se han copiado del *American National Standard Programming Language COBOL, ANSI X3.23-1985, ISO 1989-1985* y se han reproducido con la autorización del *American National Standard Programming Language COBOL, ANSI X3.23-1985, ISO 1989-1985* (copyright 1985 de American National Standards Institute). Estas copias pueden adquirirse en el American National Standard Institute, sito en 1430 Broadway, New York, New York, 10018.

La Conference On DAta SYstems Languages (CODASYL) se encarga del mantenimiento del lenguaje COBOL.

Capítulo 1. Introducción al Lenguaje de Programación COBOL/400

COmmon Business Oriented Language (COBOL) es un lenguaje de programación que se parece al inglés. Como su propio nombre indica, COBOL es especialmente eficiente para el proceso de problemas de gestión. Pone énfasis en la descripción y el manejo de ítems de datos y de registros de entrada/salida; de esta manera, se adapta bien a la gestión de grandes archivos de datos.

El lenguaje COBOL/400 proporciona muchos elementos de la Interfaz de Programación Común COBOL (CPI) de la Arquitectura de Aplicación de Sistemas IBM * (SAA*) y es el producto implantado en el sistema AS/400.

El compilador y la biblioteca COBOL/400 es un programa con licencia IBM que acepta y ejecuta programas COBOL que sigan el estándar ANSI X3.23-1985 (*American National Standard Programming Language COBOL, ANSI X3.23-1985, ISO 1989-1985*). ANSI es una organización formada por fabricantes, consumidores y grupos de intereses comunes y generales, que establece los procedimientos por los que las organizaciones acreditadas crean y mantienen unos estándares industriales voluntarios en los Estados Unidos.

Extensiones al Estándar ANSI

Para ayudar al usuario en la utilización del COBOL en el sistema AS/400, el programa con licencia COBOL/400 incluye asimismo una serie de ampliaciones IBM al estándar ANSI X3.23-1985. Entre las ampliaciones más significativas se incluyen:

- TRANSACTION I/O: puede enviar y recibir registros de una estación de trabajo.
- COPY: puede utilizar archivos descritos externamente.
- DATABASE I/O: puede utilizar entradas estándar de División de Datos y de Entorno COBOL para especificar la identificación de archivos, las definiciones de campos y las estructuras de datos. Se han añadido cláusulas en los verbos READ, WRITE, REWRITE, DELETE y START para dar soporte a la base de datos del AS/400.
- Tipos de datos ampliados: da soporte a datos computational-3 (decimales internos o decimales empaquetados) y computational-4 (binarios).
- Se da soporte a los tipos de datos booleanos y puntero.
- Tiene la opción de utilizar el apóstrofe en lugar de comillas.
- Se da soporte a las instrucciones dirigidas al compilador SKIP1/2/3, EJECT y TITLE.
- ACCEPT/DISPLAY Ampliado: proporciona soporte para la E/S de la estación de trabajo a nivel de campo.
- Cláusula LIKE: puede definir las características de un nombre de datos copiándolos de un nombre de datos definido anteriormente.

- Supresión del listado del compilador: puede suprimir de forma selectiva las partes del listado del compilador utilizando la instrucción *CBL o *CONTROL o la frase SUPPRESS de la instrucción COPY.
- Se da soporte a literales no numéricos hexadecimales.

Características del Compilador COBOL/400

Las siguientes características independientes del lenguaje están disponibles con el compilador COBOL/400:

- Comprobación de sintaxis:

  El Programa de Utilidad para la Entrada del Fuente (SEU) proporciona un comprobador de sintaxis COBOL que comprueba los errores en líneas de código mientras el usuario los entra o los cambia. Se visualizan los mensajes de error, lo que le permite corregirlos antes de la compilación.
- Opción de referencia cruzada:
  - Ofrece un listado de cada nombre de División de Datos y nombre de párrafo de División de Procedimientos.
  - Indica los números de instrucción de cada referencia para el ítem.
- Supresión de mensajes de diagnósticos por debajo del nivel especificado por el usuario.
- El distintivo del Federal Information Processing Standard (FIPS) emite mensajes que identifican elementos del lenguaje obsoletos o que no están conformes en el programa fuente COBOL. Un **programa fuente** es un conjunto de instrucciones que se escribe en un lenguaje de programación y que debe convertirse en lenguaje de máquina antes de que se ejecute el programa.
- El distintivo SAA se utiliza para destacar las funciones del programa que no se pueden transportar a otros entornos COBOL de SAA.

Utilización de la Notación de Sintaxis COBOL/400

En COBOL, los formatos básicos se presentan en un sistema uniforme de notación de sintaxis que se define en los párrafos siguientes. Esta notación está diseñada para ayudarle a escribir instrucciones fuente COBOL.

- Las palabras clave COBOL aparecen en letras mayúsculas; por ejemplo:

  ```
  PARM1
  ```

  Deben escribirse tal y como se muestra. Si no se escribe alguna palabra clave necesaria, el compilador lo considera un error.
- Las variables que representan nombres o valores suministrados por el usuario aparecen en letras minúsculas; por ejemplo:

  ```
  parmx
  ```
- Para una consulta más fácil del texto, algunas palabras aparecen seguidas de un guión y un dígito o una letra; por ejemplo:

  ```
  identifier-1
  ```

  Este sufijo no cambia la definición sintáctica de la palabra.

- Los operadores aritméticos y lógicos (+, -, *, /, **, >, <, =, >=, y <=) que aparecen en formatos de sintaxis son necesarios. Estos operadores son palabras reservadas como *caracteres especiales*. Para obtener un listado completo de palabras COBOL/400 reservadas, consulte la sección “Palabras Reservadas” de la publicación *Consulta* COBOL/400
- Todos los signos de puntuación y demás caracteres especiales que aparecen en el diagrama son requeridos por la sintaxis del formato cuando se muestran; si no los incluye, el programa dará un error.
- Deberá escribir las cláusulas necesarias y las cláusulas opcionales (cuando proceda) en el orden que muestra el diagrama, a menos que las reglas asociadas indiquen lo contrario de manera explícita.

Lectura de los Diagramas de Sintaxis

En este manual, la sintaxis se describe utilizando la estructura definida a continuación.

- Lea los diagramas de sintaxis de izquierda a derecha y de arriba a abajo, siguiendo la línea:

  ►►— Indica el inicio de una instrucción. Los diagramas de unidades sintácticas que no sean instrucciones, como pueden ser las cláusulas, frases y párrafos, también se inician con este símbolo.

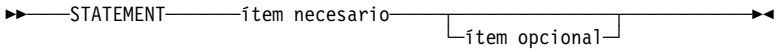
  —► Indica que la sintaxis de la instrucción continúa en la línea siguiente.

  ►— Indica que una instrucción viene de la línea anterior.

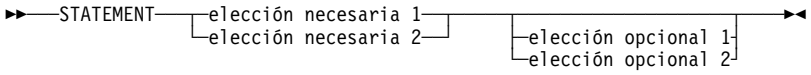
  —►◄ Indica el final de una instrucción. Los diagramas de unidades sintácticas que no sean instrucciones, como pueden ser las cláusulas, frases y párrafos, también terminan con este símbolo.

  Nota: Las instrucciones dentro de un diagrama de un párrafo completo no comenzarán con ►►— ni terminarán con —►◄ a menos que su comienzo y final coincida con el del párrafo.

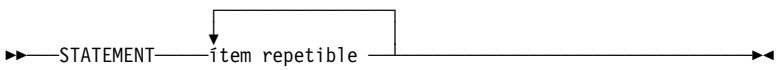
- Los ítems necesarios aparecen en la línea horizontal (la vía principal). Los ítems opcionales aparecen por debajo de la línea principal:



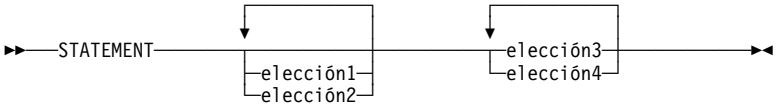
- Cuando pueda escoger entre dos o más ítems, aparecerán verticalmente, en una pila. Si *debe* escoger uno de los dos ítems, aparecerá un ítem de la pila en la línea principal. Si la elección de un ítem es opcional, aparecerá la pila entera por debajo de la línea principal:



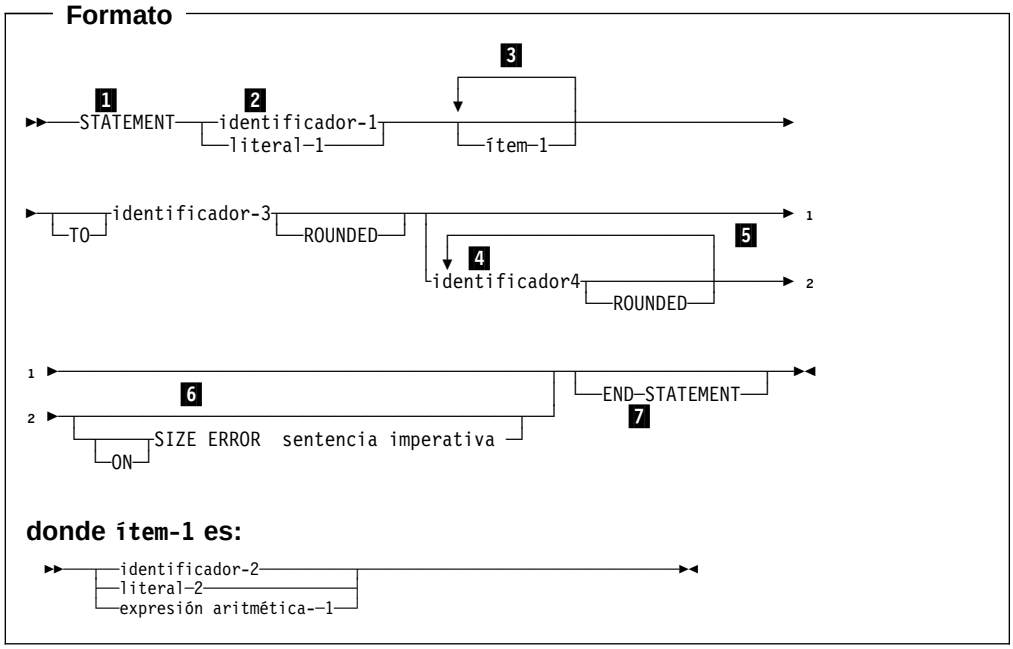
- Una flecha que vuelve hacia la izquierda por encima de un ítem indica que dicho ítem puede repetirse:



- Un flecha de repetición por encima de una pila de elecciones necesarias u opcionales indica que puede realizar más de una elección en los ítems apilados, o repetir una sola elección:



El ejemplo siguiente muestra cómo se utiliza la sintaxis:



1 La palabra clave STATEMENT debe especificarse y codificarse tal y como se muestra.

2 Este operando es necesario. Debe codificarse el *identificador-1* o el *literal-1*.

3 El operando *ítem-1* es opcional. Puede codificarse o no, según sea requerido o no por la aplicación. Si se codifica, debe repetirse, con cada entrada separada por uno o más espacios en blanco. Las selecciones de entrada permitidas para este operando se describen al final del diagrama.

4 El operando *identificador-4* es opcional. Si se especifica, debe repetirse con uno o más espacios en blanco separando cada entrada. A cada entrada se le puede asignar la palabra clave ROUNDED.

5 En los casos en los que varias líneas deban continuar más allá del margen derecho, se preserva el orden de línea de arriba a abajo.

6 La palabra clave ON es opcional para la palabra clave SIZE ERROR, que es opcional por sí misma. Si se codifica SIZE ERROR, entonces es necesario el operando *sentencia imperativa*.

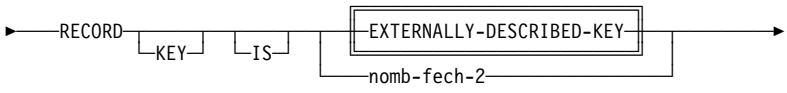
7 La palabra clave END-STATEMENT puede codificarse al final de la instrucción. No es un delimitador necesario.

Lectura de las Ampliaciones IBM

Una ampliación IBM generalmente añade o contradice una regla o restricción que la precede inmediatamente. Primero se presenta el estándar, porque algunos programadores utilizan el lenguaje COBOL/400 sin ampliaciones IBM. Después se presenta la ampliación para aquellos que *sí* las utilizan.

Las ampliaciones IBM dentro de figuras o tablas se muestran en recuadros a menos que se identifiquen explícitamente como ampliaciones.

Las cláusulas e instrucciones que aparecen dentro de diagramas de sintaxis que sean ampliaciones del lenguaje COBOL/400 para ANSI X3.23-1985 COBOL se incluyen en líneas dobles, tal y como se muestra a continuación:

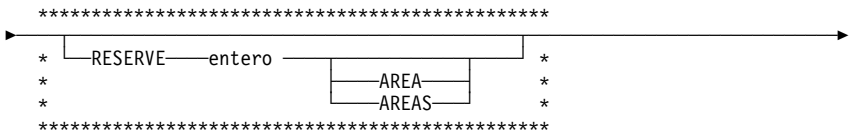


Ampliación de IBM

Las ampliaciones del lenguaje COBOL/400 para ANSI X3.23-1985 COBOL que forman parte de la descripción de texto están encerradas en líneas de Ampliación IBM, como en este párrafo.

Fin de Ampliación de IBM

Las cláusulas e instrucciones COBOL que aparecen en los diagramas que se comprueban por sintaxis, pero que el compilador COBOL/400 trata como documentación, están encerradas dentro de asteriscos, de la manera siguiente:



Códigos de Entrada del Lenguaje de Control

El recuadro que aparece en la parte inferior derecha de cada diagrama de sintaxis CL contiene los códigos de entrada que especifican el entorno en el que puede entrarse el mandato. Los códigos indican si el mandato:

- puede utilizarse en un trabajo interactivo o de proceso por lotes (fuera de un programa compilado; Trabajo:B o I)
- Puede utilizarse en un programa compilado interactivo o de proceso por lotes (Pgm:B o I)
- Puede utilizarse en un procedimiento REXX interactivo o de proceso por lotes (REXX:B o I)

- Puede utilizarse como parámetro para el mandato CL CALL, o para pasar como serie de caracteres al programa de sistema QCMDEXC (Exec).

Visión General de la Programación en COBOL/400

El usuario sigue cuatro pasos o fases principales para crear el programa COBOL/400:

- Introducción del programa fuente.
- Compilación del programa fuente.
- Depuración del programa.
- Ejecución del programa compilado.

Introducción del Programa COBOL

El Programa de Utilidad para la Entrada del Fuente (SEU) proporciona una pantalla especial que se corresponde con la codificación estándar COBOL para ayudarle a entrar un programa fuente COBOL adecuado en el sistema. El SEU proporciona también un comprobador de sintaxis COBOL que comprueba los errores de cada línea mientras el usuario los entra o los cambia. Para obtener más información sobre la introducción del programa fuente COBOL/400, consulte el Capítulo 2, “Instalación del Programa Fuente en el Sistema AS/400”. Para obtener más información sobre la utilización del SEU, consulte la *SEU Guía del Usuario y Manual de Consulta*.

Compilación del Programa COBOL

Una vez entrado el programa fuente en el sistema, el usuario deberá compilar dicho programa fuente utilizando el mandato Create Programa COBOL (CRTCBLPGM). El compilador se llama para crear un programa objeto COBOL y un listado. Un **programa objeto** es un conjunto de instrucciones que la máquina puede utilizar. Un compilador produce el programa objeto desde un programa fuente.

El usuario puede especificar varias opciones del compilador utilizando el mandato CRTCBLPGM o utilizando la instrucción PROCESS con las opciones deseadas. Cualquier opción especificada en la instrucción PROCESS altera temporalmente las opciones correspondientes al mandato CRTCBLPGM. Este proceso se explica detalladamente en el Capítulo 3, “Compilación de un Programa COBOL/400”.

Depuración del Programa COBOL

El sistema operativo OS/400 proporciona las funciones siguientes, que el usuario puede utilizar para probar y depurar programas:

- Biblioteca de prueba
- Puntos de interrupción
- Rastreos.

El compilador COBOL/400 ofrece las funciones siguientes para la comprobación y depuración de programas:

- Características de depuración
- Vuelco con formato.

Estas características le permiten supervisar las operaciones de programas específicos durante el tiempo de ejecución. Debe decidir qué es lo que debe supervisar y qué información recuperar para los propósitos de depuración.

Consulte el Capítulo 5, “Depuración del Programa” para obtener más información sobre las características de la depuración.

Ejecución del Programa COBOL

Podrá ejecutar el programa COBOL de muchas maneras, siempre en función de la persona que lo haya escrito, de la manera de escribirlo y del usuario. Podrá ejecutar un programa COBOL llamándolo desde un programa CL, desde un programa de aplicación, desde otro programa de lenguaje de alto nivel o desde un mandato creado por el usuario.

Cuando se acaba el programa, el sistema devuelve el control al que lo ha llamado.

Para más información acerca de la ejecución del programa, consulte el Capítulo 4, “Ejecución del Programa en COBOL”.

Capítulo 2. Instalación del Programa Fuente en el Sistema AS/400

Este capítulo proporciona la información necesaria para instalar el programa. Este capítulo también describe brevemente las herramientas y la metodología que necesarias para completar la instalación.

Hay dos formas de instalar un programa fuente COBOL en el sistema:

- Utilizando el Programa de Utilidad para la Entrada del Fuente (SEU). Este método se explica con detalle en este capítulo.
- Desde un disquete o cinta utilizando la función de copia del OS/400.

  Consulte la publicación *CL Reference* para obtener información adicional sobre cómo utilizar la función COPY para la entrada del programa fuente desde disquete o cinta.

Para instalar el programa fuente COBOL utilizando el SEU, especifique el mandato Arrancar Programa de Utilidad para la Entrada del Fuente (STRSEU) y especifique CBL para el parámetro TYPE. Consulte la publicación *SEU Guía del Usuario y Manual de Consulta* para obtener más información sobre el mandato STRSEU y sobre el uso del SEU.

Diseño del Programa COBOL/400

Se puede utilizar la estructura de programa de la Figura 1 en la página 10 como modelo para desarrollar programas COBOL legibles y eficientes. Observe que todas las entradas que aparecen a continuación no son necesarias; la mayoría sólo se proporcionan con finalidades informativas.

```
IDENTIFICATION DIVISION. 1
  PROGRAM-ID. nombre de programa.
  AUTHOR. entrada de comentarios.
  INSTALLATION. entrada de comentarios.
  DATE-WRITTEN. entrada de comentarios.
  DATE-COMPILED. entrada de comentarios.
  SECURITY.
*    EL párrafo SECURITY puede contener el prólogo del
*    programa. El prólogo contiene una descripción del programa
*    y puede ser tan descriptivo o breve como recomienden las
*    directrices de instalación. Se recomienda utilizar las
*    minúsculas para los comentarios; no obstante, puesto que
*    algunas impresoras pueden imprimir sólo letras en
*    mayúsculas, los comentarios pueden escribirse en
*    mayúsculas. El subrayado sirve para resaltar los
*    comentarios.

ENVIRONMENT DIVISION. 2
 CONFIGURATION SECTION. 3
  SOURCE-COMPUTER. IBM-AS400.
  OBJECT-COMPUTER. IBM-AS400.
  SPECIAL-NAMES.   REQUESTOR IS CONSOLE.
 INPUT-OUTPUT SECTION. 4
  FILE-CONTROL.
    SELECT nombre de archivo ASSIGN TO DISK nombre de archivo
      ORGANIZATION IS SEQUENTIAL
      ACCESS MODE IS SEQUENTIAL
      FILE STATUS IS nombre de datos.

DATA DIVISION. 5
 FILE SECTION.
  FD nombre de archivo
      BLOCK CONTAINS 2 RECORDS
      RECORD CONTAINS 80 CHARACTERS
      LABEL RECORDS ARE STANDARD
      DATA RECORD IS nombre de registro
 01 nombre de registro  PIC X(132).
 WORKING-STORAGE SECTION.
 77 nombre de datos PIC XX.
 LINKAGE SECTION.

PROCEDURE DIVISION. 6
 DECLARATIVES
 END DECLARATIVES.
 proceso principal SECTION.
  párrafo principal.
      instrucciones COBOL.
         STOP RUN.
```

Figura 1. Ejemplo de la Estructura del Programa COBOL/400

Identification Division **1** es la única división que debe incluirse; todas las otras divisiones son opcionales.

Environment Division **2** consta de dos secciones: Configuration Section **3** , que describe las las especificaciones generales de los sistemas objeto y fuente y Input-Output Section **4** , que define cada archivo y especifica la información que se necesita para transmitir los datos entre un medio externo y el programa COBOL.

Data Division **5** describe los archivos que se deben utilizar en el programa, así como los registros que contienen. También describe cualquier ítem de datos del almacenamiento de trabajo que sea necesario.

Procedure Division **6** se compone de declarativas opcionales, procedimientos que contienen secciones y/o párrafos, sentencias e instrucciones.

Formato del Archivo Fuente

La longitud de registro estándar de los archivos fuente es de 92 caracteres. Estos 92 caracteres están compuestos por un número de secuencia de 6 caracteres, un área de fecha de última modificación de 6 caracteres y un campo de datos de 80 caracteres.

El compilador COBOL/400 soporta una longitud de registro adicional de 102; al final del registro (posiciones 92-102), se coloca un campo de 10 caracteres que contiene información suplementaria. El compilador COBOL no utiliza esta información, pero se coloca en el extremo derecho del listado del compilador. El usuario es el responsable de situar la información en este campo. Si quiere utilizar este campo adicional, cree un archivo fuente con una longitud de registro de 102.

IBM proporciona un archivo fuente en el que se pueden almacenar registros fuente si es que no desea crear su propio archivo. Este archivo, denominado QLBLSRC, se encuentra en la biblioteca QGPL y su longitud de registro es de 92 caracteres.

Instalación del Fuente Mediante el SEU

El SEU proporciona unos formatos de pantalla especiales para COBOL. Corresponden al Formulario de Codificación COBOL que se ha diseñado para facilitar la instalación del programa fuente COBOL. La Figura 2 muestra un formato de pantalla, la relación entre los encabezamientos en el Formulario de Codificación COBOL y las etiquetas de la pantalla; también indica el lugar en el que se debe entrar el código fuente.

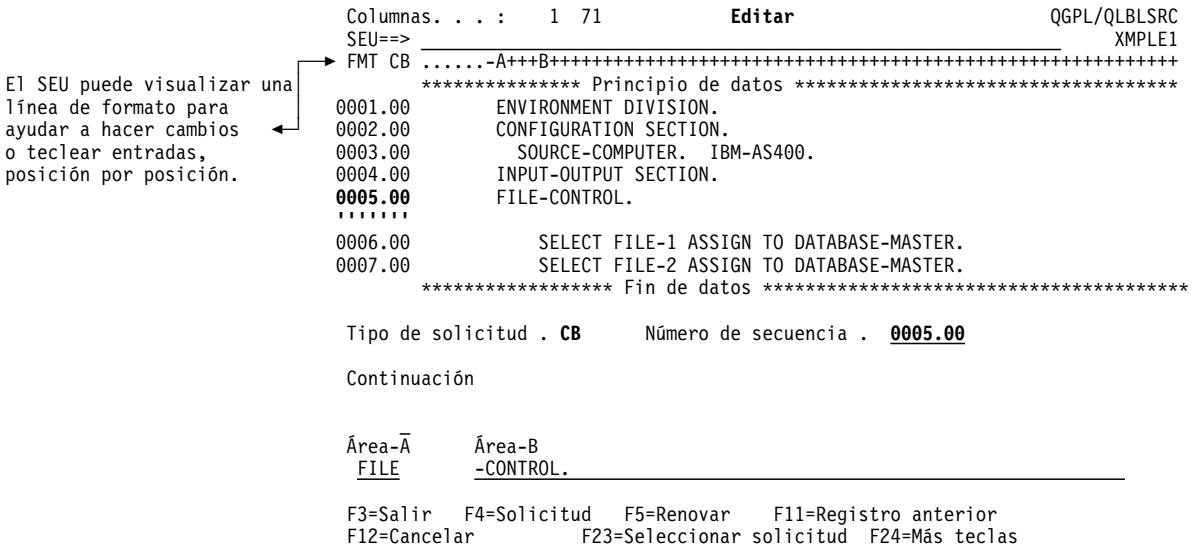


Figura 2. Formato de Pantalla SEU

Para obtener una descripción completa sobre la instalación del programa mediante el SEU, consulte el manual *SEU Guía del Usuario y Manual de Consulta*.

Utilización en el SEU del Comprobador de Sintaxis COBOL

Para utilizar el comprobador de sintaxis COBOL en el SEU, especifique el parámetro TYPE(CBL) del mandato STRSEU. El comprobador de sintaxis COBOL comprueba cada línea en busca de errores mientras introduce líneas nuevas o cambia líneas existentes. Se identifican las instrucciones fuente incorrectas y se visualizan mensajes de error, lo que permite corregir los errores antes de compilar el programa. Puesto que el comprobador de sintaxis COBOL sólo comprueba una instrucción a la vez, independientemente de las instrucciones que la preceden o la siguen, sólo pueden detectarse los errores de sintaxis dentro de los datos fuente. No se detectan errores interrelacionales como por ejemplo los nombres no definidos y las referencias incorrectas dirigidas a nombres. El compilador COBOL detecta estos errores cuando se compila el programa.

Cada vez que se entra o modifica una línea fuente, se puede comprobar la sintaxis de 496 líneas de código fuente como una unidad. La longitud de una sola unidad de comprobación de sintaxis viene determinada por la ampliación de una línea entrada o modificada:

> Una unidad de comprobación de sintaxis se extiende hacia el comienzo de un miembro fuente hasta la primera línea fuente, o hasta que se encuentra una línea que finalice en un punto.

Una unidad de comprobación se extiende hacia el final del miembro fuente hasta la última línea fuente, o hasta que se encuentra una línea que finalice en punto.

Si esta unidad tiene más de 496 líneas fuente (sin incluir las líneas de comentarios) el sistema responde con un mensaje adecuado.

Si se produce un error en una unidad de comprobación de sintaxis, toda la unidad se presenta en contraste invertido. El mensaje del final de la pantalla hace referencia al primer error de la unidad.

La comprobación de sintaxis se produce línea a línea mientras se introduce el código fuente. Las líneas que constan de instrucciones incompletas generan mensajes de error. Estos mensajes desaparecen cuando se completan las instrucciones, tal como sucede en el siguiente ejemplo:

```
ADD A
TO BCD.
```

Un mensaje de error se genera una vez que se especifica la primera línea y desaparece cuando se especifica la segunda línea, una vez completada la instrucción. Una instrucción COBOL puede extenderse en un máximo de 496 líneas. Asimismo, si una línea fuente se especifica o se modifica, puede comprobarse la sintaxis de hasta 496 líneas de códigos fuente como una unidad.

Las siguientes regulaciones se aplican a la comprobación de sintaxis de las funciones fuente COBOL:

- No se comprueba la sintaxis en el código fuente de una línea con un asterisco (*) o una barra (/) en la columna 7. Un asterisco indica una línea de comentarios; una barra inclinada indica una línea de comentarios y un salto de página.
- No se acepta ninguna opción del compilador durante la comprobación de sintaxis.

  Por ejemplo, el comprobador de sintaxis acepta las comillas dobles o simples como delimitadores no numéricos, siempre y cuando no se mezclen dentro de una unidad de comprobación de sintaxis. El comprobador de sintaxis permanece inactivo si el delimitador es el mismo que se especificará en el mandato CRTCBLPGM para compilar instrucciones fuente COBOL o en la instrucción PROCESS.
- La primera sentencia que sigue a cualquier encabezamiento de párrafo que se lista a continuación debe empezar en la misma línea que el encabezamiento del párrafo.

  ```
  PROGRAM-ID.
  AUTHOR.
  INSTALLATION.
  DATE-WRITTEN.
  DATE-COMPILED.
  SECURITY.
  SOURCE-COMPUTER.
  OBJECT-COMPUTER.
  SPECIAL-NAMES.
  ```
- La sustitución especificada por las cláusulas CURRENCY y DECIMAL-POINT del párrafo SPECIAL-NAMES no se aceptan durante la comprobación de sintaxis interactiva.

- Cuando se utiliza la cláusula REPLACING *Identificador-1* BY *Identificador-2* de la instrucción COPY y cuando alguno de los identificadores incluye modificación de referencias, SEU comprueba únicamente que los paréntesis coincidan. Para obtener más información sobre la modificación de referencias, consulte el Capítulo 11, “Consideraciones de Programación en COBOL/400”.

Sintaxis de las Instrucciones del Lenguaje de Consulta Estructurada (SQL)

La sintaxis de las instrucciones SQL en un programa fuente COBOL es la siguiente:

Si el tipo de miembro del programa fuente es SQLCBL o CICSSQLCBL, cuando el comprobador de sintaxis COBOL detecta una instrucción SQL, la instrucción se pasa al comprobador de sintaxis SQL. Si se detecta un error, se devuelve un mensaje.

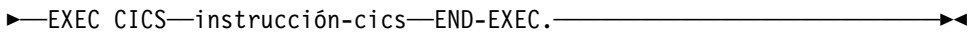
Si se detecta una instrucción SQL y el tipo de miembro no es SQLCBL ni CICSSQLCBL, se devuelve un mensaje COBOL que indica que una instrucción COBOL es errónea.

Si hay errores en la instrucción SQL incluida además de errores en las anteriores instrucciones COBOL, el mensaje de error SQL sólo se visualizará después de corregir los errores COBOL anteriores.

Para obtener más información sobre las instrucciones SQL, consulte el manual *SQL/400* Manual de Consulta*.

Sintaxis de las Instrucciones del Sistema de Control de Información del Cliente (CICS)

La sintaxis de las instrucciones CICS incluidas de un programa fuente COBOL es la siguiente:



Si el tipo de miembro del programa fuente es CICSCBL o CICSSQLCBL, cuando el comprobador de sintaxis COBOL detecta una instrucción CICS, el comprobador de sintaxis busca únicamente los errores de sintaxis elementales.

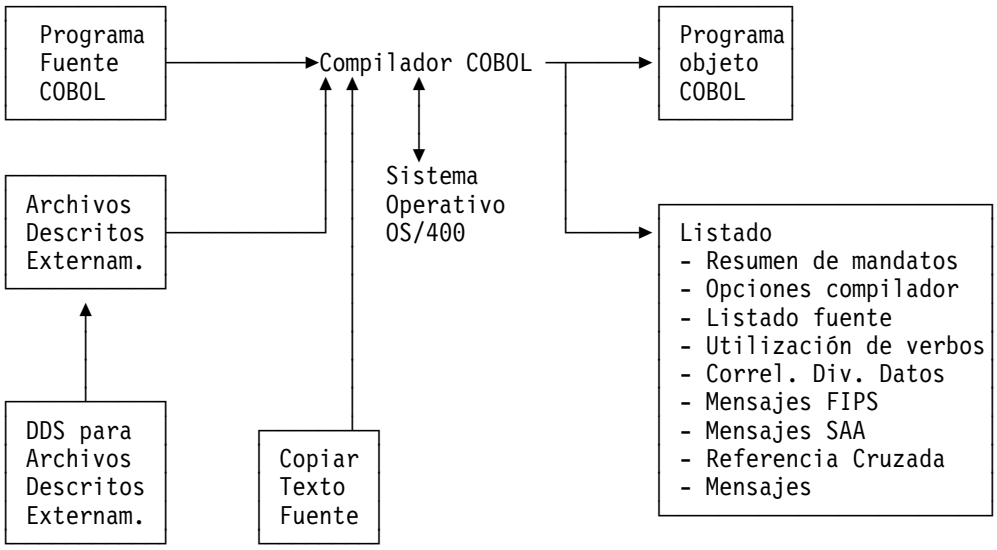
Si se detecta una instrucción CICS y el tipo de miembro no es CICSCBL ni CICSSQLCBL, se devuelve un mensaje COBOL que indica que la instrucción COBOL es errónea.

Para obtener más información sobre las instrucciones CICS/400 consulte la publicación *CICS/400 Application Programming Guide*.

Capítulo 3. Compilación de un Programa COBOL/400

Es preciso compilar el programa fuente COBOL/400 para producir un programa objeto que se pueda ejecutar. Esto se realiza con el mandato Crear Programa COBOL (CRTCBLPGM). El resultado de la compilación es un programa objeto COBOL y un listado.

Se pueden especificar diferentes opciones de compilador mediante el mandato CRTCBLPGM, o utilizando la instrucción PROCESS en el propio programa. Cualquier opción especificada en la instrucción PROCESS prevalce sobre las opciones correspondientes del mandato CRTCBLPGM. La instrucción PROCESS se trata más adelante en el apartado “Uso de la Instrucción PROCESS para Especificar las Opciones del Compilador” en la página 33.



Durante la compilación, el compilador comprueba tanto la sintaxis del programa fuente COBOL línea a línea como las relaciones entre líneas.

Utilización del Mandato Crear Programa COBOL (CRTCBLPGM)

Para compilar un programa fuente COBOL/400 a un programa objeto, debe entrar el mandato CRTCBLPGM. Este mandato llama al compilador COBOL/400. Se puede utilizar el mandato CRTCBLPGM interactivamente o en trabajos por lotes, o bien desde otros programas del sistema AS/400.

Nota de Programación: El número de entradas de la Tabla de Definición de Objetos (ODT) y la cantidad de almacenamiento que necesita un programa COBOL, varían según la cantidad y tipo de instrucciones COBOL que se utilicen en el programa. La combinación de estos factores puede provocar un aumento de los límites del tamaño interno del AS/400 para el programa. Si se produce esta anomalía, utilice la opción *NOUNREF del parámetro GENOPT. Si el problema persiste, el programa debe escribirse de nuevo.

Cuando se especifica la opción *NOUNREF, sólo se definen los nombres que estén referenciados o que sean necesarios para la estructuración de datos. Esta opción es útil cuando el programa contiene muchos identificadores sin referencia.

Si no se ha especificado CBL como el tipo de miembro fuente, el compilador emite un aviso.

Si se utiliza la instrucción COPY de Formato 2 en el programa para acceder a los archivos descritos externamente, el sistema operativo proporciona información al programa referente a dichos archivos.

Si el compilador COBOL se detiene, se emite el mensaje LBL9001

```
Compilación anómala. Programa no creado.
```

Se puede utilizar un programa de lenguaje de control que puede supervisar esta excepción utilizando el mandato Supervisar Mensaje (MONMSG).

Utilización de las Pantallas de Solicitud CRTCBLPGM

Para introducir el mandato CRTCBLPGM desde las pantallas de solicitud CRTCBLPGM, escriba `CRTCBLPGM` y pulse F4 (Solicitud) para que aparezca la primera pantalla. Los parámetros y opciones se describen a medida que aparecen por pantalla, de la página 18 a la 28. Los valores por omisión, que aparecen subrayados, se definen en primer lugar.

Cada parámetro de esta pantalla muestra un valor por omisión. Desplace el cursor más allá de los ítems en los que desee aplicar los valores por omisión. Escriba encima de cualquier ítem para establecer los distintos valores u opciones. Si no está seguro acerca del valor del parámetro, teclee un interrogante (?) en la primera posición del campo y pulse Intro o F4 (Solicitud), para recibir más información detallada. El signo de interrogación debe ir seguido de un espacio en blanco.

La Figura 3 muestra las pantallas de solicitud CRTCBLPGM. Cuando se visualiza la primera pantalla de solicitud CRTCBLPGM, tan sólo aparecen los parámetros necesarios que se han solicitado. Para visualizar los parámetros adicionales, pulse F10. Observe la primera pantalla de la Figura 3. Para consultar el recordatorio de los parámetros, tal como se muestra en la segunda y tercera pantalla de la Figura 3, avance página.

```
                     Crear Programa COBOL (CRTCBLPGM)

Escriba la elección, pulse Intro.

Programa . . . . . .   *PGMID        Nonbre, *PGMID
  Biblioteca . . . . . .    *CURLIB     Nombre, *CURLIB
Archivo fuente . . . . . .   QLBLSRC       Nombre
  Biblioteca . . . . . .    *LIBL       Nombre, *LIBL, *CURLIB
Miembro fuente . . . . . .   *PGM          Nombre, *PGM
Nivel de gravedad de generación.   29            0-29
Texto 'descripción'. . . . . .   *SRCMBRTXT


                          Parámetros Adicionales

Opciones de listado fuente . . .                *SOURCE, *NOSOURCE, *SRC...
              + para más valores
Opciones de generación . . . . .                *NOLIST, *LIST, *NOXREF...
              + para más valores

                                                                       Más...
F3=Salir  F4=Solicitud  F5=Renovar  F12=Cancelar  F13=Cómo util. esta pantalla
F24=Más teclas
```

```
                     Crear Programa COBOL (CRTCBLPGM)

Escriba la elección, pulse Intro.

Opciones de conversión . . . . .                *NOVARCHAR, *VARCHAR...

Límite de Mensaje:
  Número de mensajes . . . . . .   *NOMAX        1-9999, *NOMAX
  Gravedad límite de mensaje . .   29            0-29
Imprimir archivo . . . . . .   QSYSPRT       Nombre
  Biblioteca . . . . . .    *LIBL       Nombre, *LIBL, *CURLIB
Señalización FIPS. . . . . .
              + para más valores
Señalización SAA . . . . . .   *NOFLAG       *NOFLAG, *FLAG
Opciones de pantalla ampliada. .                *DFRWRT, *NODFRWRT...
              + para más valores
Gravedad de señalización . . . .   0             0-99
Sustituir programa . . . . . .   *YES          *NO, *YES
Release destino. . . . . .   *CURRENT      *CURRENT, *PRV, V2R1M0...
Perfil usuario . . . . . .   *USER         *USER, *OWNER
                                                                   Más...
F3=Salir  F4=Solicitud  F5=Renovar  F12=Cancelar  F13=Cómo util. esta pantalla
F24=Más teclas
```

```
                     Crear Programa COBOL (CRTCBLPGM)

Escriba la elección, pulse Intro.

Autorización . . . . . .   *LIBCRTAUT    Nombre, *LIBCRTAUT, *ALL...
Vuelco depuración compilador:
                                1             1-65535, *
                                65535         1-65535
Vuelco texto intermedio. . . . .   0             0-31









                                                                     Final
F3=Salir  F4=Solicitud  F5=Renovar  F12=Cancelar  F13=Cómo util. esta pantalla
F24=Más teclas
```

Figura 3. Pantallas de Solicitud CRTCBLPGM

Parámetros de los Mandatos CRTCBLPGM

A continuación se muestra una descripción de los parámetros para el mandato CRTCBLPGM. Primero se definen los valores por omisión que están subrayados para su identificación. Los parámetros y opciones se describen en el orden en que aparecen en la pantalla de solicitud.

Todos los nombres de objetos especificados para el mandato CRTCBLPGM deben respetar los convenios de denominación de AS/400: deben ser nombres básicos, de 10 caracteres de longitud, estar compuestos por caracteres alfanuméricos, el primero de los cuales debe ser alfabético; o bien pueden ser nombres entrecomillados de 8 caracteres de longitud.

Si desea relacionar estas descripciones de parámetros con el diagrama de sintaxis CRTCBLPGM, consulte la Figura 4 en la página 30.

Parámetro PGM:

Especifica el nombre del programa así como el nombre de biblioteca para el objeto programa COBOL que se está creando. Los valores posibles son:

***PGMID**

El nombre del objeto programa se obtiene del párrafo PROGRAM-ID del programa fuente COBOL.

nombre-programa

Entre un nombre para identificar el programa COBOL compilado. Si especifica un nombre de programa para este parámetro, y la compilación se ejecuta por lotes, el primer programa del trabajo por lotes utilizará este nombre; cualquier otro programa utilizará el nombre especificado en el párrafo PROGRAM-ID del programa fuente.

Los posibles valores de la biblioteca:

***CURLIB**

Si no especifica un nombre de biblioteca, se utilizará la biblioteca actual. Si no ha asignado una biblioteca como biblioteca actual, se utiliza QGPL.

nombre-biblioteca

Entre el nombre de la biblioteca que deberá contener el objeto programa creado.

Parámetro SRCFILE:

Especifica el nombre del archivo fuente que contiene el fuente COBOL a compilar. Los valores posibles son:

QLBLSRC

Especifica que el archivo fuente, QLBLSRC, contiene el fuente COBOL a compilar.

nombre-archivo-fuente

Entre el nombre del archivo fuente que contiene el fuente COBOL que se debe compilar. Este archivo fuente debe tener 92 caracteres de longitud de registro.

Los valores posibles de la biblioteca son:

***LIBL**

Si no especifica ningún nombre de biblioteca, el sistema busca en la lista de bibliotecas la biblioteca en la que se ubica el archivo fuente.

***CURLIB**

Se utiliza la biblioteca actual. Si no ha asignado una biblioteca como biblioteca actual, se utiliza QGPL.

nombre-biblioteca

Entre el nombre de la biblioteca que contiene el archivo fuente.

Parámetro SRCMBR:

Especifica el nombre del miembro que contiene el fuente COBOL que se debe compilar. Este parámetro sólo se puede especificar si el archivo fuente que se referencia en el parámetro SRCFILE es un archivo de base de datos: Los valores posibles son:

***PGM**

Si ha especificado un nombre de programa para el parámetro PGM, el compilador busca el programa fuente en un miembro que tenga el mismo nombre que el programa, y crea un programa objeto con el mismo nombre que el programa y miembro.

Si no ha especificado ningún nombre de programa para el parámetro PGM, el compilador busca el programa fuente en el primer miembro del archivo fuente de la base de datos, y crea un programa objeto que utiliza el nombre especificado en el párrafo PROGRAM-ID.

nombre-miembro-archivo-fuente
Entre el nombre del miembro que contiene el fuente COBOL.

Parámetro GENLVL:
Especifica el nivel de gravedad que determina si un objeto programa se crea. El nivel de gravedad se corresponde al nivel de gravedad de los mensajes que se han producido durante la compilación del programa. Si el nivel de gravedad de los mensajes de error es mayor que el valor que ha especificado, no se crea ningún objeto programa. Por ejemplo, si especifica 19 para este parámetro, no se creará ningún objeto programa si el nivel de gravedad de alguno de los mensajes es 20 o mayor.

Los valores posibles son:

29 Si se produce un error con un nivel de gravedad mayor que 29, no se creará ningún objeto programa.

Nivel-gravedad
Especifique un número de uno o dos dígitos del 0 al 29. Si se produce un error con un nivel de gravedad mayor que este nivel, no se creará ningún objeto programa.

Parámetro TEXT:
Permite introducir el texto que describe brevemente el programa y su función.

***SRCMBRTXT**
Utilice el mismo texto para el objeto programa que describe el miembro del archivo de base de datos que contiene el fuente COBOL. Si el fuente proviene de un dispositivo o de un archivo incorporado, el especificar *SRCMBRTXT tiene el mismo efecto que especificar *BLANK.

***BLANK**
No se especifica ningún texto.

descripción-texto
Introduzca el texto que describe brevemente el programa y sus funciones. La longitud del texto no puede superar los 50 caracteres y debe ir entre comillas simples. Las comillas simples no forman parte de la serie de 50 caracteres.

Parámetro OPTION:
Especifica las opciones que se deben utilizar cuando se compila el fuente COBOL. Los valores posibles son:

***SOURCE o *SRC**
El compilador produce un listado fuente compuesto por las las entradas fuente COBOL y por los mensajes de error en tiempo de compilación.

***NOSOURCE o *NOSRC**
El compilador no produce la parte fuente del listado. Si no se necesita un listado fuente, utilice esta opción, ya que de este modo la compilación se efectuará con más rapidez.

***NOXREF**
El compilador no produce un listado de referencias cruzadas para el programa fuente.

***XREF**
El compilador produce un listado de referencias cruzadas para el programa fuente.

***GEN**
El compilador crea un objeto programa una vez compilado el programa.

***NOGEN**
El compilador no crea un objeto programa una vez compilado el programa. Especifique esta opción si sólo desea obtener listados de error.

***NOSEQUENCE**
Los números de referencia no se comprueban para errores de secuencia.

***SEQUENCE**
Los números de referencia se comprueban para errores de secuencia. Los

errores de secuencia no se producen si se especifica la opción *LINENUMBER.

***NOVBSUM**
El número total de utilizaciones del verbo no se imprime.

***VBSUM**
El número total de utilizaciones del verbo se imprime.

***NONUMBER**
Los números de secuencia del archivo fuente se utilizan para los números de referencia.

***NUMBER**
Los números de secuencia suministrados por el usuario (columnas 1 a 6) se utilizan para números de referencia.

***LINENUMBER**
Los números de secuencia que el compilador crea se utilizan para los números de referencia. Esta opción combina el código fuente del programa y el código fuente introducido mediante instrucciones COPY en una secuencia numerada consecutivamente. Utilice esta opción si especifica la señalización FIPS (Federal Information Processing Standards) o la señalización SAA. *

***NOMAP**
El compilador no lista la correlación de División de Datos.

***MAP**
El compilador lista la correlación de División de Datos.

***NOOPTIONS**
No se listan las opciones en activo para esta compilación.

***OPTIONS**
Se listan las opciones en activo para esta compilación.

***QUOTE**
Especifica que el delimitador comillas (") se utiliza para literales no numéricos y para literales booleanos. También especifica que el valor de la constante figurativa QUOTE posee el valor EBCDIC de una comilla.

Nota: Los datos booleanos son una categoría de ítems de datos que están limitados a valores 1 ó 0. Un literal booleano está compuesto por caracteres booleanos entrecomillados y precedidos por una B; por ejemplo: B"1".

***APOST**
Especifica que el delimitador apóstro (') se utiliza para literales no numéricos y para literales booleanos. También especifica que el valor de la constante figurativa QUOTE posee el valor EBCDIC de un apóstrofo.

***NOSECLVL**
No se lista un texto de mensaje de segundo nivel para esta compilación.

***SECLVL**
Se lista un texto de mensaje de segundo nivel para esta compilación junto con el texto de error de primer nivel.

Nota: Se imprime el texto de error de primer nivel cada vez que se produce un error.

***PRTCORR**
El compilador inserta líneas de comentarios en los listados del compilador que indican los ítems elementales que se han incluido como resultado de utilizar la frase CORRESPONDING.

***NOPRTCORR**
El compilador no inserta líneas de comentario en el listado del compilador cuando se utiliza la frase CORRESPONDING.

***NOSRCDBG**
Esta opción determina el tipo de información que se visualiza en la estación de trabajo programable cuando se utiliza el CoOperative Development

Environment/400 para compilar programas COBOL. Véase la nota de la página 21 para obtener más información.

El compilador no produce información de depuración a nivel de fuente. Si *NOLSTDBG también está activa, el compilador tampoco produce información de error a nivel de fuente.

***SRCDBG**

Esta opción determina el tipo de información que se visualiza en la estación de trabajo programable cuando se utiliza el CoOperative Development Environment/400 para compilar programas COBOL. Véase la nota de la página 21 para obtener más información.

El programa produce tanto información de error como de depuración a nivel de fuente.

No es posible especificar *SRCDBG y *LSTDBG juntos. Especifique una de las dos.

***NOLSTDBG**

Esta opción determina el tipo de información que se visualiza en la estación de trabajo programable cuando se utiliza el CoOperative Development Environment/400 para compilar programas COBOL. Véase la nota de la página 21 para obtener más información.

El compilador no permite ver el listado, la información de error a nivel de fuente ni la información de depuración a nivel de listado.

***LSTDBG**

Esta opción determina el tipo de información que se visualiza en la estación de trabajo programable cuando se utiliza el CoOperative Development Environment/400 para compilar programas COBOL. Véase la nota de la página 21 para obtener más información.

El compilador permite ver un listado, así como información de depuración a nivel de listado. Si *NOSRCDBG también está activa, el compilador tampoco produce información de error a nivel de fuente.

No es posible especificar *SRCDBG y *LSTDBG a la vez. Especifique una de las dos.

Nota: Sólo es posible utilizar las opciones *NOSRCDBG, *SRCDBG, *NOLSTDBG y *LSTDBG si se utiliza el producto AD/Cycle CoOperative Development Environment/400 para compilar los programas. Si especifica una de estas opciones o más, pero no tiene instalado el producto CODE/400, el compilador COBOL/400 no continuará el proceso y emitirá un mensaje de error. Para obtener más información acerca de estas opciones, consulte la publicación *CODE Debug Tool User's Guide and Reference*, SC09-1622.

***PRINT**

El compilador produce un listado de spool.

***NOPRINT**

El listado no produce ningún listado de spool.

Parámetro GENOPT:

Especifica las opciones a utilizar cuando el objeto programa se ha creado. Los listados podrían utilizarse si se produce algún error en COBOL. Los valores posibles son:

***NOLIST**

No se lista ninguna IRP (representación intermedia de programa), código hexadecimal asociado ni mensajes de error.

***LIST**

Se lista la IRP, sus códigos hexadecimales asociados y cualquier mensaje de error.

***NOXREF**

No produce un listado de referencias cruzadas de los objetos definidos en la IRP.

***XREF**

Produce un listado de referencias cruzadas de todos los objetos definidos en la IRP.

***NOPATCH**
No reserva un espacio en el programa compilado para un área de parche de programa.

***PATCH**
Reserva un espacio en el programa compilado para un área de parche de programa. El área de parche de programa se puede utilizar para operaciones de depuración.

***NODUMP**
No lista la plantilla del programa.

***DUMP**
Lista la plantilla del programa.

***NOATR**
No lista los atributos de un fuente IRP.

***ATR**
Lista los atributos de un fuente IRP.

***RANGE**
En el tiempo de ejecución, el sistema verifica que los subíndices se encuentren dentro de los rangos correctos, pero no verifica los rangos de los índices. Asimismo, comprueba las modificaciones de referencia y las operaciones de subserie de caracteres generadas por el compilador.

***NORANGE**
No verifica los rangos en el tiempo de ejecución.

Nota: La opción *RANGE genera un código para comprobar los rangos subindexados. Por ejemplo, se asegura de que el usuario no intente acceder al elemento 21 de una matriz de 20 elementos.

La opción *NORANGE no genera un código para comprobar los rangos de subíndices.

Estas opciones no eliminan la comprobación del subíndice cero realizada por el sistema operativo. Si se produce el subíndice cero, el sistema operativo no permitirá su utilización y emitirá el mensaje MCH0603.

***UNREF**
Incluye ítems de datos no referenciados en el programa compilado.

***NOUNREF**
No incluye ítems de datos no referenciados en programas compilados. De este modo, se reduce el número de entradas ODT (tabla de definición de objetos) utilizadas, lo que permite la compilación de un programa más grande. Los ítems de datos no referenciados siguen apareciendo en el listado de referencias cruzadas que se ha producido a raíz de la especificación de OPTION (*XREF).

***NOOPTIMIZE**
El compilador sólo realiza optimizaciones estándar para el programa.

***OPTIMIZE**
El compilador intenta generar un programa que opere de forma más eficiente y que utilice menos almacenamiento. Sin embargo, al especificar *OPTIMIZE el tiempo necesario para compilar el programa puede aumentar considerablemente.

***NODDSFILLER**
Si no se encuentran campos coincidentes mediante una instrucción COPY DDS, no se generará ninguna descripción de campo.

***DDSFILLER**
Cuando no se encuentran campos coincidentes mediante una instrucción COPY DDS, se genera siempre una descripción de campo FILLER de carácter único "07 FILLER PIC X".

***NOSYNC**
Se comprueba la sintaxis de la cláusula SYNCHRONIZED.

***SYNC**
La cláusula SYNCHRONIZED provoca la alineación de un ítem elemental en un límite natural de almacenamiento.

***NOCRTF**
Los archivos que no están disponibles durante una operación OPEN, no se crean dinámicamente.

***CRTF**
Los archivos que no están disponibles durante una operación OPEN se crean dinámicamente. Una vez creados, el archivo obtendrá la autorización del perfil de trabajo. (Consulte la explicación de la instrucción OPEN en el manual *COBOL/400 Reference* para conocer mejor las condiciones bajo las que se pueden producir creaciones de archivos dinámicos.)

Nota: La longitud máxima de los registros de un archivo que se creará dinámicamente es 32 766.

***NODUPKEYCHK**
No comprueba las claves duplicadas de los archivos INDEXED.

***DUPKEYCHK**
Comprueba las claves duplicadas de los archivos INDEXED. (Consulte la explicación sobre la instrucción READ en el manual *COBOL/400 Reference* para comprender mejor las condiciones bajo las que se indica a un programa la existencia de registros con claves duplicadas.)

Aviso: Si se especifica esta opción, el rendimiento del compilador puede verse afectado.

***STDERR**
Se utiliza el manejo de errores estándar. Consulte el Capítulo 6, “Manejo de Errores y Excepciones COBOL/400” en la página 71 para obtener más información.

***NOSTDERR**
Se utiliza el método de manejo de errores de la Versión 1, Releases 1 y 2.

***NOEXTACCDSP**
El compilador no permite instrucciones ACCEPT o DISPLAY ampliadas.

***EXTACCDSP**
El compilador permite instrucciones ACCEPT y DISPLAY ampliadas. Consulte el Apéndice E del manual *COBOL/400 Reference* para obtener información sobre los cambios en las listas de palabras reservadas que se producen cuando se utiliza esta opción.

***NOINZDLT**
Los archivos relativos con acceso secuencial no se inicializan con registros suprimidos durante la operación CLOSE si los archivos se han abierto para OUTPUT. Es decir, la cantidad de registros escritos determinan el límite de registros. Las operaciones OPEN subsiguientes sólo permiten acceder hasta el límite de registros.

***INZDLT**
Los archivos relativos con acceso secuencial se inicializan con registros suprimidos durante la operación CLOSE si los archivos se han abierto para OUTPUT. Los registros activos de los archivos no se verán afectados. En otras palabras, el límite de registro se define como el tamaño de archivo para ulteriores operaciones de E/S.

***NOBLK**
El compilador permite bloquear sólo los archivos de acceso SEQUENTIAL que no tengan una instrucción START.

Si se especifica una cláusula BLOCK CONTAINS, esta cláusula se omite, excepto para los archivos de cinta.

***BLK**
Cuando se utilizan BLOCK CONTAINS, el compilador permite el bloqueo desde los archivos de acceso DYNAMIC y SEQUENTIAL con una instrucción START. Para los archivos RELATIVE que se han abierto para operaciones de salida, no está permitido el bloqueo.

La cláusula BLOCK CONTAINS controla el número de registros que se deben bloquear.

Cuando no se especifica ninguna cláusula BLOCK CONTAINS, el compilador permite sólo bloquear archivos de acceso SEQUENTIAL que no tengan una instrucción START. El sistema operativo determina la cantidad de registros que se deben bloquear.

***STDINZ**
El compilador inicializa los ítems de datos del usuario a los valores por omisión del sistema, siempre y cuando no dependan de una cláusula VALUE.

***NOSTDINZ**
Para los ítems de usuario que no tengan una cláusula VALUE, el compilador no inicializa los ítems de datos a los valores por omisión del sistema.

***FS21DUPKY**
El compilador informa de un estado de archivo 21 cuando se procesa un archivo indexado con claves duplicadas en modalidad dinámica o al azar, si el valor de la clave se cambia entre la instrucción READ obligatoria y una instrucción REWRITE o DELETE siguiente.

***NOFS21DUPKY**
El compilador no informa de un archivo de estado 21 cuando se procesa un archivo indexado con claves duplicadas en modalidad de acceso dinámica o al azar. Una instrucción REWRITE puede cambiar la clave de un registro.

Parámetro CVTOPT:
Especifica cómo maneja el compilador los tipos de datos SAA de tiempo, fecha e indicación horaria, así como los campos de caracteres de longitud variable que se han pasado desde archivos descritos externamente hasta el programa del usuario a través de COPY DDS. Los valores posibles son:

***NOVARCHAR**
Los campos de longitud variable se omiten y se declaran como campos FILLER.

***VARCHAR**
Los campos de longitud variable se declaran como ítems de grupo de longitud fija y son accesibles para el programa. Para obtener más información acerca de los campos de longitud variable, consulte el apartado “Declaración de Ítems de Datos utilizando Tipos de Datos de CVTOPT” en la página 137.

***NODATETIME**
Los tipos de datos de fecha, hora e indicación horaria se omiten y se declaran campos FILLER.

***DATETIME**
Los tipos de datos de fecha, hora e indicación horaria se declaran campos de longitud variable y son accesibles para el programa.

***NOGRAPHIC**
Los tipos de datos DBCS gráficos se omiten y se declaran campos FILLER.

***GRAPHIC**
Los tipos de datos DBCS gráficos de longitud variable se declaran campos alfanuméricos de longitud fija y son accesibles para el programa.

Cuando la opción *VARCHAR también se utiliza, los tipos de datos DBCS gráficos de longitud variable se declaran ítems de grupo y son accesibles para el programa. Para obtener más información sobre los tipos de datos DBCS gráficos, consulte el apartado “Campos Gráficos DBCS” en la página 139.

Parámetro MSGLMT:
Controla la compilación indicando el número máximo de mensajes de un nivel de gravedad de error determinado que pueden ocurrir antes de que se detenga la compilación.

Por ejemplo, es posible detener la compilación si se producen más de tres errores con un nivel de gravedad 20 o superior. En este

ejemplo, se especificaría 3 como el número máximo de mensajes de error, y 20 para el nivel máximo de gravedad de error. Si se producen tres errores con un nivel de gravedad 20 o superior, la compilación continúa, pero cuando se produce un cuarto, la compilación se detiene automáticamente. Si la cantidad máxima de mensajes de error no se supera ni se iguala, la compilación continúa independientemente de la cantidad de errores producidos.

límite-mensaje
Los posibles valores para el número máximo de mensajes de error son:

***NOMAX**
La compilación se completa con normalidad independientemente del número de errores producidos.

1-9999
La compilación se detiene si el número de errores del nivel de gravedad especificado supera la cantidad especificada por el usuario. Si la cantidad máxima de mensajes de error no se supera ni se iguala, la compilación continúa independientemente de la cantidad de errores producidos.

gravedad-mensaje
Los posibles valores para el nivel máximo de gravedad de error son:

29 La compilación se detiene si la cantidad de errores con un nivel de gravedad 29 o superior sobrepasa el número máximo de mensajes de error especificado.

nivel-gravedad-máximo
Especifique un número de uno o dos dígitos del 0 al 29. La compilación se detiene si la cantidad de errores con el nivel de gravedad especificado o superior sobrepasa la cantidad máxima de mensajes de error especificada por el usuario.

Parámetro PRTFILE:
Especifica el nombre del archivo al que se dirige el listado del compilador, así como la biblioteca en la que está ubicado el archivo. El archivo debe tener un longitud mínima de registro de 132. Si se especifica un archivo que contenga un registro de longitud inferior a 132, la información se pierde.

Los valores posibles son:

QSYSPRT
Si no se especifica un nombre de archivo, el listado del compilador se dirige a QSYSPRT, que es un archivo suministrado por IBM.

nombre-archivo
Entre el nombre del archivo al que se dirige el listado del compilador.

Los valores posibles de la biblioteca son:

***LIBL**
Si no se especifica un nombre de biblioteca, el sistema busca en la lista de bibliotecas *LIBL, para encontrar la biblioteca en la que se encuentra el archivo.

***CURLIB**
Se utiliza la biblioteca actual. Si no ha asignado la biblioteca como biblioteca actual, se utiliza QGPL.

nombre-biblioteca
Introduzca el nombre de la biblioteca en la que se encuentra el archivo.

Parámetro FLAGSTD:
Especifica las opciones de la señalización FIPS. (Seleccione la opción *LINENUMBER para asegurarse de que los números de referencia que se utilizan en los mensajes FIPS sean únicos.) Los valores posibles son:

***NOFIPS**
El programa fuente no dispone de la señalización FIPS.

***MINIMUM**
Señal FIPS para el subconjunto menor o mayor.

***INTERMEDIATE**
Señal FIPS para el subconjunto intermedio o mayor.

***HIGH**
Señal FIPS para el subconjunto mayor.

***NOSEG**
No se señala con FIPS el módulo opcional SEGMENTATION.

***SEG1**
Señal FIPS para el módulo opcional SEGMENTATION de nivel 1 y superior.

***SEG2**
Señal FIPS para el módulo opcional SEGMENTATION de nivel 2.

***NODEB**
El módulo opcional DEBUG no se señala con FIPS.

***DEB1**
Señal FIPS para el módulo opcional DEBUG de nivel 1 y superior.

***DEB2**
Señal FIPS para el módulo opcional DEBUG de nivel 2.

***NOOBSOLETE**
Los elementos de lenguaje obsoletos no se señalan.

***OBSOLETE**
Los elementos de lenguaje obsoletos se señalan.

Parámetro SAAFLAG:
Especifica si se desea señalar las funciones COBOL/400* que no están soportadas por COBOL SAA. (Seleccione la opción *LINENUMBER para asegurarse de que los números de referencia que se utilizan en los mensajes COBOL SAA sean únicos.) Los valores posibles son:

***NOFLAG**
No se realiza la señalización de COBOL SAA.

***FLAG**
Se realiza la señalización de COBOL SAA.

Parámetro EXTDSPOPT:
Especifica las opciones que se deben utilizar para las instrucciones ACCEPT y DISPLAY ampliadas para la estación de trabajo de E/S. Los valores posibles son:

***DFRWRT**
Las instrucciones DISPLAY ampliadas se mantienen en un almacenamiento intermedio hasta que se produce una instrucción ACCEPT ampliada, o hasta que el almacenamiento intermedio se llene.

Si no se detecta una instrucción ACCEPT ampliada antes de que el almacenamiento intermedio se llene, el contenido del almacenamiento intermedio aparece en pantalla. Cuando se detecta una instrucción ACCEPT ampliada, el contenido actual del almacenamiento intermedio aparece en pantalla.

***NODFRWRT**
Todas las instrucciones DISPLAY ampliadas se ejecutan a medida que se detectan.

***UNDSPCHR**
Las instrucciones ACCEPT y DISPLAY ampliadas manejan los caracteres que se pueden visualizar y los que no.

***NOUNDSPCHR**
Utilice esta opción cuando los datos que se deben visualizar contengan caracteres DBCS ampliados. Las instrucciones ACCEPT y DISPLAY ampliadas manejan únicamente caracteres que se pueden visualizar.

Aunque el usuario debe utilizar esta opción para estaciones de pantallas conectadas a controladores 3174 y 3274, también puede utilizarla para estaciones de trabajo locales. Si no se utiliza esta opción, los datos deben contener caracteres visualizables. Si los datos contienen valores menores que el hexadecimal 20, los resultados obtenidos no serán los esperados, ya que aparecerán errores graves e incluso algún formato de pantalla erróneo.

***ACCUPDALL**
Todos los tipos de datos se visualizan previamente en las instrucciones ACCEPT ampliadas, independientemente de la existencia de la frase UPDATE.

***ACCUPDNE**
Tan sólo los datos de edición numéricos se visualizan previamente en las instrucciones ACCEPT ampliadas que no contengan la frase UPDATE.

Parámetro FLAG:

Especifica el nivel de seguridad mínimo de los mensajes que se van a imprimir. Los valores posibles son:

0 Todos los mensajes se imprimen.

nivel-gravedad

Entre un número de uno o dos dígitos que especifique el nivel de gravedad mínimo de los mensajes que se han de imprimir. Se listan los mensajes que tienen niveles de seguridad del valor especificado o superior.

Parámetro REPLACE:

Especifica si se crea un objeto programa nuevo cuando un objeto programa del mismo nombre ya existe en la misma biblioteca. Los valores posibles son:

***YES**

Se crea un nuevo objeto programa y cualquier objeto programa existente del mismo nombre de la biblioteca especificada se traslada a la biblioteca QRPLOBJ.

***NO**

No se crea ningún objeto programa si un objeto programa del mismo nombre ya existe en la biblioteca especificada.

Parámetro TGTRLS:

Especifica el release del sistema operativo en el que el usuario pretende utilizar el objeto que se está creando. Se puede especificar un nivel de release exacto en el formato VxRxMx, donde Vx es la versión, Rx el release y Mx el nivel de modificación. Por ejemplo, V2R2M0 es versión 2, release 2 y nivel de modificación 0.

Nota: Para utilizar el objeto en el sistema destino, es preciso guardar el objeto con el nivel de release de destino especificado en el mandato de creación y a continuación restaurarlo en el sistema destino.

Los valores posibles son:

***CURRENT**

El objeto debe utilizarse en el release del sistema operativo que se ejecute en ese momento en el sistema. También es posible utilizar el objeto en un sistema con cualquier release posterior del sistema operativo instalado.

***PRV**

El objeto debe utilizarse en el release previo con un nivel de modificación 0 del sistema operativo. También es posible utilizar el objeto en un sistema con cualquier release ulterior del sistema operativo instalado.

nivel-release

Especifique el release en el formato VxRxMx. El objeto puede utilizarse en un sistema con el release especificado o con cualquier release posterior del sistema operativo instalado.

Los valores válidos dependen de la versión actual, release y del nivel de modificación, que varían con cada release nuevo.

Parámetro USRPRF:

Especifica el perfil del usuario que trabajará con el programa COBOL compilado. El perfil del propietario o del usuario de este programa sirve para ejecutar el programa y para controlar los objetos que el programa puede utilizar (incluyendo la autorización que el programa tiene para cada objeto). Este parámetro no se actualiza si el programa ya existe. Para cambiar el valor de USRPRF, suprima el programa y vuelva a compilarlo utilizando el valor correcto.

Los valores posibles son:

***USER**

El perfil de usuario del usuario de este programa se utiliza cuando se ejecuta el programa.

***OWNER**

El perfil de usuario del propietario del programa y del usuario se utilizan cuando se ejecuta el programa. Los conjuntos de autorizaciones para objetos en los dos perfiles de usuario sirven para encontrar y acceder a los objetos durante la ejecución del programa. El usuario del programa será el propietario de cualquier objeto que se cree durante el programa.

Nota: Especifique el parámetro USRPRF para reflejar los requisitos de seguridad de la instalación. Los servicios de seguridad disponibles en el sistema AS/400 se describen

con todo detalle en el manual *Seguridad Manual de Consulta*.

Parámetro AUT:

Especifica la autorización que se otorga a los usuarios que carecen de una autorización específica para el objeto programa, que no están en la lista de autorizaciones o que su grupo no tiene autorización específica para el objeto programa. Es posible alterar la autorización de todos los usuarios, o sólo de usuarios específicos, una vez creado el objeto del programa utilizando los mandatos GRTOBJAUT (Autorización de Objeto Otorgado) o RVKOBJAUT (Autorización de Objeto Revocado).

Los posibles valores son:

***LIBCRTAUT**

La autorización pública del objeto se obtiene de la palabra clave CRTAUT de la biblioteca destino (biblioteca que debe contener el objeto programa creado). Este valor se determina cuando se crea el objeto. Si el valor CRTAUT de la biblioteca cambia después de que se haya creado el objeto del programa, el nuevo valor NO afectará a los objetos existentes.

***ALL**

Proporciona autorización a todas las operaciones del objeto programa, excepto a aquéllas que hagan referencia al propietario o que estén controladas por la autorización de la gestión de listas de autorizaciones. El usuario puede controlar la existencia del objeto programa, especificar su seguridad, modificarla, así como efectuar cualquier tipo de operación, pero no puede traspasar su título de propiedad.

***CHANGE**

Proporciona a todos los datos autorización para poder realizar todo tipo de operaciones en el objeto programa, excepto aquéllas que estén restringidas para el propietario o controladas por la autorización del objeto y por la autorización de la gestión del objeto. El usuario puede cambiar el objeto y realizar funciones básicas, como por ejemplo ejecutar y depurar el objeto programa.

***USE**

Proporciona autorización operacional de objeto y autorización de lectura, así como autorización para efectuar operaciones básicas en el objeto programa, como por ejemplo ejecutar el programa. Se avisa al usuario antes de que el objeto se modifique.

***EXCLUDE**

El usuario no tiene acceso al objeto programa.

nombre-lista-autorización

Entre el nombre de una lista de autoridades de usuarios y autorizaciones a las que el programa se añade. El objeto programa está protegido por esta lista de autorizaciones y la autorización pública del objeto programa se establece a *AUTL. La lista de autorizaciones debe estar en el sistema cuando se emite el mandato CRTCBLPGM. Utilice el mandato Crear Lista de Autorizaciones (CRTAUTL) para obtener una lista de Autorizaciones propia.

Nota: Especifique el parámetro AUT para reflejar los requisitos de seguridad de la instalación. Los servicios de seguridad disponibles en el sistema AS/400 se describen con detalle en el manual *Seguridad Manual de Consulta*.

Parámetro DUMP:

Sistema de ayuda de depuración COBOL/400 de IBM para el personal de servicio de IBM.

Parámetro ITDUMP (n):

Ayuda de depuración IBM proporcionada para el personal de servicio de IBM. Este parámetro obliga al compilador a volcar el texto interno en determinados momentos durante la compilación del programa fuente.

Entrada del Mandato CRTCBLPGM desde la Línea de Mandatos

Es posible entrar el mandato CRTCBLPGM desde la línea de mandatos. Teclee CRTCBLPGM seguido de los parámetros adecuados para compilar el programa. Consulte la Figura 4 en la página 30 para cerciorarse del tipo de sintaxis adecuada que se ha de utilizar. Si no está seguro del significado de algunos parámetros, consulte las descripciones referentes a parámetros y opciones que se encuentran entre las páginas 18 y 28. Consulte los siguientes ejemplos para conocer la sintaxis correcta que es preciso utilizar para introducir el mandato CRTCBLPGM desde la línea de mandatos.

Ejemplo 1

```
CRTCBLPGM SRCFILE(QGPL/QLBLSRC) SRCMBR(SAMPLE) SAAFLAG(*FLAG)
```

Fuente Parcial para el Miembro SAMPLE

```
ID DIVISION.
PROGRAM-ID. EXAMPLE.
```

El ejemplo anterior permite crear un programa COBOL/400 desde el miembro fuente SAMPLE en el archivo QLBLSRC y en la biblioteca QGPL. El programa resultante se llama EXAMPLE. Si se especifica *FLAG para el parámetro SAAFLAG, se indica al compilador que debe identificar cualquier función que no esté soportada por COBOL SAA. En este ejemplo se han utilizado todos los valores por omisión de los parámetros a excepción de los parámetros SRCFILE, SRCMBR y SAAFLAG.

Ejemplo 2

```
CRTCBLPGM PGM(TEST) SRCFILE(SOURCE1) CVTOPT(*GRAPHIC)
```

En el ejemplo anterior, el compilador busca el archivo SOURCE1 en la lista de bibliotecas y busca el miembro llamado TEST dentro de ese mismo archivo. (El valor del parámetro SRCMBR ha tomado el valor por omisión *PGM, lo que significa que ha de buscar un miembro que tenga el mismo nombre que el programa que se ha de crear). El compilador crea un programa COBOL/400 denominado TEST a partir del programa fuente que se ha encontrado en el miembro TEST del archivo SOURCE1. El hecho de especificar *GRAPHIC para el parámetro CVTOPT indica que si DDS contiene tipos de datos DBCS gráficos, el programa COBOL deberá poder referenciarlos como campos alfanuméricos.

Entrada del Mandato CRTCBLPGM desde un Programa CL

Cuando se emite el mandato CRTCBLPGM desde un programa CL, se pueden utilizar las expresiones de concatenación de todos los valores de los parámetros. Consulte el manual *CL Reference* para obtener más información sobre las expresiones de concatenación. Consúltelo también para obtener una descripción de las reglas de denominación de objetos OS/400, así como para consultar la descripción de la sintaxis de mandatos OS/400.

Interfaz de Programación de Uso General

Se puede utilizar este mandato en QCMDEXC.

Fin de Interfaz de Programación de Uso General

Sintaxis del Mandato CRTCBLPGM

La Figura 4 muestra la sintaxis del mandato CRTCBLPGM.

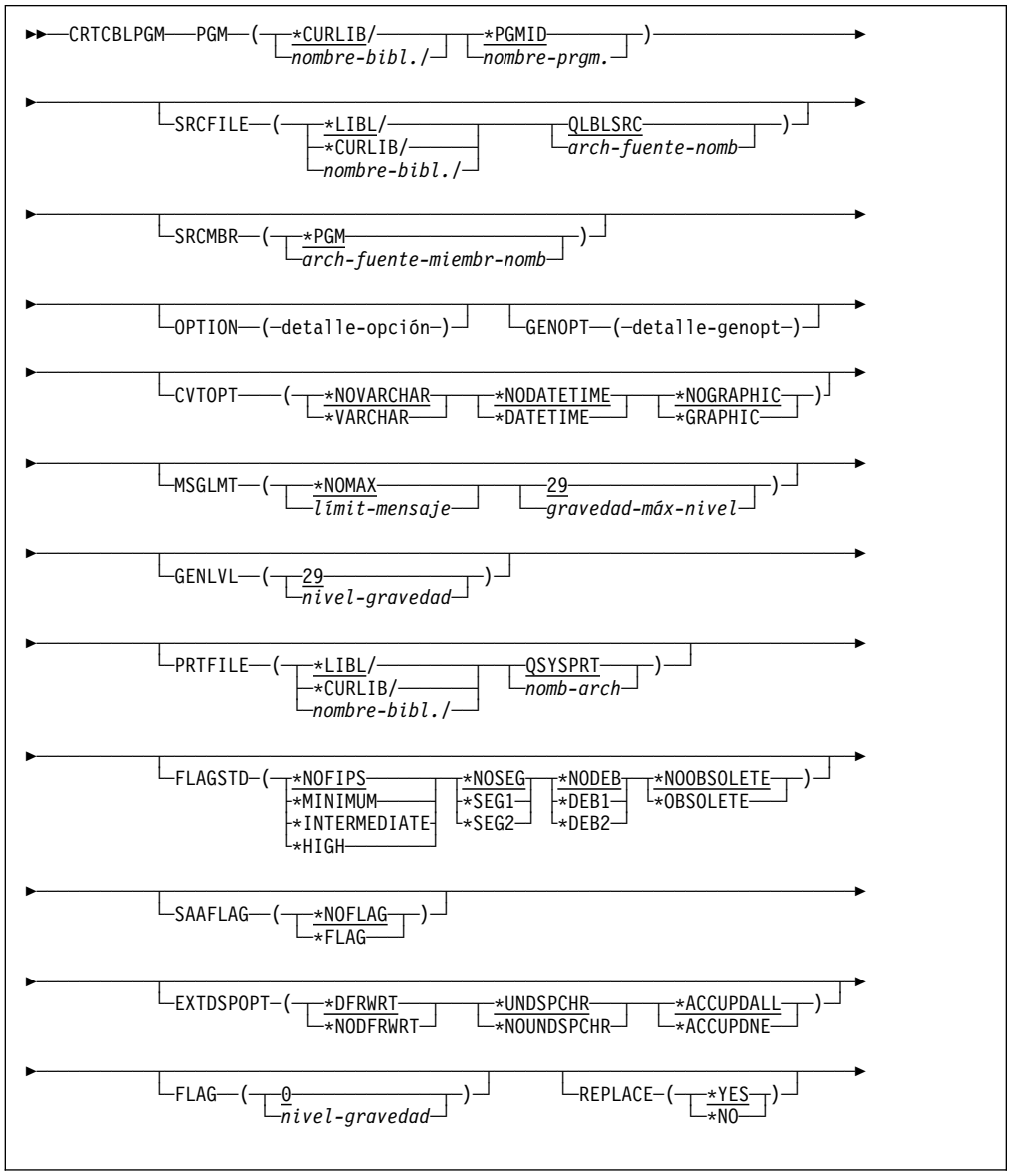


Figura 4 (Parte 1 de 5). Sintaxis del Mandatos CRTCBLPGM

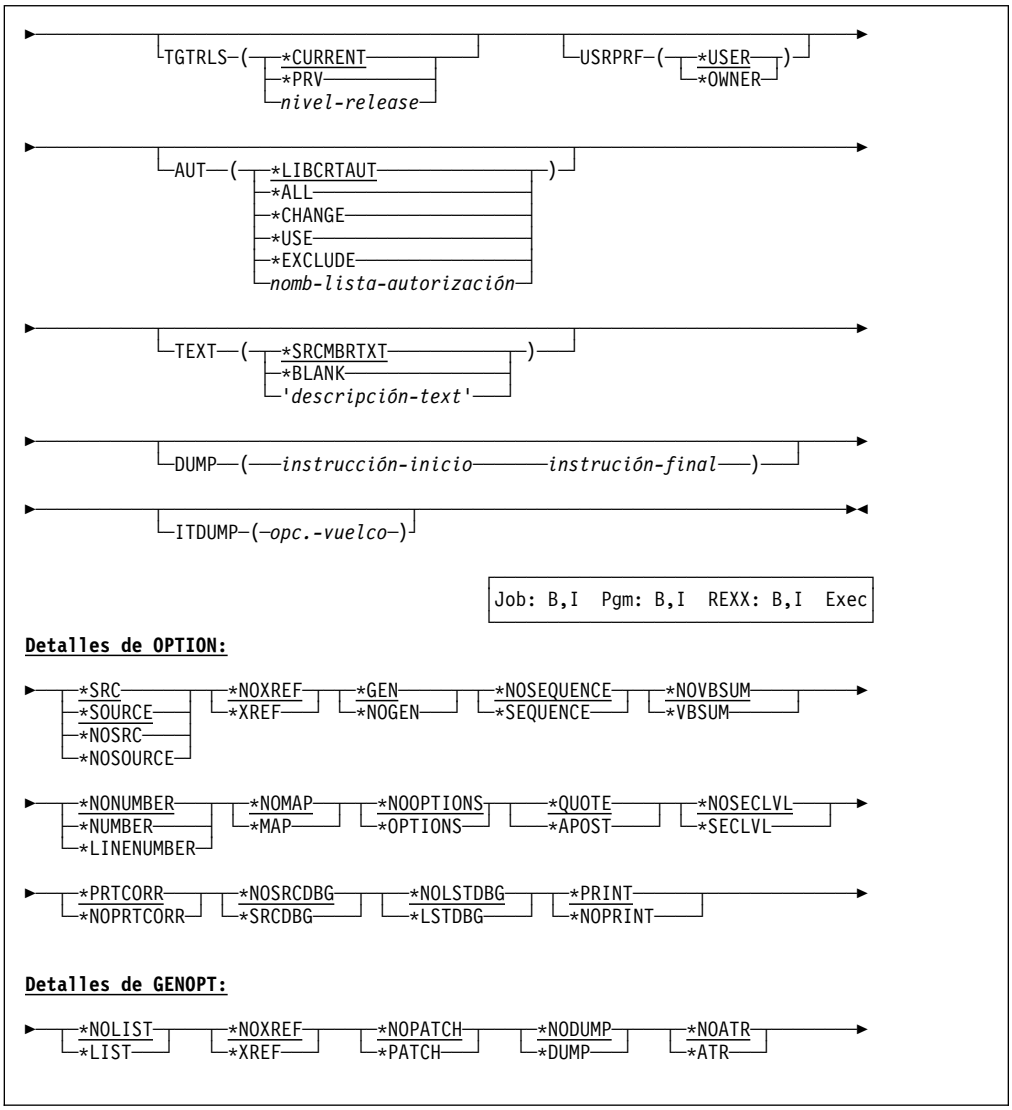


Figura 4 (Parte 2 de 5). Sintaxis del Mandatos CRTCBLPGM

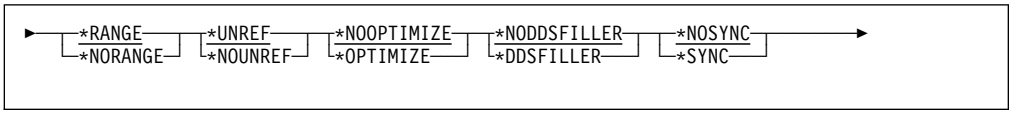


Figura 4 (Parte 3 de 5). Sintaxis del Mandatos CRTCBLPGM

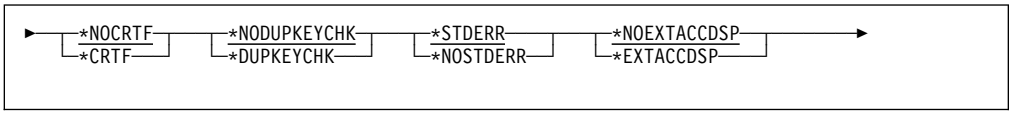


Figura 4 (Parte 4 de 5). Sintaxis del Mandatos CRTCBLPGM

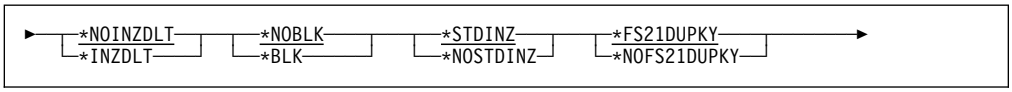


Figura 4 (Parte 5 de 5). Sintaxis del Mandatos CRTCBLPGM

Compilación del Programa Fuente para el Release Anterior

El programa COBOL/400 en un sistema AS/400 se puede compilar usando el release actual del sistema operativo OS/400 y restaurarlo en un sistema AS/400 que utilice un release anterior del sistema operativo.

El parámetro Target Release (TGTRLS) del mandato CRTCBLPGM permite especificar el nivel de release en el que desea utilizar el programa objeto. El parámetro TGTRLS dispone de tres valores posibles: *CURRENT , *PRV y *nivel-release*:

Especifique *CURRENT si el programa objeto se debe utilizar en el release del sistema operativo que en estos momentos se ejecuta en el sistema. Por ejemplo, si V2R2M0 se ejecuta en el sistema, *CURRENT significa que el usuario quiere utilizar el programa en un sistema que tiene instalado V2R2M0. Este valor es el valor por omisión.

Especifique *PRV si el programa objeto se ha de utilizar en el release anterior con un nivel de modificación 0 del sistema operativo. Por ejemplo, si V2R2M0 se ejecuta en el sistema, *PRV significa que el usuario quiere utilizar el programa en un sistema que tiene instalado V2R1M0.

Nivel-release permite especificar el nivel de release en el que se desea utilizar el programa objeto. Los valores que se pueden introducir para este parámetro, y que se modificarán en cada release nueva, dependen de la versión actual, release y nivel de modificación.

Para obtener más información acerca del parámetro TGTRLS, vea la página 27.

Es preciso tener presente las siguientes limitaciones:

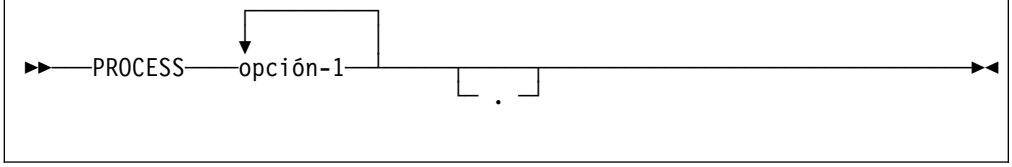
- El soporte que se compila para ser utilizado con el release anterior sólo está disponible cuando se utiliza el parámetro TGTRLS del parámetro CRTCBLPGM. Especifique *PRV o bien el nivel de release cuando se compile el programa y guárdelo mediante los mandatos CL Salvar Objeto (SAVOBJ) o Salvar Biblioteca (SAVLIB) para restaurarlo satisfactoriamente en el release anterior del sistema operativo.
- Sólo puede utilizar el parámetro TGTRLS para programas COBOL creados en el entorno System/38.
- Es posible restaurar un programa objeto o a un release anterior a un release siguiente. No puede restaurar un programa objeto en un sistema que está por debajo de más de un release. Es decir, sólo se proporciona un nivel anterior de release de compatibilidad.
- No puede utilizar las funciones que son nuevas en el release actual del sistema operativo en un programa que se ha compilado para ser utilizado en el nivel de release anterior.
- Los programas deben volverse a convertir cuando se restauran al release anterior; por lo tanto, no puede suprimir la observabilidad si los programas han de volverse a convertir.
- No deben haber bibliotecas producto en la parte del sistema de la lista de bibliotecas.

Uso de la Instrucción PROCESS para Especificar las Opciones del Compilador

La instrucción PROCESS es una parte opcional del programa fuente COBOL. La instrucción PROCESS se puede utilizar para especificar las opciones que normalmente se especifican durante la compilación. Las opciones especificadas en la instrucción PROCESS alteran temporalmente las opciones correspondientes especificadas en el mandato CL CRTCBLPGM.

El formato de la instrucción PROCESS es la siguiente:

Formato



Se aplican las siguientes reglas:

- La instrucción PROCESS debe colocarse antes de la primera instrucción fuente del programa COBOL precediendo inmediatamente al encabezamiento IDENTIFICATION DIVISION.
- La instrucción empieza con la palabra PROCESS. Las opciones pueden aparecer en más de una línea; sin embargo, sólo la primera línea puede contener la palabra PROCESS.
- La palabra PROCESS y todas las opciones deben aparecer entre las posiciones 8 y 72. La posición 7 debe estar en blanco. Las posiciones restantes pueden utilizarse como en las instrucciones fuente COBOL: de la 1 a la 6 para números de secuencia, y de la 73 a la 80 para propósitos de identificación.
- Las opciones deben estar separadas por espacios en blanco y/o comas.
- Las opciones pueden aparecer en cualquier orden. Si se especifican opciones contradictorias, por ejemplo, LIST y NOLIST, tiene preferencia la última opción encontrada.
- Si la palabra clave de la opción es correcta y la subopción es errónea, se asume la subopción por omisión.

No todos los parámetros CRTCBLPGM tienen una opción correspondiente en la sentencia PROCESS. Las tablas siguientes indican las opciones de la instrucción PROCESS disponibles y los parámetros y opciones del mandato CRTCBLPGM equivalentes. Los valores por omisión se subrayan. Las descripciones de las opciones de instrucción PROCESS corresponden a las descripciones de las opciones y parámetros que empiezan en la página 18.

Opción de Instrucción PROCESS	**CRTCBLPGM**
	Opción de parámetro GENLVL
GENLVL(nn)	nn

Opciones de Instrucción PROCESS	CRTCBLPGM
	Opciones de parámetro OPTION
GEN NOGEN	*GEN *NOGEN
NOMAP MAP	*NOMAP *MAP
NONUMBER NUMBER LINENUMBER	*NONUMBER *NUMBER *LINENUMBER
NOSECLVL SECLVL	*NOSECLVL *SECLVL
NOOPTIONS OPTIONS	*NOOPTIONS *OPTIONS
QUOTE APOST	*QUOTE *APOST
NOSEQUENCE SEQUENCE	*NOSEQUENCE *SEQUENCE
SOURCE (o SRC) NOSOURCE (o NOSRC)	*SOURCE (o *SRC) *NOSOURCE (o *NOSRC)
NOVBSUM VBSUM	*NOVBSUM *VBSUM
NOXREF XREF	*NOXREF *XREF
PRTCORR NOPRTCORR	*PRTCORR *NOPRTCORR

Opciones de Instrucción PROCESS	CRTCBLPGM
	Opciones de parámetro GENOPT
NOINZDLT INZDLT	*NOINZDLT *INZDLT
NOLIST LIST	*NOLIST *LIST
STDERR NOSTDERR	*STDERR *NOSTDERR
NODDSFILLER DDSFILLER	*NODDSFILLER *DDSFILLER
NOSYNC SYNC	*NOSYNC *SYNC
NOCRTF CRTF	*NOCRTF *CRTF
NODUPKEYCHK DUPKEYCHK	*NODUPKEYCHK *DUPKEYCHK
NOEXTACCDSP EXTACCDSP	*NOEXTACCDSP *EXTACCDSP
NOBLK BLK	*NOBLK *BLK
STDINZ NOSTDINZ	*STDINZ *NOSTDINZ
FS21DUPKEY NOFS21DUPKEY	*FS21DUPKY *NOFS21DUPKY
RANGE NORANGE	*RANGE *NORANGE
UNREF NOUNREF	*UNREF *NOUNREF

Opciones de Instrucción PROCESS	CRTCBLPGM
	Opciones de parámetro CVTOPT
NOVARCHAR VARCHAR	*NOVARCHAR *VARCHAR
NODATETIME DATETIME	*NODATETIME *DATETIME
NOCVTGRAPHIC CVTGRAPHIC	*NOGRAPHIC *GRAPHIC

Opciones de Instrucción PROCESS	CRTCBLPGM
	Opciones de parámetro FLAGSTD
NOFIPS MINIMUM INTERMEDIATE HIGH	*NOFIPS *MINIMUM *INTERMEDIATE *HIGH
NOSEG SEG1 SEG2	*NOSEG *SEG1 *SEG2
NODEB DEB1 DEB2	*NODEB *DEB1 *DEB2
NOOBSOLETE OBSOLETE	*NOOBSOLETE *OBSOLETE

Opciones de Instrucción PROCESS EXTDSPOPT(*a b c*)	CRTCBLPGM
	Opciones de parámetro EXTDSPOPT
DFRWRT NODFRWRT	*DFRWRT *NODFRWRT
UNDSPCHR NOUNDSPCHR	*UNDSPCHR *NOUNDSPCHR
ACCUPDALL ACCUPDNE	*ACCUPDALL *ACCUPDNE

Opciones de Instrucción PROCESS	CRTCBLPGM
	Opciones de parámetro SAAFLAG
NOSAAFLAG SAAFLAG	*NOFLAG *FLAG

Opción de Instrucción PROCESS	CRTCBLPGM
	Opción de parámetro FLAG
FLAG(nn)	nn

Opciones de Instrucción PROCESS	CRTCBLPGM
NOFS9MTO0M FS9MTO0M	no aplicable
NOGRAPHIC GRAPHIC	no aplicable

FS9MTO0M cambia un estado de archivo 9M a un estado de archivo 0M.

La opción GRAPHIC de la instrucción PROCESS está disponible para procesar caracteres DBCS en literales DBCS. Consulte el Apéndice F, “Soporte del Juego de Caracteres de Idiomas Internacionales de Doble Byte” en la página 355 para obtener información acerca del soporte DBCS.

La opción EXTDSPOPT de la instrucción PROCESS debe codificarse con las opciones entre paréntesis asociadas similares a la sintaxis `FLAG(nn)`. Se puede especificar más de una opción dentro de los paréntesis para la opción EXTDSPOPT. Por ejemplo, para especificar DFRWRT y UNDSPCHR, escriba

```
EXTDSPOPT(DFRWRT UNDSPCHR)
```

También es correcto especificar `EXTDSPOPT` o `EXTDSPOPT( )`.

Cuando sólo se especifica EXTDSPOPT en la instrucción PROCESS, todos los valores por omisión de las opciones adicionales están activas.

Si se especifica `EXTDSPOPT( )`, no tiene ninguna consecuencia para el programa.

Si se especifican opciones contradictorias, la última opción especificada altera temporalmente a las otras.

Compilación de varios Programas

La instrucción PROCESS se puede utilizar para separar varios programas y/o subprogramas que se han de compilar con una sola invocación del compilador. (Un **subprograma** es un programa llamado que se combina con el programa de llamada en tiempo de ejecución para producir una unidad de ejecución.) Cuando se compilan programas múltiples, todas las opciones del compilador que se han especificado en la instrucción del mandato CRTCBLPGM, amén de las opciones por omisión y de las opciones especificadas en la última instrucción PROCESS que precede al programa, estarán activas para la compilación del programa. Toda la salida del compilador se dirige a los destinos especificados por la instrucción del mandato CRTCBLPGM.

Todos los programas objeto se almacenan en la biblioteca especificada en el parámetro PGM. Si se especifica el nombre del programa para el parámetro PGM, el primer programa del trabajo por lotes tiene ese nombre, y el resto de programas utilizan el nombre especificado en el párrafo PROGRAM-ID del programa fuente.

Uso de COPY dentro de la Instrucción PROCESS

Puede utilizarse una instrucción COPY en el programa fuente siempre que pueda utilizarse una serie de caracteres o un separador. Cada instrucción COPY debe ir precedida por un espacio y seguida de un punto y un espacio. Para obtener más información acerca de la instrucción COPY, consulte la sección “Instrucción COPY” del manual *COBOL/400 Reference*.

La instrucción COPY Formato 1 puede utilizarse dentro de la instrucción PROCESS para recuperar las opciones del compilador almacenadas previamente en una biblioteca fuente e incluirlas en la instrucción COPY. COPY puede utilizarse para incluir las opciones que prevalcen sobre aquéllas que el compilador especifica como valores por omisión. Cualquier opción de la instrucción PROCESS puede recuperarse con la instrucción COPY.

Las opciones del compilador pueden indistintamente preceder y seguir a la instrucción COPY dentro de la instrucción PROCESS. La última aparición detectada de una opción prevalece sobre todas las apariciones precedentes de esa opción.

El ejemplo siguiente muestra el uso de la instrucción COPY dentro de la instrucción PROCESS. La instrucción COPY debe ir seguida de un punto. Observe también que, en este ejemplo, NOMAP prevalece sobre la opción correspondiente en el miembro de la biblioteca:

```
000001  PROCESS XREF              MYPROG
000002  COPY DEFLTS.              MYPROG
        MAP, SOURCE, LIST         DEFLTS
000004  NOMAP, FLAG(20)           MYPROG
000010  IDENTIFICATION DIVISION.  MYPROG
```

Explicación de la Salida del Compilador

La salida del compilador puede incluir:

- Un resumen de las opciones de mandatos
- Un listado de opciones: listado de las opciones activas para la compilación. Utilice OPTION(*OPTIONS).
- Un listado fuente: listado de las instrucciones que se encuentran en el programa fuente. Utilice OPTION(*SOURCE o *SRC).
- Un listado de utilizaciones de verbos: listado de los verbos COBOL y frecuencia de uso de cada uno de ellos. Utilice OPTION(*VBSUM).
- Una correlación de la División de Datos: glosario de la información generada por compilador referente a los datos. Utilice OPTION(*MAP). También se incluye una correlación de los nombres suministrados por el usuario respecto a los nombres internos generados por el compilador.
- Señalización SAA: lista de las funciones del programa que no son portables a otros entornos COBOL SAA. Utilice SAAFLAG(*FLAG).
- Mensajes FIPS: lista de mensajes para un conjunto COBOL FIPS, para cualquiera de los módulos opcionales, para todos los elementos de lenguaje obsoletos o para una combinación de un subconjunto COBOL FIPS, módulos opcionales y todos los elementos obsoletos. Consulte la información referente al “parámetro FLAGSTD” de la página 25 para un mejor conocimiento de las opciones específicas que están disponibles para la señalización FIPS.
- Un listado de referencias cruzadas. Utilice OPTION(*XREF).
- Mensajes del compilador (incluyendo las estadísticas de diagnóstico).
- Estadísticas de compilación.
- Un listado del programa generado en formato simbólico.
- Un programa objeto.

La presencia o ausencia de algunos de estos tipos de salidas del compilador viene determinada por las opciones especificadas en la instrucción PROCESS o por los mandatos CRTCBLPGM. El nivel de los mensajes de diagnóstico impresos depende de la opción FLAG.

Especificación del Formato del Listado

Una barra (/) en el área del indicador (columna 7) de una línea da como resultado un salto de página del listado del programa fuente. La línea de comentarios de la barra (/) se imprime en la primera línea de la página siguiente.

Ampliación de IBM

Si especifica la instrucción EJECT en el programa, la próxima instrucción fuente se imprime en la parte superior de la página siguiente en el listado del compilador. Esta instrucción puede escribirse en cualquier lugar del Área A o Área B y debe ser la única instrucción en la línea.

La instrucción SKIP1/2/3 origina la inserción en el listado del compilador de unas líneas en blanco. Una instrucción SKIP1/2/3 puede escribirse en cualquier lugar del Área A o B. Debe ser la única instrucción de la línea.

- SKIP1 inserta una sola línea en blanco (doble espacio)
- SKIP2 inserta dos líneas en blanco (triple espacio).
- SKIP3 inserta tres líneas en blanco (espacio cuádruple)

Cada una de las instrucciones SKIP anteriores origina una sola inserción de una, dos o tres líneas.

Una instrucción TITLE coloca un título en cada página indicada.

Las instrucciones fuente COBOL se pueden listar o suprimir de forma selectiva utilizando las instrucciones *CONTROL, *CBL o COPY:

- *CONTROL NOSOURCE y *CBL NOSOURCE suprimen el listado de las instrucciones fuente.
- *CONTROL SOURCE y *CBL SOURCE continúan el listado de las instrucciones fuente.
- Una instrucción COPY que lleva la frase SUPPRESS suprime el listado de instrucciones copiadas. Por su duración, esta instrucción prevalece sobre cualquier instrucción *CONTROL o *CBL. Si el miembro copiado contiene instrucciones *CONTROL o *CBL, la última se ejecuta una vez cuando el miembro COPY se ha procesado.

Consulte la publicación *COBOL/400 Reference* para obtener más información sobre las instrucciones EJECT, SKIP1/2/3, *CONTROL, *CBL, COPY y TITLE.

Fin de Ampliación de IBM

Caracteres de Separación de la Hora

EL parámetro TIMSEP de los mandatos relacionados con el trabajo (por ejemplo CHGJOB) especifica el carácter de separación de tiempo que se utiliza en las indicaciones de la hora que aparecen en los listados del compilador. Si el valor TIMSEP no está, el valor del sistema QTIMSEP se utiliza por omisión.

Visualización del Listado del Compilador Utilizando el SEU

El Programa de Utilidad de Entrada Fuente (SEU) permite examinar el contenido del listado del compilador en una cola de salida. Es posible revisar los resultados de una compilación anterior y hacer a la vez los cambios necesarios en el código fuente. La Figura 5 muestra una pantalla dividida del SEU que permite examinar el listado de una estación de trabajo.

```
 Columnas. . . :    1  71           Editar                       XMPLIB/QLBLSRC
 SEU==> _____________________________________________________________      XMPLE
 FMT CB ......-A+++B++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
0014.00        DATA DIVISION.
0015.00        FILE SECTION.
0016.00        FD  FILE1
0017.00            RECORD CONTAINS 56 CHARACTERS
0018.00            LABEL RECORDS ARE OMITTED
0019.00            DATA RECORD IS REB-1.
0020.00        01  REC-1 PIC X(56).
 ______________________________________________________________________________
 Columnas . . . :    1  71           Examinar   Archivo Spool   . . :      XMPLE
 SEU==> _____________________________________________________
0000.50   STMT
0000.51 *   19  MSGID: LBL1327  SEVERITY: 30  SEQNBR:  001900
0000.52         Mensaje . . . . :   'REB-1' no definido en el programa. Cláusula
0000.53           ignorada.
0000.54                          * * * * *   F I N   D E   M E N S A J E S   * *
0000.55                                          Resumen de Mensajes
0000.56   Total    Info(0-4)    Aviso(5-19)      Error(20-29)    Gravedad(30-39)

 F6=Mover línea de división  F19=Izquierda  F20=Derecha
 F21=Mandatos del sistema    F24=Más teclas
Se ha encontrado error de sintaxis.
```

Figura 5. Pantalla Dividida de Editar/Examinar del SEU

Al mismo tiempo que se examina el listado del compilador, se pueden corregir las sentencias fuente que contengan errores. Para buscar errores, escriba `F *ERR` en la línea de mandatos del SEU.

Para obtener información completa sobre cómo examinar un listado del compilador, consulte el manual *SEU Guía del Usuario y Manual de Consulta*.

Programa y Listado Ejemplo

Las siguientes páginas muestran las opciones del compilador y el listado fuente que el programa ejemplo ha producido. En el texto siguiente se incluyen las referencias a las tablas. Estas referencias están indicadas por la impresión de letras en blanco en un fondo oscuro, por ejemplo (**Z**). Las letras invertidas del texto corresponden a las letras encontradas en las tablas.

Resumen de Mandatos

Este resumen, que se obtiene después de la compilación, lista todas las opciones especificadas en el mandato CRTCBLPGM. Consulte el apartado “Utilización del Mandato Crear Programa COBOL (CRTCBLPGM)” en la página 15 para obtener más información sobre las opciones definidas por el usuario.

```
5763CB1 V3R0M5  001000            IBM SAA COBOL/400              TESTER/SAMPLE         AS400SYS  03/27/94 11:01:51     Página  1
Programa . . . . . . . . . . . . . . :   SAMPLE
  Biblioteca . . . . . . . . . . . . :     TESTER
Archivo fuente . . . . . . . . . . . :   QLBLSRC
  Biblioteca . . . . . . . . . . . . :     TESTER
Miembro fuente . . . . . . . . . . . :   SAMPLE     03/27/94 11:01:34
Nivel de gravedad de generación  . . :   29
"Descripción" del texto. . . . . . . :   *BLANK
Opciones de listado fuente . . . . . :   *NONE
Opciones de generación . . . . . . . :   *NONE
Opciones de conversión . . . . . . . :   *NONE
Límite de mensaje:
  Cantidad de mensajes.. . . . . . . :   *NOMAX
  Gravedad límite de mensaje   . . . :   29
Imprimir archivo. . . . . . . . . . . :   QSYSPRT
  Biblioteca . . . . . . . . . . . . :     *LIBL
Señalización FIPS. . . . . . . . . . :   *NOFIPS *NOSEG *NODEB *NOOBSOLETE
Señalización SAA . . . . . . . . . . :   *NOFLAG
Opciones de visualización ampliada . :
Gravedad de señalización . . . . . . :   0
Sustituir programa . . . . . . . . . :   *YES
Release de destino . . . . . . . . . :   *CURRENT
Perfil de usuario. . . . . . . . . . :   *USER
Autorización . . . . . . . . . . . . :   *LIBCRTAUT
Compilador . . . . . . . . . . . . . :   IBM SAA COBOL/400
```

Figura 6. Listado de Resumen de Mandatos

Identificación de las Opciones Activas

Si se especifica, la instrucción PROCESS se imprime en primer lugar. La Figura 7 muestra una lista de todas las opciones activas para la compilación del programa ejemplo: las opciones especificadas en el mandato CRTCBLPGM, una vez modificadas por la instrucción PROCESS. Las opciones del compilador se listan al comienzo de cada salida del compilador cuando se especifica el parámetro OPTIONS.

```
5763CB1 V3R0M5  001000            Fuente COBOL AS/400              TESTER/SAMPLE         AS400SYS  03/27/94 11:01:51     Página  2
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
    1  000100 PROCESS OPTIONS, SAAFLAG, SOURCE, MAP, XREF,                                                 03/27/94
    2  000200   FLAG(00), MINIMUM, VBSUM.
                     Opciones del Compilador COBOL Activas
                        OPTIONS
                        SOURCE
                        XREF
                        MAP
                        VBSUM
                        NONUMBER
                        NOSEQUENCE
                        GEN
                        GENLVL(29)
                        FLAG( 0)
                        MINIMUM
                        NOSEG
                        NODEB
                        NOOBSOLETE
                        SAAFLAG
                        QUOTE
                        NOSECLVL
                        NOSRCDBG
                        NOLSTDBG
                        PRINT
                        PRTCORR
                     Opciones de Generación COBOL Activas
                        NOLIST
                        UNREF
                        RANGE
                        NOATR
                        NOXREF
                        NODUMP
                        NOPATCH
                        NOOPTIMIZE
                        NODDSFILLER
                        NOSYNC
                        NOCRTF
                        NODUPKEYCHK
                        STDERR
                        NOEXTACCDSP
                        NOINZDLT
                        NOFS9MTO0M
                        NOBLK
                        STDINZ
                        FS21DUPKY
                     Opciones de Conversión COBOL Activas
                        NOVARCHAR
                        NODATETIME
                        NOGRAPHIC
```

Figura 7. Lista de Opciones Activas

Listado Fuente

La Figura 8 muestra un listado fuente. Las instrucciones del programa fuente se listan tal y como se someten. El fuente no se lista si se especifica la opción NOSOURCE. Después de la página en la que aparece listado el párrafo PROGRAM-ID, todas las páginas de salida del compilador tienen el nombre del identificador de programa listado en el encabezamiento antes del nombre del sistema.

```
5763CB1 V3R0M5  001000            Fuente COBOL AS/400              TESTER/SAMPLE         AS400SYS  03/27/94 11:01:51     Página  4
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
  A    B                                                                                  C    D          E
    3  000300 IDENTIFICATION DIVISION.
    4  000400 PROGRAM-ID.     SAMPLE.
    5  000500   AUTHOR.       PROGRAMMER NAME.
    6  000600   INSTALLATION. COBOL DEVELOPMENT CENTRE.
    7  000700   DATE-WRITTEN. 11/27/87.
    8  000800   DATE-COMPILED.  03/27/94 11:01:51   .
    9  000900 ENVIRONMENT DIVISION.
   10  001000 CONFIGURATION SECTION.
   11  001100 SOURCE-COMPUTER. IBM-AS400.                                                      03/27/94
   12  001200 OBJECT-COMPUTER. IBM-AS400.                                                      03/27/94
   13  001300 INPUT-OUTPUT SECTION.
   14  001400 FILE-CONTROL.
   15  001500     SELECT FILE-1 ASSIGN TO DISK-SAMPLE.
   16  001600 DATA DIVISION.
   17  001700 FILE SECTION.
   18  001800 FD  FILE-1
   19  001900     LABEL RECORDS ARE STANDARD
   20  002000     RECORD CONTAINS 20 CHARACTERS
   21  002100     DATA RECORD IS RECORD-1.
   22  002200 01  RECORD-1.
   23  002300   02 FIELD-A      PIC X(20).
   24  002400 WORKING-STORAGE SECTION.
   25  002500 01  FILLER.
   26  002600   05 KOUNT        PIC S9(2) COMP-3.
   27  002700   05 LETTERS      PIC X(26) VALUE "ABCDEFGHIJKLMNOPQRSTUVWXYZ".
   28  002800   05 ALPHA REDEFINES LETTERS
   29  002900                    PIC X(1)  OCCURS 26 TIMES.
   30  003000   05 NUMBR        PIC S9(2) COMP-3.
   31  003100   05 DEPENDENTS   PIC X(26) VALUE "01234012340123401234012340".
   32  003200   05 DEPEND REDEFINES DEPENDENTS
   33  003300                    PIC X(1)  OCCURS 26 TIMES.
   34  003400 COPY WRKRCD.
   35 +000010 01  WORK-RECORD.                                                              WRKRCD
   36 +000020   05 NAME-FIELD   PIC X(1).                                                  WRKRCD
   37 +000030   05 FILLER       PIC X(1)  VALUE SPACE.                                     WRKRCD
   38 +000040   05 RECORD-NO    PIC S9(3).                                                 WRKRCD
   39 +000050   05 FILLER       PIC X(1)  VALUE SPACE.                                     WRKRCD
   40 +000060   05 LOCATION     PIC A(3)  VALUE "NYC".                                     WRKRCD
   41 +000070   05 FILLER       PIC X(1)  VALUE SPACE.                                     WRKRCD
   42 +000080   05 NO-OF-DEPENDENTS                                                        WRKRCD
   43 +000090                    PIC X(2).                                                 WRKRCD
   44 +000100   05 FILLER       PIC X(7)  VALUE SPACES.                                    WRKRCD
   45  003500 77  WORKPTR USAGE POINTER.
       003600*****************************************************
       003700* EL SIGUIENTE PÁRRAFO ABRE EL ARCHIVO DE SALIDA QUE*
       003800* HA DE CREARSE E INICIALIZA LOS CONTADORES         *
       003900*****************************************************
   46  004000 PROCEDURE DIVISION.
       004100 STEP-1.
   47  004200     OPEN OUTPUT FILE-1.
   48  004300     MOVE ZERO TO KOUNT, NUMBR.
       004400*****************************************************
       004500* LOS 3 PÁRRAFOS SIGUIENTES CREAN INTERNAMENTE LOS  *
       004600* REGISTROS QUE HA DE CONTENER EL ARCHIVO, LOS      *
       004700* GRABA EN DISCO Y LOS VISUALIZA                    *
       004800*****************************************************
       004900 STEP-2.
   49  005000     ADD 1 TO KOUNT, NUMBR.
   50  005100     MOVE ALPHA  (KOUNT) TO NAME-FIELD.
   51  005200     MOVE DEPEND (KOUNT) TO NO-OF-DEPENDENTS.
   52  005300     MOVE NUMBR          TO RECORD-NO.
       005400 STEP-3.
   53  005500     DISPLAY WORK-RECORD.
   54  005600     WRITE RECORD-1 FROM WORK-RECORD.
       005700 STEP-4.
```

Figura 8 (Parte 1 de 2). Ejemplo de Listado Fuente COBOL/400

```
55  005800     PERFORM STEP-2 THRU STEP-3 UNTIL KOUNT IS EQUAL TO 26.
    005900**************************************************
    006000* EL SIGUIENTE PÁRRAFO CIERRA EL ARCHIVO ABIERTO    *
    006100* PARA SALIDA Y VUELVE A ABRIRLO PARA ENTRADA       *
    006200**************************************************
    006300 STEP-5.
56  006400     CLOSE FILE-1.
57  006500     OPEN INPUT FILE-1.
    006600**************************************************
    006700* LOS SIGUIENTES PÁRRAFOS LEEN HACIA ATRÁS EL       *
    006800* ARCHIVO Y SELECCIONAN EMPLEADOS SIN SUBORDINADOS  *
    006900**************************************************
    007000 STEP-6.
58  007100     READ FILE-1 RECORD INTO WORK-RECORD
59  007200       AT END GO TO STEP-8.
    007300 STEP-7.
60  007400     IF NO-OF-DEPENDENTS IS EQUAL TO "0"
61  007500       MOVE "Z" TO NO-OF-DEPENDENTS.
62  007600     GO TO STEP-6.
    007700 STEP-8.
63  007800     CLOSE FILE-1.
64  007900     STOP RUN.
                      * * * * *  F I N  D E  F U E N T E  * * * * *
```

Figura 8 (Parte 2 de 2). Ejemplo de Listado Fuente COBOL/400

La Figura 8 muestra los siguientes campos:

A *Número de instrucción generado por el compilador:* Los números aparecen a la izquierda del listado del programa fuente. Estos números tienen referencia en todos los listados de salida del compilador excepto en los listados de mensajes FIPS. Un número de instrucción puede abarcar varias líneas y una línea puede contener más de una instrucción.

B *Número de Referencia:* Los números aparecen a la izquierda de las instrucciones fuente. Los números que aparecen en este campo y el encabezamiento de columnas (mostrado como SEQNBR en este listado) se determinan mediante una opción especificada en el mandato CRTCBLPGM o en la instrucción PROCESS, como se muestra en la tabla siguiente:

Opción	Encabezamiento	Origen
NONUMBER	SEQNBR	Números de secuencia de archivo fuente
NUMBER	NUMBER	Números de secuencia estándar COBOL
LINENUMBER	LINNBR	Números se secuencia generados por el compilador

C *Columna del indicador de error de secuencia:* Una S en esta columna indica que la línea está fuera de la secuencia. La comprobación de secuencia se realiza en el campo de número de secuencia sólo si se especifica la opción SEQUENCE.

D *Copyname:* Copyname, tal como se especifica en la instrucción COPY COBOL, se lista para todos los registros incluidos en el programa fuente mediante esa instrucción COPY. Si se utiliza la frase DDS-ALL-FORMATS, aparece el nombre `<--ALL-FMTS` debajo de COPYNAME.

E *Campo cambiar/fecha:* Aquí se lista la fecha en que se ha modificado la línea por última vez.

Utilización de Verbos mediante el Listado de la Cuenta

La Figura 9 muestra la lista alfabética que se produce con todos los verbos utilizados en el programa fuente. También se incluye un contador para saber cuántas veces se ha utilizado un verbo. El listado se produce cuando se especifica la opción VBSUM.

```
5763CB1 V3R0M5  001000     Uso de Verbos COBOL AS/400 por Recuento     TESTER/SAMPLE    AS400SYS  03/27/94 11:01:51     Página  6
VERB                    COUNT
ADD                         1
CLOSE                       2
DISPLAY                     1
GO                          2
IF                          1
MOVE                        5
OPEN                        2
PERFORM                     1
READ                        1
STOP                        1
WRITE                       1
                  *****  F I N   D E   V E R B O   P O R   R E C U E N T O  *****
```

Figura 9. Utilización de Verbos mediante el Listado de la Cuenta

Correlación de la División de Datos

La correlación de la División de Datos se lista cuando se especifica la opción MAP. Contiene información sobre los nombres en el programa fuente COBOL. Al final de la correlación de la División de Datos se da el número de bytes necesarios para la Sección de Almacenamiento de Trabajo y la Sección de Archivos.

```
5763CB1 V3R0M5  001000      Mapa de División de Datos COBOL AS/400     TESTER/SAMPLE        AS400SYS  03/27/94 11:01:51    Página  7
 INST NIV  NOMBRE FUENTE                SECCIÓN    DISP    LONG    TIPO   I-NOMBRE   ATRIBUTOS
  F   G     H                            I         J       K       L      M          N
   18  FD  FILE-1                        FS                                .F01       DEVICE DISK, ORGANIZATION SEQUENTIAL,
                                                                                     ACCESS SEQUENTIAL, RECORD CONTAINS 20
                                                                                     CHARACTERS, LABEL RECORDS STANDARD
   22  01  RECORD-1                      FS  00000000       20  GROUP  .D00633C
   23  02  FIELD-A                       FS  00000000       20  AN     .D0063AE
   25  01  FILLER                        WS  00000000       56  GROUP  .D006420
   26  02  KOUNT                         WS  00000000        2  PACKED .D006490
   27  02  LETTERS                       WS  00000002       26  AN     .D006512  VALUE
   28  02  ALPHA                         WS  00000002        1  AN     .D0065B0  REDEFINES .D006512, DIMENSION(26)
   30  02  NUMBR                         WS  00000028        2  PACKED .D006632
   31  02  DEPENDENTS                    WS  00000030       26  AN     .D0066B4  VALUE
   32  02  DEPEND                        WS  00000030        1  AN     .D006754  REDEFINES .D0066B4, DIMENSION(26)
   35  01  WORK-RECORD                   WS  00000000       19  GROUP  .D0067D6
   36  02  NAME-FIELD                    WS  00000000        1  AN     .D00684C
   37  02  FILLER                        WS  00000001        1  AN     .D0068C0  VALUE
   38  02  RECORD-NO                     WS  00000002        3  ZONED  .D00693C
   39  02  FILLER                        WS  00000005        1  AN     .D0069C2  VALUE
   40  02  LOCATION                      WS  00000006        3  A      .D007A98  VALUE
   41  02  FILLER                        WS  00000009        1  AN     .D007B20  VALUE
   42  02  NO-OF-DEPENDENTS              WS  00000010        2  AN     .D007B9C
   44  02  FILLER                        WS  00000012        7  AN     .D007C16  VALUE
   45  77  WORKPTR                       WS  00000000       16  POINTR .D007C92
FILE SECTION utiliza 20 bytes de almacenamiento
WORKING-STORAGE SECTION utiliza 75 bytes de almacenamiento
              *****  F I N   D E   M A P A   D E   D I V I S I O N   D E   D A T O S  *****
```

Figura 10. Correlación de la División de Datos

La correlación de División de Datos muestra los campos siguientes:

F *Número de instrucción:* El número de instrucción generada por el compilador dónde se ha definido el ítem de los datos se lista para cada ítem de datos en la correlación de División de Datos.

G *Nivel de ítem de datos:* Aquí se lista el número del nivel de ítem de datos, como se especifica en el programa fuente. Los nombres de índice se identifican con una *IX* en el número de nivel y un campo de tipo en blanco.

H *Nombre fuente:* Aquí se lista el nombre de los datos, como se especifica en el programa fuente.

I *Sección:* La sección en la que se ha definido el ítem se muestra aquí mediante la utilización de los códigos siguientes:

```
FS  Sección de Archivo
WS  Sección de Almacenamiento de Trabajo
LS  Sección de Enlace
SM  Sección Clasificar/Fusionar
SR  Registro Especial
```

J *Desplazamiento:* Aquí se ofrece el desplazamiento del ítem, en bytes, desde el ítem de grupo de nivel 01.

K *Longitud:* Aquí se lista la longitud decimal del ítem.

L *Tipo:* Se muestra aquí el tipo de datos para el ítem mediante la utilización de los códigos siguientes:

GROUP	Ítem de grupo
A	Alfabético
AN	Alfanumérico
ANE	Alfanumérico editado
INDEX	Ítem de datos de índice (USAGE INDEX)
BOOLN	Booleano
ZONED	Decimal con Zona (decimal externo)
PACKED	Decimal empaquetado (decimal interno) (USAGE COMP, COMP-3 o PACKED-DECIMAL)
BINARY	Binario (USAGE COMP-4 o BINARY)
NE	Numérico editado
POINTR	Ítem de datos del puntero (USAGE POINTER)

M *Nombre interno:* Aquí se listan los nombres internos generados por el compilador y se asignan de la manera siguiente:

Nombre de archivos

El nombre interno utiliza el formato `.Fnn`, donde `.F` indica el nombre de archivo y `nn` es un número de dos dígitos único.

Nombres de Datos

El nombre interno utiliza el formato `.Dxxxxxx`, donde `.D` indica un nombre de índice o un nombre de datos, y `xxxxxx` es un valor hexadecimal de seis dígitos único. Estos nombres aparecen en el listado IRP si se genera.

N *Atributos:* Los atributos del ítem se listan de la siguiente manera:

- Para los archivos, puede darse la información siguiente:

  Tipo de dispositivo
  ORGANIZATION
  ACCESS MODE
  Información BLOCK CONTAINS
  Información RECORD CONTAINS
  Información LABEL
  Se indica RERUN

Se indica SAME AREA
Se indica CODE-SET
Se indica SAME RECORD AREA
Se indica LINAGE.

- Para los ítems de datos, los atributos indican si se ha especificado para el ítem la siguiente información:

  REDEFINES
  VALUE
  JUSTIFIED
  SYNCHRONIZED
  BLANK WHEN ZERO
  SIGN IS LEADING
  SIGN IS LEADING SEPARATE
  SIGN IS SEPARATE
  INDICATORS.

- Para los ítems de tabla, las dimensiones del elemento se listan aquí en el formato `DIMENSION (nn)`. Para cada dimensión, se da un valor OCCURS máximo. Cuando una dimensión es una variable, se lista como tal, dando los valores OCCURS más alto y más bajo.

Mensajes FIPS

Los mensajes FIPS en la Figura 11, se listan cuando se especifica el parámetro FLAGSTD. Consulte la página 25 para obtener más información sobre cómo especificar la opción para la opción FIPS. Sólo se listan los mensajes para el subconjunto FIPS solicitado, los módulos opcionales y/o los elementos obsoletos.

Nota: Se proporciona el número de secuencia y columna cada vez que se emite el mensaje.

```
5763CB1 V3R0M5  001000        Mensajes FIPS COBOL AS/400           TESTER/SAMPLE         AS400SYS  03/27/94 11:01:51     Página  8
 FIPS-ID    DESCRIPCIÓN Y NÚMEROS SECUENCIA  SEÑALIZADOS P
   Q
LBL8200     Los siguientes ítems que no se ajustan al estándar son válidos a un nivel FIPS intermedio o superior.
LBL8201     Instrucción COPY.
                003400  0008
LBL8300     Los siguientes ítems que no se ajustan al estándar son válidos a un nivel FIPS o superior.  Q
LBL8303     Párrafo DATE-COMPILED.
                000800  0010
LBL8500     Los siguientes ítems que no se ajustan al estándar son definidos por IBM o son extensiones IBM.  Q
LBL8504     Nombre de asignación en cláusula ASSIGN.
                001500  0036
LBL8518     USAGE IS COMPUTATIONAL-3.
                002600  0036
                003000  0036
LBL8520     USAGE IS POINTER.
                003500  0026
LBL8561     Instrucción COPY con bliblioteca por omisión asumida.
                003400  0019
   7 violaciones FIPS señalizadas.  R
        * * *  F I N  D E  M E N S A J E S  F I P S  C O B O L  * * *
```

Figura 11. Mensajes FIPS

Los mensajes FIPS están compuestos por los campos siguientes:

O *FIPS-ID*: Este campo lista el número del mensaje FIPS.

P *Descripción y números de referencia señalizados:* Este campo lista una descripción de la condición señalizada, seguido por una lista de los números de referencia del programa fuente donde se encuentra esta condición.

El tipo de números de referencia utilizados y sus nombres en el encabezamiento (mostrados como SEQUENCE NUMBERS en este listado) se determinan mediante una opción especificada en el mandato CRTCBLPGM o en la instrucción PROCESS, tal como se muestra en la tabla siguiente:

Opción	Encabezamiento
NONUMBER	DESCRIPCION Y NUMEROS DE SECUENCIA SEÑALIZADOS
NUMBER	DESCRIPCIÓN Y NÚMEROS SUMINISTRADOS POR USUARIO SEÑALIZADOS
LINENUMBER	DESCRIPCIÓN Y NÚMEROS DE LÍNEAS SEÑALIZADOS

Q *Ítems agrupados por nivel:* Estos encabezamientos subdividen los mensajes FIPS por nivel y categoría.

R *Violaciones FIPS señalizadas:* El número total de violaciones FIPS señalizadas se incluye al final del listado FIPS.

Mensajes SAA

La Figura 12 muestra los mensajes SAA que se listan cuando se especifica la opción de señalización SAA. Consulte el apartado Parámetro SAAFLAG en la página 26 o "Uso de la Instrucción PROCESS para Especificar las Opciones del Compilador" en la página 33 para obtener más información sobre cómo especificar esta opción.

```
5763CB1 V3R0M5  001000            Mensajes COBOL SAA            TESTER/SAMPLE         AS400SYS  03/27/94 11:01:51     Página 9
 MSGID      DESCRIPCIÓN, NÚMEROS DE SECUENCIA Y NÚMEROS DE COLUMNA SEÑALIZADOS

LBL8800     Los siguientes ítems se han señalizado como no portables por otros sistemas COBOL SAA.
LBL8801     Las opciones APOST,NUMBER,SEQUENCE,GRAPHIC,NOCRTF,NODUPKEYCHK,NOSYNC y EXTACCDSP no son COBOL SAA.
                000100  0008
LBL8809     Instrucción PROCESS.
                000100  0008
LBL8843     USAGE IS POINTER.
                003500  0026
   Se han señalizado 3 mensajes COBOL SAA.
        * * *  F I N  D E  M E N S A J E S  C O B O L  S A A  * * *
```

Figura 12. Mensajes SAA

Para obtener más información sobre la señalización SAA, consulte el apartado "Señalización SAA" en la página 349.

Listado de Referencias Cruzadas

La Figura 13 muestra el listado de referencias cruzadas que se genera cuando se especifica la opción XREF. Proporciona una lista de todas las referencias de datos y las referencias de nombres de procedimientos, por número de instrucción, dentro del programa fuente.

```
5763CB1 V3R0M5  001000    Listado de referencias cruzadas COBOL AS/400    TESTER/SAMPLE      AS400SYS  03/27/94 11:01:51   Página 10
NOMBRES (* = Nombre-procedimiento) REFERENCIAS DEFINIDAS (* = Modificado)
 S                                T    U
 ALPHA                            28   50
 DEPEND                           32   51
 DEPENDENTS                       31   32
*DUMMY-SECTION                    47
 FIELD-A                          23
 FILE-1                           18   15  47  56  57  58  63
 KOUNT                            26   48* 49* 50  51  55
 LETTERS                          27   28
 LOCATION                         40
 NAME-FIELD                       36   50*
 NO-OF-DEPENDENTS                 42   51* 60  61*
 NUMBR                            30   48* 49* 52
 RECORD-NO                        38   52*
 RECORD-1                         22   21  54*
*STEP-1                           47
*STEP-2                           49   55
*STEP-3                           53   55
*STEP-4                           55
*STEP-5                           56
*STEP-6                           58   62
*STEP-7                           60
*STEP-8                           63   59
 WORK-RECORD                      35   53  54  58*
 WORKPTR                          45
                 * * * * *  F I N  D E  R E F E R E N C I A S  C R U Z A D A S * *
```

Figura 13. Listado de Referencias Cruzadas

El listado de referencias cruzadas muestra los siguientes campos:

S *Campo de nombres:* Aquí se lista el nombre de datos o el nombre de procedimiento referenciado. Todos los nombres de procedimiento se señalizan con un * antes del nombre. Los nombres se listan por orden alfabético.

T *Campo definido:* Aquí se lista el número de instrucción donde se ha definido el nombre dentro del programa fuente.

U *Campo de referencias:* Todos los números de instrucciones se listan en la misma secuencia en que aparece la referencia al nombre en el programa fuente. Un * que sigue a un número de secuencia indica que el ítem se ha modificado en esa instrucción.

Mensajes

La Figura 14 muestra los mensajes que se generan durante la compilación del programa.

```
5763CB1 V3R0M5  001000           Mensajes COBOL AS/400           TESTER/SAMPLE        AS400SYS  03/27/94 11:01:51     Página 11
 INST                    X
*V    18  MSGID: LBL0650  GRAVEDAD: 00  NUMSEC:  001800  W
       Mensaje . . . . :   Bloqueo/Desbloqueo para archivo 'FILE-1' se
       ejecutará por código generado por compilador.  Y
                    * * * * *  F I N   D E   M E N S A J E S  * * * * *
                                      Resumen Mensajes
 Total    Info(0-4)    Aviso(5-19)      Error(20-29)    Grave(30-39)    Terminal(40-99)
 Z     1          1             0               0             0             0
Registros fuente leídos . . . . . :   79
Registros de copia leídos . . . . :   10
Miembros de copia procesados  . . :   1
Errores de secuencia  . . . . . . :   0
Mensaje de gravedad emitido más alto:   0
 LBL0901 00  Programa SAMPLE creado en biblioteca TESTER.
               * * * * *  F I N   D E   C O M P I L A C I O N  * * * * *
```

Figura 14. Mensajes de Diagnóstico

Los campos que se muestran son:

V *Número de instrucción:* Este campo lista el número de instrucciones generadas por el compilador asociadas con la instrucción del programa fuente para la que se ha emitido el mensaje. [1]

W *Número de referencia:* Aquí se emite el número de referencia. [1] Los números que aparecen en este campo y el encabezamiento de la columna (aquí se muestra como SEQNBR) vienen determinados por una opción especificada en el mandato CRTCBLPGM o en la instrucción PROCESS, tal como se muestra en la tabla siguiente:

Opción	Encabezamiento	Origen
NONUMBER	SEQNBR	Números de secuencia de archivo fuente
NUMBER	NUMBER	Números de secuencia suministrados por el usuario
LINENUMBER	LINNBR	Números se secuencia generados por el compilador

Cuando se emite un mensaje para un registro desde un archivo de copia, el número vienen precedido por el signo +.

X *MSGID* y *Nivel de Gravedad:* Estos campos contienen el número de mensaje y el nivel de gravedad asociado. Los niveles de gravedad se definen de la manera siguiente:

```
00  Informativo
10  Aviso
20  Error
30  Error Grave
40  Irrecuperable (generalmente error del usuario)
50  Irrecuperable (generalmente error del compilador)
```

Y *Mensaje:* El mensaje identifica la condición e indica la acción emprendida por el compilador.

[1] El número de instrucciones y el número de referencia no aparecen en ciertos mensajes que hacen referencia a ítems desaparecidos. Por ejemplo, si el párrafo PROGRAM-ID no aparece, el mensaje LBL0031 aparecerá en el listado sin ninguna instrucción ni número de referencia listado.

Z *Estadísticas de mensajes:* Este campo lista el numero total de mensajes y el número de mensajes por nivel de gravedad.

Los totales listados son el número de mensajes que genera el compilador para cada gravedad y no siempre son el número listado. Por ejemplo, si se especifica FLAG(10), no se lista ningún mensaje de gravedad menor de 10. Las cuentas, no obstante, indican el número que debería imprimirse si no se hubiera suprimido.

Capítulo 4. Ejecución del Programa en COBOL

Este capítulo proporciona la información necesaria para ejecutar el programa en COBOL/400.

Para ello, las formas más usuales consisten en la utilización de:

- Un mandato CALL del lenguaje de control (CL)
- Una instrucción CALL de COBOL
- Un programa de menú orientado a la aplicación
- La emisión de un mandato creado por el usuario.

Un mandato CALL del lenguaje de control puede formar parte de un trabajo por lotes, entrarlo un usuario de la estación de trabajo en forma interactiva o incluirlo en un programa CL. Un ejemplo del mandato CALL sería CALL NOMINA. NOMINA es el nombre de un programa COBOL que se llama y se ejecuta.

Un programa COBOL puede llamar a otro programa con la instrucción CALL de COBOL. (Consulte la sección “Instrucción CALL” de la publicación *COBOL/400 Reference* para más información).

Otra forma de ejecutar el programa COBOL es mediante un menú orientado a la aplicación. El usuario de la estación de trabajo puede seleccionar una opción de un menú y llamar al programa pertinente. La figura siguiente ilustra un ejemplo de un menú orientado a la aplicación.

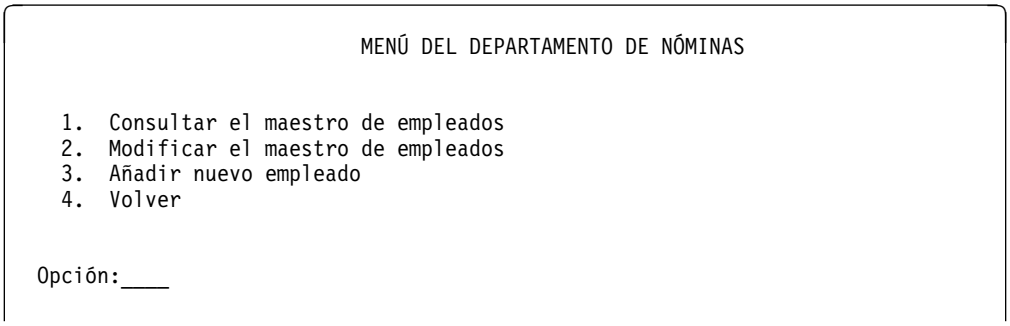


Figura 15. Ejemplo de un Menú de Aplicación

El menú mostrado en esta figura normalmente se visualiza mediante un programa de lenguaje de control en el que cada opción llama a un programa COBOL distinto.

El propio usuario puede crear también un mandato para ejecutar un programa en COBOL utilizando una definición de mandato. Una **definición de mandato** es un objeto que contiene la definición de un mandato (incluido el nombre del mandato, las descripciones de parámetro y la información de control de validez) e identifica al programa que efectúa la función solicitada por el mandato. El identificador del objeto que el sistema reconoce es *CMD.

Por ejemplo, puede crear un mandato, PAGA, que llame a un programa, NOMINA. PAYROLL es el nombre de un programa COBOL que se llama y se ejecuta. El usuario puede entrar el mandato de forma interactiva o en un trabajo por lotes. Consulte la publicación *CL Guía del Programador* para más información acerca de la utilización de la definición de mandato.

Cuando un programa COBOL finaliza normalmente, el sistema devuelve el control al llamador. El llamador puede ser un usuario de la estación de trabajo, un programa en lenguaje de control (como el programa de manejo del menú) o cualquier otro programa en COBOL.

Si un programa COBOL finaliza de forma anómala durante el tiempo de ejecución, se emite el mensaje LBE9001

```
Error id-mensaje provocó terminación del programa
```

Un programa CL puede supervisar esta excepción utilizando el mandato Supervisar Mensaje (MONMSG). Consulte la publicación *CL Reference* para más información acerca de los mandatos del lenguaje de control.

Si un programa finaliza por una razón que no sea la utilización de la instrucción STOP o finaliza de forma anómala al término del programa, el código de retorno se establece en 2. Consulte los mandatos RTVJOBA y DSPJOB en la publicación *CL Guía del Programador* para más información acerca de los códigos de retorno.

Cuando se ejecuta un trabajo por lotes que utiliza la instrucción ACCEPT, los datos de entrada se toman de la corriente de trabajos. Estos datos deben colocarse inmediatamente después del mandato CALL para el programa en COBOL. Es responsabilidad del usuario la solicitud (mediante varias instrucciones ACCEPT) de la misma cantidad de datos disponibles. Consulte la sección “Instrucción ACCEPT” de la publicación *COBOL/400 Reference* para obtener más información.

Nota: Si se solicitan más datos de los disponibles, el mandato CL que sigue a los datos se trata como dato de entrada. Si hay más datos disponibles de los que se solicitan, cada línea de datos de más se trata como un mandato CL. En cada uno de estos casos, pueden producirse resultados no deseados.

Respuesta a Mensajes de Consulta en Tiempo de Ejecución

Cuando se ejecuta un programa COBOL, pueden generarse mensajes de consulta en tiempo de ejecución. Los mensajes necesitan una respuesta del usuario antes de que el programa continúe ejecutándose.

Se puede añadir los mensajes de consulta a una lista de respuestas del sistema para proporcionar respuestas automáticas a los mensajes. Las respuestas para estos mensajes pueden especificarse de forma individual o general. Este método de respuesta a mensajes de consulta es especialmente útil para los programas por lotes que, de lo contrario, necesitarían un operador para emitir las respuestas.

Los siguientes mensajes de consulta de COBOL/400 pueden añadirse a la lista de respuestas del sistema:

LBE7200
LBE7201
LBE7203
LBE7204
LBE7205
LBE7206
LBE7207
LBE7208
LBE7209
LBE7210
LBE7211

LBE7604.

La lista de respuestas sólo se utiliza cuando un trabajo que tiene especificado el atributo de respuesta de mensajes de consulta (INQMSGRPY) como `INQMSGRPY(*SYSRPYL)` envía un mensaje de consulta.

El parámetro INQMSGRPY se produce en los mandatos de lenguaje de control siguientes:

- Cambiar Trabajo (CHGJOB)
- Cambiar Descripción de Trabajo (CHGJOBD)
- Crear Descripción de Trabajo (CRTJOBD)
- Someter Trabajo (SBMJOB).

Puede seleccionar una de estas cuatro modalidades de respuesta especificando uno de los siguientes valores para el parámetro INQMSGRPY:

SAME	No se producen cambios en la forma de enviar respuestas a mensajes de consulta
RQD	Todos los mensajes de consulta requieren una respuesta por parte del receptor de dichos mensajes
DFT	Se emite una respuesta por omisión
SYSRPYL	La lista de respuestas del sistema se comprueba para una entrada de lista de respuestas coincidente. Si se produce la coincidencia, se utiliza el valor de respuesta en esa entrada. Si no hay entrada para ese mensaje de consulta, se necesita una respuesta.

Puede utilizar el mandato Añadir Entrada de Lista de Respuestas (ADDRPYLE) para añadir las entradas en la lista de respuestas del sistema, o el mandato Trabajar con Entrada de Lista de Respuestas (WRKRPYLE) para cambiar o eliminar entradas en la lista de respuestas del sistema. Vea la publicación *CL Reference* para más detalles acerca de los mandatos ADDRPYLE y WRKRPYLE. También podrá responder a mensajes de consulta en tiempo de ejecución con un manejador de errores que haya definido. Para más información acerca de las API de manejo de errores, consulte la publicación *System Programmer's Interface Reference*.

Capítulo 5. Depuración del Programa

El lenguaje COBOL/400 y el sistema operativo OS/400 proporciona funciones para la depuración de los programas desarrollados por el usuario. Este capítulo describe aquellas funciones que le permitirán depurar los programas.

Funciones OS/400	Funciones COBOL/400
Puntos de interrupción	Características de depuración
Rastreos	Vuelco con formato

Las funciones OS/400 le permitirán probar programas a la vez que proteger los archivos de producción; podrá, asimismo, observar y depurar operaciones a medida que se ejecuta un programa. No se necesita ningún código fuente especial para utilizar las funciones OS/400.

Las funciones COBOL pueden utilizarse independientemente de las funciones OS/400 o en combinación con ellas para:

- Depurar un programa
- Producir un vuelco con formato de los contenidos de los campos, estructuras de datos, matrices y tablas.

El código fuente es necesario para la utilización de las características de depuración de COBOL y la posibilidad de vuelco con formato. Un vuelco con formato también puede obtenerse mediante una respuesta del usuario al mensaje en tiempo de ejecución.

El contenido de OPEN-FEEDBACK y de I-O-FEEDBACK proporciona información de depuración adicional. El método para la obtención de esta información se describe más adelante en este capítulo, concretamente en el apartado “Estado de Archivos y Áreas de Realimentación” en la página 108.

Mientras prueba los programas, asegúrese de que la lista de bibliotecas haya cambiado, para direccionar los programas a una biblioteca de prueba que contenga datos de prueba; de esta manera, ningún dato real existente se verá afectado.

Para evitar que los archivos de base de datos en bibliotecas de producción se modifiquen de manera no intencionada, puede especificar UPDPROD(*NO) en el mandato Arrancar Depuración (STRDBG) o utilizando el mandato Cambiar Depuración (CHGDBG). Vea la publicación *CL Reference* para más información.

Nota: Consulte el manual *CL Guía del Programador* para los mandatos CL necesarios para los programas de prueba y depuración.

No hay instrucciones especiales para la prueba en el programa que se va a probar. El programa puede ejecutarse de forma normal sin modificación alguna. Todas las funciones de prueba se especifican en el trabajo que contiene el programa, no en el mismo programa.

Las funciones de prueba sólo se aplican al trabajo en el que se especifiquen. Un programa puede utilizarse simultáneamente en dos trabajos: uno que esté en un entorno de prueba y otro que esté en un entorno normal de proceso.

Las funciones de prueba le permitirán observar las operaciones que se realizan mientras el programa se ejecuta. Estas funciones incluyen la utilización de puntos de interrupción y de rastreos. (Consulte los apartados “Utilización de Puntos de Interrupción” en la página 59 y “Utilización de un Rastreo” en la página 66 para más información).

Cómo Evitar Errores de Códigos Comunes

Los errores que más frecuentemente cometen los programadores en COBOL se pueden dividir en dos categorías: errores de tiempo de compilación y errores de tiempo de ejecución.

El compilador puede detectar los errores al compilar el programa fuente. Aunque el compilador hace correcciones basándose en ciertas suposiciones sobre los errores que encuentra, es preciso que el usuario corrija el fuente y compile de nuevo si tiene errores.

Los errores de codificación comunes incluyen:

- Descripciones de registros no coincidentes con archivos descritos externamente
- Archivos de copia no encontrados
- Faltas de ortografía
- Puntuación defectuosa, especialmente la falta de puntos
- Sintaxis incorrecta o incompleta
- Mala utilización de palabras reservadas.

A continuación viene una serie de errores que sólo aparecen cuando se ejecuta el programa:

- Imposibilidad de hacer coincidir la descripción de registros en el programa fuente con el formato de los registros reales en el archivo que debe leerse. Puede ser un error producido por el usuario (los registros son correctos, pero la descripción es incorrecta) o un error causado por la persona que ha creado los registros que lee el programa. (Por ejemplo, la descripción es correcta, pero se han entrado uno o más registros de forma incorrecta.)
- Traslado de un ítem de datos cuyo subíndice o índice es demasiado grande, es negativo o es 0. Este traslado puede recubrir y destruir parte del código o puede recoger datos defectuosos.
- Olvidar la definición de un campo de signo para ítems que pueden tener valores negativos. (En tal caso, se pierde el signo y el número negativo se convierte, por error, en positivo).
- Traslado de datos en un área demasiado pequeña para ellos, provocando un truncamiento no deseado.
- Olvidar inicializar los ítems de datos en la sección de Almacenamiento de Trabajo antes de utilizarlos. Esto puede dar como resultado un error de datos decimal.
- En un programa llamado, la coincidencia incorrecta entre la descripción de datos de la Sección de Enlace con los del llamador. O, en el programa de llamada, la identificación errónea de los datos que se van a pasar.
- Traslado de un ítem de grupo a otro ítem de grupo cuando las descripciones de datos subordinados son incompatibles.

- Especificación de USAGE para un ítem de datos redefinido que sea distinto del USAGE especificado originalmente para el ítem redefinido, y después olvidarse del cambio una vez que se haya llevado a cabo la redefinición.
- Incluir una instrucción GO TO sin nombre de procedimiento, y no inicializarlo con una instrucción ALTER antes de que el programa en ejecución alcance ese punto.
- No incluir las cláusulas AT END o INVALID KEY o los procedimientos USE en archivos descritos en el programa.
- No hacer coincidir la descripción de registro fuente del archivo TRANSACTION con la descripción de registro del formato de pantalla.

Utilización de Puntos de Interrupción

Un punto de interrupción es un número de instrucción o una etiqueta en el programa que detiene el proceso del programa, y pasa el control al usuario de la estación de trabajo o a un programa específico. Si se utiliza un número de instrucción, dicho número puede aparecer en el listado del programa fuente COBOL. Si utiliza una etiqueta como punto de interrupción, la etiqueta:

- Puede estar asociada con una función realizada por el programa COBOL. Por ejemplo,

  ```
  .OPEN
  ```

  indica la función de abrir archivo.
- Puede ser una etiqueta interna generada por el compilador del COBOL. Por ejemplo,

  ```
  .L000001
  ```

  indica la primera etiqueta generada internamente.

Nota: Para determinar las etiquetas generadas internamente para el programa, utilice el parámetro GENOPT en el mandato CRTCBLPGM para obtener un listado IRP del programa.

Cuando una instrucción de punto de interrupción va a ejecutarse en un trabajo interactivo, el sistema visualiza el punto de interrupción en el que el programa se ha parado y, si se solicitó, visualiza los valores de las variables de programa. Una vez haya conseguido esta información (en una pantalla), puede ir a la pantalla de Entrada de Mandatos e introducir mandatos del OS/400 para solicitar otras funciones (como por ejemplo la visualización o cambio de una variable, la adición de un punto de interrupción o la adición de un rastreo). Consulte la publicación *CL Guía del Programador* para más información respecto a los puntos de interrupción.

Para un trabajo por lotes, puede llamarse a un programa de punto de interrupción cuando se alcanza el punto de interrupción. La información del punto de interrupción se pasa al programa de punto de interrupción.

Ejemplo de un Programa que Utiliza Puntos de Interrupción

La Figura 16 muestra un ejemplo de un programa COBOL que utiliza puntos de interrupción. Los siguientes mandatos OS/400 añaden puntos de interrupción en las instrucciones 43 y 52. El valor de la variable KOUNT se visualiza cuando se alcanza el punto de interrupción en la instrucción 52.

Mandatos OS/400:

```
STRDBG     TESTPRT
ADDBKP     STMT(43)
ADDBKP     STMT(52)
           PGMVAR(KOUNT)
```

Los mandatos OS/400 se exponen en el manual *CL Reference*.

```
5763CB1 V3R0M5  001000            Fuente COBOL AS/400              TESTER/TESTPRT         AS400SYS  03/30/94 17:05:37     Página  2
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
    1  000100 IDENTIFICATION DIVISION.
    2  000200 PROGRAM-ID.     TESTPRT.
    3  000300   AUTHOR.       PROGRAMMER NAME.
    4  000400   INSTALLATION. COBOL DEVELOPMENT CENTRE.                                                  03/30/94
    5  000500   DATE-WRITTEN. 11/27/87.
    6  000600   DATE-COMPILED.  03/30/94 17:05:37   .
    7  000700 ENVIRONMENT DIVISION.
    8  000800 CONFIGURATION SECTION.
    9  000900 SOURCE-COMPUTER. IBM-AS400.                                                               03/30/94
   10  001000 OBJECT-COMPUTER. IBM-AS400.                                                               03/30/94
   11  001100 INPUT-OUTPUT SECTION.
   12  001200 FILE-CONTROL.
   13  001300     SELECT FILE-1 ASSIGN TO DISK-SAMPLE.
   14  001400 DATA DIVISION.
   15  001500 FILE SECTION.
   16  001600 FD  FILE-1
   17  001700     LABEL RECORDS ARE STANDARD
   18  001800     RECORD CONTAINS 20 CHARACTERS
   19  001900     DATA RECORD IS RECORD-1.
   20  002000 01  RECORD-1.
   21  002100   02 FIELD-A      PIC X(20).
   22  002200 WORKING-STORAGE SECTION.
   23  002300 01  FILLER.
   24  002400   05 KOUNT        PIC S9(2) COMP-3.
   25  002500   05 LETTERS      PIC X(26) VALUE "ABCDEFGHIJKLMNOPQRSTUVWXYZ".
   26  002600   05 ALPHA REDEFINES LETTERS
   27  002700                   PIC X(1)  OCCURS 26 TIMES.
   28  002800   05 NUMBR        PIC S9(2) COMP-3.
   29  002900   05 DEPENDENTS   PIC X(26) VALUE "01234012340123401234012340".
   30  003000   05 DEPEND REDEFINES DEPENDENTS
   31  003100                   PIC X(1)  OCCURS 26 TIMES.
   32  003200 01  WORK-RECORD.
   33  003300   05 NAME-FIELD   PIC X(1).
   34  003400   05 FILLER       PIC X(1)  VALUE SPACE.
   35  003500   05 RECORD-NO    PIC S9(3).
   36  003600   05 FILLER       PIC X(1)  VALUE SPACE.
   37  003700   05 LOCATION     PIC A(3)  VALUE "NYC".
   38  003800   05 FILLER       PIC X(1)  VALUE SPACE.
   39  003900   05 NO-OF-DEPENDENTS
   40  004000                   PIC X(2).
   41  004100   05 FILLER       PIC X(7)  VALUE SPACES.
       004200**************************************************
       004300* EL SIGUIENTE PARRAFO ABRE EL ARCHIVO DE SALIDA QUE*
       004400* HA DE CREARSE E INICIALIZA LOS CONTADORES         *
       004500**************************************************
   42  004600 PROCEDURE DIVISION.
       004700 STEP-1.
   43  004800     OPEN OUTPUT FILE-1. 1
   44  004900     MOVE ZERO TO KOUNT, NUMBR.
```

Figura 16 (Parte 1 de 2). Ejemplo de Programa COBOL que Utiliza Puntos de Interrupción

```
5763CB1 V3R0M5  001000            Fuente COBOL AS/400              TESTER/TESTPRT         AS400SYS  03/30/94 17:05:37     Página  2
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
      005000*****************************************************
      005100* LOS 3 PARRAFOS SIGUIENTES CREAN INTERNAMENTE LOS  *
      005200* REGISTROS QUE HA DE CONTENER EL ARCHIVO, LOS      *
      005300* GRABA EN DISCO Y LOS VISUALIZA                    *
      005400*****************************************************
      005500 STEP-2.
   45 005600     ADD 1 TO KOUNT, NUMBR.
   46 005700     MOVE ALPHA  (KOUNT) TO NAME-FIELD.
   47 005800     MOVE DEPEND (KOUNT) TO NO-OF-DEPENDENTS.
   48 005900     MOVE NUMBR          TO RECORD-NO.
      006000 STEP-3.
   49 006100     DISPLAY WORK-RECORD.
   50 006200     WRITE RECORD-1 FROM WORK-RECORD.
      006300 STEP-4.
   51 006400     PERFORM STEP-2 THRU STEP-3 UNTIL KOUNT IS EQUAL TO 26.
      006500*****************************************************
      006600* EL SIGUIENTE PARRAFO CIERRA EL ARCHIVO ABIERTO    *
      006700* PARA SALIDA Y VUELVE A ABRIRLO PARA ENTRADA       *
      006800*****************************************************
      006900 STEP-5.
   52 007000     CLOSE FILE-1. 2
   53 007100     OPEN INPUT FILE-1.
      007200*****************************************************
      007300* LOS SIGUIENTES PARRAFOS LEEN HACIA ATRAS EL       *
      007400* ARCHIVO Y SELECCIONAN EMPLEADOS SIN SUBORDINADOS  *
      007500*****************************************************
      007600 STEP-6.
   54 007700     READ FILE-1 RECORD INTO WORK-RECORD
   55 007800       AT END GO TO STEP-8.
      007900 STEP-7.
   56 008000     IF NO-OF-DEPENDENTS IS EQUAL TO "0"
   57 008100       MOVE "Z" TO NO-OF-DEPENDENTS.
   58 008200     GO TO STEP-6.
      008300 STEP-8.
   59 008400     CLOSE FILE-1.
   60 008500     STOP RUN.
                        * * * * *  F I N   D E   F U E N T E  * * * * *
```

Figura 16 (Parte 2 de 2). Ejemplo de Programa COBOL que Utiliza Puntos de Interrupción

1 El primer punto de interrupción se utiliza para saber dónde está el usuario en el programa. Cuando se produce la interrupción, se visualiza la información siguiente:

```
                              Visualizar Punto de Interrupción

 Sentencia/Instrucción . . . . . . . . . :   43 /0017
 Programa  . . . . . . . . . . . . . . . :   TESTPRT
 Nivel de repetición . . . . . . . . . . :   1









 Pulse Intro para continuar.

  F3=Salir Programa  F10=Entrada de mandatos
```

Figura 17. Primer Punto de Interrupción Visualizado

2 La información siguiente se visualiza como resultado de alcanzar el segundo punto de interrupción:

```
                          Visualizar Punto de Interrupción

 Sentencia/Instrucción . . . . . . . . . : 52 /0056
 Programa. . . . . . . . . . . . . . . . : TESTPRT
 Nivel de repetición . . . . . . . . . . : 1
 Posición inicial  . . . . . . . . . . . : 1
 Formato . . . . . . . . . . . . . . . . : *CHAR
 Longitud  . . . . . . . . . . . . . . . : *DCL

 Variable. . . . . . . . . . . . . . . . : 05  KOUNT
   Tipo. . . . . . . . . . . . . . . . :   PACKED
   Longitud  . . . . . . . . . . . . . :   2 0
  ' 26'




Pulse Intro para continuar.

 F3=Salir Programa  F10=Entrada de mandatos
```

Figura 18. Segundo Punto de Interrupción Visualizado

Para especificar una variable para el parámetro PGMVAR, empiece cada nombre que entre con un carácter alfanumérico (A a la Z, $, #, o @). Puede ir seguido de los caracteres (de la A a la Z, del 0 al 9, $, #, @, o _).

El ejemplo siguiente muestra cómo visualizar una variable COBOL, RECORD-NO, en el ejemplo de programa. Debido a que el sistema operativo OS/400 trata al guión como un carácter especial, RECORD-NO debe ir entre comillas.

```
STRDBG      TESTPRT
ADDBKP      STMT(58)
            PGMVAR('RECORD-NO')
```

Para visualizar el valor de un ítem de tabla, entre los números de apariciones adecuadas (subíndices) con el nombre de variable. Se permiten hasta siete dimensiones de subíndices, que deben estar separados por comas.

No utilice un nombre de índice ni un ítem de datos de índices como subíndice. Cuando se introduce un índice como subíndice, el sistema operativo utiliza el valor interno del índice como subíndice y pueden producirse resultados no deseados.

El ejemplo siguiente muestra cómo se debe especificar la variable COBOL TABLE1 con tres dimensiones.

```
PGMVAR('TABLE1(SUB1, SUB2, SUB3)')
```

Se permiten uno o más espacios en blanco después de cada coma que separa subíndices, pero la longitud total de la variable más los subíndices, paréntesis, comas y espacios en blanco especificados con la palabra clave PGMVAR no puede sobrepasar los 132 caracteres. Para más información acerca de la codificación de variables en mandatos CL, consulte el manual *CL Reference*.

Los nombres de variables pueden calificarse en el parámetro PGMVAR. Por ejemplo:

```
PGMVAR('NAME-FIELD OF WORK-RECORD')
```

Puede utilizarse otra técnica para visualizar variables que no sean ítems de una tabla multidimensional. Por ejemplo, para visualizar el campo NAME-FIELD, puede utilizar el mapa de la correlación de División de Datos COBOL para encontrar su nombre interno en COBOL (I-NAME). A continuación, utilice el listado de referencias cruzadas IRP para encontrar el numero de la Tabla de Definición de Objetos (ODT) para el nombre interno. (Consulte el apartado “Uso de la Instrucción PROCESS para Especificar las Opciones del Compilador” en la página 33 para más información sobre la obtención estos listados). La Figura 19 muestra el mapa de la correlación de División de Datos, y la Figura 20 en la página 64 muestra el listado de referencias cruzadas para el ejemplo de programa, TESTPRT.

```
 5763CB1 V3R0M5  001000      Mapa de División de Datos COBOL AS/400     TESTER/TESTPRT         AS400SYS  03/30/94 17:05:37    Página 4
  INST NIV  NOMBRE FUENTE                SECCIÓN    DISP     LONG  TIPO   I-NOMBRE   ATRIBUTOS
    16  FD  FILE-1                          FS                            .F01      DEVICE DISK, ORGANIZATION SEQUENTIAL,
                                                                                   ACCESS SEQUENTIAL, RECORD CONTAINS 20
                                                                                   CHARACTERS, LABEL RECORDS STANDARD
    20  01  RECORD-1                        FS  00000000       20  GROUP  .D00633C
    21  02  FIELD-A                        FS  00000000       20  AN     .D0063AE
    23  01  FILLER                         WS  00000000       56  GROUP  .D006420
    24  02  KOUNT                          WS  00000000        2  PACKED .D006490
    25  02  LETTERS                        WS  00000002       26  AN     .D006512  VALUE
    26  02  ALPHA                          WS  00000002        1  AN     .D0065B0  REDEFINES .D006512, DIMENSION(26)
    28  02  NUMBR                          WS  00000028        2  PACKED .D006632
    29  02  DEPENDENTS                     WS  00000030       26  AN     .D0066B4  VALUE
    30  02  DEPEND                         WS  00000030        1  AN     .D006754  REDEFINES .D0066B4, DIMENSION(26)
    32  01  WORK-RECORD                    WS  00000000       19  GROUP  .D0067D6
    33  02  NAME-FIELD                     WS  00000000        1  AN     .D00684C  1
    34  02  FILLER                         WS  00000001        1  AN     .D0068C0  VALUE
    35  02  RECORD-NO                      WS  00000002        3  ZONED  .D00693C
    36  02  FILLER                         WS  00000005        1  AN     .D0069C2  VALUE
    37  02  LOCATION                       WS  00000006        3  A      .D006A98  VALUE
    38  02  FILLER                         WS  00000009        1  AN     .D006B20  VALUE
    39  02  NO-OF-DEPENDENTS               WS  00000010        2  AN     .D006B9C
    41  02  FILLER                         WS  00000012        7  AN     .D006C16  VALUE
FILE SECTION utiliza 20 bytes de almacenamiento
WORKING-STORAGE SECTION utiliza 75 bytes de almacenamiento
                * * * * *  F I N  D E  M A P A  D E  D I V I S I O N  D E  D A T O S  * * * * *
```

Figura 19. Mapa de Correlación de División de Datos para TESTPRT

1 El I-NAME para NAME-FIELD

```
5763SS1 V3R0M5 920925    IBM COBOL/400 5763CB1 V3R0M5      IRP LISTADO PARA TESTPRT                  03/30/94 17:05:37  Página 43
ODT  Nombre ODT                     Referencia Cruzada SEQ     (* Indica donde está definido)
0184 .DMPFBH1 514*
0185 .DMPFBH2 515*
014F .DMPFBIB 452*
0148 .DMPFBLN 445*
015B .DMPFBLO 471*
0186 .DMPFBLP 512 516*
0182 .DMPFBLS 512*
014C .DMPFBL1 449* 1065 1066
014D .DMPFBL2 450*
0160 .DMPFBMF 476*
014E .DMPFBMN 451* 995 1098 1099 1118 1119
0180 .DMPFBND 509*
0150 .DMPFBOB 453*
015A .DMPFBOF 470*
0152 .DMPFBOL 458*
015F .DMPFBPO 475*
0161 .DMPFBQN 477*
0155 .DMPFBRC 461*
0153 .DMPFBRW 459*
0158 .DMPFBSC 468*
0149 .DMPFBSF 446*
014A .DMPFBSL 447*
014B .DMPFBSN 448*
0146 .DMPFBTY 443* 1097 1117
0159 .DMPFBUF 469*
0183 .DMPFBVL 513*
018B .DMPIOFB 522*
01A0 .DMPIOFS 545* 546 547
01A6 .DMPKYLN 551*
0165 .DMPNDEV 481* 1087 1145
0144 .DMPOFBS 441* 442 443 444 445 446 447 448 449 450 451 452 453 454 458 459 460 461 462 467 468 469 470 471 472 473 474 475 476
             477 478 479 480 481 482 508 509 510
01AA .DMPRCD  555*
01AC .DMPRCDN 557*
01AE .DMPRDUP 559*
01A1 .DMPRFMT 546*
01A7 .DMPRRN  552*
01A5 .DMPSRC  550*
0220 .D006A98 685*
0221 .D006B20 686*
0222 .D006B9C 687* 767 914 916
0223 .D006C16 688*
0211 .D0063AE 670*
0210 .D00633C 669* 789 904
0212 .D006420 671* 672 673 676 677
0213 .D006490 672* 753 757 761 765 815
0216 .D0065B0 675* 763
0214 .D006512 673* 674
0218 .D0066B4 677* 678
0217 .D006632 676* 754 758 769
021B .D0067D6 680* 681 682 683 684 685 686 687 688 778 789 904
021A .D006754 679* 767
021D .D0068C0 682*
021C .D00684C 681* 763  1
021F .D0069C2 684*
021E .D00693C 683* 769
```

Figura 20. Sección del Listado de Referencias Cruzadas IRP para TESTPRT

1 021C es el número ODT para NAME-FIELD

Ahora podrá utilizar el número ODT 021C (para NAME-FIELD), con los mandatos siguientes, para añadir un punto de interrupción al ejemplo de programa en la instrucción 52.

```
STRDBG      TESTPRT
ADDBKP      STMT(52)
            PGMVAR('/021C')
```

Estos mandatos se exponen en el manual *CL Reference*.

Cuando se alcanza este punto de interrupción, se visualiza lo siguiente:

```
                        Visualizar Punto de Interrupción

Sentencia/Instrucción . . . . . : 52 /0056
Programa. . . . . : TESTPRT
Nivel de repetición . . . . . : 1
Posición inicial  . . . . . : 1
Formato . . . . . : *CHAR
Longitud  . . . . . : *DCL

Variable. . . . . : /021C
  Tipo. . . . . :   CHARACTER
  Longitud. . . . . :   2
  *...+....1....+....2....+....3....+....4....+....5
 'Z'




Pulse Intro para continuar.

F3=Salir Programa  F10=Entrada de mandatos
```

Figura 21. Punto de Interrupción en la Instrucción 52

Cambio de Variables de Programa

Ahora podrá cambiar el valor de las variables de programa para alterar el proceso del programa. Puede utilizar el mandato Cambiar Variable de Programa (CHGPGMVAR) para cambiar el valor de una variable. Este proceso se explica con más detalle en el manual *CL Reference*.

Puede utilizar el mandato DSPPGMVAR para visualizar ítems de datos de puntero, pero no podrá utilizar el mandato CHGPGMVAR para cambiar ítems de datos de puntero. Para cambiar los ítems de datos de puntero, deberá utilizar los mandatos CHGHLLPTR o CHGPTR. Para más información acerca de los mandatos CHGHLLPTR y CHGPTR, consulte el manual *CL Reference*.

Consideraciones para la Utilización de Puntos de Interrupción

Debe conocer las siguientes características de los puntos de interrupción antes de utilizarlos:

- Si se elude un punto de interrupción, por ejemplo la instrucción GO TO, ese punto de interrupción no se procesa.
- Cuando se establece un punto de interrupción en una instrucción, tiene lugar el punto de interrupción antes de que se procese la instrucción.
- Las funciones de los puntos de interrupción se especifican mediante los mandatos OS/400.

  Estas funciones incluyen:

  – Añadir puntos de interrupción a programas

  – Eliminar puntos de interrupción de programas

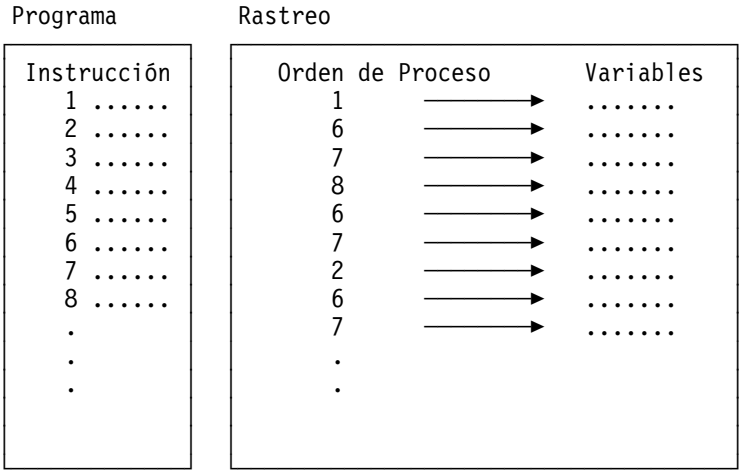
  – Visualizar información de punto de interrupción

– Reanudar la ejecución de un programa después de que se haya alcanzado (visualizado) un punto de interrupción.

Consulte la publicación *CL Guía del Programador* para la descripción de estos mandatos y para más detalles acerca de puntos de interrupción.

Utilización de un Rastreo

Un rastreo es un registro de algunas o todas las instrucciones que se ejecutan en un programa. Si se solicita, un rastreo también registra los valores de variables específicas utilizadas en las instrucciones del programa.



Un rastreo se diferencia de un punto de interrupción en que el número de instrucciones implicadas en el rastreo influye en el lugar en el que finalizará el rastreo. El sistema registra todas las instrucciones rastreadas que se han procesado. Puede solicitar una visualización de la información rastreada, que muestra la secuencia en que se han procesado las instrucciones y, si se solicita, los valores de las variables utilizadas en las instrucciones.

Especifique qué instrucción rastreará el sistema. También puede especificar que las variables se visualicen únicamente cuando su valor haya cambiado desde la ejecución de la última instrucción de rastreo.

Podrá especificar un rastreo de una instrucción en un programa, un grupo de instrucciones en un programa o todas las instrucciones de todo un programa.

Ejemplo de Utilización de un Rastreo

La Figura 22 en la página 67 muestra una parte del ejemplo de programa COBOL, TESTPRT. El mandato OS/400 siguiente añade un rastreo de las instrucciones 54 a 58 en ese programa. La variable NO-OF-DEPENDENTS sólo se registrará si su valor cambia entre las instrucciones 54 y 58:

```
ADDTRC      STMT((54 58))
            PGMVAR('NO-OF-DEPENDENTS')
            OUTVAR(*CHG)
```

Nota: Ha de introducirse STRDBG antes que la instrucción ADDTRC.

```
5763CB1 V3R0M5                   Fuente AS/400 COBOL
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
      004200*****************************************************                                 03/07/94
      004300* EL SIGUIENTE PARRAFO ABRE EL ARCHIVO DE SALIDA QUE*
      004400* HA DE CREARSE E INICIALIZA LOS CONTADORES         *
      004500*****************************************************                                 03/07/94
   42 004600 PROCEDURE DIVISION.                                                                 03/07/94
      004700 STEP-1.                                                                             03/07/94
   43 004800     OPEN OUTPUT FILE-1.                                                            03/07/94
   44 004900     MOVE ZERO TO KOUNT, NUMBR.                                                     03/07/94
      005000*****************************************************                                 03/07/94
      005100* LOS 3 PARRAFOS SIGUIENTES CREAN INTERNAMENTE LOS  *                                 03/07/94
      005200* REGISTROS QUE HA DE CONTENER EL ARCHIVO, LOS      *                                 03/07/94
      005300* GRABA EN DISCO Y LOS VISUALIZA                    *                                 03/07/94
      005400*****************************************************                                 03/07/94
      005500 STEP-2.                                                                             03/07/94
   45 005600     ADD 1 TO KOUNT, NUMBR.                                                         03/07/94
   46 005700     MOVE ALPHA  (KOUNT) TO NAME-FIELD.                                             03/07/94
   47 005800     MOVE DEPEND (KOUNT) TO NO-OF-DEPENDENTS.                                       03/07/94
   48 005900     MOVE NUMBR          TO RECORD-NO.                                              03/07/94
      006000 STEP-3.                                                                             03/07/94
   49 006100     DISPLAY WORK-RECORD.                                                           03/07/94
   50 006200     WRITE RECORD-1 FROM WORK-RECORD.                                               03/07/94
      006300 STEP-4.                                                                             03/07/94
   51 006400     PERFORM STEP-2 THRU STEP-3 UNTIL KOUNT IS EQUAL TO 26.
      006500*****************************************************
      006600* EL SIGUIENTE PARRAFO CIERRA EL ARCHIVO ABIERTO    *
      006700* PARA SALIDA Y VUELVE A ABRIRLO PARA ENTRADA       *
      006800*****************************************************
      006900 STEP-5.
   52 007000     CLOSE FILE-1.
   53 007100     OPEN INPUT FILE-1.
      007200*****************************************************
      007300* LOS SIGUIENTES PARRAFOS LEEN HACIA ATRAS EL       *
      007400* ARCHIVO Y SELECCIONAN EMPLEADOS SIN SUBORDINADOS  *
      007500*****************************************************
      007600 STEP-6.
   54 007700     READ FILE-1 RECORD INTO WORK-RECORD
   55 007800       AT END GO TO STEP-8.
      007900 STEP-7.
   56 008000     IF NO-OF-DEPENDENTS IS EQUAL TO "0"
   57 008100       MOVE "Z" TO NO-OF-DEPENDENTS.
   58 008200     GO TO STEP-6.
      008300 STEP-8.
   59 008400     CLOSE FILE-1.
   60 008500     STOP RUN.
                        * * * * *  F I N  D E L  F U E N T E  * * * * *
```

Figura 22. Ejemplo de un Programa COBOL que Utiliza un Rastreo

La Figura 23 en la página 68 es un ejemplo de un listado de información rastreada. Esta información se produce mediante el mandato Visualizar Datos de Rastreo (DSPTRCDTA):

```
DSPTRCDTA OUTPUT(*PRINT) CLEAR(*YES)
```

Este mandato se explica en el manual *CL Reference*.

```
5763SS1 V3R0M5                        Visualizar Datos de Rastreo
Trabajo. . :   DSP02        Usuario. . :   PGMRS        Número . . . . :   004122
                Sentencia/
Programa        Instrucción           Nivel de Recursividad   Número de secuencia
TESTPRT         54                           1                           1
 Posición inicial  . . . . . . . . . . . : 1
 Longitud  . . . . . . . . . . . . . . . : *DCL
 Formato . . . . . . . . . . . . . . . . : *CHAR
 Variable  . . . . . . . . . . . . . . . : 05  NO-OF-DEPENDENTS
   Tipo  . . . . . . . . . . . . . . . . :   CHARACTER
   Longitud  . . . . . . . . . . . . . . :   2
   *...+....1....+....2....+....3....+....4....+....5
  '0 '
                Sentencia/
Programa        Instrucción           Nivel de Recursividad   Número de Secuencia
TESTPRT         56                           1                           2
TESTPRT         57                           1                           3
TESTPRT         58                           1                           4
 Posición inicial  . . . . . . . . . . . : 1
 Longitud  . . . . . . . . . . . . . . . : *DCL
 Formato . . . . . . . . . . . . . . . . : *CHAR
*Variable  . . . . . . . . . . . . . . . : 05  NO-OF-DEPENDENTS
   Tipo  . . . . . . . . . . . . . . . . :   CHARACTER
   Longitud  . . . . . . . . . . . . . . :   2
   *...+....1....+....2....+....3....+....4....+....5
  'Z '
                Sentencia/
Programa        Instrucción           Nivel de Recursividad   Número de Secuencia
TESTPRT         54                           1                           5
TESTPRT         56                           1                           6
 Posición inicial  . . . . . . . . . . . : 1
 Longitud  . . . . . . . . . . . . . . . : *DCL
 Formato . . . . . . . . . . . . . . . . : *CHAR
*Variable  . . . . . . . . . . . . . . . : 05  NO-OF-DEPENDENTS
   Tipo  . . . . . . . . . . . . . . . . :   CHARACTER
   Longitud  . . . . . . . . . . . . . . :   2
   *...+....1....+....2....+....3....+....4....+....5
  '1 '
                Sentencia/
Programa        Instrucción           Nivel de Recursividad   Número de Secuencia
TESTPRT         58                           1                           7
TESTPRT         54                           1                           8
TESTPRT         56                           1                           9
 Posición inicial  . . . . . . . . . . . : 1
 Longitud  . . . . . . . . . . . . . . . : *DCL
 Formato . . . . . . . . . . . . . . . . : *CHAR
*Variable  . . . . . . . . . . . . . . . : 05  NO-OF-DEPENDENTS
   Tipo  . . . . . . . . . . . . . . . . :   CHARACTER
   Longitud  . . . . . . . . . . . . . . :   2
   *...+....1....+....2....+....3....+....4....+....5
  '2 '
                Sentencia/
Programa        Instrucción           Nivel de Recursividad   Número de Secuencia
TESTPRT         58                           1                          10
TESTPRT         54                           1                          11
TESTPRT         56                           1                          12
```

Figura 23. Listado de Visualizar Datos de Rastreo

Consideraciones para la Utilización de un Rastreo

Antes de utilizar los rastreos debe tener un conocimiento de las siguientes características de los rastreos:

- Las instrucciones eludidas, por ejemplo la instrucción GO TO, no se incluyen en el rastreo.

- Las funciones de rastreo se especifican mediante los mandatos OS/400 en el trabajo que contiene el programa rastreado. Estas funciones incluyen la adición de peticiones de rastreo en un programa, eliminación de peticiones de rastreo de un programa, eliminación de datos recogidos de rastreos anteriores, visualización de información de rastreos y visualización de los rastreos que se han especificado para el programa.
- Además de los números de instrucción, en el campo de salida STMT del rastreo puede aparecer el nombre de rutinas generadas por COBOL.

Consulte la publicación *CL Guía del Programador* para más información acerca de los rastreos.

Utilización de un Conmutador para Depuración en Tiempo de Ejecución

Se proporciona un conmutador en tiempo de ejecución para el recurso de depuración COBOL. Este conmutador activa el código de depuración generado cuando se especifica WITH DEBUGGING MODE. Cuando el conmutador está activado, se activan todas las secciones de depuración compiladas; cuando está desactivado (estado por omisión) se desactivan los procedimientos declarativos USE FOR DEBUGGING. Consulte el Apéndice B, “Características de Depuración” en la página 329 para más información acerca de las características de depuración COBOL y la utilización del conmutador en tiempo de ejecución.

Utilización de un Vuelco con Formato COBOL

Algunos mensajes en tiempo de ejecución COBOL le permiten obtener una opción de vuelco con formato COBOL seleccionando D o F. El vuelco con formato (selección D) incluye la información actual acerca de los archivos del programa, los contenidos de los campos, las estructuras de los datos, matrices y tablas para las variables de datos COBOL definidas por el usuario.

Si elige la opción F, el vuelco también incluye una lista de campos generados por el compilador y sus contenidos.

Tanto la opción D como la opción F volcarán los primeros 256 caracteres de las variables del programa. Cualquier variable mayor que 256 caracteres se truncará.

Si no quiere un vuelco, especifique C (cancelar *sin* vuelco). La respuesta C también es la respuesta por omisión para todos los mensajes de consulta COBOL que permiten un vuelco.

Para más información acerca de las modalidades de respuesta, consulte “Respuesta a Mensajes de Consulta en Tiempo de Ejecución” en la página 54.

La salida para el vuelco se envía al archivo de impresora suministrado por IBM, QPPGMDMP.

Consulte el Apéndice H, “Ejemplo de Vuelco con Formato COBOL” en la página 393 para ver un ejemplo de un vuelco con formato.

Capítulo 6. Manejo de Errores y Excepciones COBOL/400

Este capítulo describe el manejo de errores del COBOL/400 y su utilización. También define la relación entre el manejo de errores y el proceso de verbos de E/S.

El compilador COBOL/400 proporciona dos métodos para el manejo de errores: estándar y no estándar. El manejo de errores estándar no está disponible en los compiladores con un release anterior a la Versión 1 Release 3.

Manejo de Errores Estándar

El manejo de errores estándar ofrece compatibilidad extra con otros compiladores IBM COBOL (tales como VS COBOL II), además de otros compiladores que no son IBM COBOL. Es muy útil durante el proceso de instrucciones de E/S, puesto que captura errores graves, que de lo contrario, pasarían inadvertidos.

Una característica muy importante del manejo de errores estándar es la emisión de un mensaje en tiempo de ejecución cuando se produce un error durante el proceso de una instrucción de E/S si no hay ninguna frase AT END/INVALID KEY, ningún procedimiento USE ni ninguna cláusula FILE STATUS en la instrucción SELECT para el archivo.

¡Sensibilidad del Release!

El manejo de errores estándar se introdujo en la Versión 1 Release 3 como una opción *por omisión*. Para obtener el manejo de errores que se utilizó en releases anteriores, especifique *NOSTDERR como opción de generación del mandato CRTCBLPGM, o NOSTDERR en la instrucción PROCESS.

Visión General del Manejo de Errores

Cuando se ejecuta un programa COBOL, pueden producirse varios tipos de error. La instrucción COBOL que está activa en el momento en que se produce un error provoca que se ejecuten ciertas cláusulas o frases COBOL.

Durante las operaciones aritméticas, los errores típicos son errores de tamaño (MCH1210) y errores de datos decimales (MCH1202); la frase correspondiente para el manejo de errores es la frase SIZE ERROR.

COBOL no detecta directamente la mayoría de errores MCH; los detecta el sistema operativo y dan como resultado los mensajes del sistema. COBOL supervisa estos mensajes, estableciendo bits internos que determinan si se ha de ejecutar una instrucción imperativa SIZE ERROR o emitir un mensaje en tiempo de ejecución (LBE7200) para finalizar el programa.

COBOL detecta errores que se producen mediante la división por cero durante una operación aritmética. Si el COBOL los detecta, estos errores provocan que se ejecute la instrucción imperativa SIZE ERROR.

El mensaje del sistema MCH1210 se produce cuando se desplaza un campo numérico a un receptor que es demasiado pequeño. COBOL supervisa este error

y también da como resultado la ejecución de la instrucción imperativa SIZE ERROR.

LBE7200 es un mensaje en tiempo de ejecución que se emite generalmente cuando se produce un error grave no supervisado en el programa COBOL. También puede emitirse bajo *NOSTDERR, cuando se produce un error en ausencia de un manejador de errores apropiado.

El mensaje del sistema MCH1202 es un ejemplo típico de un error grave no supervisado. Este tipo de error produce como resultado el mensaje en tiempo de ejecución COBOL LBE7200 (o LBE7204 si el error se produce en un programa llamado por un programa COBOL). Los mensajes del sistema MCH3601 y MCH0601 son otros ejemplos de errores graves no supervisados.

Para operaciones de E/S, hay varias frases y cláusulas importantes para el manejo de errores. Son las frases AT END/INVALID KEY y NO DATA (codificadas a nivel de la instrucción COBOL), el procedimiento USE y la cláusula FILE STATUS (codificada a nivel de archivo). Durante las operaciones aritméticas y de E/S, el sistema detecta los errores y envía mensajes; es entonces cuando COBOL supervisa los mensajes. Al igual que sucede con un error resultante de una división por cero, COBOL detecta algunos errores durante una operación de E/S. Sin tener en cuenta cómo se detecta un error durante una operación de E/S, el resultado siempre será un estado de archivo interno en el que el primer carácter no será cero, será un mensaje en tiempo de ejecución o ambas cosas.

Interfaz de programación de uso general

Utilización de las Interfaces del Programa de Aplicación (API) para el Manejo de Errores

Es posible utilizar las API COBOL/400 para controlar el manejo de errores dentro de los programas. Estas API son Recuperar Manejador de Errores COBOL (QLRRTVCE) y Establecer Manejador de Errores COBOL (QLRSETCE).

La API Recuperar Manejador de Errores COBOL (QLRRTVCE) permite recuperar el nombre del programa actual o pendiente de manejo de errores COBOL. Se puede llamar mediante cualquier lenguaje de programación.

La API Establecer Manejador de Errores COBOL (QLRSETCE) permite especificar la identidad de un programa de manejo de errores COBOL. Se puede llamar mediante cualquier lenguaje de programación.

También es posible utilizar la API Cambiar Programa Principal COBOL (QLRCHGCM) para crear unidades de ejecución múltiples que tengan su propio manejador de error.

Para obtener más información sobre estas API, consulte el manual *System Programmer's Interface Reference*.

Fin de Interfaz de programación de uso general

Estado de archivo Interno y Externo

Debe proporcionar un entrada FILE-CONTROL para especificar la organización y el método de acceso para cada archivo que utiliza el programa COBOL. También puede codificar en esta entrada una cláusula FILE STATUS.

La cláusula FILE STATUS designa uno o dos ítems de datos (codificados en la sección WORKING-STORAGE) para mantener una copia del resultado de una operación de E/S. La copia del primero de estos ítems se denomina estado de archivo externo. Si utiliza el archivo TRANSACTION, se dispone de otro registro del resultado llamado código de retorno externo, que está compuesto por los códigos de retorno principal y secundario externos.

COBOL guarda sus propias copias de estos dos ítems de datos, que se almacenan en el Bloque de Información de Archivos (FIB) COBOL. En este capítulo, el *estado de archivo* y *códigos de retorno* (*principal*/*secundario*) hacen referencia a copias de COBOL a no ser que se especifique lo contrario.

Durante el proceso de una instrucción de E/S, el estado de archivo puede actualizarse en una de las tres maneras que se describen a continuación. El contenido del estado del archivo determina los procedimientos del manejo que errores que deben ejecutarse.

Los procedimientos de manejo de errores toman el control después de una operación de entrada o salida no satisfactoria, que se indica mediante cualquier estado del archivo en el que el primer carácter no sea cero. Antes de que se ejecute cualquiera de estos procedimientos, el estado del archivo se copia en el estado del archivo externo.

El estado del archivo se establece mediante una de las tres maneras siguientes:

- Método A (todos los archivos):

  COBOL comprueba el contenido de variables en los bloques de control de archivos. Si el contenido no es el esperado, se establece un estado de archivo que no sea cero. La mayoría de estados de archivo que se establecen de esta forma son el resultado de la comprobación del Bloque de Información de Archivos (FIB) y del Bloque de Control de Archivos (UFCB).

- Método B (archivos de transacción):

  COBOL comprueba los códigos de retorno principal y secundario del sistema. Si el código de retorno principal no es cero, el código de retorno (compuesto por códigos de retorno principal y secundario) se transforma en estado de archivo. Si el código de retorno principal es cero, el Método A o C puede establecer el estado de archivo.

  Observe que para las operaciones READ, WRITE y REWRITE de subarchivo, sólo se aplican los Métodos A y C.

  Para obtener una lista de códigos de retorno y sus correspondientes estado de archivo, consulte el apartado “Resumen del soporte de estructuras de archivos y valores claves de estado” en la publicación *COBOL/400 Reference*.

- Método C (todos los archivos):

  El sistema envía un mensaje cuando el COBOL llama la gestión de datos para realizar una operación de E/S. COBOL supervisa estos mensajes y establece un estado de archivo concordante.

COBOL supervisa específicamente un mensaje generando supervisores de mensajes en el programa objeto producido en tiempo de compilación. La generación del supervisor de mensajes se basa en los tipos de archivos (como por ejemplo, tipo de organización y tipo de acceso) que especifique en un programa. Así pues, un mensaje que se supervisa específicamente en un programa puede caer bajo el manejador de E/S genérico que está en otro programa. En este capítulo hay más información sobre la generación del supervisor de mensajes.

COBOL supervisa la mayoría de mensajes que el sistema envía en respuesta a una operación de E/S. Las excepciones típicas de E/S dan como resultado mensajes CPF que empiezan por “CPF4” o “CPF5”, y COBOL realiza la supervisión específica para éstas.

Para obtener una lista de mensajes para los que el COBOL efectúa la supervisión específica, consulte el apartado “Resumen de Soporte de la Estructura de Archivos y Valores Clave del Estado” en la publicación *COBOL/400 Reference*.

Detección de Errores Generales

Estado del Archivo

001

Inicio de la operación de E/S
– Método A: Compruebe el contenido de las variables de los bloques de control de archivos.
(Por ejemplo, compruebe que se ha abierto el archivo adecuadamente y en la modalidad apropiada.)

¿Se ha establecido el estado de archivo interno diferente de 00?
Sí **No**

002

Llame la gestión de datos para realizar la operación de E/S
– Método C: Supervise los mensajes enviados por la gestión de datos y establezca el estado de archivo interno de acuerdo a los mismos.
– Método A: Compruebe los bloques de información del sistema y establezca el estado de archivo interno si no lo hizo utilizando el Método C.

¿Es un archivo de transacciones?
Sí **No**

003

Establezca el estado de archivo externo
– Transfiera el estado de archivo interno al estado de archivo externo (especificado en la cláusula de estado de archivo). Basándose en el estado de archivo interno, ejecute el código para el manejo de errores.

004

Compruebe los códigos de retorno principal y secundario del sistema
– Método B: Si la gestión de datos ha enviado un mensaje, convierta los códigos de retorno principal y secundario asociados al mensaje del sistema en estado de archivo interno.

Continúe en el Paso 003

005

Continúe en el Paso 003

Generación del Supervisor de Mensajes

Un supervisor de mensajes proporciona a un programa una forma de manejar mensajes enviados por el sistema o por otro programa. Un supervisor de mensajes puede manejar uno o más mensajes.

En algunos aspectos, un supervisor de mensajes se parece a un procedimiento USE. Un supervisor de mensajes especifica una acción que se ha de llevar a cabo cuando se produce un error durante el proceso de una instrucción de la interfaz (MI) de la misma forma que un procedimiento USE especifica las acciones que ha que llevar a cabo en respuesta a un error de E/S. Observe que el mensaje del sistema señala el error de instrucción MI y que la instrucción COBOL está compuesta por una o más instrucciones MI.

A diferencia de un procedimiento USE (que puede no estar activo durante un programa completo), un supervisor de mensajes COBOL se activa tan pronto como se inicia el programa. Los supervisores de mensajes establecen los estados de archivo e indican las condiciones SIZE ERROR, END-OF-PAGE y OVERFLOW.

Los supervisores de mensajes que genera el COBOL se agrupan en varios conjuntos, generados bajo ciertas condiciones dentro del programa COBOL. La siguiente tabla proporciona las directrices generales concernientes a la generación de supervisores de mensajes:

Tabla 1 (Página 1 de 3). Generación de supervisores de mensajes

Causa del supervisor de mensajes	**Miembros de ejemplo del conjunto de mensajes supervisados**
Codifique una cláusula de estado de archivo	• No se encontró archivo, estado archivo externo 35 • Condición de error permanente, estado archivo externo 30 • Modalidad OPEN no válida, estado archivo externo 37 • No hay registro siguiente, mensaje del sistema CPF5183 (parte del estado de archivo externo 46) • Tipo de acceso no definido o no autorizado, estado archivo externo 91 • Error lógico, estado archivo externo 92 (excepto para mensajes del sistema CPF4740 y CPF5070) • Registro bloqueado, estado archivo externo 9D • OPEN con control de compromiso anómalo, estado archivo externo 9P • WRITE no válido, mensajes del sistema CPF5018 y CPF5272 (parte del estado de archivo externo 24).
Codifique una frase AT END	• Manejador fin de archivo, mensajes del sistema CPF5001 y CPF5025 • No se encontró archivo, estado archivo externo 35

Tabla 1 (Página 2 de 3). Generación de supervisores de mensajes	
Causa del supervisor de mensajes	**Miembros de ejemplo del conjunto de mensajes supervisados**
Especifique un subarchivo en el programa	• Último registro grabado en el subarchivo, estado archivo externo 9M • No se encontró registro de subarchivo, mensaje del sistema CPF5020 (parte del estado archivo externo 23) • Violación del límite de subarchivo, mensajes del sistema CPF5021 y CPF5043 (parte del estado de archivo externo 24). Una **violación del límite** es un intento de escribir fuera de los límites de un archivo secuencial.
Codifique una instrucción READ de subarchivo con la frase NEXT MODIFIED	• No hay registros de subarchivo modificados, estado archivo externo 12.
Utilice un archivo secuencial indexado	• No hay supervisor específico (Método A), establezca estados de archivo interno 21 y 22.
Hay una operación READ con clave	• Mensajes del sistema CPF5006 y CPF5013 (parte de estado archivo externo 23).
Hay una operación WRITE secuencial	• Violación de límite, mensaje del sistema CPF5116 (parte de estado archivo externo 34).
Hay una operación REWRITE secuencial indexada	• No hay supervisor específico (método A), establezca los estados de archivo interno 21, 43, 44 y 9S.
Hay TRANSACTION I/O	• Tiempo de espera READ, mensaje del sistema CPF4743, establezca estado archivo externo 00 • No hay datos durante el READ, mensaje del sistema CPF4742, establezca bit NO DATA • No hay dispositivos adquiridos, mensaje del sistema CPF5070 (parte de estado archivo externo 92) • No hay dispositivos invitados/adquiridos, mensaje del sistema CPF4740 (parte de estado archivo externo 92 y estado archivo externo 10) • Cancelar trabajo, estado archivo externo 9A • WRITE falló, estado archivo externo 9I • Error temporal, estado archivo externo 9N.
Especifique una cláusula de formato en una instrucción de E/S	• Nombre de formato no válido/no encontrado, estado archivo interno 9K.

Tabla 1 (Página 3 de 3). Generación de supervisores de mensajes	
Causa del supervisor de mensajes	**Miembros de ejemplo del conjunto de mensajes supervisados**
No hay ninguna E/S (incluyendo las operaciones ACCEPT/DISPLAY ampliadas) en el programa.	• Manejador de excepciones END-OF-PAGE (mensaje del sistema CPF5004) • Error de comprobación de nivel, estado archivo externo 39 • Manejador de excepciones genérico, estado archivo externo 90 • Discrepancia del indicador (mensaje en tiempo de ejecución LBE7421, mensaje del sistema CPF4238) • Hacer caso omiso de COMMIT o ROLLBACK (mensaje del sistema CPF8350). • Clave duplicada, estado archivo externo 22. • Cambio no válido de dirección READ DYNAMIC, estado de archivo interno 9U, mensaje del sistema CPF5184.
Nota: Para obtener una lista de los mensajes supervisados que se desactivan bajo un estado de archivo determinado, consulte el apartado “Resumen de soporte de estructura de archivos y valores clave del estado” de la publicación *COBOL/400 Reference*.	

Finalización de un Programa COBOL

Hay tres elementos que pueden provocar la finalización de un programa COBOL:

Una instrucción COBOL (EXIT PROGRAM, STOP RUN o GOBACK)

Una respuesta a un mensaje de consulta

Una instrucción STOP RUN o EXIT PROGRAM implícita.

Se supone una instrucción STOP RUN cuando un programa principal COBOL carece de una instrucción ejecutable (EXIT PROGRAM implícito para un subprograma COBOL), es decir, cuando el proceso se desactiva al llegar a la última instrucción un programa.

Los mensajes de consulta pueden emitirse en respuesta a una instrucción COBOL (un literal STOP), pero se emiten generalmente cuando se produce un error grave en un programa, o cuando una operación COBOL no finaliza satisfactoriamente. (Por ejemplo LBE7205, LBE7207 y LBE7208).

Hay cuatro respuestas comunes a un mensaje de consulta COBOL: C, D, F y G (cancelar, cancelar y volcar, cancelar y vuelco total, continuar). Los tres primeros originan (como pasos finales) un STOP RUN implícito seguido de un mensaje de escape LBE9001. LBE9001 indica que el programa finaliza debido a un mensaje.

Una instrucción STOP RUN implícita o explícita, o una instrucción GOBACK que aparezca en un programa principal, provoca la finalización de toda la unidad de ejecución COBOL. Si se emite un mensaje de escape (LBE9001) como paso final de una unidad de ejecución, el llamador del primer programa COBOL puede supervisarlo. (Esto es debido a que el primer programa COBOL que se ha de llamar se convierte en el programa principal).

Si una unidad de ejecución COBOL está compuesta por varios programas COBOL y no COBOL, el programa principal COBOL puede emitir el mensaje de escape. Así pues, cualquier programa no COBOL que se llame después del programa principal no puede supervisar el mensaje de escape.

Códigos de Retorno

Cuando se especifica un archivo TRANSACTION en el programa, la cláusula FILE STATUS de la instrucción SELECT puede contener dos nombres de datos: el estado de archivo externo y el código de retorno (principal y secundario). Tal y como se describe en el apartado “Estado de archivo Interno y Externo” en la página 73, un estado de archivo puede establecerse en una de las tres formas descritas; no obstante, el sistema establece los códigos de retorno después de cualquier transacción de E/S que llame la gestión de datos. Por lo tanto, la mayoría de condiciones de error que dan como resultado un mensaje del sistema también poseen un código de retorno asociado.

Los códigos de retorno son similares a los valores del estado de archivo. Es decir, los mensajes CPF que envía el sistema se agrupan juntos bajo supervisores de mensaje, y cada supervisor de mensajes establece uno o más estados de archivo.

De forma similar, los mensajes CPF se agrupan juntos, y cada grupo de mensajes genera el mismo código de retorno principal. (El código de retorno secundario no tiene que ser el mismo).

La principal diferencia entre los estados de archivo y los códigos de retorno es que la agrupación de mensajes CPF es distinta.

Aunque COBOL sólo establece códigos de retorno para archivos TRANSACTION, otros tipos de archivos (como por ejemplo los archivos de impresora) también los establecen. Se puede acceder a los códigos de retorno para estos archivos mediante una operación ACCEPT desde I-O-FEEDBACK.

Modelos Estándar y no Estándar de Manejo de Errores

La figuras 24 y 25 muestran dos modelos de manejo de errores diferentes.

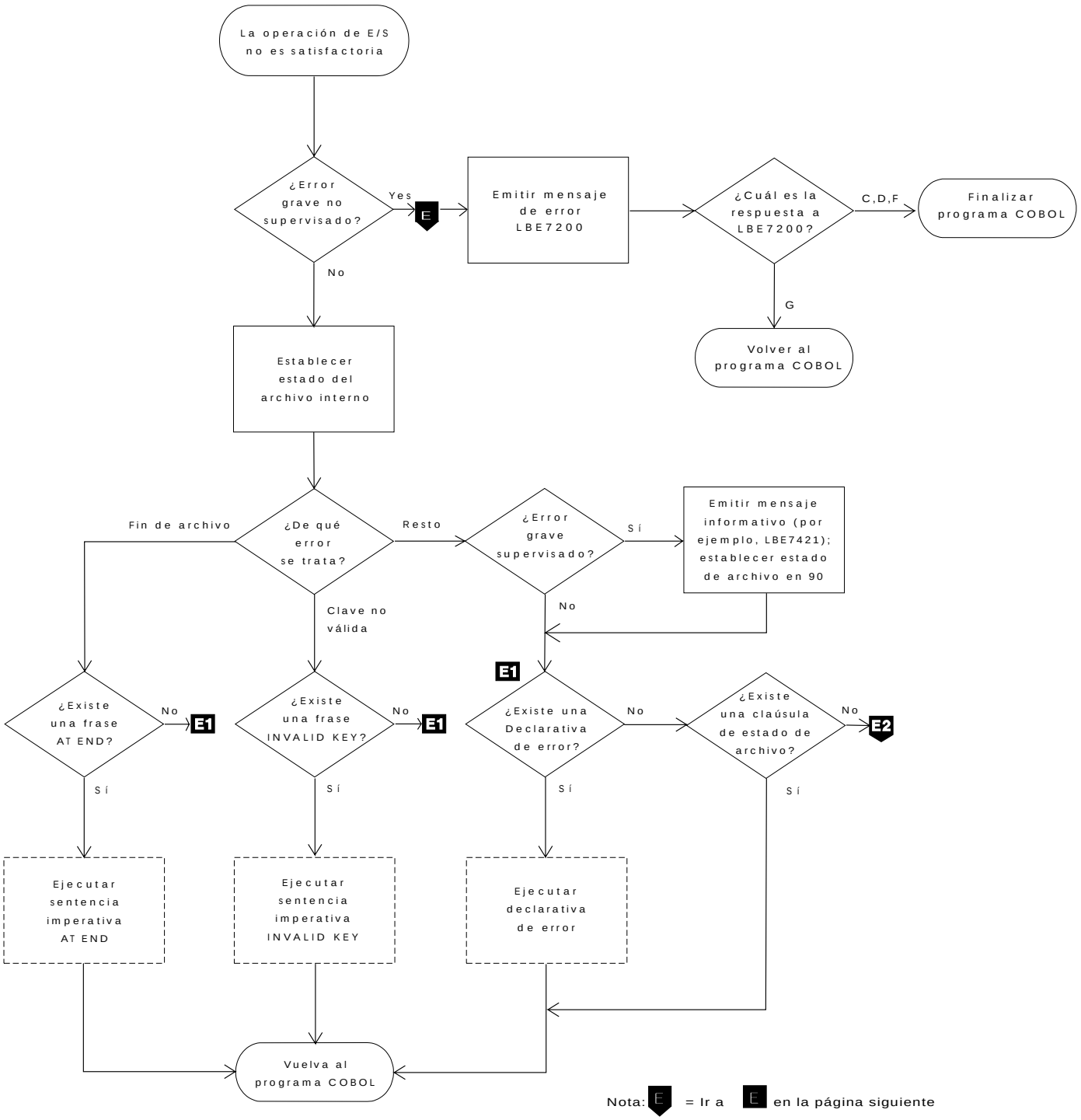


Figura 24 (Parte 1 de 2). Manejo de Error Estándar (por Omisión)

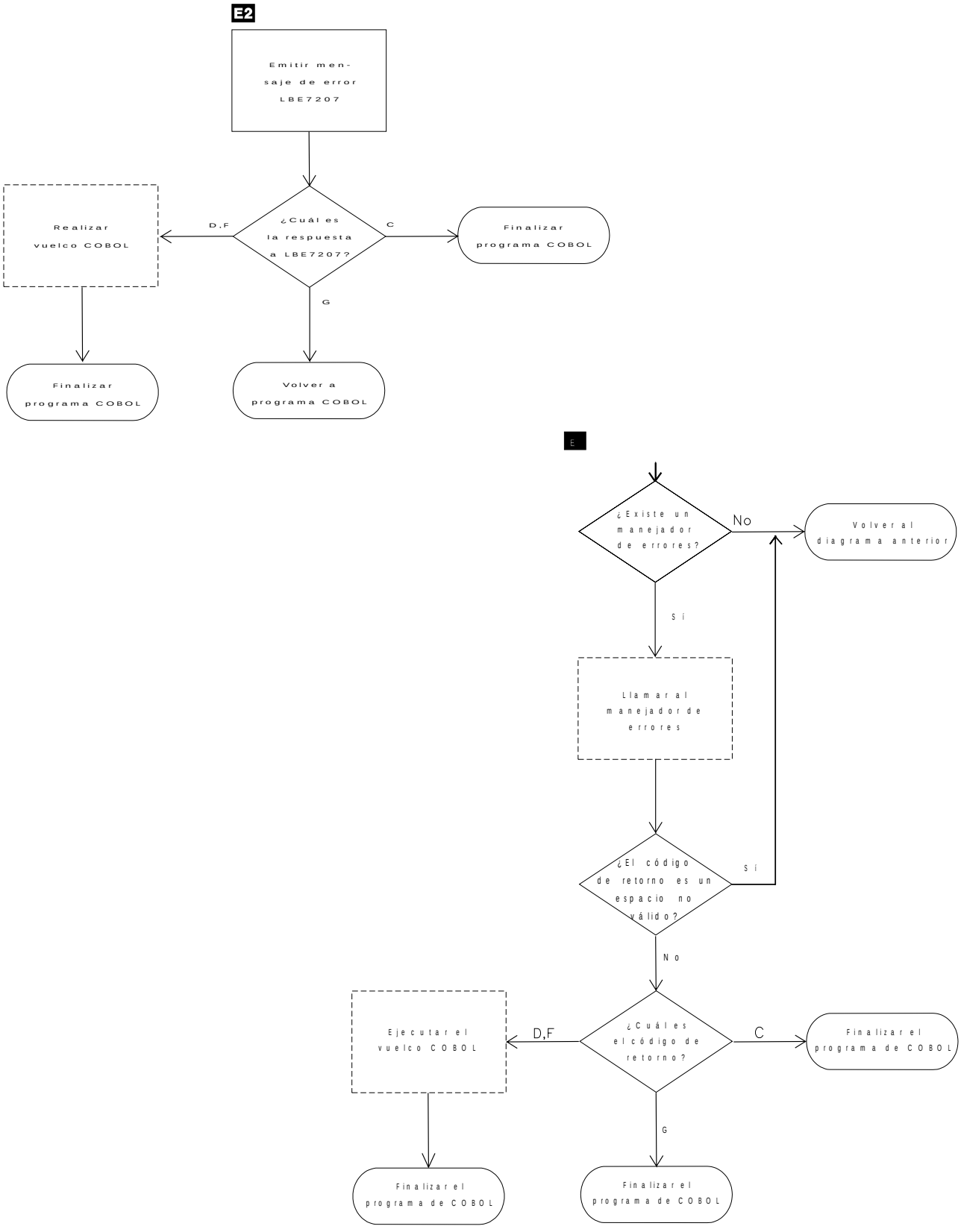


Figura 24 (Parte 2 de 2). Manejo de Error Estándar (por Omisión)

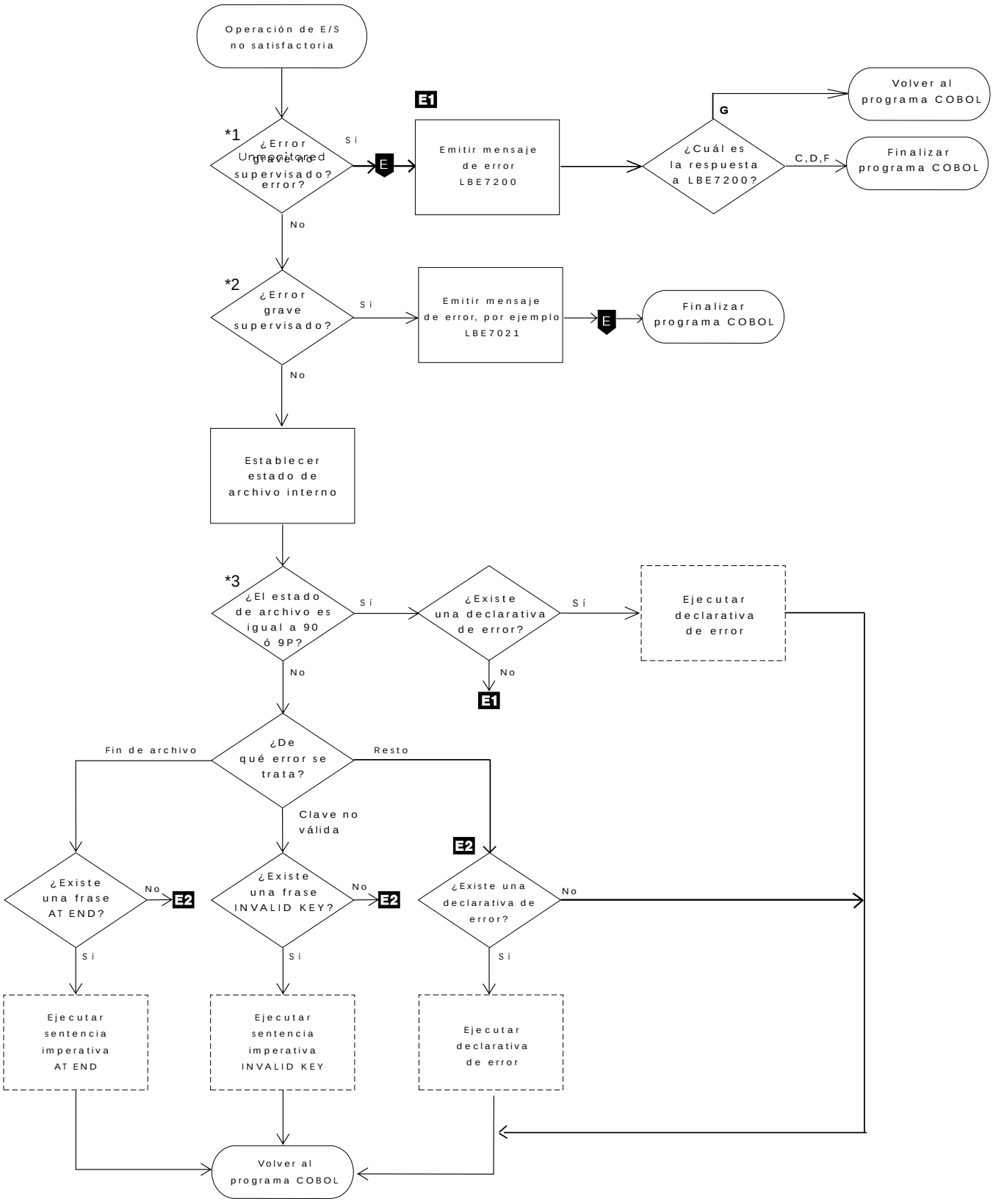


*Figura 25. Manejo de error no estándar (disponible por la opción *NOSTDERR)*

Pueden producirse otras excepciones de E/S que COBOL no tenga previstas. Estas excepciones también dan como resultado los mensajes CPF4xxx y CPF5xxx, pero no hay ninguna supervisión específica para ellos. Por el contrario, son detectados por medio de un manejador de error de E/S genérico. Este manejador de error supervisa algunos rangos de los mensajes CPF4xxx y CPF5xxx; establece el estado de archivo 90 y sigue a la rama *Sí* desde la posición *3 en la Figura 25 en la página 82.

Puede producirse algún excepción de E/S que se supervisa específicamente y que, según el modelo de manejo de errores no estándar, es suficientemente grave como para detener el programa. En esta situación no se establece ningún estado de archivo.

Estas excepciones de E/S dan como resultado mensajes de escape COBOL específicos seguidos por la finalización del programa siguen a la rama *Sí* desde la posición *2 en la Figura 25.

Ejemplo: CPF4238 - discrepancia INDARA entre el programa y el archivo

Hay supervisión específica para este mensaje, y el resultado es el mensaje de error LBE7021 seguido por una finalización del programa.

Otros mensajes COBOL que entran en esta categoría son LBE7020 y LBE7022.

Durante una operación de E/S, puede producirse un problema que el sistema no esperaba. Estos problemas generalmente dan como resultado varios mensajes (como por ejemplo los que comienzan por “MCH”) que se desactivan fuera del rango CPF4xxx y CPF5xxx. Estos errores, conocidos como errores graves no supervisados, siguen a la rama *Sí* desde la posición *1 en la Figura 25. Estos errores se manejan mediante un supervisor de mensajes multiuso y dan como resultado una finalización del programa COBOL. No se establece ningún estado de archivo

Efectos de *STDERR y *NOSTDERR en el Estado de Archivos

- Efectos de los mensajes LBE742x y LBE702x:

  Con *STDERR, el estado de archivo 90 se establece siguiendo la emisión de mensajes LBE742x. Entonces, el programa continúa si hay un procedimiento USE o una cláusula FILE STATUS.

  Con *NOSTDERR, los mensajes LBE702x provocan que el programa finalice sin establecer un estado de archivo.

- Finalización de un programa debido al estado archivo 9P o 90:

  Con *STDERR, un estado de archivo 9P o 90 que surge de un error de E/S (señalado mediante los mensajes CPF4xxx y CPF5xxx) no hace que el programa finalice mientras haya un procedimiento USE o una cláusula FILE STATUS. Si no existe ninguno, se emite un mensaje de error LBE7207.

  Con *NOSTDERR, un estado de archivo 9P ó 90 en la ausencia de un procedimiento USE provoca la emisión del mensaje de error LBE7200 y la finalización del programa.

- Emisión de un mensaje de error para cualquier estado de archivo cuyo primer carácter no sea cero cuando no hay manejador de error ni la cláusula FILE STATUS:

Con *STDERR, cualquier estado de archivo cuyo primer carácter no sea cero cuando no hay la frase AT END/INVALID KEY, el procedimiento USE, ni la cláusula FILE STATUS, origina la emisión de un mensaje de consulta LBE7207 (con las opciones de respuesta C, D, F y G).

Con *NOSTDERR, cualquier estado de archivo cuyo primer carácter no sea cero cuando no hay ninguna frase AT END/INVALID ni procedimiento USE permite al programa continuar a no ser que ya haya finalizado.

Proceso de Verbos de E/S

El siguiente diagrama muestra cuándo se ejecutan el procedimiento USE y las instrucciones imperativas (NOT) AT END, (NOT) INVALID KEY y NO DATA. Este proceso se mantiene desde la Versión 1 Release 3 y es *independiente* del método de manejo de errores que se elija (*STDERR o *NOSTDERR).

Observe que el estado de archivo que se muestra aquí hace referencia al estado de archivo interno.

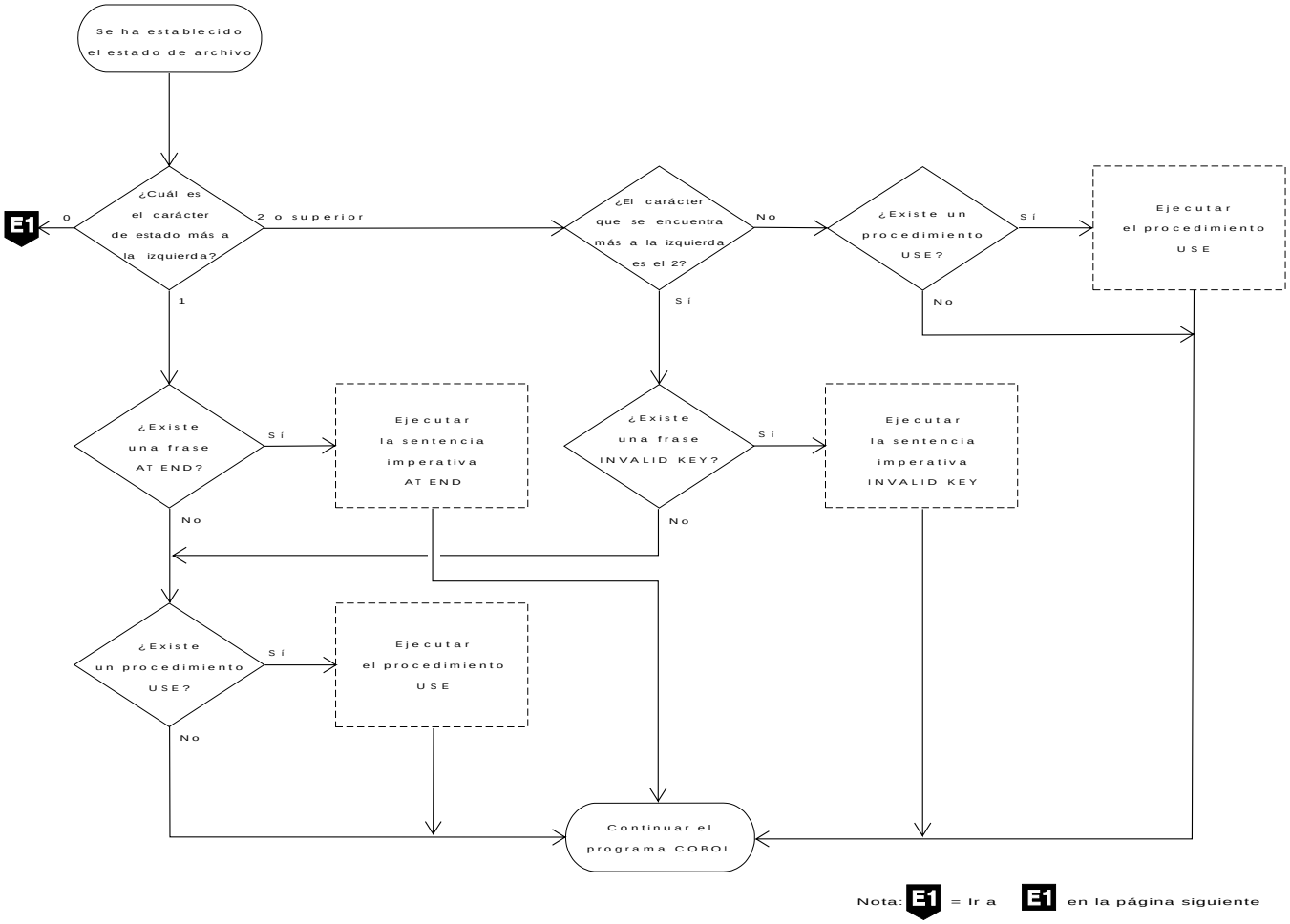


Figura 26 (Parte 1 de 2). Proceso de Verbos de E/S

Figura 26 (Parte 2 de 2). Proceso de Verbos de E/S

Nota: Siga las partes del diagrama que se apliquen en sus instrucciones.

Excepciones Comunes y Algunas de sus Causas

Error de datos decimales MCH1202:

- Se ha utiliza un ítem numérico elemental como un fuente cuando se han almacenado anteriormente datos no válidos en él. El ítem debe tener una cláusula VALUE, o debe utilizarse una instrucción MOVE para inicializar su valor.
- Se ha realizado un intento para colocar los datos no numéricos en un ítem numérico.
- Se han grabado datos defectuosos a un subarchivo al principio del programa. Los datos del subarchivo no se validarán hasta que no aparezcan en pantalla. Es por eso que se puede producir el error 1202 en la WRITE de un registro de control del subarchivo, pero en realidad los datos defectuosos se han colocado antes en el subarchivo.

Excepciones del puntero MCH0601:

- Parte de un ítem de sección de enlace fuera del espacio asignado.

Por ejemplo, si se establece la dirección de ítem de sección de enlace y uno o más de estos ítems de datos elementales se extiende fuera del espacio con una MOVE al ítem de datos elementales, se emite MCH0601.

Para obtener más información sobre cómo utilizar los punteros, consulte el apartado “Utilización de Punteros en un Programa COBOL/400” en la página 295.

Alineación del puntero MCH0602:

- La alineación del puntero de la Sección del almacenamiento de trabajo del programa de llamada no coincide con la alineación de la Sección de enlace del programa llamado. La alineación tiene como límite 16 bytes.

  Para obtener más información sobre cómo utilizar los punteros, consulte la “Utilización de Punteros en un Programa COBOL/400” en la página 295.

Error del puntero MCH3601:

- Se realiza una referencia a un registro o un campo dentro de un registro y se ha cerrado el archivo asociado o nunca se ha abierto.

  Por ejemplo, OPEN para el archivo no ha sido satisfactorio y se ha intentado efectuar el proceso de cualquier instrucción de E/S para ese archivo. El estado de archivo debe comprobarse antes de que se intente cualquier otra E/S.

Fin de solicitudes CPF2415:

- Se ha intentado acceder a la entrada desde la corriente de entrada de trabajos mientras el sistema se ejecutaba en modalidad por lotes y no hay entrada disponible.

Recuperación de una Anomalía

Recuperación con control de compromiso

Cuando se vuelve a arrancar el sistema después de una anomalía, los archivos bajo el control de compromiso se restauran automáticamente a su estado en el último límite de compromiso. Para obtener información adicional acerca del control de compromiso, consulte el apartado “Consideraciones sobre Control de Compromiso” en la página 99.

En una anomalía de trabajo (debido a un error del usuario o del sistema) los archivo bajo control de compromiso se restauran como parte de la finalización de trabajo al estado de archivo del límite de compromiso anterior.

Debido a que los archivos bajo control de compromiso se retrotraen después de una anomalía del sistema o del proceso, esta característica puede utilizarse como ayuda en la reiniciación del sistema. Se puede crear un registro separado para almacenar datos que pueden utilizarse si fuera necesario al reiniciar un trabajo. Estos datos de reinicio pueden incluir ítems como totales, contadores, valores de claves de registro, valores de claves relativos y otras informaciones importantes del proceso de una aplicación.

Si mantiene los datos de reinicio mencionados anteriormente en un archivo bajo control de compromiso, los datos de reinicio también se almacenarán permanen-

temente en la base de datos cuando se emite una instrucción COMMIT. Cuando se produce una ROLLBACK después de una anomalía de trabajo o de proceso, puede recuperar un registro de la ampliación de proceso realizado satisfactoriamente antes de la anomalía. Observe que el método anterior sólo es una técnica de programación sugerida, y no siempre será lo idóneo, dependiendo de la aplicación.

Recuperación del Archivo TRANSACTION

En algunos casos, se pueden recuperar errores de E/S en archivos TRANSACTION sin la intervención del operador, o sin desactivar/activar las estaciones de trabajo o los dispositivos de comunicaciones.

En errores de E/S potencialmente recuperables en archivos TRANSACTION, el sistema inicia la acción además de los pasos que han de tomarse en el programa de aplicación para intentar la recuperación del error. Para obtener más información sobres las acciones que realiza el sistema, consulte la publicación *Remote Work Station Guide*.

Examinando el estado de archivo después de un operación de E/S, el programa de aplicación puede determinar si es posible efectuar una recuperación de un error de E/S en el archivo TRANSACTION. Si la Clave de estado de archivo tiene un valor 9N, el programa de aplicación podrá recuperar el error de E/S. Un procedimiento de recuperación debe codificarse como parte del programa de aplicación; además, varía dependiendo de si el archivo TRANSACTION ha adquirido un dispositivo único o de si los dispositivos múltiples estaban conectados.

Para un archivo con un dispositivo adquirido:

1. Cierre el archivo TRANSACTION con el error de E/S.
2. Vuelva a abrir el archivo.
3. Procese los pasos necesarios para volver a intentar la operación de E/S anómala. Esto puede implicar cierto número de pasos, dependiendo del tipo de dispositivo de programa utilizado. (Por ejemplo, si la última operación de E/S fue READ, deben repetirse una o más instrucciones WRITE que se procesaron antes que la instrucción READ.) Para obtener más información acerca de los procedimientos de recuperación, consulte la publicación *ICF Programmer's Guide*.

Para un archivo de pantalla con múltiples dispositivos adquiridos:

1. Procese DROP del dispositivo de programa que ha provocado el error de E/S en el archivo TRANSACTION.
2. Procese ACQUIRE del mismo dispositivo de programa.
3. Consulte el paso 3 descrito anteriormente.

Para un archivo ICF con múltiples dispositivos adquiridos:

1. Procese ACQUIRE del mismo dispositivo de programa.
2. Consulte el paso 3 descrito anteriormente.

Para un archivo de pantalla con múltiples dispositivos adquiridos:

Los intentos de recuperar un programa de aplicación deben intentarse normalmente sólo una vez.

Si falla el intento de recuperación:

- Si el archivo tiene conectado un solo dispositivo de programa, termine el programa mediante el proceso de la instrucción STOP RUN, EXIT PROGRAM o GOBACK e intente localizar el origen del error.
- Si el archivo tiene múltiples dispositivos de programa adquiridos, es posible que desee efectuar una de las siguientes acciones:
  - Continuar con el proceso sin el dispositivo del programa que ha provocado el error de E/S en el archivo TRANSACTION y volver a adquirir más tarde el dispositivo.
  - Terminar el programa.

Para obtener una descripción sobre los códigos de retorno principales y secundarios que pueden servir de ayuda para diagnosticar errores de E/S en el archivo TRANSACTION, consulte el manual *ICF Programmer's Guide* o la publicación *Guía para la Gestión de Datos*.

La Figura 27 en la página 89 muestra un ejemplo de un procedimiento de recuperación de errores.

IBM International Business Machines

AS/400 ESPECIFICACIONES DE DESCRIPCION DE DATOS

GX21-9891-0 UM/050*
Impreso en España
*El número de hojas por área de relleno puede variar ligeramente.

Archivo		Instrucciones de Grabación	Signo	Descripción	Página de
Programador	Fecha		Tecla		

A

Columnas: Número de Secuencia (1–5) · Tipo de Formulario (6) · And/Or/Coment. (A/O/*) (7) · Condicionamiento: No (N) (8), Nombre Condición: Indicador (9–10), No (N) (11), Indicador (12–13), No (N) (14), Indicador (15–16) · Tipo Nombre o Espec (b/R/H/J/K/S/O) (17) · Reservado (18) · Nombre (19–28) · Referencia (R) (29) · Longitud (30–34) · Tipo Datos/Desplazamiento Teclado (35) · Posiciones Decimales (36–37) · Utilización (b/O/I/B/H/M/N/P) (38) · Ubicación: Línea (39–41), Pos (42–44) · Funciones (45–80)

```
     A* ARCHIVO DE PANTALLA PARA EJEMPLO DE RECUPERACION DE ERROR
     A*
     A                                      INDARA
     A          R FORMAT1                   CF01(01 'FIN DE PROGRAMA')
     A*
     A                                 12 28'INTRODUCIR ENTRADA '
     A            INPUTFLD       5   I 12 42
     A                                 28 26'F1 = TERMINAR'
     A
     A
     A
     A
     A
     A
     A
     A
     A
     A
     A
     A
```

Figura 27. Ejemplo de un Procedimiento de Recuperación de Errores -- DDS

```
5763CB1 V3R0M5                  Fuente COBOL AS/400
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
    1  000100 IDENTIFICATION DIVISION.                                                          02/01/94
    2  000200 PROGRAM-ID.  RECOVERY.                                                            02/05/94
    3  000300 ENVIRONMENT DIVISION.                                                             02/01/94
    4  000400 CONFIGURATION SECTION.                                                            02/01/94
    5  000500 SOURCE-COMPUTER. IBM-AS400.                                                       02/02/94
    6  000600 OBJECT-COMPUTER. IBM-AS400.                                                       02/02/94
    7  000700 INPUT-OUTPUT SECTION.                                                             02/01/94
    8  000800 FILE-CONTROL.                                                                     02/01/94
    9  000900     SELECT RECOVFILE                                                              02/05/94
   10  001000         ASSIGN TO WORKSTATION-RECVFILE-SI                                         03/22/94
   11  001100         ORGANIZATION IS TRANSACTION                                               02/05/94
   12  001200         ACCESS MODE IS SEQUENTIAL                                                 02/01/94
   13  001300         FILE STATUS IS STATUS-FLD, STATUS-FLD-2                                   02/05/94
   14  001400         CONTROL-AREA IS CONTROL-FLD.                                              02/05/94
   15  001500     SELECT PRINTER-FILE                                                           02/05/94
   16  001600         ASSIGN TO PRINTER-QPRINT.                                                 02/05/94
       001700                                                                                   02/01/94
   17  001800 DATA DIVISION.                                                                    02/01/94
   18  001900 FILE SECTION.                                                                     02/01/94
   19  002000 FD  RECOVFILE                                                                     02/05/94
   20  002100     LABEL RECORDS ARE OMITTED                                                     02/05/94
   21  002200     DATA RECORD IS RECOV-REC.                                                     02/05/94
   22  002300 01  RECOV-REC.                                                                   02/05/94
   23  002400     COPY DDS-ALL-FORMATS OF RECVFILE.                                             03/22/94
   24 +000001       05  RECVFILE-RECORD PIC X(5).                                 <-ALL-FMTS
      +000002*  INPUT FORMAT:FORMAT1    FROM FILE RECVFILE   OF LIBRARY COBNATEX  <-ALL-FMTS
      +000003*                                                                    <-ALL-FMTS
   25 +000004       05  FORMAT1-I     REDEFINES RECVFILE-RECORD.                 <-ALL-FMTS
   26 +000005           06 INPUTFLD          PIC X(5).                           <-ALL-FMTS
      +000006* OUTPUT FORMAT:FORMAT1    FROM FILE RECVFILE   OF LIBRARY COBNATEX  <-ALL-FMTS
      +000007*                                                                    <-ALL-FMTS
      +000008*      05  FORMAT1-O     REDEFINES RECVFILE-RECORD.                 <-ALL-FMTS
       002500
   27  002600 FD  PRINTER-FILE.
   28  002700 01  PRINTER-REC.
   29  002800    05  PRINTER-RECORD             PIC X(132).
       002900
   30  003000 WORKING-STORAGE SECTION.
       003100
   31  003200 01  I-O-VERB                     PIC X(10).
   32  003300 01  STATUS-FLD                   PIC X(2).
   33  003400    88  NO-ERROR                    VALUE "00".
   34  003500    88  ACQUIRE-FAILED              VALUE "9H".
   35  003600    88  TEMPORARY-ERROR             VALUE "9N".
   36  003700 01  STATUS-FLD-2                 PIC X(4).
   37  003800 01  CONTROL-FLD.
   38  003900    05  FUNCTION-KEY               PIC X(2).
   39  004000    05  PGM-DEVICE-NAME            PIC X(10).
   40  004100    05  RECORD-FORMAT              PIC X(10).
   41  004200 01  END-INDICATOR                PIC 1   INDICATOR 1
   42  004300                                    VALUE B"0".
   43  004400    88  END-NOT-REQUESTED           VALUE B"0".
   44  004500    88  END-REQUESTED               VALUE B"1".
   45  004600 01  USE-PROC-FLAG                PIC 1
   46  004700                                    VALUE B"0".
   47  004800    88  USE-PROC-NOT-EXECUTED       VALUE B"0".
   48  004900    88  USE-PROC-EXECUTED           VALUE B"1".
   49  005000 01  RECOVERY-FLAG                PIC 1
   50  005100                                    VALUE B"0".
   51  005200    88  NO-RECOVERY-DONE            VALUE B"0".
   52  005300    88  RECOVERY-DONE               VALUE B"1".
   53  005400 01  HEADER-LINE.
   54  005500    05  FILLER                     PIC X(60)
   55  005600                                    VALUE SPACES.
   56  005700    05  FILLER                     PIC X(72)
   57  005800                                    VALUE "ERROR REPORT".
   58  005900 01  DETAIL-LINE.
   59  006000    05  FILLER                     PIC X(15)
   60  006100                                    VALUE SPACES.
   61  006200    05  DESCRIPTION                PIC X(25)
   62  006300                                    VALUE SPACES.
   63  006400    05  DETAIL-VALUE               PIC X(92)
   64  006500                                    VALUE SPACES.
```

Figura 28 (Parte 1 de 3). Ejemplo de un Procedimiento de Recuperación de Errores

```
5763CB1 V3R0M5                  Fuente COBOL AS/400
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
   65  006600 01  MESSAGE-LINE.
   66  006700    05  FILLER                    PIC X(15)
   67  006800                                   VALUE SPACES.
   68  006900    05  DESCRIPTION               PIC X(117)
   69  007000                                   VALUE SPACES.
   70  007100 PROCEDURE DIVISION.
       007200 DECLARATIVES.
       007300 HANDLE-ERRORS SECTION.
       007400     USE AFTER STANDARD ERROR PROCEDURE ON RECOVFILE. 1
       007500 DISPLAY-ERROR.
   71  007600     SET USE-PROC-EXECUTED TO TRUE.
   72  007700     WRITE PRINTER-REC FROM HEADER-LINE AFTER ADVANCING PAGE.
   73  007800     MOVE "ERROR OCCURRED IN" TO DESCRIPTION OF DETAIL-LINE.
   74  007900     MOVE I-O-VERB TO DETAIL-VALUE OF DETAIL-LINE.
   75  008000     WRITE PRINTER-REC FROM DETAIL-LINE AFTER ADVANCING 5 LINES.
   76  008100     MOVE "FILE STATUS =" TO DESCRIPTION OF DETAIL-LINE.
   77  008200     MOVE STATUS-FLD TO DETAIL-VALUE OF DETAIL-LINE. 2
   78  008300     WRITE PRINTER-REC FROM DETAIL-LINE AFTER ADVANCING 2 LINES.
   79  008400     MOVE "EXTENDED FILE STATUS =" TO DESCRIPTION OF DETAIL-LINE.
   80  008500     MOVE STATUS-FLD-2 TO DETAIL-VALUE OF DETAIL-LINE.
   81  008600     WRITE PRINTER-REC FROM DETAIL-LINE AFTER ADVANCING 2 LINES.
   82  008700     MOVE "CONTROL-AREA =" TO DESCRIPTION OF DETAIL-LINE.
   83  008800     MOVE CONTROL-FLD TO DETAIL-VALUE OF DETAIL-LINE.
   84  008900     WRITE PRINTER-REC FROM DETAIL-LINE AFTER ADVANCING 2 LINES.
       009000 CHECK-ERROR.
   85  009100     IF TEMPORARY-ERROR AND NO-RECOVERY-DONE THEN
   86  009200         MOVE "***ERROR RECOVERY BEING ATTEMPTED***" 3
       009300             TO  DESCRIPTION OF MESSAGE-LINE
   87  009400         WRITE PRINTER-REC FROM MESSAGE-LINE
       009500             AFTER ADVANCING 3 LINES
   88  009600         PERFORM ERROR-RECOVERY
       009700     ELSE
   89  009800         IF RECOVERY-DONE THEN 4
   90  009900             MOVE "***ERROR AROSE FROM RETRY AFTER RECOVERY***"
       010000                 TO  DESCRIPTION OF MESSAGE-LINE
   91  010100             WRITE PRINTER-REC FROM MESSAGE-LINE
       010200                 AFTER ADVANCING 3 LINES
   92  010300             MOVE "***PROGRAM TERMINATED***"
       010400                 TO  DESCRIPTION OF MESSAGE-LINE
   93  010500             WRITE PRINTER-REC FROM MESSAGE-LINE
       010600                 AFTER ADVANCING 2 LINES
   94  010700             GO TO ERROR-EXIT
       010800         ELSE
   95  010900             SET NO-RECOVERY-DONE TO TRUE.
   96  011000     MOVE "***EXECUTION CONTINUES***"
       011100         TO DESCRIPTION OF MESSAGE-LINE.
   97  011200     WRITE PRINTER-REC FROM MESSAGE-LINE
       011300         AFTER ADVANCING 2 LINES.
   98  011400     GO TO END-OF-DECLARATIVES.
       011500 ERROR-RECOVERY.
   99  011600     SET RECOVERY-DONE TO TRUE.
  100  011700     DROP PGM-DEVICE-NAME FROM RECOVFILE.
  101  011800     ACQUIRE PGM-DEVICE-NAME FOR RECOVFILE. 5
       011900 ERROR-EXIT.
  102  012000     CLOSE RECOVFILE
       012100           PRINTER-FILE.
       012200 END-OF-DECLARATIVES.
       012300 END DECLARATIVES.
       012400
       012500 MAIN-PROGRAM SECTION.
       012600 MAINLINE.
  103  012700     MOVE "OPEN" TO I-O-VERB.
  104  012800     OPEN I-O     RECOVFILE
       012900          OUTPUT  PRINTER-FILE.
  105  013000     PERFORM I-O-PARAGRAPH UNTIL END-REQUESTED. 6
  106  013100     CLOSE RECOVFILE
       013200           PRINTER-FILE.
  107  013300     STOP RUN.
       013400 I-O-PARAGRAPH.
  108  013500     MOVE "WRITE" TO I-O-VERB.
  109  013600     SET USE-PROC-NOT-EXECUTED TO TRUE.
  110  013700     WRITE RECOV-REC FORMAT IS "FORMAT1"
       013800          INDICATOR IS END-INDICATOR.
  111  013900     IF USE-PROC-EXECUTED AND RECOVERY-DONE THEN 7
  112  014000         GO TO I-O-PARAGRAPH.
```

Figura 28 (Parte 2 de 3). Ejemplo de un Procedimiento de Recuperación de Errores

```
5763CB1 V3R0M5                  Fuente COBOL AS/400
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
  113  014100     MOVE "READ" TO I-O-VERB.
  114  014200     SET USE-PROC-NOT-EXECUTED TO TRUE.
  115  014300     SET NO-RECOVERY-DONE TO TRUE.
  116  014400     READ RECOVFILE FORMAT IS "FORMAT1"
       014500         INDICATOR IS END-INDICATOR. 8
  117  014600     IF NO-ERROR THEN
  118  014700         PERFORM SOME-PROCESSING.
       014800 SOME-PROCESSING.
  119  014900     (INSERT SOME DATABASE PROCESSING, FOR EXAMPLE).
                        * * * * *  F I N  D E  F U E N T E  * * * * *
```

Figura 28 (Parte 3 de 3). Ejemplo de un Procedimiento de Recuperación de Errores

1 Define el proceso que se produce cuando hay un error de E/S en RECOVFILE.

2 Imprime información para ayudar en el diagnóstico del problema.

3 Si el estado de archivo es igual a 9N (error transitorio) y no se ha intentado ninguna recuperación anterior de error para esta operación de E/S, se intenta ahora la recuperación del error.

4 Para evitar un bucle en un programa, ahora no se intenta la recuperación si se ha intentado anteriormente.

5 La recuperación consiste en desactivar y volver a adquirir el dispositivo de programa en el que se ha producido el error de E/S.

6 La tarea principal del programa consiste en escribir y leer en un dispositivo hasta que el usuario indique un final de programa pulsando F1.

7 Si la operación WRITE ha sido anómala pero se ha efectuado la recuperación, se intenta de nuevo WRITE.

8 Si la operación READ ha sido anómala, el proceso continuará escribiendo otra vez en el dispositivo, e intentando luego de nuevo READ.

Capítulo 7. Gestión de Archivos y Datos

Este capítulo contiene información general acerca de la gestión de archivos y datos que le pueden ser útiles al crear aplicaciones COBOL/400.

Este capítulo describe:

- Las características de los programas COBOL/400 en el sistema AS/400, tanto independientes de dispositivos como dependientes de dispositivos
- Funciones de spooling de entrada y salida
- Consideraciones para la alteración temporal del sistema
- Consideraciones para el bloqueo de archivos y registros
- Control de compromiso
- Bloqueo y desbloqueo de registros
- Estado de archivos y áreas de realimentación
- Información general acerca de la utilización de archivos descritos externamente y archivos descritos por el programa en un programa COBOL/400
- La instrucción COPY de Formato 2 (Opción DD, DDR, DDS o DDSR).

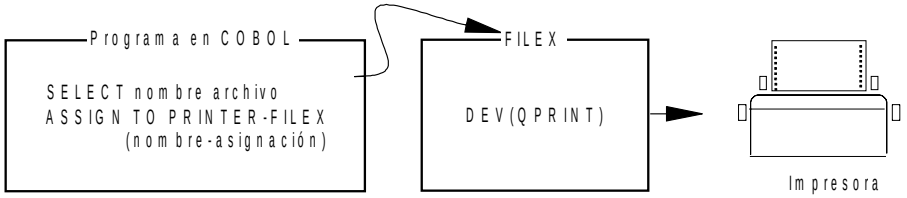
El número máximo de archivos que el usuario puede definir y abrir en una serie de archivos utilizados por un programa COBOL es de 99. Si utiliza opciones de visualización ampliada, el número máximo es 98. Para más información acerca de la especificación de las opciones de visualización ampliada, consulte la página 23 .

Dependencia e Independencia de Dispositivo

El elemento clave para todas las operaciones de E/S en el sistema AS/400 es el archivo. Todos los archivos utilizados se definen para el sistema operativo. El sistema operativo mantiene una descripción de cada archivo utilizado por un programa.

Los archivos se mantienen en línea y sirven como enlace de conexión entre un programa y el dispositivo utilizado para E/S. La asociación de dispositivos reales se realiza cuando se procesa el archivo. En determinados casos, este tipo de control de E/S permite que el usuario cambie el atributo del archivo (y en algunos casos, el dispositivo) utilizado en un programa sin cambiar dicho programa.

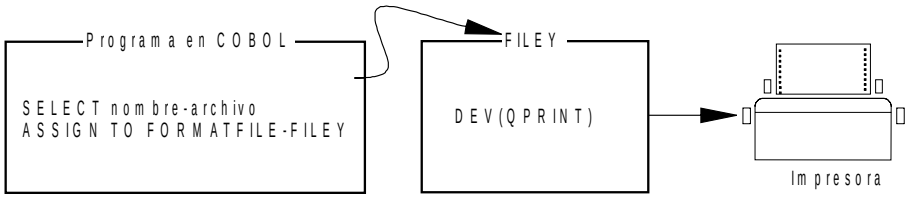
En el lenguaje COBOL/400, el nombre de archivo especificado en la entrada ASSIGNMENT-NAME de la cláusula ASSIGN de la entrada de control del archivo se utiliza para direccionar al archivo. Este nombre del archivo se direcciona a la descripción de archivos del sistema:



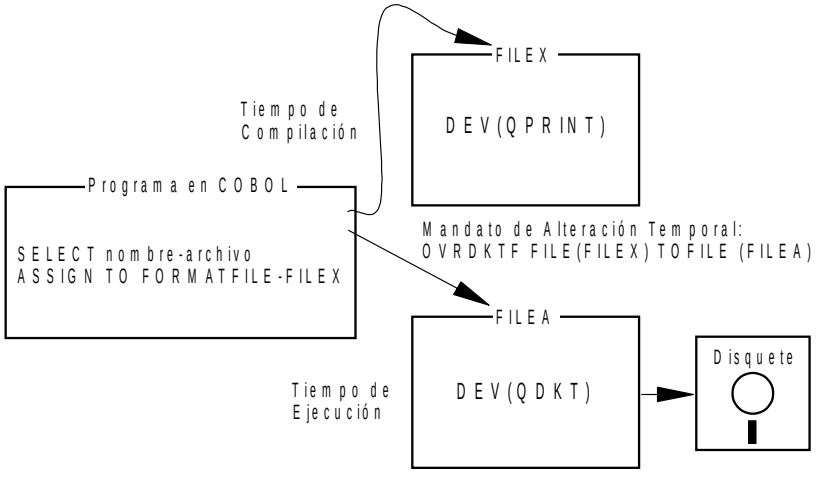
El nombre de dispositivo COBOL en la cláusula ASSIGN define las funciones COBOL que pueden procesarse en el archivo seleccionado. En tiempo de compilación, ciertas funciones COBOL son válidas sólo para un nombre de dispositivo COBOL específico; en este aspecto, COBOL es dependiente de dispositivo. Los ejemplos siguientes son de dependencia de dispositivo:

- Las operaciones SUBFILE sólo son válidas para un dispositivo WORKSTATION.
- Los indicadores sólo son válidos para dispositivos WORKSTATION o FORMATFILE.
- LINAGE sólo es válida para el dispositivo PRINTER.
- OPEN INPUT WITH NO REWIND sólo es válida para el dispositivo TAPEFILE.

Por ejemplo, supongamos que el nombre de archivo FILEY se asocia en el programa COBOL con el dispositivo FORMATFILE. El dispositivo FORMATFILE es un tipo de dispositivo independiente. Por lo tanto, no hay ninguna especificación de línea o control válida en el programa COBOL en la instrucción WRITE ADVANCING. Cuando el programa se ejecuta, el dispositivo de E/S real se especifica en la descripción de FILEY. Por ejemplo, el dispositivo puede ser una impresora; únicamente se utilizaría la línea por omisión y el control de páginas o aquellos definidos en el DDS:



Los mandatos CL pueden utilizarse para alterar temporalmente un parámetro en la descripción de archivo especificada o para volver a dirigir un archivo en tiempo de compilación o ejecución. La redirección de archivos permite que el usuario especifique un archivo en tiempo de compilación y otro archivo en tiempo de ejecución:



En el ejemplo anterior, el mandato Alterar Temporalmente a Archivo de Disquete (OVRDKTF) permite que el programa se ejecute con un archivo de dispositivo completamente distinto al que se especificó en tiempo de compilación.

No todas las redirecciones de archivos o alteraciones temporales son válidas. En tiempo de ejecución, se produce una comprobación para asegurar que las especificaciones dentro del programa COBOL sean válidas para el archivo a procesar. El sistema operativo OS/400 permite algunas redirecciones de archivos incluso si el programa contiene dispositivos específicos. Por ejemplo, si el nombre del dispositivo COBOL es PRINTER y el archivo real que utiliza el programa no es una impresora, el sistema operativo ignora las especificaciones de salto y espaciado de impresión del COBOL.

Hay otras redirecciones de archivo que no están permitidas por el sistema operativo y que provocan la terminación del programa. Por ejemplo, si el nombre de dispositivo COBOL es DATABASE o DISK y se especifica una operación READ por clave en el programa, el programa se termina si el archivo real que utiliza el programa no es un archivo de disco o de base de datos.

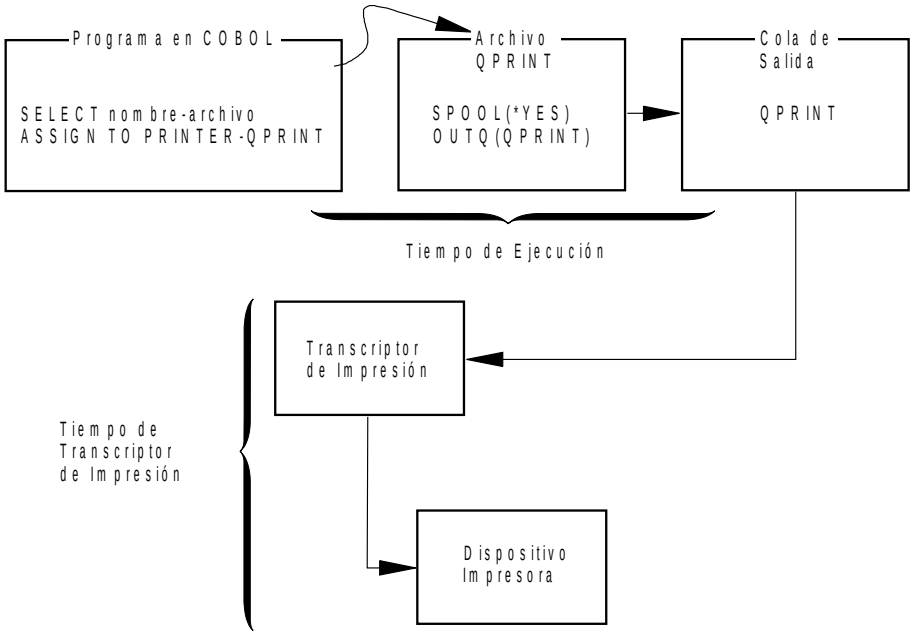
Consulte el apartado “Consideraciones acerca de la Alteración Temporal del Sistema” en la página 97 para obtener información más detallada acerca de las redirecciones y las alteraciones temporales de archivos válidas.

Spooling

El sistema AS/400 proporciona la utilización de funciones de spool de entrada y salida. Cada descripción de archivo AS/400 contiene un atributo de spool que determina si se utiliza el spooling para el archivo en tiempo de ejecución. El programa COBOL no sabe que está utilizando el spool. El lector o el transcriptor de spool determinan el dispositivo físico real desde el que se lee un archivo o al que se graba un archivo. Consulte la publicación *Guía para la Gestión de Datos* para obtener una información más detallada acerca del spooling.

Spool de Salida

El spool de salida es válido para los trabajos por lotes e interactivos. La descripción del archivo que se especifica en COBOL mediante el nombre del sistema contiene la especificación para el spool, tal como se muestra en el ejemplo siguiente:

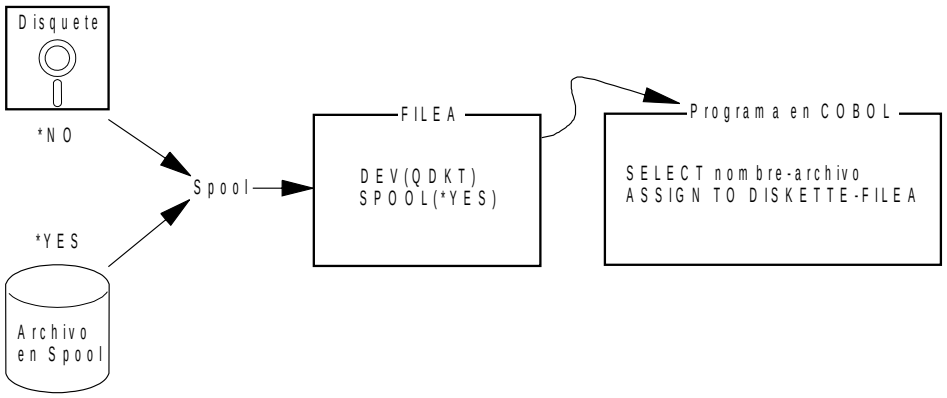


Los mandatos de alteración temporal de archivos pueden utilizarse en tiempo de ejecución para alterar temporalmente las funciones que se especifican en la descripción de archivos, como el número de copias a imprimir. Además, el soporte de spooling de AS/400 le permitirá redirigir un archivo una vez se ha ejecutado el programa. Por ejemplo, podrá dirigir la salida de impresora a un dispositivo distinto, como puede ser un disquete.

Spool de Entrada

El spool de entrada sólo es válido para los archivos de datos incorporados en trabajos por lotes. Si los datos de entrada que lee COBOL provienen de un archivo en spool, COBOL no sabe desde qué dispositivo se han puesto en spool los datos.

Los datos se leen desde un archivo incorporado en spool:



Consulte la publicación *Guía para la Gestión de Datos* para obtener más información acerca de los archivos de datos incorporados.

Consideraciones acerca de la Alteración Temporal del Sistema

Debe especificar cualquier alteración temporal antes de que el programa COBOL abra el archivo. El sistema utiliza el mandato de alteración temporal de archivo para determinar el archivo que se va a abrir y los atributos del archivo.

La forma más sencilla de alterar temporalmente un archivo es alterar temporalmente algunos atributos de dicho archivo. Por ejemplo, se especifica FILE(OUTPUT) con COPIES(2) cuando se crea un archivo de impresora. Antes de ejecutar un programa COBOL, el número de copias impresas de salida puede cambiarse a 3. El mandato de alteración temporal es el siguiente:

```
OVRPRTF  FILE(OUTPUT)  COPIES(3)
```

Otra manera de alterar temporalmente un archivo es redirigiendo el programa COBOL para que acceda a un archivo distinto. Cuando la alteración temporal redirige el programa a un archivo del *mismo* tipo (como, por ejemplo, de un archivo de impresora a otro archivo de impresora) el archivo se procesa de la misma manera que el archivo original.

Cuando la alteración temporal redirige el programa a un archivo de tipo *diferente*, el archivo alterado temporalmente se procesa de la misma forma que se procesaría el archivo original. Las especificaciones dependientes de dispositivo en el programa COBOL se ignoran, y el sistema toma los valores por omisión.

No todas las redirecciones de archivos son válidas. Por ejemplo, un archivo indexado para un programa COBOL sólo puede alterarse temporalmente a otro archivo indexado con una vía de acceso por clave.

Al alterar temporalmente un archivo de base de datos para procesar todos los miembros se puede conseguir el proceso de múltiples miembros para un archivo de base de datos. Observe las excepciones siguientes:

- No se puede alterar temporalmente un archivo fuente de base de datos utilizado para un programa COBOL para procesar todos los miembros. Al especificar OVRDBF MBR(*ALL) se acaba la compilación.
- No se puede alterar temporalmente un archivo de base de datos utilizado para una instrucción COPY para procesar todos los miembros. Al especificar OVRDBF MBR(*ALL) se ignorará la instrucción COPY.

El programador COBOL debe asegurarse de que las alteraciones temporales se apliquen convenientemente. Para más información acerca de las redirecciones de archivo válidas, las características dependientes de dispositivo ignoradas y los valores por omisión asumidos, consulte la publicación *Guía para la Gestión de Datos*.

Bloqueo de Archivos y Registros por COBOL

El sistema operativo permite que un estado de bloqueo (exclusivo, de lectura admisible exclusiva, compartido sin actualización o compartido para lectura) se sitúe en un archivo utilizado durante el paso de un trabajo. El archivo puede situarse en un estado de bloqueo con el mandato Asignar Objeto (ALCOBJ).

Por omisión, el sistema operativo coloca los estados de bloqueo siguientes en archivos de base de datos cuando los programas COBOL abren los archivos:

Tipo OPEN	Estado de Bloqueo
INPUT	Compartido para lectura
I/O	Compartido para actualización
EXTEND	Compartido para actualización
OUTPUT	Compartido para actualización

La modalidad EXTEND es un método para añadir registros al final de un archivo secuencial cuando se abre el archivo.

El estado de bloqueo compartido para lectura permite que otro usuario abra el archivo con un estado de bloqueo compartido para lectura, compartido para actualización, compartido sin actualización y de lectura admisible exclusiva, pero el usuario no puede especificar el uso exclusivo del archivo. El estado de bloqueo compartido para actualización permite que otro usuario abra el archivo con un estado de bloqueo compartido para lectura o compartido para actualización.

El sistema operativo coloca el bloqueo compartido para lectura en un archivo de dispositivo y un estado de bloqueo de lectura admisible exclusiva en el dispositivo. Otro usuario puede abrir el archivo pero no puede utilizar el mismo dispositivo.

Nota: Cuando un programa COBOL abre un archivo físico para OUTPUT, ese archivo estará sujeto a un bloqueo exclusivo durante el tiempo necesario para borrar el miembro.

Para más información acerca de la asignación de recursos y estados de bloqueo, consulte la publicación *Guía para la Gestión de Datos*.

Bloqueo y Liberación de Registros

Cuando COBOL lee un registro de base de datos y el archivo se abre para E/S, se coloca un bloqueo en ese registro de forma que otro programa no pueda actualizarlo. Es decir, otro programa puede leer el registro si abre un archivo para entrada, pero no si abre el archivo para E/S.

Para más información acerca de la duración del bloqueo de registro con o sin control de compromiso, consulte la Tabla 2 en la página 101.

Para evitar que la instrucción READ bloquee registros en archivos abiertos en modalidad de E/S (actualización), puede utilizar la frase NO LOCK. La instrucción READ WITH NO LOCK desbloquea los registros bloqueados por una instrucción READ anterior. Para más información acerca de esta frase, consulte la sección de la instrucción READ en la publicación *COBOL/400 Reference*.

Para un archivo lógico basado en un archivo físico, el bloqueo se coloca en el registro del archivo físico. Si un archivo lógico se basa en más de un archivo físico, se coloca un bloqueo en un registro en cada archivo físico.

Este bloqueo no sólo se aplica para otros programas, sino también para el programa original si intenta actualizar el mismo registro físico subyacente a través de un segundo archivo.

Nota: Cuando un archivo con organización indexada o relativa se abre para E/S, utilizando acceso al azar o dinámico, una operación de E/S anómala en

cualquier verbo de E/S excepto WRITE también desbloquea el registro. Una operación WRITE no se considera una operación de actualización; por lo tanto, no se libera el bloqueo de registro.

Para más información acerca de la liberación de registros de base de datos leídos para la actualización, consulte la publicación *Guía para la Gestión de Datos*.

Posibilidad de Compartir una Vía de Datos Abierta

Si ya ha abierto un archivo a través de otro programa en el paso de direccionamiento, el programa COBOL puede utilizar la misma Vía de de Datos Abierta (ODP) para acceder al archivo.

Nota: Los pasos de direccionamiento se describen en la publicación *Programación: Guía para la Gestión de Trabajos*; un trabajo contiene, generalmente, un solo paso de direccionamiento.

Las reglas siguientes se aplican para las ODP compartidas:

1. Debe especificar SHARE(*YES) en el mandato que crea el archivo, en un mandato de cambio, o en un mandato de alteración temporal para el archivo.
2. Una vez que un programa abre por primera vez una ODP compartida y permanece abierta, las operaciones OPEN siguientes dentro del mismo paso de direccionamiento se ejecutan más rápidamente que las operaciones OPEN estándar. La velocidad de otras operaciones de E/S que no sean la de apertura no se verá afectada.
3. La utilización del archivo dentro de programas diferentes debe ser coherente. Por ejemplo, si un programa que no sea COBOL realiza una operación READ PREVIOUS utilizando E/S bloqueada, la instrucción COBOL READ podría recuperar el registro que precede a la posición del archivo actual y no el registro que sigue a la posición del archivo actual.

Consideraciones sobre Control de Compromiso

El control de compromiso es una función que permite:

- La sincronización de cambios para archivos de base de datos dentro del mismo trabajo
- La cancelación de los cambios que no deben entrarse de modo permanente en la base de datos
- El bloqueo de registros que están modificándose hasta que finalicen los cambios
- Las técnicas para la recuperación de anomalías del trabajo o del sistema.

En algunas aplicaciones, sería deseable sincronizar los cambios para los registros de base de datos. Si el programa determina que los cambios son válidos, estos se efectúan permanentemente en la base de datos (se procesa una instrucción COMMIT). Si los cambios no son válidos, o si se produce un problema durante el proceso, los cambios pueden cancelarse (se procesa una instrucción ROLLBACK). Tenga en cuenta que cuando se borra un archivo después de abrirse para OUTPUT, el proceso de una instrucción ROLLBACK no restaura los registros borrados para el archivo. Los cambios realizados en los registros de un archivo que *no* está bajo el control de compromiso siempre son permanentes. A tales cambios no les afectan nunca las instrucciones COMMIT o ROLLBACK posteriores.

Cada punto en el que se procese satisfactoriamente una instrucción COMMIT o ROLLBACK es un límite de compromiso. Si todavía no se ha emitido ninguna instrucción COMMIT o ROLLBACK en un programa, la primera apertura de cualquier archivo bajo el control de compromiso crea un límite de compromiso. El compromiso o retroacción de cambios afecta únicamente a aquellos cambios que se hayan hecho desde el anterior límite de compromiso.

La sincronización de cambios en los límites de compromiso hace más fáciles los procedimientos de arranque o de recuperación después de una anomalía. Para más información, consulte el apartado “Recuperación de una Anomalía” en la página 86.

Cuando el control de compromiso se utiliza para archivos de base de datos, los registros de dichos archivos están sujetos a un nivel de bloqueo superior LCKLVL (*ALL) o a un nivel de bloqueo inferior LCKLVL(*CHG). Con un nivel de bloqueo inferior (*CHG), se bloquean todos los registros que se cambian (o vuelven a grabarse, se suprimen o se añaden en archivos bajo el control de compromiso) se bloquean hasta que se procesa satisfactoriamente una instrucción COMMIT o ROLLBACK. Con un nivel de bloqueo superior (*ALL), se bloquean *todos* los registros accedidos, tanto para entrada como para salida, hasta que se procesa satisfactoriamente una instrucción COMMIT o una ROLLBACK. Para ambos niveles de bloqueo de registro, ningún otro trabajo puede modificar los datos en registros bloqueados hasta que la instrucción COMMIT o la ROLLBACK se haya completado satisfactoriamente. (Un registro bloqueado sólo puede modificarse dentro del mismo trabajo y a través del mismo archivo físico o lógico.)

El nivel de bloqueo también controla si pueden leerse los registros bloqueados. Con un nivel de bloqueo superior (*ALL), no podrá leer registros bloqueados en un archivo de base de datos. Con un nivel de bloqueo inferior (*CHG), podrá leer registros bloqueados en un archivo de base de datos, siempre que el archivo se abra como INPUT en el trabajo, o que se abra como E/S y se utilice READ WITH NO LOCK.

| Puede obtenerse un tercer nivel de bloqueo especificando LCKLVL(*CS), en el que
| se bloquean todos los registros accedidos desde archivos bajo control de compro-
| miso. Los registros que no se actualizan o se suprimen sólo se bloquean hasta
| que se haya accedido a un registro distinto. Los registros actualizados, añadidos o
| suprimidos se bloquean hasta que la transacción se ha comprometido o retrotraído.

El resto de los trabajos, en los que los archivos *no* están bajo control de compromiso, siempre pueden leer registros bloqueados, prescindiendo del nivel de bloqueo utilizado, con tal de que los archivos se abran como INPUT. Debido a que en algunos casos es posible que otros trabajos lean registros bloqueados, puede accederse a los datos *antes de que estén permanentemente comprometidos para una base de datos*. Si se procesa una instrucción ROLLBACK *después* de que otro trabajo haya leído registros bloqueados, los datos accedidos no reflejarán el contenido de la base de datos.

La Tabla 2 muestra las consideraciones sobre el bloqueo de registros para archivos con y sin control de compromiso.

Tabla 2. Consideraciones sobre el bloqueo de registros con y sin control de compromiso

VERBO	MODALIDAD APERTURA	NIVEL BLOQUEO		DURACIÓN DEL BLOQUEO DE REGISTROS
				Siguiente Operación E/S / COMMIT o ROLLBACK
DELETE	I-O	Sin control de compromiso		DELETE
		Con control de compromiso	*CHG	
			*ALL	
READ	INPUT	Sin control de compromiso		READ
		Con control de compromiso	*CHG	
			*ALL	
READ WITH NO LOCK	I-O	Sin control de compromiso		READ
		Con control de compromiso	*CHG	
			*ALL	
READ	I-O	Sin control de compromiso		READ
		Con control de compromiso	*CHG	
			*ALL	
REWRITE	I-O	Sin control de compromiso		REWRITE
		Con control de compromiso	*CHG	
			*ALL	
START	INPUT	Sin control de compromiso		START
		Con control de compromiso	*CHG	
			*ALL	
START	I-O	Sin control de compromiso		START
		Con control de compromiso	*CHG	
			*ALL	
WRITE	I-O	Sin control de compromiso		WRITE
		Con control de compromiso	*CHG	
			*ALL	
WRITE	OUTPUT	Sin control de compromiso		WRITE
		Con control de compromiso	*CHG	
			*ALL	

Un archivo bajo control de compromiso puede cerrarse o abrirse sin afectar al estado de los cambios realizados desde el último límite de compromiso. Debe emitirse aún una instrucción COMMIT para realizar los cambios permanentes, o emitir una ROLLBACK para cancelar los cambios. Una instrucción COMMIT, cuando se procesa, deja los archivos en el mismo estado, abierto o cerrado, que tenían antes del proceso.

Todos los archivos bajo el control de compromiso dentro del mismo trabajo deben registrarse por diario en el mismo diario. Para más información acerca de la gestión de diario y de sus funciones relacionadas, así como acerca del control de compromiso, consulte la publicación *Advanced Backup and Recovery Guide*.

El control de compromiso también debe especificarse fuera del lenguaje COBOL mediante el lenguaje de control OS/400 (CL). El mandato Arrancar Control de Compromiso (STRCMTCTL) establece la posibilidad de control de compromiso y establece el nivel de bloqueo de registro a nivel superior (*ALL) o a nivel inferior (*CHG). El mandato STRCMTCTL no inicia automáticamente el control de compromiso de un archivo. Dicho archivo también debe especificarse en la cláusula COMMITMENT CONTROL del párrafo I-O-CONTROL en el programa COBOL. El

entorno de control de compromiso se termina normalmente utilizando el mandato Finalizar Control de Compromiso (ENDCMTCTL). Esto hace que se cancelen todos los cambios no comprometidos para archivos de base de datos bajo el control de compromiso. (Se procesa una instrucción ROLLBACK). Consulte la publicación *CL Reference* para más información acerca de los mandatos STRCMTCTL y ENDCMTCTL.

Para más información acerca del control de compromiso, consulte la publicación *Advanced Backup and Recovery Guide*.

Nota: La capacidad de evitar la lectura de datos no comprometidos que hayan sido cambiados es una función de control de compromiso y sólo está disponible si se trabaja bajo control de compromiso. La extensión de control de compromiso no modifica el soporte normal de la base de datos (sin compromiso), y permite leer registros bloqueados cuando se lee un archivo abierto únicamente para entrada. Intente utilizar los archivos de forma coherente. Normalmente, los archivos deben ejecutarse siempre bajo el control de compromiso o no ejecutarse nunca bajo control de compromiso.

La Figura 29 en la página 103 muestra una utilización posible de control de compromiso en un entorno bancario. El programa procesa las transacciones para la transferencia de fondos de una cuenta a otra. Si no se producen problemas durante la transacción, se comprometen los datos para el archivo de base de datos. Si la transferencia no puede realizarse debido a un número de cuenta incorrecto o a fondos insuficientes, se emite una instrucción ROLLBACK para cancelar los cambios.

IBM International Business Machines

AS/400 ESPECIFICACIONES DE DESCRIPCION DE DATOS

GX21-9691-0 UM/050*
Impreso en España
*El número de hojas por área de relleno puede variar ligeramente.

Archivo		Instrucciones de Grabación	Signo							
Programador	Fecha		Tecla							

Descripción	Página de

A

Número de Secuencia	Tipo de Formulario	And/Or/Comment. (A/O/*)	Condicionamiento: No (N)	Nombre Condición: Indicador	No (N)	Indicador	No (N)	Indicador	Tipo Nombre o Espec (b/R/H/J/K/S/O)	Reservado	Nombre	Referencia (R)	Longitud	Tipo Datos/Desplazamiento Teclado	Posiciones Decimales	Utilización (b/O/I/B/H/M/N/P)	Ubicación: Línea	Pos	Funciones
1 2 3 4 5	6	7	8	9 10	11 12 13		14 15 16		17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
	A	*		NOMBRE			DE		E		ARCHIVO		PANTALLA		DE		SOL	ICI	TUD 'ACCTFMTS'
	A	*																	
	A																	1	INDARA
	A								R		ACCTPMT								
	A																		TEXT('SOLICITUD CUENTA CLIENTE')
	A																		
	A																		CA01(15 'FIN DE PROGRAMA')
	A																		PUTRETAIN OVERLAY
	A																1	3	'ACTUALIZ MAESTRA CUENTA'
	A																3	3	'NUMERO CUENTA ORIGEN'
	A										ACCTFROM		5	Y	0	I	3	23	CHECK(ME)
	A			99															ERRMSG('NUMERO CUENTA ORIGEN +
	A																		NO VALIDO' 99)
	A			98															ERRMSG('FONDOS INSUFICIENTES EN +
	A																		CUENTA ORIGEN' 98)
	A																4	3	'NUMERO CUENTA DESTINO'
	A										ACCTTO		5	Y	0	I	4	23	CHECK(ME)
	A			97															ERRMSG('NUMERO CUENTA DESTINO +
	A																		NO VALIDO' 97)
	A																5	3	'IMPORTE TRANSFERIDO'
	A										TRANSAMT		10	Y	02	I	5	23	
	A								R		ERRFMT								
	A			96													6	5	'ESTADO ARCHIVO NO VALIDO'
	A			96													7	5	'TECLA DE GRAVACIÓN NO VALIDA'
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		

Figura 29. Ejemplo de Utilización de Control de Compromiso --DDS

```
5763CB1 V3R0M5                  Fuente AS/400 COBOL
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
    1  000100 IDENTIFICATION DIVISION.                                                              02/01/94
    2  000200 PROGRAM-ID.     ACCOUNT.                                                              02/04/94
    3  000300   AUTHOR.       PROGRAMMER NAME.                                                      01/27/94
    4  000400   INSTALLATION. COBOL DEVELOPMENT CENTRE.                                            01/27/94
    5  000500   DATE-WRITTEN. 02/02/88.                                                            02/04/94
    8  000080   DATE-COMPILED.  05/24/92 14:02:39   .                                             03/01/94
    7  000700 ENVIRONMENT DIVISION.                                                               01/27/94
    8  000800 CONFIGURATION SECTION.                                                              01/27/94
    9  000900 SOURCE-COMPUTER. IBM-AS400.                                                         01/27/94
   10  001000 OBJECT-COMPUTER. IBM-AS400.                                                         01/27/94
   11  001100 INPUT-OUTPUT SECTION.                                                               01/27/94
   12  001200 FILE-CONTROL.                                                                       01/27/94
   13  001300     SELECT ACCOUNT-FILE  ASSIGN TO DATABASE-ACCTMST                                 02/04/94
   14  001400         ORGANIZATION IS INDEXED                                                     02/04/94
   15  001500         ACCESS IS DYNAMIC                                                           02/04/94
   16  001600         RECORD IS EXTERNALLY-DESCRIBED-KEY                                          02/04/94
   17  001700         FILE STATUS IS ACCOUNT-FILE-STATUS.                                         02/04/94
   18  001800     SELECT DISPLAY-FILE ASSIGN TO WORKSTATION-ACCTFMTS-SI  1                        02/04/94
   19  001900         ORGANIZATION IS TRANSACTION.                                                02/04/94
       002000*****************************************************************                 02/04/94
   20  002100 CONTROL DE E-S                                                                      02/04/94
   21  002200     CONTROL DE COMPROMISO PARA ACCOUNT-FILE.  2                                     02/04/94
       002300*****************************************************************                 02/04/94
   22  002400 DATA DIVISION.                                                                      02/04/94
   23  002500 FILE SECTION.                                                                       02/04/94
   24  002600 FD  ACCOUNT-FILE                                                                    02/04/94
   25  002700     LABEL RECORDS ARE STANDARD.                                                     02/04/94
   26  002800 01  ACCOUNT-RECORD.                                                                02/04/94
   27  002900     COPY DDS-ALL-FORMATS OF ACCTMST.                                                02/04/94
   28 +000001       05  ACCTMST-RECORD PIC X(82).                                 <-ALL-FMTS
      +000002*   FORMATO E-S:ACCNTREC  DESDE ARCHIVO ACCTMST    DE BIBLIOTECA XMPLIB       <-ALL-FMTS
      +000003*                                                                               <-ALL-FMTS
      +000004*DEFINICIONES CLAVE PARA FORMATO DE REGISTRO  ACCNTREC                         <-ALL-FMTS
      +000005*  NÚMERO              NOMBRE             RECUPERACIÓN  TIPO    SECALT       <-ALL-FMTS
      +000006*   0001   ACCNTKEY                        ASCENDENTE  CON SIGNO  NO        <-ALL-FMTS
   29 +000007       05  ACCNTREC      REDEFINES ACCTMST-RECORD.                          <-ALL-FMTS
   30 +000008           06 ACCNTKEY          PIC S9(5).                                 <-ALL-FMTS
   31 +000009           06 NAME              PIC X(20).                                 <-ALL-FMTS
   32 +000010           06 ADDR              PIC X(20).                                 <-ALL-FMTS
   33 +000011           06 CITY              PIC X(20).                                 <-ALL-FMTS
   34 +000012           06 STATE             PIC X(2).                                  <-ALL-FMTS
   35 +000013           06 ZIP               PIC S9(5).                                 <-ALL-FMTS
   36 +000014           06 BALANCE           PIC S9(8)V9(2).                            <-ALL-FMTS
       003000
   37  003100 FD  DISPLAY-FILE
   38  003200     LABEL RECORDS ARE STANDARD.
   39  003300 01  DISPLAY-REC.
   40  003400     COPY DDS-ALL-FORMATS OF ACCTFMTS.
   41 +000001       05  ACCTFMTS-RECORD PIC X(20).                                <-ALL-FMTS
      +000002*  FORMATO ENTRADA:ACCTPMT    DESDE ARCHIVO ACCTFMTS   DE BIBLIOTECA XMPLIB  <-ALL-FMTS
      +000003*                           SOLICITUD CUENTA CLIENTE                          <-ALL-FMTS
   42 +000004       05  ACCTPMT-I     REDEFINES ACCTFMTS-RECORD.                         <-ALL-FMTS
   43 +000005           06 ACCTFROM          PIC S9(5).                                 <-ALL-FMTS
   44 +000006           06 ACCTTO            PIC S9(5).                                 <-ALL-FMTS
   45 +000007           06 TRANSAMT          PIC S9(8)V9(2).                            <-ALL-FMTS
      +000008* FORMATO SALIDA:ACCTPMT    DESDE ARCHIVO ACCTFMTS   DE BIBLIOTECA XMPLIB    <-ALL-FMTS
      +000009*                           SOLICITUD CUENTA CLIENTE                          <-ALL-FMTS
      +000010*      05  ACCTPMT-O     REDEFINES ACCTFMTS-RECORD.                         <-ALL-FMTS
      +000011*  FORMATO ENTRADA:ERRFMT     DESDE ARCHIVO ACCTFMTS   DE BIBLIOTECA XMPLIB  <-ALL-FMTS
      +000012*                                                                            <-ALL-FMTS
      +000013*      05  ERRFMT-I      REDEFINES ACCTFMTS-RECORD.                         <-ALL-FMTS
      +000014* FORMATO SALIDA:ERRFMT     DESDE ARCHIVO ACCTFMTS   DE BIBLIOTECA XMPLIB    <-ALL-FMTS
      +000015*                                                                            <-ALL-FMTS
      +000016*      05  ERRFMT-O      REDEFINES ACCTFMTS-RECORD.                         <-ALL-FMTS
   46  003500 WORKING-STORAGE SECTION.
   47  003600 77  ACCOUNT-FILE-STATUS           PIC X(2).
```

Figura 30 (Parte 1 de 3). Ejemplo de la Utilización del Control de Compromiso

```
5763CB1 V3R0M5                  AS/400 COBOL Source
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
   48  003700 77  IND-ON                      PIC 1    VALUE B"1".
   49  003800 77  IND-OFF                     PIC 1    VALUE B"0".
   50  003900 01  DISPFILE-INDICS.
   51  004000     COPY DDS-ALL-FORMATS-INDIC OF ACCTFMTS. 3
   52 +000001       05  ACCTFMTS-RECORD.                                    <-ALL-FMTS
      +000002*  FORMATO ENTRADA:ACCTPMT    DESDE ARCHIVO ACCTFMTS   DE BIBLIOTECA XMPLIB   <-ALL-FMTS
      +000003*                           SOLICITUD CUENTA CLIENTE                          <-ALL-FMTS
   53 +000004           06 ACCTPMT-I-INDIC.                                  <-ALL-FMTS
   54 +000005                07 IN15         PIC 1  INDIC 15.               <-ALL-FMTS
      +000006*                           FIN DE PROGRAMA                                   <-ALL-FMTS
   55 +000007                07 IN97         PIC 1  INDIC 97.               <-ALL-FMTS
      +000008*                           NÚMERO DE CUENTA DESTINO NO VÁLIDO                <-ALL-FMTS
   56 +000009                07 IN98         PIC 1  INDIC 98.               <-ALL-FMTS
      +000010*                           FONDOS INSUFICIENTES EN CUENTA ORIGEN             <-ALL-FMTS
   57 +000011                07 IN99         PIC 1  INDIC 99.               <-ALL-FMTS
      +000012*                           NÚMERO DE CUENTA ORIGEN NO VÁLIDO                 <-ALL-FMTS
      +000013* FORMATO SALIDA:ACCTPMT    DESDE ARCHIVO ACCTFMTS   DE BIBLIOTECA XMPLIB    <-ALL-FMTS
      +000014*                           SOLICITUD CUENTA CLIENTE                          <-ALL-FMTS
   58 +000015           06 ACCTPMT-O-INDIC.                                  <-ALL-FMTS
   59 +000016                07 IN97         PIC 1  INDIC 97.               <-ALL-FMTS
      +000017*                           NÚMERO DE CUENTA DESTINO NO VÁLIDO                <-ALL-FMTS
   60 +000018                07 IN98         PIC 1  INDIC 98.               <-ALL-FMTS
      +000019*                           FONDOS INSUFICIENTES EN CUENTA ORIGEN             <-ALL-FMTS
   61 +000020                07 IN99         PIC 1  INDIC 99.               <-ALL-FMTS
      +000021*                           NÚMERO DE CUENTA ORIGEN NO VÁLIDO                 <-ALL-FMTS
      +000022*  FORMATO ENTRADA:ERRFMT     DESDE ARCHIVO ACCTFMTS   DE BIBLIOTECA XMPLIB   <-ALL-FMTS
      +000023*                                                                            <-ALL-FMTS
      +000024*          06 ERRFMT-I-INDIC.                                   <-ALL-FMTS
      +000025* FORMATO SALIDA:ERRFMT     DESDE ARCHIVO ACCTFMTS   DE BIBLIOTECA XMPLIB    <-ALL-FMTS
      +000026*                                                                            <-ALL-FMTS
   62 +000027           06 ERRFMT-O-INDIC.                                   <-ALL-FMTS
   63 +000028                07 IN95         PIC 1  INDIC 95.               <-ALL-FMTS
   64 +000029                07 IN96         PIC 1  INDIC 96.               <-ALL-FMTS
       004100
   65  004200 PROCEDURE DIVISION.
       004300 DECLARATIVES.
       004400 ERROR-SECTION SECTION.
       004500     USE AFTER STANDARD EXCEPTION PROCEDURE ON ACCOUNT-FILE.
       004600 ERROR-PARAGRAPH.
   66  004700     IF ACCOUNT-FILE-STATUS IS NOT EQUAL "23" THEN
   67  004800         MOVE IND-ON TO IN96 OF ERRFMT-O-INDIC     4
       004900     ELSE
   68  005000         MOVE IND-ON TO IN95 OF ERRFMT-O-INDIC. 5
   69  005100     WRITE DISPLAY-REC FORMAT IS "ERRFMT"
       005200          INDICATORS ARE ERRFMT-O-INDIC.
   70  005300     READ DISPLAY-FILE.
   71  005400     CLOSE DISPLAY-FILE
       005500           ACCOUNT-FILE.
   72  005600     STOP RUN.
       005700 END DECLARATIVES.
       005800 MAIN-PROGRAM SECTION.
       005900 MAINLINE.
   73  006000     OPEN I-O DISPLAY-FILE
       006100          I-O ACCOUNT-FILE.
   74  006200     MOVE ZEROS TO ACCTPMT-I-INDIC
       006300                   ACCTPMT-O-INDIC.
   75  006400     PERFORM WRITE-READ-DISPLAY.
   76  006500     PERFORM VERIFY-ACCOUNT-NO UNTIL IN15 EQUAL IND-ON.
   77  006600     CLOSE DISPLAY-FILE
       006700           ACCOUNT-FILE.
   78  006800     STOP RUN.
       006900 VERIFY-ACCOUNT-NO.
   79  007000     PERFORM VERIFY-TO-ACCOUNT.
   80  007100     IF IN97 OF ACCTPMT-O-INDIC EQUAL IND-OFF THEN
   81  007200         PERFORM VERIFY-FROM-ACCOUNT.
   82  007300     PERFORM WRITE-READ-DISPLAY.
       007400 VERIFY-FROM-ACCOUNT.
   83  007500     MOVE ACCTFROM TO ACCNTKEY.
   84  007600     READ ACCOUNT-FILE
   85  007700         INVALID KEY MOVE IND-ON TO IN99 OF ACCTPMT-O-INDIC.
   86  007800     IF IN99 OF ACCTPMT-O-INDIC EQUAL IND-ON THEN 6
       007900*                                            *
       008000         ROLLBACK
       008100*                                            *
   87  008200     ELSE
   88  008300         PERFORM UPDATE-FROM-ACCOUNT.
```

Figura 30 (Parte 2 de 3). Ejemplo de la Utilización del Control de Compromiso

```
5763CB1 V3R0M5                  Fuente AS/400 COBOL
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
      008400 VERIFY-TO-ACCOUNT.
   89 008500     MOVE ACCTTO TO ACCNTKEY.
   90 008600     READ ACCOUNT-FILE
   91 008700         INVALID KEY MOVE IND-ON TO IN97 OF ACCTPMT-O-INDIC. 7
   92 008800     IF IN97 OF ACCTPMT-O-INDIC EQUAL IND-ON THEN
      008900*                                              *
      009000         ROLLBACK 8
      009100*                                              *
   93 009200     ELSE
   94 009300         PERFORM UPDATE-TO-ACCOUNT.
      009400 UPDATE-TO-ACCOUNT.
   95 009500     ADD TRANSAMT TO BALANCE.
   96 009600     REWRITE ACCOUNT-RECORD.
      009700 UPDATE-FROM-ACCOUNT.
   97 009800     SUBTRACT TRANSAMT FROM BALANCE.
   98 009900     REWRITE ACCOUNT-RECORD.
   99 010000     IF BALANCE IS LESS THAN 0 THEN
  100 010100         MOVE IND-ON TO IN98 OF ACCTPMT-O-INDIC
      010200*                                              *
      010300         ROLLBACK 9
      010400*                                              *
  101 010500     ELSE
      010600*                                              *
      010700         COMMIT. 10
      010800*                                              *
  102 010900 WRITE-READ-DISPLAY.
  103 011000     WRITE DISPLAY-REC FORMAT IS "ACCTPMT"
      011100         INDICATORS ARE ACCTPMT-O-INDIC. 11
  104 011200     MOVE ZEROS TO ACCTPMT-I-INDIC
      011300                   ACCTPMT-O-INDIC.
  105 011400     READ DISPLAY-FILE RECORD
      011500         INDICATORS ARE ACCTPMT-I-INDIC.
      011600
                       * * * * *  F I N  D E L  F U E N T E  * * * * *
```

Figura 30 (Parte 3 de 3). Ejemplo de la Utilización del Control de Compromiso

1 Se proporciona un área de indicadores separada para el programa.

2 La cláusula COMMITMENT CONTROL especifica los archivos que han de ponerse bajo control de compromiso. Todos los archivos que se nombren en esta cláusula quedan afectados por los verbos COMMIT y ROLLBACK.

3 La instrucción COPY de Formato 2 con el atributo de indicador INDIC, define las entradas de descripción de datos en WORKING-STORAGE para los indicadores que el programa va a utilizar.

4 Se activa IN96 si hay un estado de archivo no válido.

5 Se activa IN95 si hay una condición INVALID KEY en la operación REWRITE.

6 Se activa IN99 si el número de cuenta entrado para la cuenta a la que se transfiere el dinero no es válido.

7 Se activa IN97 si el número de cuenta entrado para la cuenta a la que debe transferirse el dinero no es válido.

8 Si se produce la condición INVALID KEY en la instrucción READ, se utiliza una ROLLBACK y el bloqueo de registro colocado en el registro después de la primera instrucción READ se libera.

9 Si no se permite la transferencia de fondos (se ha activado un indicador), se procesa la instrucción ROLLBACK. Se cancelan todos los cambios realizados en archivos de base de datos bajo control de compromiso.

10 Si la transferencia de fondos ha sido válida (no se han activado indicadores), se procesa la instrucción COMMIT, y todos los cambios realizados en los archivos de base de datos bajo control de compromiso serán permanentes.

11 La frase INDICATORS es necesaria para las opciones en la pantalla de estación de trabajo que se controlan mediante los indicadores.

Desbloqueo de Registros de Entrada y Bloqueo de Registros de Salida

Un **bloque** contiene más de un registro. Para aumentar el rendimiento en las operaciones de entrada y salida, el compilador COBOL genera códigos para desbloquear registros de entrada y bloquear registros de salida si se producen todas las condiciones siguientes:

1. Se especifica *NOBLK (con o sin una cláusula BLOCK CONTAINS) y si se dan **todas** las condiciones siguientes:
   a. Para el archivo se especifica ACCESS IS SEQUENTIAL.
   b. El archivo sólo se abre para INPUT o OUTPUT en ese programa.
   c. El archivo se asigna a DISK, DATABASE, DISKETTE o TAPEFILE.
   d. No se especifican instrucciones START para el archivo.

   En la organización RELATIVE, no se efectúa bloqueo para OPEN OUTPUT.

   Si se especifica BLOCK CONTAINS, se ignorará excepto para archivos de cinta. En los archivos de cinta, la cláusula BLOCK CONTAINS controla el número de registros que se han de bloquear. Si no se especifica BLOCK CONTAINS, el sistema determinará el número de registros que se han de bloquear. En el caso de archivos de DISKETTE, el sistema determinará siempre el número de registros que se han de bloquear.
2. Si *BLK se especifica con BLOCK CONTAINS y se dan **todas** las condiciones siguientes:
   a. ACCESS IS SEQUENTIAL o ACCESS IS DYNAMIC están especificadas en el archivo.
   b. El archivo se abre sólo para INPUT o OUTPUT en dicho programa.
   c. El archivo se asigna a DISK, DATABASE, DISKETTE o TAPEFILE.

   En la organización RELATIVE, no se efectúa agrupación para OPEN OUTPUT.

   La cláusula BLOCK CONTAINS controla el número de registros que se han de bloquear. En el caso de archivos de DISKETTE, el sistema determinará siempre el número de registros que se han de bloquear.

Incluso si se cumplen todas las condiciones anteriores, ciertas restricciones OS/400 pueden provocar la imposibilidad de procesar bloqueos y desbloqueos. En estos casos, no se producirán mejoras en el rendimiento.

Si se está utilizando archivos de organización accedidos e indexados de forma dinámica, se puede utilizar READ PRIOR y READ NEXT para realizar agrupaciones. Al utilizar READ PRIOR y READ NEXT para realizar agrupaciones, no se puede cambiar la dirección mientras aún queden registros en la agrupación. Para borrar los registros de una agrupación, especifique una operación al azar (por

ejemplo, READ o START), o utilice una operación secuencial READ FIRST o READ LAST.

Si se produce un cambio de dirección ilegal, el resultado será un estado de archivo 9U. No será posible ninguna E/S más hasta que el archivo se cierre y se vuelva a abrir.

El bloqueo puede alterarse temporalmente en tiempo de ejecución especificando `SEQONLY(*NO)` para el mandato OVRDBF.

Si se utiliza BLOCK CONTAINS en los archivos de base de datos, y si el factor de bloqueo cero se especifica o se calcula, el sistema determinará el factor de bloqueo.

Hay algunos casos en los que el factor de bloqueo especificado por el usuario puede cambiarse. Consulte la publicación *Guía para la Base de Datos* para más información acerca de estas situaciones.

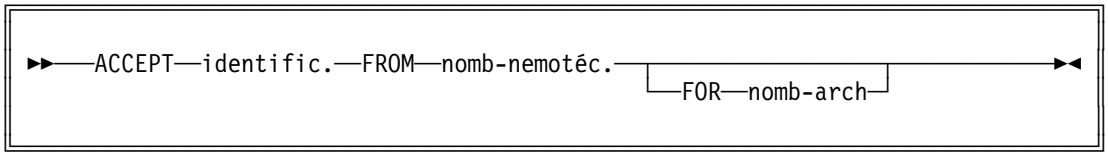
El área de realimentación de E/S contendrá el número de registros en un bloque en el que se grabe o se lea dicho bloque de registros. El área I/O-FEEDBACK no se actualiza después de cada lectura o grabación de archivos en la que COBOL agrupe y desagrupe varios registros. La actualización se produce cuando se lee o se graba el siguiente bloque. Consulte el apartado “I/O FEEDBACK” en la publicación *COBOL/400 Reference* para más información.

Para los archivos de base de datos, puede que no vea los cambios cuando se producen, si los cambios se producen en programas distintos. Para obtener una descripción del efecto de la agrupación en los cambios en archivos de base de datos, consulte el análisis sobre el proceso sólo secuencial en la publicación *Guía para la Base de Datos*.

Estado de Archivos y Áreas de Realimentación

Para transferir datos (áreas OPEN-FEEDBACK o I-O-FEEDBACK) asociados con un archivo abierto a un identificador, utilice el formato siguiente:

Instrucción ACCEPT – Formato 3 – Área de realimentación



Consulte la sección “Instrucción ACCEPT” de la publicación *COBOL/400 Reference* para más información acerca de la especificación de esta instrucción. Consulte la sección “Formatos de datos de atributos” de la publicación *COBOL/400 Reference* para más información acerca de las áreas OPEN-FEEDBACK e I-O-FEEDBACK.

Consulte, asimismo, la publicación *Guía para la Gestión de Datos* para obtener más información acerca de OPEN-FEEDBACK e I-O-FEEDBACK y del diseño y la descripción de las áreas de datos contenidas en las áreas de realimentación.

Cuando se especifica la cláusula FILE STATUS, el sistema traslada un valor al ítem de datos de clave de estado después de cada solicitud de entrada/salida que, explícita o implícitamente, se refiera a dicho archivo. Este valor de dos caracteres indica el estado de ejecución de la instrucción. Cuando el compilador genera el código para bloquear los registros de salida o desbloquear los registros de entrada, los valores del estado del archivo causados por excepciones del OS/400 sólo se establecen cuando se procesa el bloque. Para más información acerca de la agrupación de registros, consulte el apartado “Desbloqueo de Registros de Entrada y Bloqueo de Registros de Salida” en la página 107.

El área I/O-FEEDBACK no se actualiza después de cada lectura o grabación de archivos en la que el COBOL agrupe y desagrupe varios registros.

Es posible que no vea los cambios de los archivos de base de datos en el momento en que se producen, si dichos cambios se realizan en programas distintos. Para obtener una descripción del efecto de la agrupación en los cambios en archivos de base de datos, consulte el análisis sobre el proceso sólo secuencial en la publicación *Guía para la Base de Datos*.

Descripciones de Archivos

Todos los archivos en el sistema AS/400 se definen en el sistema operativo OS/400. La amplitud hasta la que pueden definirse los archivos presenta las siguientes diferencias:

- Un **archivo descrito por programa** se describe a nivel de campo dentro del programa COBOL en la División de Datos. La descripción del archivo para el sistema operativo incluye la información acerca del tipo de archivo y de la longitud de los registros en el archivo.
- Un **archivo descrito externamente** se describe a nivel de campo para el sistema operativo mediante el IDDU, los mandatos SQL/400* o las DDS. Si crea un archivo (por ejemplo, utilizando el mandato CRTPF) sin especificar las DDS para éste, dicho archivo todavía tendrá una descripción de campo. El campo único tendrá el mismo nombre que el archivo, y tendrá la longitud de registro especificada en el mandato de crear.

  La descripción incluye la información acerca del tipo de archivo, como por ejemplo de base de datos o de dispositivo, y una descripción de cada campo y sus atributos. El archivo debe crearse antes de que se compile el programa.

Tanto los archivos descritos externamente como los archivos descritos por programa deben definirse en el programa COBOL dentro de las secciones INPUT-OUTPUT SECTION y FILE SECTION. Las descripciones de registros en la sección FILE SECTION para los archivos descritos externamente pueden definirse con la instrucción COPY de Formato 2.

Las funciones dependientes de dispositivo, como por ejemplo el control de formularios, no las obtiene la operación COPY de Formato 2. Sólo se obtienen las descripciones a nivel de campo.

Cuando se especifica EXTERNALLY-DESCRIBED-KEY como RECORD KEY, los campos que componen RECORD KEY se obtienen también en las DDS.

Para más información acerca de la instrucción COPY de Formato 2, consulte la Figura 37 en la página 118 y el texto que la acompaña.

Nota: El proceso real del archivo dentro de la División de Procedimientos es el mismo, tanto si el archivo está descrito externamente como si está descrito por programa.

Archivos Descritos por Programa

Los registros y campos para un archivo descrito por programa se describen mediante la codificación de descripciones de registro en la Sección de Archivo del programa COBOL en lugar de utilizar la instrucción COPY de Formato 2.

El archivo debe existir en el sistema antes de que el programa pueda ejecutarse, excepto cuando utilice la creación dinámica de archivos, especificando GENOPT (*CRTF) en el mandato CRTCBLPGM. Para más información, consulte la descripción del parámetro GENOPT en la página 23 o la instrucción OPEN en la *COBOL/400 Reference*. Para crear un archivo, utilice uno de los mandatos Crear Archivo, que podrá encontrar en la publicación *CL Reference*.

Las DDS pueden utilizarse junto con los mandatos de Crear Archivo. En un archivo indexado COBOL, debe crearse una vía de acceso por claves. Especifique una clave en las DDS cuando se cree el archivo. La clave de registro en COBOL debe coincidir con la clave definida cuando se creó el archivo.

Archivos Descritos Externamente

Los archivos descritos externamente ofrecen las siguientes ventajas en comparación con los archivos descritos por programa:

- Menos codificación en programas COBOL. Si varios programas utilizan el mismo archivo, los campos pueden definirse una sola vez en el sistema operativo, y luego pueden utilizarlos todos los programas. Esto elimina la necesidad de codificar una descripción de registro para cada programa que utiliza el archivo.
- Menos actividad de mantenimiento cuando se cambia el formato de registro del archivo. Podrá actualizar con frecuencia los programas cambiando el formato de registro del archivo y recompilando a continuación los programas que utilizan el archivo cambiando cualquier codificación en el programa.
- Mejor documentación. Los programas que utilizan los mismos archivos utilizan formatos de registro y nombres de campos coherentes.
- Cualquier edición que ha de procesarse en archivos de salida descritos externamente pueden especificarse en las DDS.

La descripción externa de un archivo incluye:

- Las especificaciones del formato de registro que contenga una descripción de los campos en un registro
- Las especificaciones de vía de acceso que describen cómo se recuperan los registros.

Estas especificaciones vienen de la descripción de archivos externos y del mandato OS/400 que utilice el usuario para crear el archivo.

Podrá utilizar un archivo descrito externamente en un programa convirtiéndolo en un archivo descrito por programa (codificando la descripción del registro en el fuente). En este caso, el compilador no copia la descripción externa a nivel de campo del archivo en tiempo de compilación. Ello le resultará útil durante las conversiones, puesto que un programa ya existente podrá utilizar un archivo descrito por programa a la vez que un programa nuevo utiliza un archivo descrito externamente para referirse al mismo archivo.

La Figura 31 muestra el modo en que los programas COBOL pueden relacionarse con los archivos del sistema AS/400, utilizando las descripciones externas del archivo de las DDS.

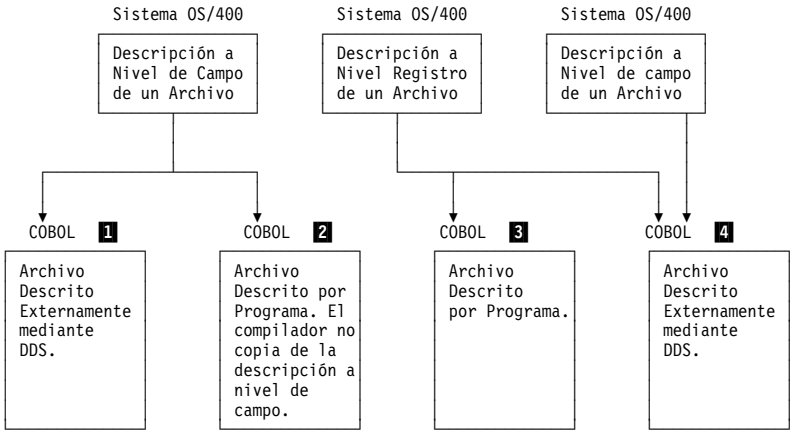


Figura 31. Ejemplo que Muestra Cómo se Relaciona el COBOL con Archivos del AS/400

1 El programa COBOL utiliza la descripción a nivel de campo de un archivo definido en el sistema operativo. El usuario de COBOL codificó una instrucción COPY de Formato 2 para la descripción de registro. En tiempo de compilación, el compilador copia de la descripción externa a nivel de campo y la convierte en una descripción de registro COBOL sintácticamente correcta. El archivo debe existir en tiempo de compilación.

2 Un archivo descrito externamente se utiliza como archivo descrito por programa en el programa COBOL. Toda la descripción de registro para el archivo se codifica en el programa COBOL. Este archivo no tiene que existir en tiempo de compilación.

3 Se describe un archivo en el sistema operativo sólo hasta el nivel de registro. Toda la descripción de registro debe codificarse en el programa COBOL. Este archivo no tiene que existir en tiempo de compilación.

4 Puede especificarse un nombre de archivo para tiempo de compilación, y puede especificarse un nombre de archivo distinto para tiempo de ejecución. Una instrucción COPY de Formato 2 genera la descripción de registro del archivo en tiempo de compilación. En tiempo de ejecución, puede utilizarse una lista de bibliotecas distinta o un mandato de alteración temporal del archivo de forma que el programa acceda a otro archivo. La descripción de archivo copiada en tiempo de compilación se utiliza para describir los registros de entrada utilizados en tiempo de ejecución.

Nota: Para los archivos descritos externamente, los dos formatos de archivo deben ser los mismos. De lo contrario, se produce un error de comprobación de nivel.

Especificaciones de Descripción de Datos (DDS)

Puede utilizar las Especificaciones de Descripción de Datos (DDS) para describir los archivos a nivel de campo en el sistema operativo. En las DDS, cada formato de registro en un archivo descrito externamente se identifica mediante un nombre de formato de registro exclusivo.

Las especificaciones del formato de registro describen los campos en un registro y la ubicación de los campos en un registro. Los campos se ubican en el registro en el orden especificado en las DDS. La descripción de campo generalmente incluye el nombre de campo, el tipo de campo (carácter, binario, decimal externo o decimal interno) y la longitud de campo (incluido el número de posiciones decimales en un campo numérico). En lugar de especificarse en el formato de registro para un archivo físico o lógico, los atributos de campo pueden definirse en un archivo de referencia de campo. (Consulte la Figura 32 en la página 113).

Las claves para un formato de registro se especifican en las DDS. Cuando utiliza la instrucción COPY de Formato 2, se genera una tabla de comentarios en el listado del programa fuente mostrando cómo se definen las claves en las DDS para el formato.

Además, las palabras clave de las DDS pueden utilizarse para:

- Especificar los códigos de edición para un campo (EDTCDE)
- Especificar palabras de edición para un campo (EDTWRD)
- Especificar que no se permiten los valores de clave duplicados para el archivo (UNIQUE)
- Especificar una descripción de texto para un formato de registro o un campo (TEXT).

Consulte la publicación *DDS Reference* para obtener una lista completa de las palabras clave de las DDS válidas para un archivo de base de datos.

IBM International Business Machines

AS/400 ESPECIFICACIONES DE DESCRIPCION DE DATOS

GX21-8891-0 UM/0850*
Impreso en España
*El número de hojas por área de relleno puede variar ligeramente.

Archivo		Instrucciones de Grabación	Signo								Descripción	Página de
Programador	Fecha		Teclа									

A

Número de Secuencia	Tipo de Formulario	Condicionamiento: And/Or/Comment. (A/O/*)	Condicionamiento: No (N)	Condicionamiento: Nombre Condición: Indicador	Condicionamiento: Nombre Condición: No (N)	Condicionamiento: Nombre Condición: Indicador	Condicionamiento: Nombre Condición: No (N)	Condicionamiento: Nombre Condición: Indicador	Tipo Nombre o Base (b/R/H/I/K/S/O)	Reservado	Nombre	Referencia (R)	Longitud	Tipo Dato/Desplazamiento Teclado	Posiciones Decimales	Utilización (b/O/I/B/H/M/N/P)	Ubicación: Línea	Ubicación: Pos	Funciones
	A	*	*	FLDREF						R	DSTREF	T	EXTO CONSULTA CAMPO DE APLICACIONES DE DISTRIBUCIÓN						
	A										DSTREF								TEXT('CONSULTA CAMPO DISTRIBUCION')
	A	*		CAMPOS USUALMENTE UTILIZADOS COMO CONSULTA															
1	A										BASDAT		6		0				EDTCDE(Y)
	A																		TEXT('CAMPO DATOS BASE)
	A	*		CAMPOS UTILIZADOS POR ARCH. MAESTRO CLIENTES															
2	A										CUST		5						CHECK(MF)
	A																		COLHDG('CLIENTE' 'NUMERO')
	A										NAME		20						COLHDG('NOMBRE DE CLIENTE')
3	A										ADDR	R							REFFLD(NOMBRE)
	A																		COLHDG('DIRECCION DE CLIENTE')
	A										CITY	R							REFFLD(NOMBRE)
	A																		COLHDG('CIUDAD DE CLIENTE')
2	A										STATE		2						CHECK(MF)
	A																		COLHDG('PROVINCIA')
	A										SRHCOD		6						CHECK(MF)
	A																		COLHDG('BUSQUEDA' 'CODIGO')
	A																		TEXT('CODIGO BUSQUEDA NUM CLIENTE')
2	A										ZIP		5		0				CHECK(MF)
	A																		COLHDG('CODIGO' 'POSTAL')
4	A										CUSTYP		1		0				RANGE(1 5)
	A																		COLHDG('CLIEN' 'TIPO')
	A																		TEXT('TIPO CLIENTE 1=GOB 2=PLAN 3=B+
	A																		US 4=PT 5=OTR')
5	A										ARBAL		8		2				COLHDG('CUENT POR COBRAR' 'SALDO')
	A																		EDTCDE(J)
6	A										ORDBAL	R							REFFLD(ARBAL)
	A																		COLHDG('IMP EN A/R' 'ARCH PEDIDOS')
	A										LSTAMT	R							REFFLD(ARBAL)
	A																		COLHDG('ULT' 'IMP' 'PAGADO')
7	A																		TEXT('ULT IMPORTE PAGADO EN A/R')
	A																		
	A																		
	A																		
	A																		

Figura 32. Ejemplo de Archivo de Referencia de Campos

Este ejemplo de un archivo de referencia de campos muestra las definiciones de los campos que utiliza el archivo CUSMSTL (lógico maestro de clientes), que se muestra en la Figura 33 en la página 115. El archivo de referencia de campos normalmente contiene las definiciones de campos que utilizan otros archivos. El texto siguiente describe algunas entradas para este archivo de referencia de campos.

1 El campo BASDAT se edita mediante el código de edición Y, tal como se indica mediante la palabra clave EDTCDE (Y). Si este campo se utiliza en un archivo de salida descrito externamente para un programa COBOL, el campo generado por COBOL es compatible con el tipo de datos especificado en las DDS. El campo se edita cuando se escribe el registro. Cuando el campo se utiliza en un archivo de salida descrito por programa, la compa-

tibilidad con los campos DDS en el archivo es responsabilidad del usuario. Cuando las DDS no se utilizan para crear el archivo, la edición adecuada del campo en el programa COBOL también es responsabilidad del usuario.

2 La entrada CHECK(MF) especifica que el campo es un campo de relleno obligatorio cuando se introduce desde una estación de trabajo de pantalla. El relleno obligatorio implica que todos los caracteres del campo deben introducirse desde la estación de trabajo de pantalla.

3 Los campos ADDR y CITY comparten los mismos atributos especificados para el campo NAME, tal y como se indica en la palabra clave REFFLD.

4 La palabra clave RANGE, que se especifica para el campo CUSTYP, asegura que los únicos números válidos que pueden entrarse en este campo desde una estación de trabajo de pantalla van del 1 al 5.

5 La palabra clave COLHDG proporciona un encabezamiento de columna para el campo si se utiliza mediante las Herramientas para el Desarrollo de Aplicaciones (Appl Dev Tools).

6 El campo ARBAL se edita mediante el código de edición J, como se indica mediante la palabra clave EDTCDE(J).

7 Se proporciona una descripción de texto (palabra clave TEXT) para algunos campos. La palabra clave TEXT se utiliza a efectos de documentación y aparece en varios listados.

Especificaciones COBOL para Archivos Descritos Externamente Utilizando las DDS

Puede incorporar la descripción del archivo en el programa codificando la instrucción COPY de Formato 2. La información de la descripción externa se recupera mediante el compilador COBOL, y se genera la estructura de datos COBOL.

Las páginas siguientes proporcionan ejemplos de la utilización de las DDS y el código COBOL que resultará de la utilización de una instrucción COPY de Formato 2. Consulte el apartado “Instrucción COPY de Formato 2 (Opción DD, DDR, DDS o DDSR)” en la página 118 para una descripción detallada de la instrucción COPY de Formato 2.

- La Figura 33 en la página 115 muestra las DDS para un archivo lógico y la Figura 34 en la página 116 muestra el código COBOL generado.
- La Figura 35 en la página 117 describe el mismo archivo pero incluye la palabra clave ALIAS, y la Figura 36 en la página 118 muestra el código COBOL generado.

El proceso de archivo real dentro de la División de Procedimientos es el mismo para los archivos descritos por programa y descritos externamente.

IBM International Business Machines

AS/400 ESPECIFICACIONES DE DESCRIPCION DE DATOS

GX21-9891-0 UM/050*
Impreso en España
*El número de hojas por área de relleno puede variar ligeramente.

Archivo		Instrucciones de Grabación	Signo	Descripción	Página de
Programador	Fecha		Tecla		

A

Número de Secuencia	Tipo de Formulario	And/Or/Coment. (A/O/*)	No (N)	Indicador	No (N)	Indicador	No (N)	Indicador	Tipo Nombre o Espec (b/R/H/J/K/S/O)	Reservado	Nombre	Referencia (R)	Longitud	Tipo Datos/Desplazamiento Teclado	Posiciones Decimales	Utilización (b/O/I/B/H/M/N/P)	Línea	Pos	Funciones
1	A	*	*	LOGICAL							CUSMSTL	ARCHIVO MAESTRO DE CLIENTE							
	A																	**2**	UNIQUE
	A	*						**3**	R		CUSREC								PFILE(CUSMSTP)
	A																		TEXT('REGISTRO MAESTRO CLIENTE')
	A										CUST								
	A										NAME								
	A										ADDR								
	A										CITY								
	A										STATE								
	A										ZIP								
	A										SRHCOD								
	A										CUSTYP								
	A										ARBAL								
	A										ORDBAL								
	A										LSTAMT								
	A										LSTDAT								
	A										CRDLMT								
	A										SLSYR		**5**						
	A										SLSLYR								
	A							**4**	K		CUST								
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		

Figura 33. Ejemplo de las Especificaciones de Descripción de Datos para un Archivo Lógico

1 Se define un archivo lógico para procesar el archivo físico maestro de clientes (CUSMSTP) y denominado CUSMSTL.

2 La palabra clave UNIQUE indica que no se permiten los valores de clave duplicados para este archivo.

3 Se define un formato de registro (CUSREC) para el archivo CUSMSTL, que se basa en el archivo físico CUSMSTP.

4 El campo CUST se identifica como campo de clave para este archivo.

5 Si no se especifican los atributos de campo (como, por ejemplo, la longitud, tipo de datos y posiciones decimales) en las DDS para un archivo lógico, los atributos se obtienen del campo correspondiente en el archivo físico. Cualquier atributo de campo especificado en las DDS para el archivo lógico

altera temporalmente los atributos para el campo correspondiente en el archivo físico. La definición de los campos en el archivo físico puede referirse a un archivo de referencia de campo. Un archivo de referencia de campos es un archivo de descripción de datos que consta de nombres de campos y de sus definiciones, como el tamaño y el tipo. Cuando se utiliza un archivo de referencia de campos, los mismos campos utilizados en varios formatos de registros tienen que definirse sólo una vez en el archivo de referencia de campos. Para más información acerca de un archivo de referencia de campos, consulte la publicación *Guía para la Base de Datos*.

La Figura 32 en la página 113 muestra un ejemplo de un archivo de referencia de campos que define los atributos de los campos utilizados en el archivo de base de datos. Consulte la publicación *Guía para la Base de Datos* para obtener más información acerca de los archivos de referencia de campos.

```
 01  CUS-MASTER.
     COPY DDS-CUSREC OF CUSLIB-CUSTMAST.
*FORMATO E-S: CUSREC DE ARCHIVO CUSTMAST DE BIBL. CUSLIB       CUSREC
*                   REGISTRO MAESTRO CLIENTE                  CUSREC
*LAS DEFINICIONES DE CLAVE PARA FORMATO DE REGISTRO CUSREC     CUSREC
*NUMERO   NOMBRE  RECUPERAC.  TIPO    SECALT                  CUSREC
*0001    CLIENTE  ASCENDENTE  AN      NO                      CUSREC
    05   CUSREC.                                              CUSREC
         06 CUST     PIC X(5).                                CUSREC
*                NUMERO CLIENTE                               CUSREC
         06 NAME     PIC X(20).                               CUSREC
*                NOMBRE CLIENTE                               CUSREC
         06 ADDR     PIC X(20).                               CUSREC
*                DIRECCION CLIENTE                            CUSREC
         06 CITY     PIC X(20).                               CUSREC
*                CIUDAD CLIENTE                               CUSREC
         06 STATE    PIC X(2).                                CUSREC
*                ABREVIATURA PROVINCIA                        CUSREC
         06 ZIP      PIC S9(5)     COMP-3.                    CUSREC
*                CODIGO POSTAL                                CUSREC
         06 SHRCOD   PIC X(6).                                CUSREC
*                CODIGO BUSQUEDA NOMBRE CLIENTE               CUSREC
         06 CUSTYP   PIC 9(1).                                CUSREC
*                TIPO CLIENTE                                 CUSREC
         06 ARBAL    PIC S9(6)V9(2)     COMP-3.               CUSREC
*                SALDO CUENTAS PEND.                          CUSREC
```

Figura 34. Ejemplo de los Resultados de la Instrucción COPY de Formato 2 (DDS)

IBM International Business Machines

AS/400 ESPECIFICACIONES DE DESCRIPCION DE DATOS

GX21-9891-0 UM/050*
Impreso en España
*El número de hojas por área de relleno puede variar ligeramente.

Archivo		Instrucciones de Grabación	Signo	Descripción	Página de
Programador	Fecha		Tecla		

A

Número de Secuencia (1-5)	Tipo de Formulario (6)	And/Or/Coment. (A/O/*) (7)	No (N) (8)	Condicionamiento: Nombre Condición – Indicador / No (N) / Indicador / No (N) / Indicador (9-16)	Tipo Nombre o Espec (b/R/H/J/K/S/O) (17)	Reservado (18)	Nombre (19-28)	Referencia (R) (29)	Longitud (30-34)	Tipo Datos/Desplazamiento Tecleado (35)	Posiciones Decimales (36-37)	Utilización (b/O/I/B/H/M/N/P) (38)	Ubicación: Línea (39-41)	Ubicación: Pos (42-44)	Funciones (45-80)
	A	*	*	LOGICAL			CUSMSTL		ARCHIVO MAESTRO CLIENTES						
	A														UNIQUE
	A	*			R		CUSREC								PFILE(CUSMSTP)
	A														TEXT('REGISTRO MAESTRO CLIENTE')
	A						CUST								ALIAS(NUMERO_CLIENTE)
	A						NAME							1	ALIAS(NOMBRE_CLIENTE)
	A						ADDR								ALIAS(DIRECCION)
	A						CITY								
	A						STATE								
	A						ZIP								
	A						SRHCOD								ALIAS(CODIGO_BUSQUEDA)
	A						CUSTYP								ALIAS(TIPO_CLIENTE)
	A						ARBAL								ALIAS(SALDO_CUENTAS_POR COBRAR)
	A						ORDBAL								
	A						LSTAMT								
	A						LSTDAT								
	A						CRDLMT								
	A						SLSYR								
	A						SLSLYR								
	A				K		CUST								
	A														
	A														
	A														
	A														
	A														
	A														
	A														
	A														
	A														
	A														
	A														
	A														
	A														
	A														
	A														

Figura 35. Ejemplo de las Especificaciones de Descripción de Datos con ALIAS

1 Este es el nombre asociado con la palabra clave ALIAS, que se incluirá en el programa. Disponible en la opción ALIAS de DDS, un alias es un nombre alternativo que sirve para que un nombre de datos de hasta 30 caracteres se incluya en un programa COBOL/400.

```
 01  CUS-MASTER.
     COPY DD-CUSREC OF CUSLIB-CUSTMAST.
*FORMATO E-S: CUSREC DESDE CUSTMAST DE BIBLIOTECA CUSLIB       CUSREC
*                  REGISTRO MAESTRO CLIENTE                   CUSREC
*LAS DEFINICIONES DE CLAVE PARA FORMATO DE REGISTRO CUSREC     CUSREC
*NUMERO        NOMBRE      RECUPERAC.  TIPO         SECALT     CUSREC
*0001     NUMERO-CLIENTE   ASCENDENTE  AN           NO        CUSREC
                                                              CUSREC
    05  CUSREC.                                              CUSREC
        06 CUSTOMER-NUMBER  PIC X(5).                         CUSREC
*               NUMERO CLIENTE                                CUSREC
        06 CUSTOMER-NAME    PIC X(20).                        CUSREC
*               NOMBRE CLIENTE                                CUSREC
        06 ADDRESS          PIC X(20).                        CUSREC
*               DIRECCION CLIENTE                             CUSREC
        06 CITY             PIC X(20).                        CUSREC
*               CIUDAD CLIENTE                                CUSREC
        06 STATE            PIC X(2).                         CUSREC
*               ABREVIATURA PROVINCIA                         CUSREC
        06 ZIP              PIC S9(5)    COMP-3.              CUSREC
*               CODIGO POSTAL                                 CUSREC
        06 SEARCH-CODE      PIC X(6).                         CUSREC
*               CODIGO BUSQUEDA NOMBRE CLIENTE                CUSREC
        06 CUSTOMER-TYPE    PIC 9(1)                          CUSREC
*               TIPO CLIENTE                                  CUSREC
        06 ACCT-REC-BALANCE PIC S9(6)V9(2)   COMP-3.          CUSREC
*               SALDO CUENTAS PEND.                           CUSREC
```

Figura 36. Ejemplo de los Resultados de la Instrucción COPY de Formato 2 (DD) con la Palabra Clave ALIAS

Ampliación de IBM

Instrucción COPY de Formato 2 (Opción DD, DDR, DDS o DDSR)

Para una información general acerca de los formatos de la instrucción COPY, vea la publicación *COBOL/400 Reference*.

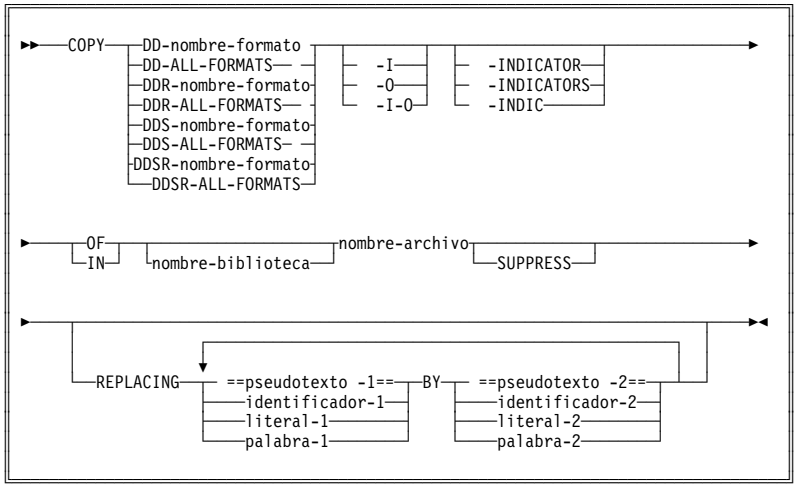


Figura 37. Instrucción COPY – Formato 2 – Conversión DDS

Puede utilizar la instrucción COPY de Formato 2 (opción DD, DDR, DDS o DDSR) para crear instrucciones de División de Datos COBOL para describir un archivo existente en el sistema. Estas descripciones se basan en la versión del archivo existente en tiempo de compilación. No utilizan ninguna sentencia fuente DDS del archivo. Consulte la sección “Instrucción COPY” de la publicación *COBOL/400 Reference* para más información acerca de la instrucción COPY

Nota: El término *instrucción COPY de Formato 2* indicará la instrucción COPY de Formato 2 (opción DD, DDR, DDS o DDSR) a través de este manual.

La instrucción COPY de Formato 2 sólo se utiliza en la División de Datos. Debe asegurarse que un ítem a nivel de grupo que tiene un número de nivel menor que 05 precede a la instrucción.

La opción DD se utiliza para hacer referencia a los nombres ALIAS (alternativos). La especificación de un nombre de ALIAS en las DDS permite que un nombre de datos de hasta 30 caracteres se incluya en el programa COBOL.

Cuando se utiliza la opción DD, cualquier nombre de ALIAS presente sustituye a los nombres de campos de las DDS correspondientes. Todos los subrayados en los nombres ALIAS se convierten en guiones antes de producirse alguna sustitución.

La opción DDR hace todo lo que hace la opción DD. También copia los nombres de campo de formato DDS internos, sustituyendo los caracteres COBOL no válidos (@, #, $ y _) por los caracteres COBOL válidos A, N, D y - según proceda. Esta opción también elimina cualquier subrayado desde el final de los nombres de campo.

La opción DDS copia los nombres de campos de formato interno DDS. Para ver ejemplos de claves y nombres de claves que pueden generarse cuando utilice la opción DDS de la instrucción COPY de Formato 2, consulte las páginas 127 a 133.

La opción DDSR hace todo lo que hace la opción DDS. También copia los nombres de campo de formato DDS internos, sustituyendo los caracteres COBOL no válidos (@, #, $, y _) por los caracteres COBOL válidos A, N, D, y -, según proceda. Esta opción también elimina cualquier subrayado desde el final de los nombres de campo.

El ejemplo siguiente muestra el efecto de la opción DDR o DDSR en nombres de campos COBOL no válidos:

Nombre de campo original	**Nombre de campo modificado**
FLD_A	FLD-A
NUMBER#1	NUMBERN1
POINT@7	POINTA7
BALANCE$	BALANCED

Cuando la cláusula RECORD KEY especifica EXTERNALLY-DESCRIBED-KEY, sólo puede copiarse un formato a la vez bajo una FD. Por ejemplo, si todos los formatos de un archivo se copian bajo una FD, no puede especificarse otra instrucción COPY de Formato 2 para el mismo archivo bajo dicha FD.

El nombre de formato es el nombre de la definición de formato de registro DDS que debe convertirse en entradas de descripción de datos COBOL. El nombre de formato debe seguir las reglas para la formación de cualquier nombre COBOL/400.

Si no se se especifica -I ni -O, se asume -I-O.

Si se especifica el nombre de formato sin el atributo Indicador, y los formatos -I y -O deben generarse, cada formato de registro se genera como una redefinición de un ítem elemental 05 definido como:

- El tamaño del formato de registro más grande que se generará.

Si se especifica ALL-FORMATS (sin el atributo Indicador), cada formato de registro se genera como una redefinición de un ítem elemental 05 definido como:

- El tamaño del formato de registro más grande en el archivo, si aparece la instrucción COPY en la Sección de Archivos
- El tamaño del formato de registro más grande que se generará, si aparece la instrucción COPY fuera de la Sección de Archivos.

Cuando se especifica el atributo Indicador, no se produce ninguna redefinición. En su lugar, cada formato genera una estructura de datos separada.

Puede encontrarse mas información acerca del atributo de Indicador en la sección “Atributo de Indicador de la Instrucción COPY de Formato 2” en la página 124.

El nombre de biblioteca es opcional. Si no se especifica, se utiliza la lista de bibliotecas de trabajo actual como valor por omisión.

El nombre de archivo es el nombre de un archivo del sistema AS/400. Las entradas DDS generadas representan el formato de registro definido en el archivo. El archivo debe crearse antes de que se compile el programa.

Si el archivo es un archivo de base de datos, se genera un único formato de E-S.

Para el resto de tipos de archivos, la descripción generada varía de la manera siguiente:

- Si se especifica -I, las entradas de descripción de datos generadas contienen:
  - Los campos de entrada y entrada/salida para un formato que no sea de subarchivo, o
  - Los campos de entrada, salida, y entrada/salida para un formato de subarchivo.
- Si se especifica -O, las entradas de descripción de datos generadas contienen:
  - Los campos de salida y entrada/salida para un formato que no sea de subarchivo, o
  - Los campos de entrada, salida, y entrada/salida para un formato de subarchivo.

Nota: Los registros de subarchivo sólo con campos de salida o entrada/salida, sin indicadores de campo especificados, generan formatos I/O.

Si se necesita un área de almacenamiento separada en WORKING-STORAGE para cada formato, debe especificarse una instrucción COPY individual para cada formato.

Por ejemplo, si asume que el archivo CUSTMASTER contiene dos formatos CUSADR y CUSTDETL, pueden especificarse las instrucciones COPY siguientes:

```
    SELECT FILE-X
    ASSIGN TO DATABASE-CUSTMASTER.
⋮
FD   FILE-X
     LABEL RECORDS ARE STANDARD.
01   FILE-X-RECS.
     COPY DDS-ALL-FORMATS OF
        CUSTMASTER-QGPL.   (Vea Nota 1.)
⋮
WORKING-STORAGE SECTION.
01   ADR-REC.
     COPY DDS-CUSTADR OF
        CUSTMASTER.     (Vea Nota 2.)
01   DETAIL-REC.
     COPY DDS-CUSTDETL OF
        CUSTMASTER.     (Vea Nota 2.)
```

Notas:

1. Esta instrucción COPY sólo genera un área de almacenamiento para todos los formatos.
2. Estas instrucciones COPY generan áreas de almacenamiento separadas.

Indicadores

Los indicadores son ítems de datos Booleanos que pueden tener los valores B"0" ó B"1".

Cuando define un formato de registro para un archivo utilizando las DDS, puede condicionar las opciones utilizando indicadores; los indicadores también pueden utilizarse para reflejar respuestas particulares. Estos indicadores se conocen como OPTION y RESPONSE, respectivamente. Los indicadores de opción proporcionan opciones tales como espaciado, subrayado y permiso o petición de transferencia de datos desde un programa a una impresora o dispositivo de pantalla. Los indicadores de respuesta proporcionan información de respuesta a un programa desde un dispositivo, como por ejemplo las teclas de función pulsadas por un usuario de estación de trabajo, y si se han introducido los datos.

Pueden utilizarse indicadores con archivos TRANSACTION y FORMATFILE, pero nunca con archivos de base de datos.

Estructuras de Datos Generadas

Las diferentes palabras clave DDS influyen en la creación de los diversos tipos de estructuras de datos.

Estructuras de Nivel de Formato (Registro)

Al principio de cada formato, se genera una tabla de comentarios en el listado del programa fuente. Estos comentarios proporcionan detalles de los archivos utilizados durante la compilación del programa. Si hay claves de registros para el archivo, los comentarios también se generan para mostrar cómo se definen las claves en las DDS. A continuación aparece un listado de las entradas de clave de registro que aparecen en la tabla y el encabezamiento de la tabla.

Encabezamiento	Posible entrada
NÚMERO	Número del campo de clave
NOMBRE	Nombre del campo de clave
RECUPERACION	ASCENDING, DESCENDING
TIPO	ZONE, DIGIT, SIGNED, ABSVAL, AN (alfanumérico), N (numérico) J (ítem DBCS), DDS - L (fecha), DDS - T (hora), DDS - Z (indicación de la hora), DDS - G (gráfico de longitud variable), VARLEN (carácter de longitud variable o ítem DBCS entre paréntesis), G VARLEN (gráfico DBCS de longitud variable)
SECALT	NO, YES

Si se requiere la redefinición para permitir la generación de varios formatos, se genera un nombre a nivel de grupo de la forma siguiente:

```
05 nombre-archivo-RECORD
   PIC X(tamaño del mayor registro).
```

Para cada formato, se asigna un nombre a nivel de grupo de la manera siguiente:

- INPUT

  05 nombre-formato-I

- OUTPUT

  05 nombre-formato-O

- Formato I/O

  05 nombre-formato

Estructuras de Campos de Datos

Los nombres de campo, las definiciones PICTURE y las cláusulas de utilización numérica se obtienen directamente de los nombres de campos de formato interno DDS (o nombres ALIAS en el caso de la opción DD o DDR) y las representaciones de tipo de datos. Los nombres de campo y las definiciones PICTURE se construyen de la manera siguiente:

```
06  nombre-campo PIC
```

Nota: Consulte la Figura 38 en la página 123 para la definición COBOL PICTURE adecuada.

Tabla 3. Estructura del campo de datos

DDS		DIVISIÓN DE DATOS COBOL n=longitud total de campo (pos. DDS 30-34) m=número de decimales (pos. DDS 36 y 37)	
Tipo de datos (pos. 35)	Formatos	Si pos. DDS 36 y 37 están en blanco	Si pos. DDS 36 y 37 no están en blanco
ARCHIVOS FÍSICOS, LÓGICOS, DE IMPRESORA Y DE COMUNICACIONES			
b(En blanco)	Por omisión	PIC X(n) [2]	PIC S9(n-m)V9(m)
P	Decimal empaquetado	PIC S9(n) COMP-3	PIC S9(n-m)V9(m) COMP-3
S	Caracteres numéricos decimales/marcados con zona	PIC S9(n)	PIC S9(n-m)V9(m)
B	Binario	PIC S9(n) COMP-4	PIC S9(n-m)V9(m) COMP-4
F	Coma flotante[1]		
	precisión simple	PIC 9(5) COMP-4	PIC 9(5) COMP-4
	precisión doble	PIC 9(10) COMP-4	PIC 9(10) COMP-4
A	Carácter	PIC X(n) [2]	—
H	Datos hexadecimales	PIC X(n)	—
L	Fecha [3]	PIC X(n)	—
T	Hora [3]	PIC X(n)	—
Z	Indicación horaria [3]	PIC X(n)	—
J	Datos únicamente DBCS	PIC X(n)	—
E	Cualquier tipo de datos DBCS	PIC X(n)	—
O	Datos DBCS abiertos	PIC X(n)	—
G	Datos DBCS gráficos	PIC X(2n) [2]	—
ARCHIVOS DE PANTALLA			
b(En blanco)	Por omisión	PIC X(n)	PIC S9(n-m)V9(m)
X	Sólo alfabético	PIC X(n)	—
N	Desplazamiento numérico	PIC X(n)	PIC S9(n-m)V9(m)
Y	Sólo numérico	—	PIC S9(n-m)V9(m)
I	Entrada de teclado no habilitada	PIC X(n)	PIC S9(n-m)V9(m)
W	Katakana	PIC X(n)	—
A	Desplazamiento alfanumérico	PIC X(n)	—
D	Sólo dígitos	PIC X(n)	PIC S9(n)
F	Coma flotante [1]		
	precisión simple	PIC 9(5) COMP-4	PIC 9(5) COMP-4
	precisión doble	PIC 9(10) COMP-4	PIC 9(10) COMP-4
M	Carácter sólo numérico	PIC X(n)	—
S	Desplazamiento numérico marcado	—	PIC S9(n-m)V9(m)
E	Cualquier tipo de datos DBCS	PIC X(n)	—
J	Sólo DBCS	PIC X(n)	—
O	Datos DBCS abiertos	PIC X(n)	—
G	Datos DBCS gráficos	PIC X(2n)	—

[1] COBOL considera los campos de coma flotante como de RELLENO. Véase "Campos de coma flotante".
[2] En DDS, si el campo tiene un atributo VARLEN, se añaden dos bytes adicionales al comienzo del campo.
[3] Ítems FILLER por omisión. Véase "Campos de Fecha, Hora y de indicación horaria".

Figura 38. Estructuras de Campos de Datos

Estructuras de Indicador

Si se solicitan los indicadores, y existen en el formato, se genera un nombre de grupo adicional (nivel 06) al principio de la estructura, seguido de las entradas (nivel 07) para los indicadores individuales relevantes.

```
06   nombre-formato(-I o -O)-INDIC.
     07  INxx PIC 1 INDIC xx.
```

donde xx es el número de indicador

Por ejemplo:

```
06   SAMPLE1-I-INDIC.
     07   IN01 PIC 1 INDIC 01.
     07   IN04 PIC 1 INDIC 04.
     07   IN05 PIC 1 INDIC 05.
     07   IN07 PIC 1 INDIC 07.
06   FLD1 PIC  ...  .
06   FLD2 PIC  ...  .
```

Atributo de Indicador de la Instrucción COPY de Formato 2

El atributo de indicador especifica si las entradas de descripción de datos se generan para los indicadores.

Si se especifica el atributo de Indicador, se generan las entradas de descripción de datos para los indicadores, pero no para campos de datos. Una entrada a nivel de grupo 05 se genera del siguiente modo:

- Si la copia (COPY) es para una única estructura (por ejemplo, COPY DDS-nombre-formato-INDIC)

  05 nombre-formato-I. (o -O, si es pertinente)

- Si la copia (COPY) es para varias estructuras (por ejemplo, COPY DDS-ALL-FORMATS-INDIC)

  05 nombre-archivo-RECORD.

Las entradas de descripción de datos que se generan se determinan según el atributo de utilización (I, O, o I-O) que se especifique o asuma en la instrucción COPY.

- Si se especifica ...I-INDICATOR..., se generan las entradas de descripción de datos para los indicadores de entrada (respuesta) para los indicadores utilizados en el área de registro de entrada.
- Si se especifica ...O-INDICATOR..., se generan las entradas de descripción de datos para los indicadores de salida (opción) para los indicadores utilizados en el área de registro de salida.
- Si se especifica o se asume ...I-O-INDICATOR..., se generan las entradas de descripción de datos separadas para indicadores de entrada y salida (respuesta y opción) de los indicadores utilizados en las áreas de registro de entrada y salida.

Si no se especifica el atributo Indicador, la generación de entradas de descripción de datos para indicadores dependerá de si el archivo tenía la palabra clave INDARA especificada en las DDS en el momento en que se creó.

- Si no se especificó INDARA, las entradas de descripción de datos se generan tanto para campos de datos como de indicadores.
- Si INDARA se especificó, las entradas de descripción de datos se generan solamente para campos de datos, y no para indicadores.

Generación de Formatos I/O

Cuando todas las descripciones de campo sean idénticas, y haya solicitado campos INPUT o OUTPUT implícita o explícitamente, sólo se genera un conjunto de descripciones de campo. Este tipo de descripción se anota con una línea de comentario que reza “FORMATO I-O: nombre-formato”. Ni -I ni -O se añaden al formato de registro.

Nota: Esto siempre ocurre en los archivos de base de datos porque todas las descripciones de campos de un archivo de base de datos son idénticas.

Por ejemplo:

```
 01 RCUSREC.
    COPY DDS-CUSREC-I OF CUSFILE.
*   FORMATO E-S: CUSREC DESDE ARCHIVO CUSFILE DE BIBL. CUSLIB  CUSREC
*   DEFINICIONES DE CLAVE PARA FORMATO DE REGISTRO CUSREC
*   NUMERO NOMBRE RECUPERACION TIPO SECALT
*    0001 ARBAL ASCENDING SIGNED NO
*    0002 AREACD DESCENDING ABSVAL NO
       05   CUSREC.
       06   ARBAL          PIC S9(7)V9(2)      COMP-3          CUSREC
       06   AREACD         PIC S9(3)           COMP-3.         CUSREC
       06   BOSTAZ         PIC X(1).                           CUSREC
       06   CNTCT          PIC X(15).                          CUSREC
       06   CRCHKZ         PIC S9(2).                          CUSREC
       06   CSTAT          PIC X(1).                           CUSREC
       06   CUSTNZ         PIC S9(6).                          CUSREC
       06   DLORD          PIC S9(6).                          CUSREC
       06   DSCPCZ         PIC S9(2)V9(3)      COMP-3.         CUSREC
       06   INDUS          PIC S9(2).                          CUSREC
       06   NAME1          PIC X(25).                          CUSREC
       06   NAME2          PIC X(25).                          CUSREC
       06   NAME3          PIC X(25).                          CUSREC
       06   NAME4          PIC X(25).                          CUSREC
       06   PHONE          PIC S9(7)           COMP-3.         CUSREC
       06   PRICIZ         PIC S9(2).                          CUSREC
       06   SHPINZ         PIC X(25).                          CUSREC
       06   SLSMAZ         PIC X(3).                           CUSREC
       06   TAXCDZ         PIC S9(2).                          CUSREC
       06   TERMSZ         PIC S9(2).                          CUSREC
```

Figura 39. Ejemplo de Copia DDS que Muestra Formatos I/O

Redefinición de Formatos

Preste especial atención a la cláusula REDEFINES que puede generarse para las frases ALL-FORMATS o -I-O. Debido a que todos los formatos se redefinen en la misma área (generalmente un área de almacenamiento intermedio), varios nombres de campo pueden describir la misma área de almacenamiento, y pueden producirse resultados imprevisibles si no se vuelve a inicializar todo el área del formato antes de cada operación de salida.

Los ítems de datos que están subordinados a los ítems de datos especificados en una instrucción MOVE CORRESPONDING no se corresponden y no se trasladan cuando contienen una cláusula REDEFINES o están subordinados a un ítem redefinitorio.

Para evitar la reinicialización, se pueden utilizar varias instrucciones COPY de Formato 2 que utilicen sufijos -I y -O para crear áreas de almacenamiento separadas en la Sección de Almacenamiento de Trabajo (Working Storage) de cada

formato o tipo de formato (entrada o salida). Con estos formatos de registro pueden utilizarse instrucciones READ INTO y WRITE FROM.
Por ejemplo:

```
   FD ORDER-ENTRY-SCREEN ...
   01 ORDER-ENTRY-RECORD ...
 ⋮
   WORKING-STORAGE SECTION.
   01 ORDSFL-I-FORMAT.
      COPY DDS-ORDSFL-I OF DOESCR.
   01 ORDSFL-O-FORMAT.
      COPY DDS-ORDSFL-O OF DOESCR.
 ⋮
   PROCEDURE DIVISION.
 ⋮
   READ SUBFILE ORDER-ENTRY-SCREEN NEXT MODIFIED RECORD
      INTO ORDSFL-I-FORMAT FORMAT IS "ORDSFL"
      AT END SET NO-MODIFIED-SUBFILE-RCD TO TRUE.
 ⋮
   MOVE CORR ORDSFL-I TO ORDSFL-O.
   REWRITE SUBFILE ORDER-ENTRY-RECORD FROM ORDSFL-O-FORMAT
                              FORMAT IS "ORDSFL" ...
 ⋮
```

Ejemplos de Generación de Claves

IBM International Business Machines

AS/400 ESPECIFICACIONES DE DESCRIPCION DE DATOS

GX21-9891-0 UM/050*
Impreso en España
*El número de hojas por área de relleno puede variar ligeramente.

Archivo		Instrucciones de Grabación	Signo	Descripción	Página de
Programador	Fecha		Tecla		

A

Número de Secuencia | Tipo de Formulario | And/Or/Coment. (A/O/*) | Condicionamiento: No (N), Indicador, No (N), Indicador, No (N), Indicador (Nombre Condición) | Tipo Nombre o Espec (b/R/H/J/K/S/O) | Reservado | Nombre | Referencia (R) | Longitud | Tipo Datos/Desplazamiento Teclado | Posiciones Decimales | Utilización (b/O/I/B/H/M/N/P) | Ubicación: Línea, Pos | Funciones

```
     A*  ARCHIVO FISICO PF1 PARA EJEMPLOS DE GENERACION DE CLAVES
     A*
     A          R PFRECORC
     A*
     A            MTH            2
     A*           DAY            2
     A            YEAR           4
     A            ITEM           8
     A
     A
     A          K MTH
     A          K DAY
     A
     A
     A
     A
     A
     A
     A
     A
     A
     A
     A
     A
     A
     A
     A
     A
     A
     A
     A
     A
     A
     A
     A
```

Figura 40. Especificaciones de Descripción de Datos para un Archivo Físico

El archivo físico de la Figura 40 constituye una base para los ejemplos que aparecen a continuación. Cada ejemplo hace referencia a un archivo lógico (derivado del archivo físico) que especifica EXTERNALLY-DESCRIBED-KEY en su cláusula SELECT.

Ejemplo que Utiliza la Palabra Clave CONCAT

IBM International Business Machines

AS/400 ESPECIFICACIONES DE DESCRIPCION DE DATOS

GX21-9891-0 UM/050*
Impreso en España
*El número de hojas por área de relleno puede variar ligeramente.

Archivo		Instrucciones de Grabación	Signo	Descripción	Página de
Programador	Fecha		Tecla		

```
     A* ARCHIVO LOGICO LF1 PARA EJEMPLOS DE CONCATENAR CLAVES
     A*
     A          R RECORD1                   PFILE(PF1)
     A*
     A            DATE                      CONCAT(MES DIA AÑO)
     A*
     A          K MTH
     A          K DAY
     A
     A
     A
     A
     A
     A
     A
     A
     A
     A
     A
     A
     A
     A
     A
     A
     A
     A
     A
     A
     A
     A
     A
     A
     A
```

Figura 41. Especificaciones de Descripción de Datos que Utilizan la Palabra CONCAT

Para el archivo lógico de la Figura 41, COPY DDS genera claves y nombres de claves derivados del archivo físico:

```
 FD  LF1 LABEL RECORDS ARE STANDARD.
 01   LOG-RECORD.
             COPY DDS-ALL-FORMATS OF LF1.
       05  LF1-RECORD PIC X(8).
*    FORMATO E-S:RECORD1   DESDE ARCHIVO LF1    DE BIBLIOTECA COPYDDS
*
*DEFINICIONES DE CLAVE PARA FORMATO DE REGISTRO  RECORD1
*  NUMERO              NOMBRE             RECUPERAC.    TIPO    SECALT
*   0001   MTH-DDS                         ASCENDENTE     AN      NO
*          NOMBRE DE CLAVE ORIGINADO DEL ARCHIVO FISICO
*   0002   DAY-DDS-DDS                     ASCENDENTE     AN      NO
*          NOMBRE DE CLAVE ORIGINADO DEL ARCHIVO FISICO
       05  RECORD1       REDEFINES LF1-RECORD.
           06 DATE-DDS              PIC X(8).
           06 FILLER REDEFINES DATE-DDS.
              07 MTH-DDS            PIC X(2).
              07 DAY-DDS-DDS        PIC X(2).
              07 FILLER             PIC X(4).
```

Figura 42. Ejemplo que Utiliza la Palabra Clave CONCAT

La instrucción COPY añade el sufijo -DDS a los nombres de campos MTH y DATE porque MTH es una clave que tiene su origen en el archivo físico, y DATE es una palabra reservada COBOL. La instrucción COPY añade el sufijo -DDS dos veces en el nombre de campo DAY porque DAY es a la vez una clave que se origina en el archivo físico y una palabra reservada COBOL.

Observe que si mueve la instrucción COPY de la Sección de Archivos a la Sección de Almacenamiento de Trabajo o a la Sección de Enlace, los campos subordinados a DATE-DDS ya no están disponibles:

```
 WORKING-STORAGE SECTION.
 01   WRK-RECORD.
             COPY DDS-ALL-FORMATS OF LF1.
       05  LF1-RECORD PIC X(8).
*    FORMATO E-S:RECORD1   DESDE ARCHIVO LF1    DE BIBLIOTECA COPYDDS
*
       05  RECORD1       REDEFINES LF1-RECORD.
           06 DATE-DDS              PIC X(8).
```

Figura 43. Ejemplo que Utiliza la Palabra Clave CONCAT-- Sección de Almacenamiento de Trabajo

Ejemplo que Utiliza la Palabra Clave RENAME

IBM International Business Machines

AS/400 ESPECIFICACIONES DE DESCRIPCION DE DATOS

GX21-9891-0 UM/050*
Impreso en España
*El número de hojas por área de relleno puede variar ligeramente.

Archivo		Instrucciones de Grabación	Signo	Descripción	Página de
Programador	Fecha		Tecla		

```
     A*  ARCHIVO LOGICO LF2 PARA EJEMPLOS REDENOMINAR PALABRA CLAVE
     A*
     A          R RECORD2                   PFILE(PF1)
     A*
     A            MONTH                     RENAME(MES)
     A*
     A          K MTH
```

Figura 44. Especificaciones de Descripción de Datos que Utilizan la Palabra RENAME

Para el archivo lógico descrito en la Figura 44, COPY DDS genera una clave y un nombre de clave derivados del archivo físico:

```
 FD  LF2 LABEL RECORDS ARE STANDARD.
 01   LOG-RECORD.
             COPY DDS-ALL-FORMATS OF LF2.
       05  LF2-RECORD PIC X(2).
*    FORMATO E-S:RECORD2   DESDE ARCHIVO LF2      DE BIBLIOTECA COPYDDS
*
*DEFINICIONES DE CLAVE PARA FORMATO DE REGISTRO  RECORD2
*  NUMERO              NOMBRE             RECUPERAC.    TIPO    SECALT
*   0001   MTH-DDS                         ASCENDENTE     AN      NO
*          NOMBRE DE CLAVE ORIGINADO DEL ARCHIVO FISICO
       05  RECORD2       REDEFINES LF2-RECORD.
           06 MONTH                 PIC X(2).
           06 MTH-DDS REDEFINES MONTH PIC X(2).
```

Figura 45. Utilización de la Palabra Clave RENAME

La instrucción COPY añade el sufijo -DDS al nombre de campo MTH porque MTH es una clave que tiene su origen en el archivo físico.

Ejemplo que Utiliza la Palabra Clave SST

IBM International Business Machines

AS/400 ESPECIFICACIONES DE DESCRIPCION DE DATOS

GX21-9891-0 UM/050*
Impreso en España
*El número de hojas por área de relleno puede variar ligeramente.

Archivo		Instrucciones de Grabación	Signo	Descripción	Página de
Programador	Fecha		Tecla		

Número de Secuencia (1-5)	Tipo de Formulario (6)	And/Or/Coment. (A/O/*) (7)	Condicionamiento: No (N) (8)	Indicador (9-10)	No (N) (11)	Indicador (12-13)	No (N) (14)	Indicador (15-16)	Tipo Nombre o Espec (b/R/H/J/K/S/O) (17)	Reservado (18)	Nombre (19-28)	Referencia (R) (29)	Longitud (30-34)	Tipo Datos/Desplazamiento Teclado (35)	Posiciones Decimales (36-37)	Utilización (b/O/I/B/H/M/N/P) (38)	Ubicación: Línea (39-41)	Pos (42-44)	Funciones (45-80)
	A	*	ARCHIVO LOGICO LF3 PARA EJEMPLOS DE PALABRA CLAVE SST																
	A	*																	
	A								R		RECORD3								PFILE(PF1)
	A	*																	
	A										YY					I			SST(AÑO 2 2)
	A	*																	
	A								K		YY								
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		

Figura 46. Especificaciones de Descripción de Datos que Utilizan la Palabra Clave SST

Para el archivo lógico descrito en la Figura 46 en la página 132, COPY DDS genera las especificaciones siguientes:

```
 FD  LF3 LABEL RECORDS ARE STANDARD.
 01   LOG-RECORD.
            COPY DDS-ALL-FORMATS OF LF3.
       05  LF3-RECORD PIC X(2).
*    FORMATO E-S:RECORD3   DESDE ARCHIVO LF3    DE BIBL. COPYDDS
*
*DEFINICIONES DE CLAVE PARA FORMATO DE REGISTRO  RECORD3
*  NUMERO              NOMBRE             RECUPERAC.    TIPO    SECALT
*   0001   YY                             ASCENDENTE      AN      NO
       05  RECORD3       REDEFINES LF3-RECORD.
           06 YY                    PIC X(2).
```

Figura 47. Utilización de la Palabra Clave SST

La instrucción COPY no añade ningún sufijo al nombre de campo YY porque YY no es una clave que se origine en el archivo físico ni es una palabra reservada COBOL.

Notas Adicionales sobre Nombres de Campos y de Formatos

Si el nombre de campo generado es una palabra reservada COBOL el sufijo -DDS se añade al nombre de campo.

La frase REPLACING no puede utilizarse para cambiar el nombre de un campo de clave cuando se utiliza EXTERNALLY-DESCRIBED-KEY.

Campos de Coma Flotante

COBOL trata los campos de coma flotante como FILLER. Los campos pueden contener valores de coma flotante colocados fuera de COBOL. Se genera una definición COMP-4 para mantener el alineamiento adecuado en el registro, pero los datos *no* están en formato binario. No debe hacerse ningún intento de utilizar datos de coma flotante para el proceso en el programa COBOL.

No se permiten campos de clave de coma flotante. En casos en los que existan algunos formatos con un campo de clave de coma flotante y otros formatos que no lo tengan, utilice una o más instrucciones COPY de Formato 2 con nombres de formato específicos; no utilice la opción ALL-FORMATS.

Nota: Si no especifica el orden de clasificación en el programa, puede crear un registro que contenga campos de coma flotante en el programa COBOL moviendo LOW-VALUES a la totalidad del registro antes de mover los valores de los campos que no sean de coma flotante. Esto les dará un valor cero a los campos de coma flotante en el registro. Tenga en cuenta que el método anterior sólo es recomendable si los campos de coma flotante válidos con un valor cero son apropiados para su aplicación particular.

Frase REPLACING en la Instrucción COPY de Formato 2

La frase REPLACING puede utilizarse para sustituir cualquier fuente generada en COBOL, incluyendo los números de nivel y el nombre de formato. Observe la excepción siguiente:

- Cuando se especifica RECORD KEY IS EXTERNALLY-DESCRIBED-KEY, la frase REPLACING no puede cambiar el nombre de un campo que es una clave.

Por ejemplo:

```
5763CB1 V3R0M5                  Fuente AS/400 COBOL
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
      001500*                                                                                        03/25/94
      001600* COPY DDS  S I N  OPCIÓN REPLACING                                                      03/25/94
      001700*                                                                                        03/25/94
  14  001800 COPY DDS-CUSMST OF CUSMSTP.                                                            03/25/94
     +000001*   FORMATO E-S:CUSMST   DESDE ARCHIVO CUSMSTP  DE BIBL. COBNATEX            CUSMST
     +000002*                         REGISTRO MAESTRO CLIENTE                          CUSMST
  15 +000003       05  CUSMST.                                                          CUSMST
  16 +000004           06 CUST             PIC X(5).                                    CUSMST
     +000005*                         NÚMERO CLIENTE                                    CUSMST
  17 +000006           06 NAME             PIC X(25).                                   CUSMST
     +000007*                         NOMBRE CLIENTE                                    CUSMST
  18 +000008           06 ADDR             PIC X(20).                                   CUSMST
     +000009*                         DIRECCIÓN CLIENTE                                 CUSMST
  19 +000010           06 CITY             PIC X(20).                                   CUSMST
     +000011*                         CIUDAD CLIENTE                                    CUSMST
  20 +000012           06 STATE            PIC X(2).                                    CUSMST
     +000013*                         PROVINCIA                                         CUSMST
  21 +000014           06 ZIP              PIC S9(5)        COMP-3.                     CUSMST
     +000015*                         CÓDIGO POSTAL                                     CUSMST
```

Figura 48. COPY DDS sin la Opción REPLACING

```
5763CB1 V3R0M5                  Fuente AS/400 COBOL
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
      001900*                                                                                        03/25/94
      002000* COPY DDS C O N  OPCION REPLACING                                                       03/25/94
      002100*                                                                                        03/25/94
  31  002200 COPY DDS-CUSMST OF CUSMSTP                                                             03/25/94
  32  002300      REPLACING NAME BY ADDR-LINE-1                                                     03/25/94
  33  002400                ADDR BY ADDR-LINE-2                                                     03/25/94
  34  002500                CITY BY ADDR-LINE-3.                                                    03/25/94
     +000001*   FORMATO E-S:CUSMST   DESDE ARCHIVO CUSMSTP  DE BIBL. COBNATEX            CUSMST
     +000002*                         REGISTRO MAESTRO CLIENTE                          CUSMST
  35 +000003       05  CUSMST.                                                          CUSMST
  36 +000004           06 CUST             PIC X(5).                                    CUSMST
     +000005*                         NÚMERO CLIENTE                                    CUSMST
  37 +000006           06 ADDR-LINE-1             PIC X(25).                           CUSMST
     +000007*                         NOMBRE CLIENTE                                    CUSMST
  38 +000008           06 ADDR-LINE-2             PIC X(20).                           CUSMST
     +000009*                         DIRECCIÓN CLIENTE                                 CUSMST
  39 +000010           06 ADDR-LINE-3             PIC X(20).                           CUSMST
     +000011*                         CIUDAD CLIENTE                                    CUSMST
  40 +000012           06 STATE            PIC X(2).                                    CUSMST
     +000013*                         PROVINCIA                                         CUSMST
  41 +000014           06 ZIP              PIC S9(5)        COMP-3.                     CUSMST
     +000015*                         CÓDIGO POSTAL                                     CUSMST
```

Figura 49. COPY DDS con la Opción REPLACING

Fin de Ampliación de IBM

Vía de Acceso

La descripción de un archivo descrito externamente contiene la vía de acceso que describe cómo se recuperan los registros del archivo. Los registros pueden recuperarse basándose en una vía de acceso en secuencia de llegada (sin clave) o en una vía de acceso en secuencia de claves.

La vía de acceso del orden de llegada se basa en el orden en que se almacenan los registros en el archivo. Los registros sólo se añaden al final del archivo.

Para la vía de acceso de secuencia por claves, el orden de recuperación de los registros del archivo se basa en el contenido de los campos de clave definidos en las DDS del archivo. Por ejemplo, en las DDS mostradas en la Figura 33 en la página 115, CUST se define como campo clave. La vía de acceso en secuencia de claves se actualiza siempre que se añaden o se suprimen registros, o cuando se cambia el contenido de un campo clave.

Consulte la publicación *Guía para la Base de Datos* para una descripción completa de las vías de acceso para un archivo de base de datos descrito externamente.

Claves de Registro y Claves Comunes

Para una vía de acceso en secuencia de claves, pueden definirse uno o más campos en las DDS para utilizarlos como campos clave para un formato de registro. No todos los tipos de registro de un archivo necesitan los mismos campos clave. Por ejemplo, un registro de cabecera de pedido puede tener el campo ORDER definido como campo de clave, y los registros de detalle de pedido pueden tener los campos ORDER y LINE definidos como campos de clave.

La clave para un archivo se determina mediante las claves válidas para los tipos de registros en ese archivo. La clave del archivo se determina de la manera siguiente:

- Si todos los tipos de registro en un archivo tienen el mismo número de campos de clave definidos en las DDS que sean idénticos en sus atributos, la *clave del archivo* se compone de todos los campos en la clave para los tipos de registro. (Los campos correspondientes no tienen que tener el mismo nombre). Por ejemplo, si el archivo tiene tres tipos de registro y la clave para cada tipo de registro consta en los campos A, B y C, la clave del archivo consta de los campos A, B y C. Es decir, la clave del archivo es la misma que la clave de los registros.
- Si todos los tipos de registro en el archivo no tienen los mismos campos de clave, la clave del archivo consta de los campos de clave *comunes* para todos los tipos de registro. Por ejemplo, un archivo tiene tres tipos de registro y los campos de clave se definen de la manera siguiente:
  - REC1 contiene el campo de clave A.
  - REC2 contiene los campos de clave A y B.
  - REC3 contiene los campos de clave A, B y C.

  Luego la clave del archivo es el campo A, el campo de clave común para todos los tipos de registro.
- Si no hay ningún campo clave común para todos los tipos de registro, cualquier referencia por clave para el archivo devolverá siempre el primer registro en el archivo.

En COBOL, deberá especificar RECORD KEY para un archivo indexado para identificar el registro que desee procesar. COBOL compara el valor de clave con el valor del archivo o registro y procesa la operación especificada en el registro cuya clave coincida con el valor RECORD KEY.

Cuando se especifica RECORD KEY IS EXTERNALLY-DESCRIBED-KEY:

- Si se especifica la frase FORMAT, el compilador construye el argumento de búsqueda a partir de los campos de clave en el área de registro para el formato especificado.
- Si no se especifica la frase FORMAT, el compilador construye el argumento de búsqueda a partir de los campos de clave en el área de registro del primer formato de registro definido en el programa para ese archivo.

Nota: Para un archivo que contiene múltiples campos de clave para procesarlos en COBOL, los campos de clave deben ser contiguos en el formato de registro que utilice el programa COBOL, excepto cuando se especifique RECORD KEY IS EXTERNALLY-DESCRIBED-KEY.

Alterar Temporalmente o Añadir Funciones COBOL a la Descripción Externa

Además de colocar la descripción externa de archivo en el programa mediante la instrucción COPY de Formato 2, podrá utilizar también la definición y redefinición de registro estándar para describir archivos externos o para proporcionar una definición de grupo para una serie de campos. Es responsabilidad del programador el asegurar que las definiciones descritas por programa sean compatibles con las definiciones externas del archivo.

Comprobación de Nivel

Cuando un programa COBOL/400 utiliza un archivo descrito externamente, el sistema operativo proporciona una función de comprobación de nivel (LVLCHK). Esta función asegura que el formato no se ha modificado desde el momento de la compilación.

El compilador siempre proporciona la información requerida por la comprobación de nivel cuando se utiliza un archivo descrito externamente (es decir, cuando una descripción de registro del archivo se ha definido utilizando la instrucción COPY de Formato 2). Sólo se comprueba el nivel de aquellos formatos que ha copiado la instrucción COPY de Formato 2 bajo el FD para un archivo. La función de comprobación de nivel se iniciará en tiempo de ejecución basándose en la selección realizada en los mandatos de crear, cambiar o alterar temporalmente archivo. La opción por omisión en el mandato de crear archivo es pedir la comprobación de nivel. Si se solicitó la comprobación de nivel, se produce según el formato de registro cuando se abre el archivo. Si se produce un error en la comprobación de nivel, COBOL establece el estado de archivo a 39 en tiempo de OPEN.

Si no se ha solicitado comprobación de nivel, y el archivo se vuelve a crear utilizando un formato existente, puede que los programas COBOL que utilicen dicho formato no trabajen sin recompilación, dependiendo de los cambios en el formato. Por ejemplo,

- Un cambio de claves provocará, con toda certeza, una anomalía en el programa o en cualquier instrucción I/O
- Un cambio en la longitud de registro provocará una anomalía de cualquier REWRITE
- Un cambio en el diseño de registro puede provocar diversos errores en el proceso de dicho registro.

Deberá extremar las precauciones cuando utilice programas COBOL sin comprobación de nivel o sin recompilar los programas.

Nota: El compilador no proporciona la comprobación de nivel para los archivos descritos por programa.

Para obtener más información acerca de la comprobación de nivel, consulte la publicación *Guía para la Gestión de Datos*.

Declaración de Ítems de Datos utilizando Tipos de Datos de CVTOPT

El compilador COBOL/400 sirve para convertir campos de longitud variable desde archivos descritos externamente, así como tipos de datos de bases de datos SAA en ítems estándar de datos COBOL. Los tipos de datos SAA que se pueden convertir son datos de fecha, hora, indicación horaria y DBCS gráficos. COBOL/400 proporciona soporte limitado para estos tipos de datos.

Campos de Longitud Variable

Es posible trasladar un campo de longitud variable a un programa si se especifica *VARCHAR en el parámetro CVTOPT del mandato CRTCBLPGM, o bien la opción VARCHAR de la instrucción PROCESS. Cuando se especifica *VARCHAR, el programa COBOL/400 convertirá un campo de longitud variable desde un archivo descrito externamente en un ítem de grupo COBOL/400.

A continuación se muestra un ejemplo de este ítem de grupo:

```
06  ITEM1.
    49  ITEM1-LENGTH     PIC S9(4) COMP-4.
    49  ITEM1-DATA       PIC X(n).
```

donde `n` representa la longitud máxima del campo de longitud variable. Dentro del programa, `PIC S9(4) COMP-4` se considera como cualquier otra declaración de este tipo, y `PIC X(n)` como un estándar alfanumérico.

Puesto que el valor máximo que ITEM1-LENGTH puede mantener es 9 999, ésta será la longitud del campo de longitud variable más largo que se pueda escribir desde un programa COBOL.

Cuando no se especifica *VARCHAR, se hace caso omiso de los campos de longitud variable y se declaran como campos FILLER en programas COBOL/400. Si se especifica *NOVARCHAR, el ítem se declara como sigue:

```
06  FILLER     PIC x(n+2).
```

Para obtener más información referente a la sintaxis, consulte el parámetro CVTOPT en la página 24.

El programa puede realizar cualquier tipo de operaciones de carácter válido en la parte de datos generada; no obstante, debido a la estructura del campo, la longitud de la parte debe estar formada por datos binarios válidos. Estos datos no son válidos si son negativos o mayores que la longitud máxima de campo.

Si los dos primeros bytes del campo carecen de un número binario válido, se producirá un error si se intenta efectuar WRITE o REWRITE a un registro que contenga el campo (o UPDATE o PUT el campo en una base de datos), y se devuelve el estado de archivo 90.

Las siguientes condiciones tienen lugar si se especifican campos de longitud variable:

- Si se detecta un campo de longitud variable cuando se extrae un campo para un archivo descrito externamente o para una estructura de datos descritos externamente, dicho campo se declarará como un campo de caracteres de longitud fija en un programa COBOL/400.
- Para los campos de caracteres de un solo byte, la longitud del campo COBOL/400 declarado equivale a la longitud del campo DDS más dos bytes.
- Para campos de caracteres DBCS gráficos, la longitud del campo COBOL/400 declarado equivale al doble de la longitud del campo DDS más dos bytes. Para obtener más información sobre tipos de datos gráficos, consulte el apartado “Campos Gráficos DBCS” en la página 139. Los dos bytes extra del campo COBOL/400 contienen un número binario que representa la longitud actual del campo de longitud variable. La Figura 50 muestra la longitud de campo COBOL/400 de los campos de longitud variable.

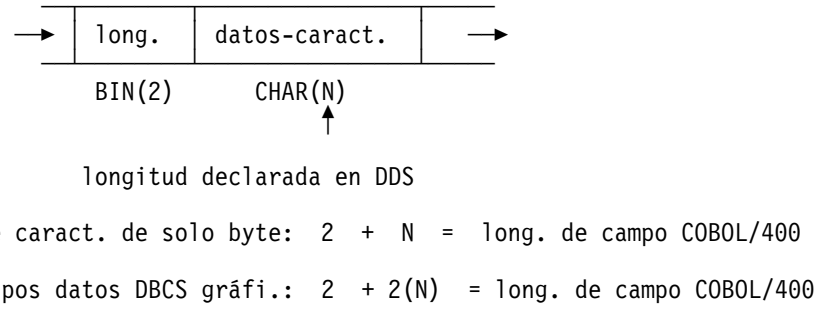


Figura 50. Longitud de Campo COBOL/400 de un Campo de Longitud Variable

- El programa COBOL/400 puede realizar cualquier operación válida de cálculo de caracteres en el campo de longitud fija declarado. Sin embargo, debido a la estructura del campo, los primeros dos caracteres del campo deben contener datos binarios válidos (datos de longitud fija actuales no válidos son no numéricos, inferiores a 0 o mayores que la longitud de campo DDS.) Se produce un error en una operación de entrada o salida si los dos primeros bytes del campo contienen datos de longitud de campo no válidos; entonces se devuelve el estado de archivo 90.
- Si no se especifica *VARCHAR, se pueden producir errores durante la realización de las operaciones WRITE en campos de longitud variable, ya que no se puede asignar ningún valor a FILLER. El campo de dos bytes puede tener un valor (por ejemplo `X'4040'`) que proporciona una longitud que sobrepase el rango permitido de este campo. Esta acción origina un error de E/S.

Si desea observar un ejemplo de un programa que utilice campos de longitud variable, consulte el apartado “Ejemplos” en la página 141.

Campos de fecha, hora e indicación horaria

Los campos de fecha, hora e indicación de la hora se introducen en el programa únicamente si se especifica la opción *DATETIME del parámetro CRTCBLPGM CVTOPT, o bien la opción DATETIME de la instrucción PROCESS. Para obtener más información sobre la sintaxis del parámetro CVTOPT, consulte la página 24.

Si no se especifica *DATETIME, se hace caso omiso de los campos de fecha, hora e indicación de la hora y se declaran campos FILLER en el programa COBOL/400.

Los campos de fecha, hora e indicación de la hora se introducen en el programa COBOL/400 como campos de caracteres de longitud variable. El programa COBOL/400 puede realizar cualquier operación válida en los campos de longitud variable. Esta operaciones se basan en las reglas COBOL estándar para ítems de datos alfanuméricos.

Los tipos de datos de fecha, hora e indicación de la hora disponen cada uno de ellos de un formato propio.

Si se actualiza un campo que contiene información referente a la hora, fecha o a la indicación horaria, y la información actualizada debe transferirse a la base de datos, el formato del campo debe ser exactamente el mismo que el del campo que fue recuperado desde la base de datos. Si no se utiliza el mismo formato, se producirá un error. Para obtener más información referente formatos válidos para cada tipo de datos, consulte el manual *DDS Reference*.

Si se intenta efectuar WRITE sobre un registro antes de trasladar el valor adecuado a un campo de fecha, hora o indicación de la hora, la operación WRITE no será satisfactoria y se devolverá el estado de archivo 90.

Si se declaran en el programa como FILLER los ítems fecha, hora e indicación de la hora, no intente efectuar WRITE sobre registros que contengan estos campos, puesto que no es posible establecer valores que puedan ser aceptados por el sistema.

Campos con posibilidad de nulos

Aunque el programa puede procesar campos con posibilidad de nulos, los valores nulos no están soportados. Las operaciones READ, SORT y MERGE pueden realizarse en campos con posibilidad de nulos, pero se producirá un error si estos campos contienen realmente valores nulos.

Campos Gráficos DBCS

El tipo de datos DBCS gráficos es una serie de caracteres donde cada carácter está representado por 2 bytes. Este tipo de datos no contienen caracteres de desplazamiento a teclado ideográfico (SO) ni caracteres de desplazamiento a teclado estándar (SI). La diferencia entre un dato de un solo byte y los datos DBCS gráficos se muestra en la siguiente figura:

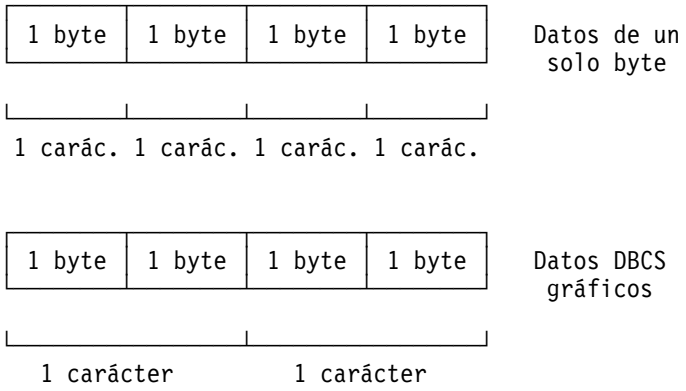


Figura 51. Comparación de Datos de un Solo Byte y Datos Gráficos

Los datos DBCS gráficos se introducen en el programa COBOL/400 solamente si se especifica el valor *GRAPHIC en el parámetro CVTOPT del mandato CRTCBLPGM, o bien la opción CVTGRAPHIC de la instrucción PROCESS. Si no se especifican datos gráficos DBCS, se hace caso omiso de los datos gráficos y se declaran campos FILLER en el programa COBOL/400. Para obtener más información sobre la sintaxis del parámetro CVTOPT, consulte la página 24.

Las siguientes condiciones se aplican cuando se especifican los datos DBCS gráficos:

- Los datos DBCS gráficos se copian en el programa COBOL/400 como un campo alfanumérico de longitud fija.
- Cada *carácter* de datos DBCS gráficos tiene una longitud de 2 bytes.
- Cada *campo* de datos DBCS gráficos de longitud fija tiene una longitud que es el doble del número de caracteres del campo. Para obtener una descripción de la longitud de campo de los campos de datos gráficos de longitud variable, consulte el apartado “Campos de Longitud Variable” en la página 137.
- El programa COBOL/400 puede realizar cualquier operación válida en los campos de longitud variable.

Campos Gráficos DBCS de Longitud Variable

Es posible utilizar campos de longitud variable en combinación con tipos de datos DBCS gráficos para especificar datos DBCS gráficos de longitud variable. Para especificar datos DBCS gráficos de longitud variable, especifique `*VARCHAR` y `*GRAPHIC` para el parámetro CVTOPT del mandato CRTCBLPGM, o bien las opciones `VARCHAR` y `CVTGRAPHIC` para la instrucción PROCESS.

Si especifica tanto `CVTOPT(*NOVARCHAR *NOGRAPHIC)` como `CVTOPT(*NOVARCHAR *GRAPHIC)` y el compilador detecta un ítem de datos DBCS gráficos de longitud variable, el programa resultante contendrá los siguientes elementos:

```
             06 FILLER            PIC X(2n+2).
      *        (Campo de longitud variable)
```

donde `n` es el número de caracteres del campo DDS.

Si se especifica CVTOPT(*VARCHAR *NOGRAPHIC) y el compilador detecta un ítem de datos DBCS gráficos de longitud variable, el programa resultante contendrá los siguientes elementos:

```
           06 NAME
*               (Campo de longitud variable)
              49 NAME-LENGTH        PIC S9(4) COMP-4.
*                           (Número de caracteres de 2 bytes)
              49 FILLER             PIC X(2n).
*                           (Campo gráfico)
```

donde n es el número de caracteres del campo DDS.

Si se especifica CVTOPT(*VARCHAR *GRAPHIC) y el compilador detecta un ítem de datos DBCS gráficos de longitud variable, el programa resultante contendrá los siguientes elementos:

```
           06 NAME
*               (Campo de longitud variable)
              49 NAME-LENGTH        PIC S9(4) COMP-4.
*                           (Número de caracteres de 2 bytes)
              49 NAME-DATA          PIC X(2n).
*                           (Campo gráfico)
```

donde n es el número de caracteres del campo DDS.

Ejemplos

La Figura 52 muestra un ejemplo de archivo DDS que define un ítem de datos DBCS gráficos de longitud variable. La Figura 53 en la página 142 muestra un programa COBOL/400 que utiliza una instrucción DDS COPY, así como el listado resultante que se obtiene cuando se compila el programa.

```
A          R SAMPLEFILE
A*
A            VARITEM      100          VARLEN
A*
A            TIMEITEM       T          TIMFMT(*HMS)
A            DATEITEM       L          DATFMT(*YMD)
A            TIMESTAMP      Z
A*
A            GRAPHITEM    100G
A            VGRAPHITEM   100G          VARLEN
```

Figura 52. Archivo DDS que Define un Campo de Datos Gráficos de Longitud Variable

```
5763CB1 V3R0M5  001000            IBM SAA COBOL/400 TESTER/PGM1         AS400SYS  04/24/94 08:55:54     Página   1
Programa . . . . . . . . . . . . . . :   PGM1
  Biblioteca . . . . . . . . . . . . :     TESTER
Archivo fuente . . . . . . . . . . . :   QLBLSRC
  Biblioteca . . . . . . . . . . . . :     TESTER
Miembro fuente . . . . . . . . . . . :   PGM1     04/24/94 08:23:06
Nivel de gravedad de generación  . . :   29
Texto 'descripción'  . . . . . . . . :   Ejemplos de tipos de datos
Opciones de listado fuente . . . . . :   *NONE
Opciones de generación . . . . . . . :   *NONE
Opciones de conversión . . . . . . . :   *VARCHAR *DATETIME *GRAPHIC
Límite de mensaje:
  Cantidad de mensajes.. . . . . . . :   *NOMAX
  Gravedad límite de mensaje   . . . :   29
Imprimir archivo. . . .  . . . . . . :   QSYSPRT
  Biblioteca . . . . . . . . . . . . :     *LIBL
Señalización FIPS. . . . . . . . . . :   *NOFIPS *NOSEG *NODEB *NOOBSOLETE
Señalización SAA . . . . . . . . . . :   *NOFLAG
Opciones de visualización ampliada . :
Gravedad de señalización . . . . . . :   0
Sustituir programa . . . . . . . . . :   *YES
Release de destino . . . . . . . . . :   *CURRENT
Perfil de usuario. . . . . . . . . . :   *USER
Autorización . . . . . . . . . . . . :   *LIBCRTAUT
Compilador . . . . . . . . . . . . . :   IBM SAA COBOL/400
5763CB1 V3R0M5  001000           Fuente COBOL AS/400              TESTER/PGM1         AS400SYS  04/24/94 08:55:54     Página  2
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
    1  000100 Identification division.                                                                  01/02/94
    2  000200   Program-id.  pgm1.                                                                    02/13/94
    3  000300 Environment division.                                                                     01/02/94
    4  000400 Configuration section.                                                                    01/02/94
    5  000500   Source-computer.  ibm-as400.                                                          01/02/94
    6  000600   Object-computer.  ibm-as400.                                                          01/02/94
    7  000700 Input-output section.                                                                     01/02/94
    8  000800   File-control.                                                                         01/02/94
    9  000900     Select file1                                                                        04/23/94
   10  001000       assign to database-samplefile                                                     02/13/94
   11  001100       organization is sequential                                                        04/23/94
   12  001200       access is sequential                                                              04/23/94
   13  001300       file status is fs1.                                                               04/23/94
   14  001400 Data division.                                                                          01/02/94
   15  001500 File section.                                                                           01/02/94
   16  001600 fd  file1.                                                                              01/02/94
   17  001700 01  record1.                                                                           01/02/94
   18  001800 copy dds-all-formats of samplefile.                                                     02/13/94
   19 +000001       05  SAMPLEFILE-RECORD PIC X(546).                                <-ALL-FMTS
      +000002*    FORMATO E-S:SAMPLEFILE   DESDE ARCHIVO  SAMPLEFILE  DE BIBLO TESTER      <-ALL-FMTS
      +000003*                                                                          <-ALL-FMTS
   20 +000004       05  SAMPLEFILE    REDEFINES SAMPLEFILE-RECORD.                      <-ALL-FMTS
   21 +000005           06 VARITEM.                                                     <-ALL-FMTS
      +000006*              (Campo de longitud variable)                                <-ALL-FMTS
   22 +000007              49 VARITEM-LENGTH     PIC S9(4) COMP-4.                     <-ALL-FMTS
   23 +000008              49 VARITEM-DATA       PIC X(100).                           <-ALL-FMTS
   24 +000009           06 TIMEITEM              PIC X(8).                             <-ALL-FMTS
      +000010*                 (Campo de hora)                                        <-ALL-FMTS
   25 +000011           06 DATEITEM              PIC X(8).                             <-ALL-FMTS
      +000012*                 (Campo de fecha)                                       <-ALL-FMTS
   26 +000013           06 TIMESTAMP             PIC X(26).                            <-ALL-FMTS
      +000014*                 (Campo de indicación de la hora)                       <-ALL-FMTS
   27 +000015           06 GRAPHITEM             PIC X(200).                           <-ALL-FMTS
      +000016*                 (Campo gráfico)                                        <-ALL-FMTS
   28 +000017           06 VGRAPHITEM.                                                 <-ALL-FMTS
      +000018*              (Campo de longitud variable)                              <-ALL-FMTS
   29 +000019              49 VGRAPHITEM-LENGTH  PIC S9(4) COMP-4.                     <-ALL-FMTS
      +000020*                 (Número de caracteres de 2 bytes)                      <-ALL-FMTS
   30 +000021              49 VGRAPHITEM-DATA    PIC X(200).                           <-ALL-FMTS
      +000022*                 (Campo gráfico)                                        <-ALL-FMTS
   31  001900 working-storage section.                                                            04/22/94
   32  002000 77  fs1      pic x(2).                                                             04/23/94
   33  002100 Procedure division.                                                                 01/09/94
       002200 Mainline.                                                                           01/02/94
   34  002300     stop run.                                                                       01/02/94
                      * * * * *   F I N   D E   F U E N T E   * * * * *
5738CB1 V2R2M0  001000            Mensajes COBOL AS/400             TESTER/PGM1         AS400SYS  04/24/94 08:55:54     Página   3
 INST
*   16  MSGID: LBL0650  GRAVEDAD: 00  NUMSEC:  001600
       Mensaje . . . . :   Bloqueo/Desbloqueo para archivo 'FILE-1' se
         se realizará por medio del código generado por el compilador.
                    * * * * *   F I N   D E   M E N S A J E S   * * * * *
                                      Resumen Mensajes
  Total    Info(0-4)    Aviso(5-19)      Error(20-29)    Grave(30-39)    Terminal(40-99)
     1          1              0                0               0               0
 Registros fuente leídos . . . . . . :   23
 Registros de copia leídos . . . . . :   22
 Miembros de copia procesados  . . . :   1
 Errores de secuencia  . . . . . . . :   0
 Mensaje de gravedad emitido más alto:   0
  LBL0901 00  Programa PGM1 creado en biblioteca TESTER.
               * * * * *   F I N   D E   C O M P I L A C I O N   * * * * *
```

Figura 53. Programa COBOL/400 que utiliza Ítems de Datos DBCS Gráficos de Longitud Variable

Consideraciones acerca de los Datos de Sistemas Cruzados

Los identificadores de juegos de caracteres codificados (CCSID) le ayudarán a mantener la integridad de los datos de caracteres en los sistemas.

La Arquitectura de Representación de Datos de Caracteres (CDRA) define los valores CCSID para identificar los puntos de código utilizados para representar caracteres y para convertir dichos códigos de manera que conserven sus significados.

Como consecuencia de la conversión CDRA, es posible que pueda tener caracteres de sustitución (X'3F') en sus datos. Si escribe dichos caracteres en una pantalla, se producirán una serie de resultados imprevisibles.

Para obtener más información acerca de las CCSID y CDRA, consulte las publicaciones *Operación del Sistema*, SC10-9280 (SC41-3203) y *Guía para la Gestión de Datos*.

Capítulo 8. Archivos Transaction

Ampliación de IBM

Esta capítulo describe las ampliaciones del lenguaje COBOL/400 que dan soporte a las estaciones de trabajo y las comunicaciones programa a programa.

La organización de archivos TRANSACTION permite que un programa COBOL comunique interactivamente con:

- Uno o más usuarios en una estación de trabajo
- Uno o más programas en un sistema remoto
- Uno o más dispositivos en un sistema remoto.

El sistema AS/400 le permite comunicar con un programa o dispositivo (como, por ejemplo, tipos de comunicaciones Asíncronas) en un sistema remoto. Para una descripción detallada de tales dispositivos, consulte la publicación *ICF Programmer's Guide*.

Archivos Transaction Descritos por Programa

En general, los archivos TRANSACTION en COBOL se describen externamente. Sin embargo, si estos archivos se describen por programa, sólo pueden realizarse formatos simples de pantalla. Todas las descripciones a nivel de campo se definen en el programa COBOL

No envíe datos internos (empaquetados) o binarios (COMP, COMP-3 o COMP-4) a una estación de pantalla como datos de salida. Tales datos pueden contener caracteres de control de estación de pantalla que pueden provocar resultados imprevisibles.

Consulte la publicación *Guía para la Gestión de Datos* para más información acerca de la utilización de la definición de mandato.

Archivos Transaction Descritos Externamente

Un archivo en COBOL utiliza un archivo descrito externamente que contiene información de archivo y una descripción de los campos en los registros. Los registros en este archivo pueden describirse en el programa COBOL mediante el Formato 2 de la instrucción COPY.

La instrucción COPY con el Formato 2

Las instrucciones COPY con Formato 2 se utilizan para generar instrucciones en la División de Datos COBOL dentro de los programas fuente para describir archivos que existen en el sistema.

Nota: El término *Instrucción COPY con formato 2* se utiliza en este manual para describir la instrucción COPY (opción DD, DDR, DDS o DDSR).

Para más información acerca de la instrucción COPY con Formato 2, consulte el apartado "Instrucción COPY de Formato 2 (Opción DD, DDR, DDS o DDSR)" en la página 118.

Especificaciones de Descripción de Datos (Data Description Specifications)

Las **Especificaciones de Descripción de Datos** (DDS) son una descripción de la base de datos del usuario o de los archivos de dispositivo que se entran en el sistema de forma fija. La descripción se utiliza para crear archivos.

Además de las descripciones de campo (como, por ejemplo, los nombres y atributos de los campos), las especificaciones de descripción de datos (DDS) para un archivo de dispositivo de pantalla:

- Especifican las entradas de número de línea y número de posición para cada campo y constante con el fin de dar formato a la colocación del registro en la pantalla.
- Especifican las funciones de atención, como pueden ser las de campos de subrayado y resaltado, contraste invertido o un cursor parpadeante.
- Especifican la comprobación de validez para los datos entrados en la estación de trabajo de pantalla. Las funciones para la comprobación de validez incluyen:
  - Detectar campos en los que se requieren datos
  - Detectar los campos de relleno obligatorio
  - Detectar los tipos de datos incorrectos
  - Detectar los datos con un rango específico
  - Comprobar los datos para una entrada válida
  - Realizar la verificación de comprobación de dígitos de los módulos 10 u 11.
- Controlan las funciones de gestión de pantallas (por ejemplo, en qué momento se van a borrar, dar formato previamente o retener cuando se visualicen nuevos datos).
- Asocian indicadores del 01 al 99 con claves de función designadas como de tipo `CA` o `CF`. Si una tecla de función se diseña como `CF`, tanto el registro de datos modificados como el indicador de respuesta se devuelven al programa. Si una tecla de función se designa como `CA`, el indicador de respuesta se devuelve al programa, pero el registro de datos contiene, generalmente, valores por omisión para campos sólo de entrada y valores grabados en el formato para campos de entrada/salida ocultos. Para más información acerca de los tipos de teclas de función `CF` y `CA`, consulte la publicación *DDS Reference*.
- Asignan un código de edición (palabra clave EDTCDE) o palabra de edición (palabra clave EDTWRD) a un campo para especificar cómo deben visualizarse los valores del campo.
- Especifican subarchivos.

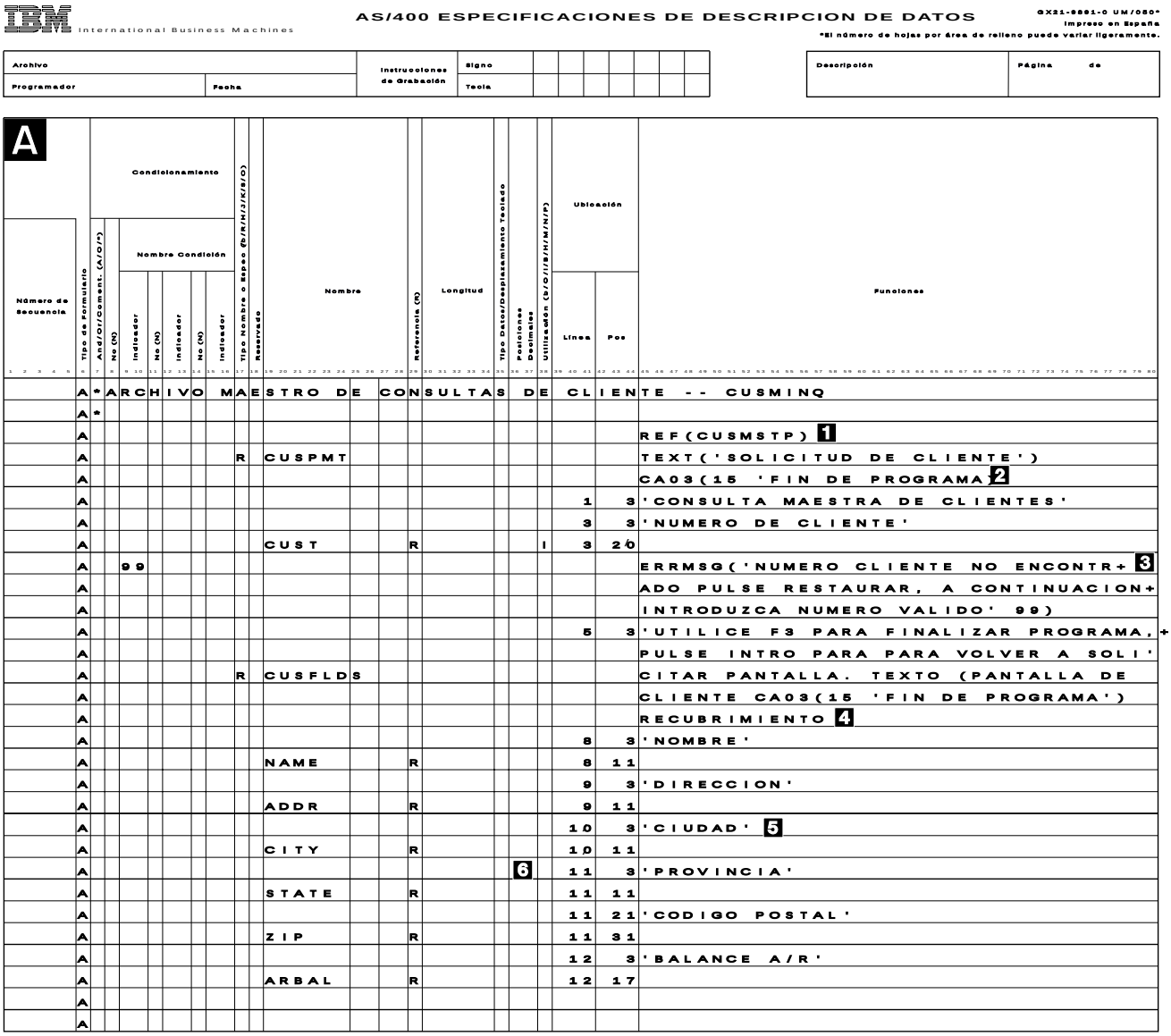
Los **Datos de formato de pantalla** definen o describen una pantalla. Un formato de registro de dispositivo de pantalla contiene tres tipos de campos:

- *Campos de Entrada:* Los campos de entrada pasan desde un dispositivo al programa cuando éste lee un registro. Los campos de entrada pueden inicializarse con un valor por omisión; si no se cambia el valor por omisión, éste pasa al programa. Los campos de entrada sin inicializar se visualizan como espacios en blanco en los que el usuario de estación de trabajo puede introducir datos.

- *Campos de Salida:* Los campos de salida pasan desde el programa al dispositivo cuando aquél graba un registro a una pantalla. El programa o el formato de registro del archivo de dispositivo puede proporcionar campos de salida.
- *Campos de Entrada/Salida:* Un campo de entrada/salida es un campo de salida que puede cambiarse y convertirse en campo de entrada. Los campos de entrada/salida pasan *desde* el programa cuando el programa graba un registro a una pantalla y pasan *al* programa cuando éste lee un registro de una pantalla. Los campos de entrada/salida se utilizan cuando el usuario va a cambiar o actualizar los datos que se graban a la pantalla desde el programa.

Para una descripción detallada de un archivo de comunicaciones de datos, consulte la publicación *ICF Programmer's Guide*. Para más información acerca de los archivos de pantalla descritos externamente, vea la publicación *Guía para la Gestión de Datos*. Para una lista de las palabras clave de las especificaciones de descripciones de datos válidos (DDS), vea la publicación *DDS Reference*.

La Figura 54 muestra un ejemplo de las DDS para un archivo de dispositivo de pantalla:



AS/400 ESPECIFICACIONES DE DESCRIPCION DE DATOS

GX21-8891-0 UM/050* Impreso en España

*El número de hojas por área de relleno puede variar ligeramente.

Archivo | Instrucciones de Grabación | Signo | Descripción | Página de

Programador | Fecha | Teclo

A

Número de Secuencia	Tipo de Formulario	Condicionamiento: And/Or/Comment (A/O/*)	Condicionamiento: No (N)	Condicionamiento: Nombre Condición: Indicador	Condicionamiento: Nombre Condición: No (N)	Condicionamiento: Nombre Condición: Indicador	Condicionamiento: Nombre Condición: No (N)	Condicionamiento: Nombre Condición: Indicador	Tipo Nombre o Espacio (B/R/H/J/K/S/O)	Reservado	Nombre	Referencia (R)	Longitud	Tipo Datos/Desplazamiento Teclado	Posiciones Decimales	Utilización (B/O/I/S/H/M/N/P)	Ubicación: Línea	Ubicación: Pos	Funciones
	A	*																	ARCHIVO MAESTRO DE CONSULTAS DE CLIENTE -- CUSMINQ
	A	*																	
	A																		REF(CUSMSTP) 1
	A								R		CUSPMT								TEXT('SOLICITUD DE CLIENTE')
	A																		CA03(15 'FIN DE PROGRAMA') 2
	A																1	3	'CONSULTA MAESTRA DE CLIENTES'
	A																3	3	'NUMERO DE CLIENTE'
	A										CUST		R			I	3	20	
	A			99															ERRMSG('NUMERO CLIENTE NO ENCONTR+ 3
	A																		ADO PULSE RESTAURAR, A CONTINUACION+
	A																		INTRODUZCA NUMERO VALIDO' 99)
	A																5	3	'UTILICE F3 PARA FINALIZAR PROGRAMA,+
	A																		PULSE INTRO PARA PARA VOLVER A SOLI'
	A								R		CUSFLDS								CITAR PANTALLA. TEXTO (PANTALLA DE
	A																		CLIENTE CA03(15 'FIN DE PROGRAMA')
	A																		RECUBRIMIENTO 4
	A																8	3	'NOMBRE'
	A										NAME		R				8	11	
	A																9	3	'DIRECCION'
	A										ADDR		R				9	11	
	A																10	3	'CIUDAD' 5
	A										CITY		R				10	11	
	A														6		11	3	'PROVINCIA'
	A										STATE		R				11	11	
	A																11	21	'CODIGO POSTAL'
	A										ZIP		R				11	31	
	A																12	3	'BALANCE A/R'
	A										ARBAL		R				12	17	
	A																		
	A																		

Figura 54. Ejemplo de las Especificaciones de Descripción de Datos para un Archivo de Dispositivo de Pantalla

Esta archivo de dispositivo de pantalla contiene dos formatos de registro: CUSPMT y CUSFLDS.

1 Los atributos para los campos de este archivo se definen en el archivo de referencia de campo CUSMSTP. Por ejemplo, EDTCDE(J) se define en CUSMSTP para el campo ARBAL.

2 La tecla F3 se asocia con el indicador 15, con el que el usuario finaliza el programa.

3 La palabra clave ERRMSG identifica el mensaje de error que se visualiza si el indicador 99 se establece en el programa que utiliza este formato de registro.

4 La palabra clave OVERLAY se utiliza para el formato de registro CUSFLDS de forma que el registro CUSPMT en la pantalla no se borre cuando el registro CUSFLDS se grabe en la pantalla.

5 Las constantes como por ejemplo ‘Nombre’, ‘Dirección’ y ‘Ciudad’ describen los campos que graba el programa.

6 Las entradas de línea y posición identifican dónde se graban los campos o constantes en la pantalla.

Proceso de un Archivo Transaction Descrito Externamente

Cuando se procesa un archivo TRANSACTION descrito externamente, el sistema operativo transforma los datos del programa al formato especificado para el archivo y visualiza los datos. Cuando los datos se pasan al programa, se transforman al formato utilizado por el programa.

El sistema operativo proporciona información de control del dispositivo para realizar operaciones de entrada/salida para el dispositivo. Cuando se solicita un registro de entrada desde el dispositivo, el sistema operativo emite la petición y luego extrae la información de control del dispositivo de los datos antes de que se pasen éstos al programa. Además, el sistema operativo puede pasar indicadores al programa indicando qué campos, si los hay, se cambian en el registro.

Cuando el programa solicita una operación de salida, pasa el registro de salida al sistema operativo. El sistema operativo proporciona la información de control del dispositivo necesaria para visualizar el registro. También añade cualquier información constante especificada para el formato del registro cuando se visualiza el registro.

Cuando el registro pasa a un programa, los campos se disponen de acuerdo con el orden especificado en las DDS. El orden en que se visualizan los campos se basa en las posiciones de pantalla (números y posiciones de línea) asignadas a los campos en las DDS. Por lo tanto, el orden en que se especifican los datos en las DDS y el orden en que aparecen en la pantalla no tiene por qué ser el mismo.

Utilización de Indicadores con Archivos Transaction

Los indicadores son ítems de datos Booleanos que pueden tener los valores B"0" ó B"1".

Cuando se define un formato de registro para un archivo utilizando las DDS, se pueden condicionar las opciones utilizando indicadores; los indicadores también pueden utilizarse para reflejar respuestas particulares. Estos indicadores se conocen como OPTION y RESPONSE, respectivamente.

Los indicadores de opción proporcionan opciones tales como espaciado, subrayado y permiso o petición de transferencia de datos desde un programa a una impresora o dispositivo de pantalla. Los indicadores de respuesta proporcionan información de respuesta a un programa desde un dispositivo, como por ejemplo las teclas de función pulsadas por un usuario de estación de trabajo y si se han introducido los datos.

Los indicadores pueden pasarse con registros de datos dentro de un área de registro, o fuera del área de registro en un área de indicadores separada.

Indicadores en un Área de Indicadores Separada

Si especifica la palabra clave a nivel de campo INDARA en las DDS, todos los indicadores definidos en el formato de registro o en los formatos para ese archivo se pasan al programa y desde el programa en un área de indicadores separada, no en el área de registro. Para información sobre cómo especificar la palabra clave INDARA, vea la publicación *DDS Reference*.

La entrada de control de archivo que tenga la palabra clave INDARA especificada en las DDS debe tener el atributo de área de indicadores separada, SI, como parte del nombre de asignación.

Las ventajas de utilizar un área de indicadores separada son las siguientes:

- El número y orden de indicadores utilizados en una instrucción de E/S para cualquier formato de registro en un archivo no tiene que coincidir con el número y orden de indicadores especificados en las DDS para dicho formato de registro.
- El programa asocia el número del indicador en una entrada de descripción de datos con el indicador apropiado.

Indicadores en el Área de Registro

Si la palabra clave INDARA no se utiliza en las DDS del archivo, los indicadores se crean en el área de registro. Cuando los indicadores se definen en un formato de registro para un archivo, se leen, regraban y graban con los datos en el área de registro.

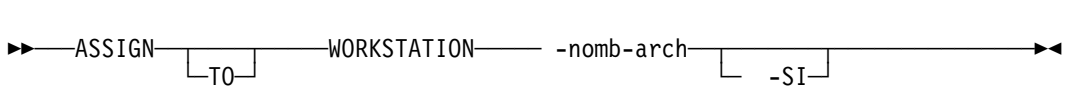
El número y orden de indicadores definido en las DDS para un formato de registro para un archivo determina el número y orden en que deben codificarse en el programa las entradas de descripción de datos para los indicadores en el formato de registros.

La entrada de control de archivo que no tiene la palabra clave INDARA especificada en las DDS asociadas *no* deberá tener el atributo de área de indicadores separada, SI, como parte del nombre de asignación.

Si una instrucción COPY de Formato 2 se utiliza para copiar indicadores en un programa fuente, estos indicadores se definen en el orden en el que están especificados en las DDS para el archivo.

Cláusula ASSIGN y Atributo de Área de Indicadores Separada

Formato



Las reglas para la cláusula ASSIGN son las siguientes:

- El dispositivo debe ser WORKSTATION

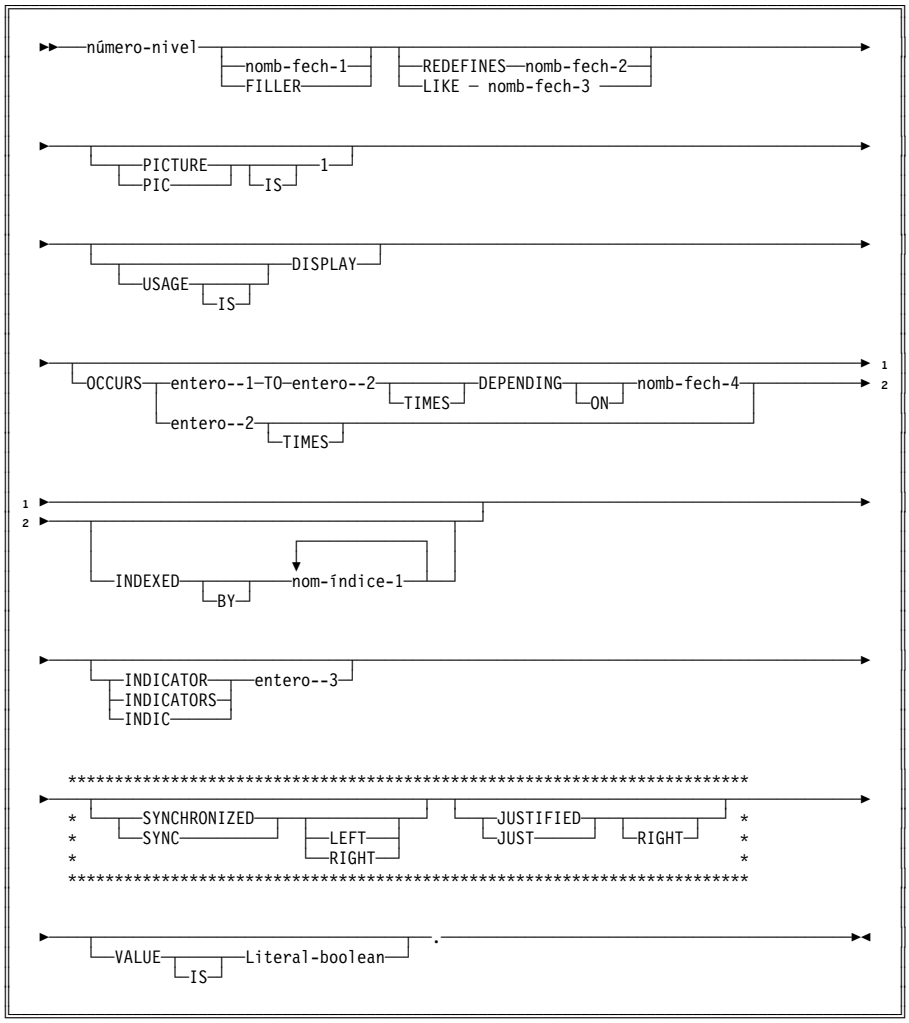
- Si se codifica -SI, el *nombre de archivo* debe referirse a un archivo que tiene la palabra clave a nivel de archivo INDARA especificada en sus DDS.

Para más información acerca de la cláusula ASSIGN, consulte el apartado “Cláusula ASSIGN” en la página 181.

Entrada de Descripción de Datos–Datos Booleanos

Cuando utiliza indicadores en un programa COBOL, debe describirlos como ítems de datos Booleanos utilizando la entrada de descripción de datos para datos booleanos.

Formato 4 - Datos Booleanos



Consideraciones Especiales

A continuación se dan las consideraciones especiales para las cláusulas utilizadas con datos booleanos. El resto de reglas para las cláusulas son las mismas que aquéllas descritas en la sección “Estructura del Programa COBOL” de la publicación *COBOL/400 Reference*.

Cláusula PICTURE: Se define un nombre de datos booleanos elemental mediante una cláusula PICTURE que contiene sólo un 1.

Cláusula USAGE: USAGE debe definirse implícita o explícitamente como DISPLAY.

Cláusula OCCURS: Cuando las cláusulas OCCURS e indicador INDICATOR se especifican a un nivel elemental, se define una tabla de ítems de datos booleanos con cada elemento de la tabla correspondiente a un indicador externo. El primer elemento de la tabla corresponde al número de indicador especificado en la cláusula INDICATOR; el segundo elemento corresponde al indicador que sigue secuencialmente al indicador especificado mediante la cláusula INDICATOR.

Por ejemplo, si se codifica lo siguiente, SWITCHES (1) corresponde al indicador 16, SWITCHES (2) corresponde al indicador 17... y SWITCHES (10) corresponde al indicador 25:

```
07     SWITCHES       PIC 1
                      OCCURS 10 TIMES
                      INDICATOR 16.
```

Cláusula INDICATOR: Si los campos del indicador están en un área de indicador separada, la cláusula INDICATOR asocia un indicador definido en las DDS con un ítem de datos booleanos. Si los campos de indicador están en un área de registro, se comprueba la sintaxis de la cláusula INDICATOR, aunque se trata como un comentario.

El `Entero-3` tiene que ser un valor del 1 al 99.

La cláusula INDICATOR sólo debe especificarse a nivel elemental.

Cláusula VALUE: La cláusula VALUE especifica el contenido inicial de un ítem de datos booleanos. Los valores permitidos para los literales booleanos son B"0", B"1" y ZERO.

Cláusula LIKE: No se puede utilizar esta cláusula para cambiar la longitud de un ítem de datos.

Frase INDICATORS

Cuando la frase INDICATORS se utiliza en las instrucciones READ, REWRITE y WRITE (vea la Figura 57 en la página 158), especifica los indicadores que se van a leer, regrabar y grabar.

El identificador especificado en la frase INDICATORS puede ser uno de los siguientes:

- Un ítem elemental de datos booleanos

- Un ítem elemental de grupo con ítems de datos booleanos subordinados a él. (Los ítems de datos booleanos pueden estar en cualquier lugar del grupo, pero son los únicos ítems que el usuario puede leer, grabar o regrabar).

  El identificador no puede estar subordinado a un ítem que esté sujeto a una cláusula OCCURS.

Indicadores en un Área de Indicadores Separada

Si se especifica INDARA en las DDS para el archivo, la utilización de los indicadores referenciados en la frase INDICATORS está basada en el número del indicador.

- En una instrucción READ, sólo se actualizan los números de indicador de respuesta referenciados por la frase INDICATORS. Los indicadores especificados en las DDS para el formato, pero no referenciados por la frase INDICATORS, se ignoran. No se modifican los indicadores referenciados por la frase INDICATORS pero no especificados en las DDS.
- En una instrucción WRITE o REWRITE, sólo se utilizan los indicadores de opción referenciados mediante la frase INDICATORS. Los indicadores especificados en las DDS para el formato pero *no* referenciados por la frase INDICATORS, se consideran como **DESACTIVADOS**. Se ignoran los indicadores referenciados por la frase INDICATORS pero no utilizados en las DDS para el formato.

Si no se especifica la frase INDICATORS, ocurre lo siguiente:

- En la instrucción READ, no se actualizan los indicadores.
- En una instrucción WRITE o REWRITE, los indicadores se tratan como si estuvieran **DESACTIVADOS**.

Indicadores en el Área de Registro

Si no se especifica INDARA en las DDS para el archivo, el tamaño del identificador en la frase INDICATORS de una instrucción de E/S (consulte la Figura 57 en la página 158) debe ser igual al número de opción o indicadores de respuesta definidos en las DDS para ese formato.

- En una instrucción READ, el tamaño del identificador debe ser igual al número de los indicadores de respuesta.
- En una instrucción REWRITE o WRITE, el tamaño del identificador debe ser igual al número de los indicadores de opción.

Los contenidos del identificador no se comprueban, pero se copian al registro o desde el comienzo del registro, de byte en byte; los números de indicador se ignoran.

Si se omite la frase INDICATORS, se siguen pasando al área de registro los datos de los campos de indicadores en el registro. La frase INDICATORS sólo se utiliza para copiar indicadores a un área de registros antes de una instrucción WRITE o REWRITE, o fuera de un área de registros después de una instrucción READ.

Programas Ejemplo con Indicadores

Esta sección contiene ejemplos de programas en COBOL/400 que ilustran la utilización de indicadores en un área de registro o un área de indicadores separada.

Todos los programas realizan lo siguiente:

1. Determinan la fecha actual.
2. Si es el primer día del mes, activan un indicador de opción que hace que aparezca un campo de salida y parpadee.
3. Le permite pulsar las teclas de función para finalizar el programa, activar los indicadores de respuesta y llamar a programas para escribir informes diarios o mensuales.

La Figura 56 en la página 156 muestra un programa que utiliza los indicadores en el área de registro pero no utiliza la frase INDICATORS en ninguna instrucción de E/S. La Figura 55 en la página 155 muestra las DDS asociadas para el archivo.

La Figura 57 en la página 158 muestra un programa que utiliza indicadores en el área de registro y la frase INDICATORS en las instrucciones de E/S. Las DDS asociadas para la Figura 57 son la Figura 55 en la página 155.

La Figura 59 en la página 161 muestra un programa que utiliza indicadores en un área de indicadores separada, definida en WORKING-STORAGE utilizando la instrucción COPY Formato 2. La Figura 58 en la página 160 muestra las DDS asociadas para el archivo.

La Figura 60 en la página 163 muestra un programa que utiliza indicadores en un área de indicadores separada, definida en una tabla WORKING-STORAGE. Las DDS asociadas para el archivo son las mismas que en la Figura 58 en la página 160.

IBM International Business Machines

AS/400 ESPECIFICACIONES DE DESCRIPCION DE DATOS

GX21-9891-0 UM/050*
Impreso en España
*El número de hojas por área de relleno puede variar ligeramente.

Archivo		Instrucciones de Grabación	Signo	Descripción	Página de
Programador	Fecha		Tecla		

A

Número de Secuencia (1-5)	Tipo de Formulario (6)	And/Or/Coment. (A/O/*) (7)	No (N) (8)	Indicador (9-10)	No (N) (11)	Indicador (12-13)	No (N) (14)	Indicador (15-16)	Tipo Nombre o Espec (b/R/H/J/K/S/O) (17)	Reservado (18)	Nombre (19-28)	Referencia (R) (29)	Longitud (30-34)	Tipo Datos/Desplazamiento Teclado (35)	Posiciones Decimales (36-37)	Utilización (b/O/I/B/H/M/N/P) (38)	Ubicación Línea (39-41)	Ubicación Pos (42-44)	Funciones (45-80)
	A	* DDS DE ARCHIVO DE PANTALLA PARA EJEMPLOS DE INDICADORES																	
	A	*																	1
	A						2		R		FORMAT1							3	CF03(99 'FIN DE PROGRAMA')
	A																		CF05(51 'INFORME DIARIO')
	A																		CF09(52 'INFORME MENSUAL')
	A	*																	
	A													4			10	10	'NUMERO DE DEPARTAMENTO: '
	A										DEPTNO		5			I	10	32	
	A			5	01												20	26	'PRODUCIR INFORMES MENSUALES'
	A																		DSPATR(BL)
	A	*																	
	A														6		24	01	'F5 = INFORME DIARIO'
	A																24	26	'F9 = INFORME MENSUAL'
	A																24	53	'F3 = TERMINAR'
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		

Figura 55. Ejemplo de un Programa que Utiliza Indicadores en el Área de Registros sin Utilizar la Frase INDICATORS en la Sentencia–Datos de E/S: Especificaciones de Descripción

1 No se utiliza la palabra clave INDARA; los indicadores se almacenan en el área de registro con los campos de datos.

2 Se especifica un formato de registro, FORMAT1.

3 Se asocian tres indicadores con tres teclas de función. El indicador 99 se activará cuando pulse F3, y así sucesivamente.

4 Se define un campo para la salida.

5 El indicador 01 se define para hacer que el campo de constante asociado parpadee si está activo el indicador.

6 Las definiciones de teclas de función (F) se documentan en la pantalla de estación de trabajo.

```
5763CB1 V3R0M5                  Fuente AS/400 COBOL
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA  FECH/CAM
    1  000100 IDENTIFICATION DIVISION.                                                              03/09/94
    2  000200   PROGRAM-ID.     XMPLE71.                                                           03/22/94
       000300*    PROGRAMA EJEMPLO CON INDICADORES EN ÁREA DE REGISTRO.                             03/09/94
    3  000400   AUTHOR.       PROGRAMMER NAME.                                                    03/09/94
    4  000500   INSTALLATION. TORONTO COBOL DEVELOPMENT CENTRE.                                   03/09/94
    5  000600   DATE-WRITTEN. 12/08/88.                                                           03/09/94
    6  000070   DATE-COMPILED.  05/24/94 11:02:36   .
    7  000800 ENVIRONMENT DIVISION.                                                              03/09/94
    8  000900   CONFIGURATION SECTION.                                                           03/09/94
    9  001000   SOURCE-COMPUTER. IBM-AS400.                                                      03/25/94
   10  001100   OBJECT-COMPUTER. IBM-AS400.                                                      03/25/94
   11  001200   INPUT-OUTPUT SECTION.                                                            03/09/94
   12  001300   FILE-CONTROL.                                                                    03/09/94
   13  001400     SELECT DISPFILE                                                                03/09/94
   14  001500       ASSIGN TO WORKSTATION-DSPFILEX  1                                            03/09/94
   15  001600       ORGANIZATION IS TRANSACTION                                                  03/09/94
   16  001700       ACCESS       IS SEQUENTIAL.                                                  03/09/94
   17  001800 DATA DIVISION.                                                                    03/09/94
   18  001900   FILE SECTION.                                                                    03/09/94
   19  002000   FD  DISPFILE                                                                     03/09/94
   20  002100     LABEL RECORDS ARE OMITTED                                                      03/09/94
   21  002200     DATA RECORD IS DISP-REC.                                                       03/09/94
   22  002300   01  DISP-REC.                                                                   03/09/94
   23  002400       COPY DDS-ALL-FORMATS OF DSPFILEX.  2                                         03/09/94
   24 +000001       05  DSPFILEX-RECORD PIC X(8).                                 <-ALL-FMTS
      +000002*FORMATO ENTRADA:FORMAT1  DESDE ARCHIVO DSPFILEX DE BIBLIOTECA XMPLIB  <-ALL-FMTS
      +000003*                                                                    <-ALL-FMTS
   25 +000004       05  FORMAT1-I     REDEFINES DSPFILEX-RECORD.                  <-ALL-FMTS
   26 +000005           06 FORMAT1-I-INDIC.                                       <-ALL-FMTS
   27 +000006                07 IN99         PIC 1  INDIC 99.  3                 <-ALL-FMTS
      +000007*                          FIN DE PROGRAMA                           <-ALL-FMTS
   28 +000008                07 IN51         PIC 1  INDIC 51.                    <-ALL-FMTS
      +000009*                          INFORME DIARIO                            <-ALL-FMTS
   29 +000010                07 IN52         PIC 1  INDIC 52.                    <-ALL-FMTS
      +000011*                          INFORME MENSUAL                           <-ALL-FMTS
   30 +000012           06 DEPTNO            PIC X(5).                           <-ALL-FMTS
      +000013*FORMATO SALIDA:FORMAT1  DESDE ARCHIVO DSPFILEX DE BIBLIOTECA XMPLIB  <-ALL-FMTS
      +000014*                                                                    <-ALL-FMTS
   31 +000015       05  FORMAT1-O     REDEFINES DSPFILEX-RECORD.                  <-ALL-FMTS
   32 +000016           06 FORMAT1-O-INDIC.                                       <-ALL-FMTS
   33 +000017                07 IN01         PIC 1  INDIC 01.                    <-ALL-FMTS
       002500
   34  002600 WORKING-STORAGE SECTION.
   35  002700 01  CURRENT-DATE.
   36  002800    05  CURR-YEAR                 PIC 9(2).
   37  002900    05  CURR-MONTH                PIC 9(2).
   38  003000    05  CURR-DAY                  PIC 9(2).
   39  003100 01  INDIC-AREA.  4
   40  003200    05  IN01                      PIC 1.
   41  003300       88  NEW-MONTH  5            VALUE B"1".
   42  003400    05  IN51                      PIC 1.
   43  003500       88  WANT-DAILY             VALUE B"1".
   44  003600    05  IN52                      PIC 1.
   45  003700       88  WANT-MONTHLY           VALUE B"1".
   46  003800    05  IN99                      PIC 1.
   47  003900       88  NOT-END-OF-JOB         VALUE B"0".
   48  004000       88  END-OF-JOB             VALUE B"1".
   49  004100 PROCEDURE DIVISION.
       004200 XAMPLE3-MAIN.
   50  004300     OPEN I-O DISPFILE.
   51  004400     ACCEPT CURRENT-DATE FROM DATE.
   52  004500     SET NOT-END-OF-JOB TO TRUE.
   53  004600     PERFORM DISPLAY-SCREEN THRU READ-AND-PROCESS-SCREEN
       004700             UNTIL END-OF-JOB.
   54  004800     CLOSE DISPFILE.
   55  004900     STOP RUN.
       005000 DISPLAY-SCREEN.
   56  005100     MOVE ZEROS TO INDIC-AREA.  6
   57  005200     IF CURR-DAY = 01 THEN
   58  005300        SET NEW-MONTH TO TRUE.  7
   59  005400     MOVE CORR INDIC-AREA TO FORMAT1-O-INDIC.  8
   60  005500     WRITE DISP-REC FORMAT IS "FORMAT1".  9
       005600 READ-AND-PROCESS-SCREEN.
   61  005700     MOVE ZEROS TO INDIC-AREA.
   62  005800     READ DISPFILE FORMAT IS "FORMAT1".  10
```

Figura 56 (Parte 1 de 2). Ejemplo de un Programa que Utiliza Indicadores en el Área de Registros sin la Frase INDICATORS en la Sentencia–COBOL de E/S: Programa Fuente

```
5763CB1 V3R0M5                   Fuente AS/400 COBOL
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA  FECH/CAM
   63  005900     MOVE CORR FORMAT1-I-INDIC TO INDIC-AREA.  11
   64  006000     IF WANT-DAILY THEN
   65  006100        CALL "DAILY" USING DEPTNO  12
       006200     ELSE
   66  006300        IF WANT-MONTHLY THEN
   67  006400           CALL "MONTHLY" USING DEPTNO.
                       * * * * *  F I N  D E  F U E N T E  * *
5763CB1 V3R0M5                   Mensajes AS/400 COBOL
 INST
              * * * * *  F I N  D E  M E N S A J E S  * * * * *
                                    Resumen Mensajes
 Total    Info(0-4)    Aviso(5-19)      Error(20-29)    Grave(30-39)    Terminal(40-99)
     0          0             0                0               0                0
Registros fuente leídos . . . . . :   64
Registros de copia leídos . . . . :   17
Miembros de copia procesados  . . :   1
Errores de secuencia  . . . . . . :   0
Mensaje de gravedad emitido más alto:   0
 LBL0901 00  Programa XMPLE71 creado en biblioteca XMPLIB.
1          * * * * *  F I N  D E  C O M P I L A C I O N  * * * * *
```

Figura 56 (Parte 2 de 2). Ejemplo de un Programa que Utiliza Indicadores en el Área de Registros sin la Frase INDICATORS en la Sentencia–COBOL de E/S: Programa Fuente

1 El atributo del área de indicadores separada, SI, no se codifica en la cláusula ASSIGN.

2 La instrucción COPY con Formato 2 define los campos de datos y los indicadores en el área de registro.

3 Dado que el archivo no tiene un área de indicadores separada, los indicadores de respuesta y de opción se definen en el orden en que se utilizan en las DDS, y los números de indicadores se tratan como documentación.

4 Todos los indicadores utilizados por el programa se definen mediante nombres significativos en las entradas de descripción de datos en WORKING-STORAGE. Aquí se omiten los números de los indicadores porque no tienen efecto.

5 Para cada indicador, se asocia un nombre significativo de condición de nivel 88 con un valor para dicho indicador.

6 Inicializar el nivel de grupo a cero.

7 Si es el primer día del mes, se activa IN01 en WORKING-STORAGE.

8 Los indicadores adecuados a la salida de FORMAT1 se copian en el área de registros.

9 FORMAT1 se graba en la pantalla de estación de trabajo con datos y valores del indicador en el área de registro.

La frase INDICATORS no es necesaria porque no hay área de indicadores separada y se han establecido los valores del indicador en el área de registro mediante la instrucción anterior MOVE CORRESPONDING.

10 FORMAT1 se lee desde la pantalla de estación de trabajo, incluyendo datos e indicadores.

11 Los indicadores de respuesta para FORMAT1 se copian del área de registros para las entradas de descripción de datos en WORKING-STORAGE.

12 Si se pulsa F5, se procesa una llamada de programa.

```
5763CB1 V3R0M5                  Fuente AS/400 COBOL
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
    1  000100 IDENTIFICATION DIVISION.                                                              03/07/94
    2  000200   PROGRAM-ID.     XMPLE713.                                                          03/22/94
       000300*  PROGRAMA EJEMPLO - ARCHIVO CON INDICADORES EN ÁREA DE REGISTRO                      03/07/94
    3  000400   AUTHOR.       PROGRAMMER NAME.                                                    03/07/94
    4  000500   INSTALLATION. TORONTO COBOL DEVELOPMENT CENTRE.                                   03/07/94
    5  000600   DATE-WRITTEN. 12/10/88.                                                           03/07/94
    6  000070   DATE-COMPILED.  05/24/94 11:04:34   .
    7  000800 ENVIRONMENT DIVISION.                                                              03/07/94
    8  000900   CONFIGURATION SECTION.                                                           03/07/94
    9  001000   SOURCE-COMPUTER. IBM-AS400.                                                      03/07/94
   10  001100   OBJECT-COMPUTER. IBM-AS400.                                                      03/07/94
   11  001200   INPUT-OUTPUT SECTION.                                                            03/07/94
   12  001300   FILE-CONTROL.                                                                    03/07/94
   13  001400     SELECT DISPFILE                                                                03/07/94
   14  001500       ASSIGN TO WORKSTATION-DSPFILEX  1                                            03/22/94
   15  001600       ORGANIZATION IS TRANSACTION                                                  03/07/94
   16  001700       ACCESS       IS SEQUENTIAL.                                                  03/07/94
   17  001800 DATA DIVISION.                                                                    03/07/94
   18  001900   FILE SECTION.                                                                    03/07/94
   19  002000   FD  DISPFILE                                                                     03/07/94
   20  002100     LABEL RECORDS ARE OMITTED                                                      03/07/94
   21  002200     DATA RECORD IS DISP-REC.                                                       03/07/94
   22  002300   01  DISP-REC.                                                                   03/07/94
   23  002400       COPY DDS-ALL-FORMATS OF DSPFILEX.  2                                         03/22/94
   24 +000001       05  DSPFILEX-RECORD PIC X(8).                                <-ALL-FMTS
      +000002*FORMATO ENTRADA:FORMAT1  DESDE ARCHIVO DSPFILEX DE BIBLIOTECA XMPLIB  <-ALL-FMTS
      +000003*                                                                    <-ALL-FMTS
   25 +000004       05  FORMAT1-I     REDEFINES DSPFILEX-RECORD.                 <-ALL-FMTS
   26 +000005           06 FORMAT1-I-INDIC.                                       <-ALL-FMTS
   27 +000006                07 IN99         PIC 1  INDIC 99.  3                 <-ALL-FMTS
      +000007*                          FIN DE PROGRAMA                           <-ALL-FMTS
   28 +000008                07 IN51         PIC 1  INDIC 51.                    <-ALL-FMTS
      +000009*                          INFORME DIARIO                            <-ALL-FMTS
   29 +000010                07 IN52         PIC 1  INDIC 52.                    <-ALL-FMTS
      +000011*                          INFORME MENSUAL                           <-ALL-FMTS
   30 +000012           06 DEPTNO            PIC X(5).                           <-ALL-FMTS
      +000013*FORMATO SALIDA:FORMAT1  DESDE ARCHIVO DSPFILEX DE BIBLIOTECA XMPLIB  <-ALL-FMTS
      +000014*                                                                    <-ALL-FMTS
   31 +000015       05  FORMAT1-O     REDEFINES DSPFILEX-RECORD.                 <-ALL-FMTS
   32 +000016           06 FORMAT1-O-INDIC.                                       <-ALL-FMTS
   33 +000017                07 IN01         PIC 1  INDIC 01.                    <-ALL-FMTS
       002500
   34  002600 WORKING-STORAGE SECTION.
   35  002700 01  CURRENT-DATE.
   36  002800     05  CURR-YEAR                PIC 9(2).
   37  002900     05  CURR-MONTH               PIC 9(2).
   38  003000     05  CURR-DAY                 PIC 9(2).
       003100
   39  003200 77  IND-OFF                      PIC 1      VALUE B"0".
   40  003300 77  IND-ON                       PIC 1      VALUE B"1".
       003400
   41  003500 01  RESPONSE-INDICS.
   42  003600     05  END-OF-PROGRAM           PIC 1.  4
   43  003700     05  DAILY-REPORT             PIC 1.
   44  003800     05  MONTHLY-REPORT           PIC 1.
   45  003900 01  OPTION-INDICS.
   46  004000     05  NEW-MONTH                PIC 1.
       004100
   47  004200 PROCEDURE DIVISION.
       004300 XMPLE3-MAIN.
   48  004400     OPEN I-O DISPFILE.
   49  004500     ACCEPT CURRENT-DATE FROM DATE.
   50  004600     MOVE IND-OFF TO END-OF-PROGRAM.
   51  004700     PERFORM DISPLAY-SCREEN THRU READ-AND-PROCESS-SCREEN
       004800             UNTIL END-OF-PROGRAM = IND-ON.
   52  004900     CLOSE DISPFILE.
   53  005000     STOP RUN.
       005100
       005200 DISPLAY-SCREEN.
   54  005300     MOVE ZEROS TO OPTION-INDICS.
   55  005400     IF CURR-DAY = 01 THEN  5
   56  005500        MOVE IND-ON TO NEW-MONTH.
   57  005600     WRITE DISP-REC FORMAT IS "FORMAT1"  6
       005700                    INDICATORS ARE OPTION-INDICS.
       005800
```

Figura 57 (Parte 1 de 2). Ejemplo de un Programa que Utiliza Indicadores en el Área de Registros y la Frase INDICATORS en las Instrucciones de E/S–Programa Fuente COBOL

```
5763CB1 V3R0M5                   Fuente AS/400 COBOL
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
      005900 READ-AND-PROCESS-SCREEN.
   58 006000     MOVE ZEROS TO RESPONSE-INDICS.
   59 006100     READ DISPFILE FORMAT IS "FORMAT1"  7
      006200                   INDICATORS ARE RESPONSE-INDICS.  8
   60 006300     IF DAILY-REPORT = IND-ON THEN
   61 006400        CALL "DAILY" USING DEPTNO  9
      006500     ELSE
   62 006600        IF MONTHLY-REPORT = IND-ON THEN
   63 006700           CALL "MONTHLY" USING DEPTNO.
                        * * * * *  F I N  D E  F U E N T E  * *
5763CB1 V3R0M5                   Mensajes AS/400 COBOL
 INST
                     * * * * *  E N D  O F  M E S S A G E S  * * * * *
                                    Resumen Mensajes
 Total    Info(0-4)    Aviso(5-19)      Error(20-29)    Grave(30-39)    Terminal(40-99)
     0          0            0                0               0               0
Registros fuente leídos . . . . . :   67
Registros de copia leídos . . . . :   17
Miembros de copia procesados  . . :   1
Errores de secuencia  . . . . . :   0
Mensaje de gravedad más alta emitido:   0
 LBL0901 00  Programa XMPLE713 creado en biblioteca XMPLIB.
          * * * * *  F I N  D E  C O M P I L A C I O N  * * * * *
```

Figura 57 (Parte 2 de 2). Ejemplo de un Programa que Utiliza Indicadores en el Área de Registros y la Frase INDICATORS en las Instrucciones de E/S–Programa Fuente COBOL

1 El atributo del área de indicadores separada, SI, no se codifica en la cláusula ASSIGN.

2 La instrucción COPY con Formato 2 define los campos de datos y los indicadores en el área de registro.

3 Dado que el archivo no tiene un área de indicadores separada, los indicadores de respuesta y de opción se definen en el orden en que se utilizan en las DDS, y los números de indicadores se tratan como documentación.

4 Todos los indicadores que utiliza el programa se definen con nombres significativos en entradas de descripción de datos en WORKING-STORAGE. Aquí se omiten los números de los indicadores debido a que no tienen efecto. Los indicadores deben definirse en el orden necesario mediante el archivo de pantalla.

5 Si es el primer día del mes, se activa IN01 en WORKING-STORAGE.

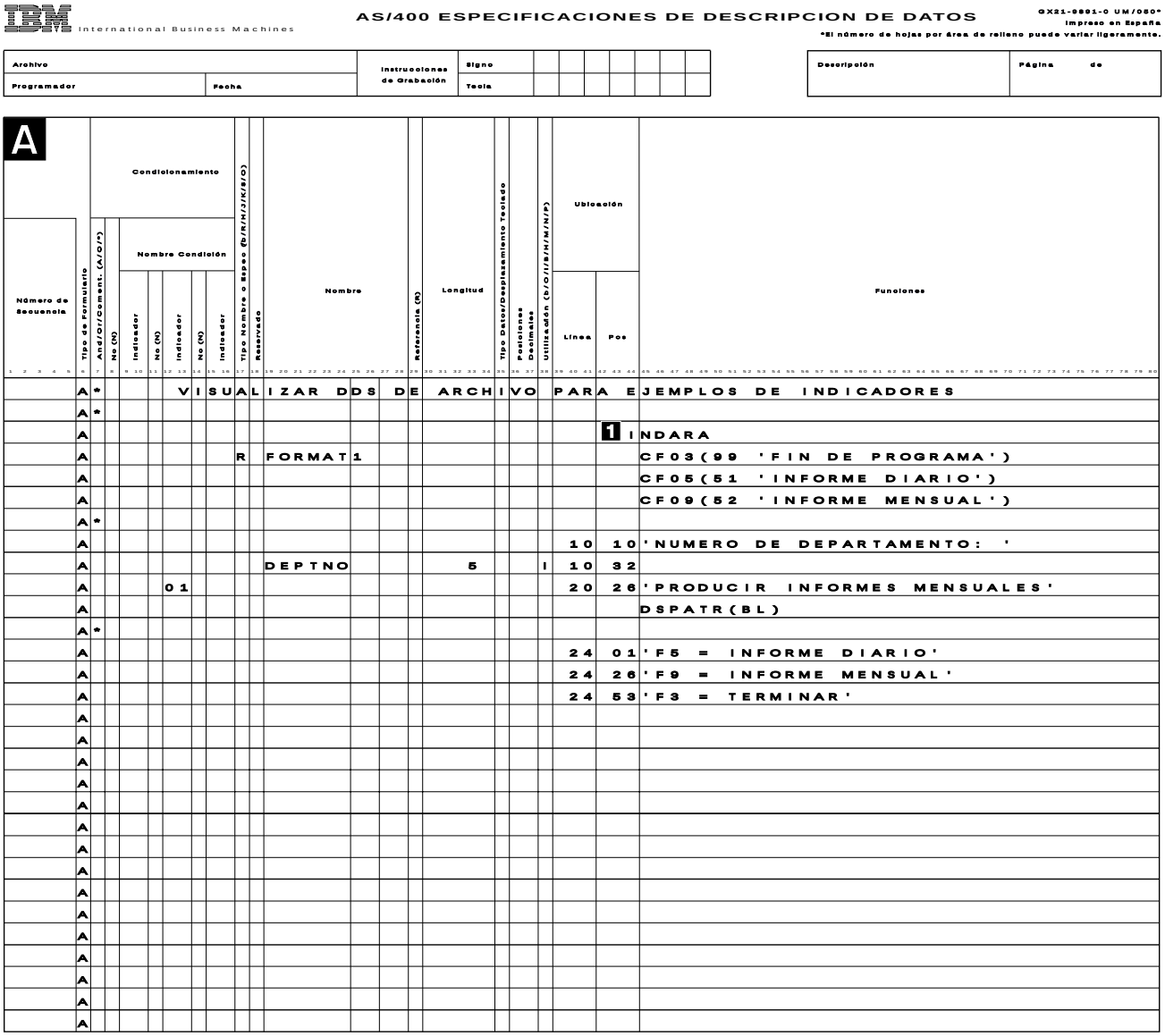
6 FORMAT1 se escribe en la pantalla de estación de trabajo:

- La frase INDICATORS hace que el contenido de la variable OPTION-INDICS se copie en el comienzo del área de registro.
- Los valores de datos y de indicador se escriben en la pantalla de estación de trabajo.

7 FORMAT1 se lee desde la pantalla de estación de trabajo, incluyendo datos e indicadores.

8 La frase INDICATORS hace que los bytes se copien desde el comienzo del área de registro a RESPONSE-INDICS.

9 Si se pulsa F5, se procesa una llamada de programa.



```
     A*     VISUALIZAR DDS DE ARCHIVO PARA EJEMPLOS DE INDICADORES
     A*
     A                                      INDARA
     A          R FORMAT1                   CF03(99 'FIN DE PROGRAMA')
     A                                      CF05(51 'INFORME DIARIO')
     A                                      CF09(52 'INFORME MENSUAL')
     A*
     A                                  10 10'NUMERO DE DEPARTAMENTO: '
     A            DEPTNO         5   I  10 32
     A     01                           20 26'PRODUCIR INFORMES MENSUALES'
     A                                      DSPATR(BL)
     A*
     A                                  24 01'F5 = INFORME DIARIO'
     A                                  24 26'F9 = INFORME MENSUAL'
     A                                  24 53'F3 = TERMINAR'
     A
     A
     A
     A
     A
     A
     A
     A
     A
     A
     A
     A
     A
     A
     A
```

Figura 58. Ejemplo de un Programa que Utiliza Indicadores en una Área de Separada de Indicadores, Definido en WORKING-STORAGE mediante la Sentencia COPY, Formato DDS

1 Se especifica la palabra clave INDARA; los indicadores se almacenan en un área de indicadores separada, no en el área de registro. Excepto para esta especificación, las DDS para este archivo son las mismas que las mostradas en la Figura 55 en la página 155.

```
5763CB1 V3R0M5                  Fuente AS/400 COBOL
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
    1  000100 IDENTIFICATION DIVISION.                                                          03/09/94
    2  000200 PROGRAM-ID.     XMPLE717.                                                         03/22/94
       000300* PROGRAMA EJEMPLO - ARCHIVO CON ÁREA DE INDICADORES SEPARADA                     03/09/94
    3  000400 AUTHOR.       PROGRAMMER NAME.                                                  03/09/94
    4  000500 INSTALLATION. TORONTO COBOL DEVELOPMENT CENTRE.                                 03/09/94
    5  000600 DATE-WRITTEN. 12/08/88.                                                         03/09/94
    6  000070 DATE-COMPILED.  05/24/94 12:53:17   .
    7  000800 ENVIRONMENT DIVISION.                                                           03/09/94
    8  000900 CONFIGURATION SECTION.                                                          03/09/94
    9  001000 SOURCE-COMPUTER. IBM-AS400.                                                     03/09/94
   10  001100 OBJECT-COMPUTER. IBM-AS400.                                                     03/09/94
   11  001200 INPUT-OUTPUT SECTION.                                                          03/09/94
   12  001300 FILE-CONTROL.                                                                  03/09/94
   13  001400     SELECT DISPFILE                                                            03/09/94
   14  001500         ASSIGN TO WORKSTATION-DSPFILE-SI [1]                                   03/22/94
   15  001600         ORGANIZATION IS TRANSACTION                                            03/09/94
   16  001700         ACCESS IS SEQUENTIAL.                                                  03/09/94
       001800                                                                                03/09/94
   17  001900 DATA DIVISION.                                                                 03/09/94
   18  002000 FILE SECTION.                                                                  03/09/94
   19  002100 FD  DISPFILE                                                                   03/09/94
   20  002200     LABEL RECORDS ARE OMITTED                                                  03/09/94
   21  002300     DATA RECORD IS DISP-REC.                                                   03/09/94
   22  002400 01  DISP-REC.                                                                 03/09/94
   23  002500     COPY DDS-ALL-FORMATS OF DSPFILE. [2]                                       03/22/94
   24 +000001       05  DSPFILE-RECORD PIC X(5).                                 <-ALL-FMTS
      +000002*FORMATO ENTRADA:FORMAT1 DESDE ARCHIVO DSPFILE  DE BIBLIOTECA XMPLIB  <-ALL-FMTS
      +000003*                                                                    <-ALL-FMTS
   25 +000004       05  FORMAT1-I     REDEFINES DSPFILE-RECORD.                  <-ALL-FMTS
   26 +000005           06 DEPTNO            PIC X(5).                           <-ALL-FMTS
      +000006*FORMATO SALIDA:FORMAT1 DESDE ARCHIVO DSPFILE   DE BIBLIOTECA XMPLIB  <-ALL-FMTS
      +000007*                                                                    <-ALL-FMTS
      +000008*      05  FORMAT1-O     REDEFINES DSPFILE-RECORD.                  <-ALL-FMTS
       002600
   27  002700 WORKING-STORAGE SECTION.
   28  002800 01  CURRENT-DATE.
   29  002900    05  CURR-YEAR                PIC 9(2).
   30  003000    05  CURR-MONTH               PIC 9(2).
   31  003100    05  CURR-DAY                 PIC 9(2).
       003200
   32  003300    77 IND-OFF                   PIC 1 VALUE B"0".
   33  003400    77 IND-ON                    PIC 1 VALUE B"1".
   34  003500 01  DISPFILE-INDICS.
   35  003600     COPY DDS-ALL-FORMATS-INDIC OF DSPFILE. [3]
   36 +000001       05  DSPFILE-RECORD.                                          <-ALL-FMTS
      +000002*FORMATO ENTRADA:FORMAT1 DESDE ARCHIVO DSPFILE  DE BIBLIOTECA XMPLIB  <-ALL-FMTS
      +000003*                                                                    <-ALL-FMTS
   37 +000004           06 FORMAT1-I-INDIC.                                       <-ALL-FMTS
   38 +000005                07 IN51         PIC 1  INDIC 51. [4]                <-ALL-FMTS
      +000006*                          INFORME DIARIO                            <-ALL-FMTS
   39 +000007                07 IN52         PIC 1  INDIC 52.                    <-ALL-FMTS
      +000008*                          INFORME MENSUAL                           <-ALL-FMTS
   40 +000009                07 IN99         PIC 1  INDIC 99.                    <-ALL-FMTS
      +000010*                          FIN DE PROGRAMA                           <-ALL-FMTS
      +000011*FORMATO SALIDA:FORMAT1  DESDE ARCHIVO DSPFILE  DE BIBLIOTECA XMPLIB  <-ALL-FMTS
      +000012*
   41 +000013           06 FORMAT1-O-INDIC.
   42 +000014                07 IN01         PIC 1  INDIC 01.
       003700
   43  003800 PROCEDURE DIVISION.
       003900
       004000 MAIN-PROCESS.
       004100
   44  004200     OPEN I-O DISPFILE.
   45  004300     ACCEPT CURRENT-DATE FROM DATE.
   46  004400     MOVE IND-OFF TO IN99 IN FORMAT1-I-INDIC.
   47  004500     PERFORM DISPLAY-SCREEN THRU READ-AND-PROCESS-SCREEN
       004600         UNTIL IN99 IN FORMAT1-I-INDIC = IND-ON.
   48  004700     CLOSE DISPFILE.
   49  004800     STOP RUN.
       004900
       005000 DISPLAY-SCREEN.
       005100
   50  005200     MOVE ZEROS TO FORMAT1-O-INDIC.
   51  005300     IF CURR-DAY = 01 THEN
   52  005400        MOVE IND-ON TO IN01 IN FORMAT1-O-INDIC. [5]
```

Figura 59 (Parte 1 de 2). Listado COBOL que Utiliza Indicadores en un Área de Indicadores Separada

```
5763CB1 V3R0M5                   Fuente AS/400 COBOL
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
   53  005500     WRITE DISP-REC FORMAT IS "FORMAT1"
       005600                    INDICATORS ARE FORMAT1-O-INDIC.  6
       005700
       005800 READ-AND-PROCESS-SCREEN.
       005900
   54  006000     MOVE ZEROS TO FORMAT1-I-INDIC.
   55  006100     READ DISPFILE FORMAT IS "FORMAT1"
       006200                   INDICATORS ARE FORMAT1-I-INDIC.  7
   56  006300     IF IN51 IN FORMAT1-I-INDIC = IND-ON THEN
   57  006400         CALL "DAILY" USING DEPTNO  8
       006500     ELSE
   58  006600         IF IN52 IN FORMAT1-I-INDIC = IND-ON THEN
   59  006700             CALL "MONTHLY" USING DEPTNO.
                        * * * * *   F I N   D E   F U E N T E  * *
5763CB1 V3R0M5                   Mensajes AS/400 COBOL
 INST
*   23  MSGID: LBL0600  GRAVEDAD: 10  NUMSEC:  000250
       Mensaje . . . . :   No se encontraron campos OUTPUT para formato FORMAT1.
              * * * * *   F I N   D E   M E N S A J E S   * * * * *
                                    Resumen Mensajes
 Total    Info(0-4)    Aviso(5-19)      Error(20-29)    Grave(30-39)    Terminal(40-99)
     1          0             1                0               0                0
Registros fuente leídos . . . . . :   67
Registros de copia leídos . . . . :   22
Miembros de copia procesados  . . :   2
Errores de secuencia  . . . . . . :   0
Mensaje de gravedad más alta emitido:   10
 LBL0901 00  Programa XMPLE717 creado en biblioteca XMPLIB.
          * * * * *   F I N   D E   C O M P I L A C I O N   * * * * *
```

Figura 59 (Parte 2 de 2). Listado COBOL que Utiliza Indicadores en un Área de Indicadores Separada

1 El atributo del área de indicadores separada, SI, se especifica en la cláusula ASSIGN.

2 La instrucción COPY con Formato 2 genera descripciones de datos en el área de registro solamente para campos de datos. Las entradas de descripción de datos para los indicadores no se generan debido a que el área de indicadores separada se ha especificado para el archivo.

3 La instrucción COPY con Formato 2, con el atributo INDICATOR, INDIC, define las entradas de descripción de datos en WORKING-STORAGE para todos los indicadores utilizados en las DDS para el formato de registro del archivo.

4 Debido a que el archivo tiene un área de indicadores separada, los números de los indicadores utilizados en las entradas de descripción de datos no se tratan como documentación.

5 Si es el primer día del mes, se activa IN01 (01) en el área de indicadores separada para FORMAT1.

6 La frase INDICATORS es necesaria para enviar los valores de los indicadores a la pantalla de estación de trabajo.

7 La frase INDICATORS es necesaria para recibir valores de los indicadores desde la pantalla de estación de trabajo. Si pulsa F5, se activa IN51.

8 Si se ha activado IN51, se procesa una llamada de programa.

```
5763CB1 V3R0M5                  Fuente AS/400 COBOL
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
    1  000100 IDENTIFICATION DIVISION.                                                          01/22/94
    2  000200 PROGRAM-ID.     XMPLE720.                                                         03/22/94
       000300*    PROGRAMA EJEMPLO                                                              01/22/94
       000400*    ARCHIVO CON ÁREA DE IND.SEPARADA EN ÁREA ALMACENAMIENTO                       01/22/94
    3  000500 AUTHOR.       PROGRAMMER NAME.                                                    01/22/94
    4  000600 INSTALLATION. TORONTO COBOL DEVELOPMENT CENTRE.                                   01/22/94
    5  000700 DATE-WRITTEN. 12/08/88.                                                           01/22/94
    6  000080 DATE-COMPILED.  05/24/94 12:46:00   .
    7  000900 ENVIRONMENT DIVISION.                                                             01/22/94
    8  001000 CONFIGURATION SECTION.                                                            01/22/94
    9  001100 SOURCE-COMPUTER. IBM-AS400.                                                       01/22/94
   10  001200 OBJECT-COMPUTER. IBM-AS400.                                                       01/22/94
   11  001300 INPUT-OUTPUT SECTION.                                                             01/22/94
   12  001400 FILE-CONTROL.                                                                     01/22/94
   13  001500     SELECT DISPFILE                                                               01/22/94
   14  001600         ASSIGN TO WORKSTATION-DSPFILE-SI [1]                                      03/22/94
   15  001700         ORGANIZATION IS TRANSACTION                                               01/22/94
   16  001800         ACCESS IS SEQUENTIAL.                                                     01/22/94
       001900                                                                                   01/22/94
   17  002000 DATA DIVISION.                                                                    01/22/94
   18  002100 FILE SECTION.                                                                     01/22/94
   19  002200 FD  DISPFILE                                                                      01/22/94
   20  002300     LABEL RECORDS ARE OMITTED                                                     01/22/94
   21  002400     DATA RECORD IS DISP-REC.                                                      01/22/94
   22  002500 01  DISP-REC.                                                                    01/22/94
   23  002600     COPY DDS-ALL-FORMATS OF DSPFILE. [2]                                          03/22/94
   24 +000001       05  DSPFILE-RECORD PIC X(5).                                 <-ALL-FMTS
      +000002*FORMATO ENTRADA:FORMAT1 DESDE ARCHIVO DSPFILE  DE BIBLIOTECA XMPLIB  <-ALL-FMTS
      +000003*                                                                    <-ALL-FMTS
   25 +000004       05  FORMAT1-I     REDEFINES DSPFILE-RECORD.                  <-ALL-FMTS
   26 +000005           06 DEPTNO            PIC X(5).                           <-ALL-FMTS
      +000006*FORMATO SALIDA:FORMAT1 DESDE ARCHIVO DSPFILE   DE BIBLIOTECA XMPLIB  <-ALL-FMTS
      +000007*                                                                    <-ALL-FMTS
      +000008*      05  FORMAT1-O     REDEFINES DSPFILE-RECORD.                  <-ALL-FMTS
       002700
   27  002800 WORKING-STORAGE SECTION.
   28  002900 01  CURRENT-DATE.
   29  003000    05  CURR-YEAR                 PIC 9(2).
   30  003100    05  CURR-MONTH                PIC 9(2).
   31  003200    05  CURR-DAY                  PIC 9(2).
       003300
   32  003400 01  INDIC-AREA.
   33  003500    05 INDIC-TABLE  OCCURS 99      PIC 1    INDICATOR 1. [3]
   34  003600      88  IND-OFF                           VALUE B"0".
   35  003700      88  IND-ON                            VALUE B"1".
       003800
   36  003900 01  DISPFILE-INDIC-USAGE.
   37  004000    05  IND-NEW-MONTH              PIC 9(2) VALUE 01.
   38  004100    05  IND-DAILY                  PIC 9(2) VALUE 51. [4]
   39  004200    05  IND-MONTHLY                PIC 9(2) VALUE 52.
   40  004300    05  IND-EOJ                    PIC 9(2) VALUE 99.
       004400
   41  004500 PROCEDURE DIVISION.
       004600
       004700 XMPLE-MAIN.
   42  004800     OPEN I-O DISPFILE.
   43  004900     ACCEPT CURRENT-DATE FROM DATE.
   44  005000     SET IND-OFF (IND-EOJ) TO TRUE.
   45  005100     PERFORM DISPLAY-SCREEN THRU READ-AND-PROCESS-SCREEN
       005200         UNTIL IND-ON (IND-EOJ).
   46  005300     CLOSE DISPFILE.
   47  005400     STOP RUN.
       005500
       005600 DISPLAY-SCREEN.
       005700
   48  005800     MOVE ZEROS TO INDIC-AREA.
   49  005900     IF CURR-DAY = 01 THEN
   50  006000        SET IND-ON (IND-NEW-MONTH) TO TRUE. [5]
   51  006100     WRITE DISP-REC FORMAT IS "FORMAT1"
       006200                    INDICATORS ARE INDIC-TABLE. [6]
       006300
       006400 READ-AND-PROCESS-SCREEN.
       006500
   52  006600     READ DISPFILE FORMAT IS "FORMAT1"
       006700                   INDICATORS ARE INDIC-TABLE. [7]
```

Figura 60 (Parte 1 de 2). Ejemplo de un Programa que Utiliza Indicadores en una Área Separada de Indicadores, Definido en una Tabla en WORKING-STORAGE

```
5763CB1 V3R0M5                   Fuente AS/400 COBOL
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
   53  006800     IF IND-ON (IND-DAILY) THEN 8
   54  006900         CALL "DAILY" USING DEPTNO
       007000     ELSE
   55  007100         IF IND-ON (IND-MONTHLY) THEN
   56  007200             CALL "MONTHLY" USING DEPTNO.
                        * * * * *   F I N   D E   F U E N T E  * *
5763CB1 V3R0M5                   Mensajes AS/400 COBOL
 INST
*   23  MSGID: LBL0600  GRAVEDAD: 10  NUMSEC:  000260
       Mensaje . . . . :   No se encontraron campos OUTPUT para formato FORMAT1.
              * * * * *   F I N   D E   M E N S A J E S   * * * * *
                                    Resumen Mensajes
 Total    Info(0-4)    Aviso(5-19)      Error(20-29)    Grave(30-39)    Terminal(40-99)
     1          0            1                0               0               0
Registros fuente leídos . . . . . :   72
Registros de copia leídos . . . . :   8
Miembros de copia procesados  . . :   1
Errores de secuencia  . . . . . :   0
Mensaje de gravedad más alta emitido:   10
 LBL0901 00  Programa XMPLE720 creado en biblioteca XMPLIB.
          * * * * *   F I N   D E   C O M P I L A C I O N   * * * * *
```

Figura 60 (Parte 2 de 2). Ejemplo de un Programa que Utiliza Indicadores en una Área Separada de Indicadores, Definido en una Tabla en WORKING-STORAGE

1 El atributo del área de indicadores separada, SI, se especifica en la cláusula ASSIGN.

2 La instrucción COPY con Formato 2 genera campos en el área de registros sólo para campos de datos.

3 Se define una tabla de 99 ítems de datos booleanos en WORKING-STORAGE. La cláusula INDICATOR para esta entrada de descripción de datos hace que estos ítems de datos se asocien con los indicadores 1 a 99 respectivamente. La utilización de esta tabla puede tener como resultado un rendimiento mejorado si se compara con la utilización de un ítem de grupo con múltiples entradas subordinadas para indicadores individuales.

4 Se define una serie de ítems de datos en WORKING-STORAGE para proporcionar nombres de subíndices significativos con los que referirse a la tabla de indicadores. La utilización de tales ítems de datos no es necesaria.

5 Si es el primer día del mes, se activa INDIC-TABLE (01) en el área de indicadores separada para FORMAT1.

6 La frase INDICATOR es necesaria para enviar los valores de los indicadores a la pantalla de estación de trabajo.

7 La frase INDICATOR es necesaria para recibir valores de los indicadores desde la pantalla de estación de trabajo. Si se pulsa F5, se activará INDIC-TABLE (51).

8 Si se ha activado INDIC-TABLE (51), se llama al programa DAILY.

Subarchivos

Pueden especificarse subarchivos en las DDS para un archivo de pantalla que permitan al usuario manejar varios registros del mismo tipo en una pantalla. Consulte la Figura 61 en la página 165 para ver un ejemplo de una pantalla de subarchivo. Un **subarchivo** es un grupo de registros que se leen desde un dispositivo de pantalla o se graban a dicho dispositivo. El programa procesa un registro a la vez,

pero el sistema operativo y la estación de trabajo envían y reciben bloques de registros. Si se transmiten más registros de los que pueden visualizarse en la pantalla a la vez, el operador de la estación de trabajo puede pasar de una página del bloque de registros a otra sin devolver el control al programa.

Los registros que se han de incluir en un subarchivo se especifican en las DDS para el archivo. El número de registros que pueden contenerse en un subarchivo también debe especificarse en las DDS. Un archivo puede contener más de un subarchivo: sin embargo, en un dispositivo sólo pueden estar activos doce subarchivos de forma simultánea. Pueden visualizarse doce subarchivos en un dispositivo a la vez.

Las DDS de un subarchivo constan de dos formatos de registros: un formato de registros de subarchivos y un formato de registros de control de subarchivos.

El formato de registro de subarchivo contiene las descripciones de campo para los registros en el subarchivo. Las especificaciones del formato de registro de subarchivo en una operación READ, WRITE o REWRITE hace que se procese el registro del subarchivo especificado, pero no afecta directamente a los datos visualizados.

La especificación del formato de registro de control del subarchivo en la instrucción READ o WRITE hace que se realicen las operaciones de lectura física, escritura o configuración de un subarchivo. La Figura 62 en la página 167 muestra un ejemplo de las DDS para un formato de registro de subarchivo, y la Figura 63 en la página 169 muestra un ejemplo de las DDS para un formato de registro de control de subarchivo.

Para la descripción de la visualización de los registros en un subarchivo, y para la descripción de las palabras clave que pueden especificarse para un subarchivo, consulte las publicaciones *Guía para la Gestión de Datos* y *DDS Reference*.

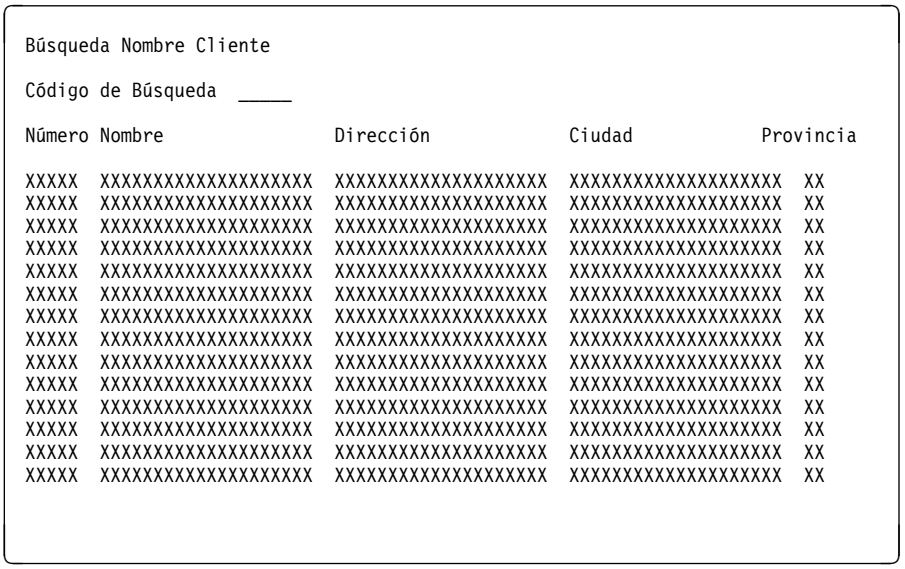


Figura 61. Pantalla de Subarchivo

Para utilizar un subarchivo en un archivo de pantalla de un programa COBOL deberá especificar la frase SUBFILE con la operación de entrada/salida. Las operaciones válidas de subarchivo son:

- READ SUBFILE nombre-archivo RECORD
- WRITE SUBFILE nombre-registro
- REWRITE SUBFILE nombre-registro.

Los subarchivos pueden procesarse secuencialmente con la instrucción READ SUBFILE NEXT MODIFIED, o procesarse al azar especificando un valor de clave relativa. Una clave relativa es un número sin signo que el sistema puede utilizar directamente para ubicar un registro en un archivo.

El archivo TRANSACTION debe ser un archivo descrito externamente. En COBOL, todos los accesos al subarchivo se realizan con un número de registro relativo. Si las frases SUBFILE se utilizan con un archivo TRANSACTION, la sentencia SELECT en la División de Entorno debe indicar ACCESS MODE IS DYNAMIC y debe especificar la clave RELATIVE KEY a utilizar.

Si un archivo de pantalla adquiere más de un dispositivo de pantalla, habrá un subarchivo separado para cada dispositivo de pantalla individual. Si se creó un subarchivo para un dispositivo de pantalla particular adquirido por un archivo TRANSACTION, todas las operaciones de entrada que hacen referencia a un formato de registro para el subarchivo se realizan sobre el subarchivo que pertenece a dicho dispositivo. Consulte el análisis de la frase TERMINAL en la página 191 de este capítulo para saber determinar cuál es el dispositivo que se utiliza. Cualquier operación que haga referencia al nombre del formato de registro que no está designado como un subarchivo se procesa como una operación de entrada/salida dirigida directamente al dispositivo de pantalla.

Utilización de Subarchivos

Algunas utilizaciones típicas de los subarchivos incluyen:

Utilización	Significado
Sólo Visualizar	El usuario de la estación de trabajo revisa la pantalla.
Visualizar con Selección	El usuario solicita más información acerca de uno de los ítems en la pantalla.
Modificación	El usuario modifica uno o más registros.
Sólo Entrada (sin comprobación de validez)	Se utiliza un subarchivo para una función de entrada de datos.
Sólo Entrada (con comprobación de validez)	Se utiliza un subarchivo para una función de entrada de datos, y se comprueban los registros.
Combinación de Tareas	Puede utilizarse un subarchivo como visualización con modificación.

IBM International Business Machines

AS/400 ESPECIFICACIONES DE DESCRIPCION DE DATOS

GX21-8081-0 UM/080* Impreso en España
*El número de hojas por área de relleno puede variar ligeramente.

Archivo		Instrucciones de Grabación	Signo							Descripción	Página de
Programador	Fecha		Tecla								

A

Número de Secuencia	Tipo de Formulario	Condicionamiento: And/Or/Coment. (A/O/*)	Condicionamiento: No (N)	Condicionamiento: Nombre Condición: Indicador	No (N)	Indicador	No (N)	Indicador	Tipo Nombre o Espac (b/R/H/J/K/S/C)	Reservado	Nombre	Referencia (R)	Longitud	Tipo Datos/Desplazamiento Teclado	Posiciones Decimales	Utilización (b/O/I/B/H/M/N/P)	Ubicación: Línea	Ubicación: Pos	Funciones
	A	*									DDS PARA EL ARCHIVO DE DISPOSITIVO DE PANTALLA PAYUPDTD								
	A	*									ACTUALIZACION DE PAGO INTERACTIVO DE CUENTAS POR COBRAR								
	A	*																	
	A								R		SUBARCH1							1	SFL
	A																		TEXT('SUBARCH PARA PAGO CLIENTE')
	A	*																	
	A										ACPPMT		4	A		I	5	4	TEXT('ACEPTAR PAGO')
	A																2		VALORES('*SI' '*NO') 3
	A			51															DSPATR(RI MDT)
	A		N	51															DSPATR(ND PR)
	A	*																	
	A										CUST		5			B	5	15	TEXT('NUMERO CLIENTE')
	A			52													4		DSPATR(RI)
	A			53															DSPATR(ND)
	A			54															DSPATR(PR)
	A	*																	
	A										AMPAID		8		02	B	5	24	TEXT('IMPORTE PAGADO')
	A																		CHECK(FE) 5
	A																		AUTO(RAB) 6
	A																		CMP(GT 0) 7
	A			52															DSPATR(RI)
	A			53															DSPATR(ND)
	A			54															DSPATR(PR)
	A	*																	
	A										ECPMSG		31	A		O	5	37	TEXT('MENSAJE DE EXCEPCION')
	A			52															DSPATR(RI)
	A			53															DSPATR(ND)
	A			54															DSPATR(BL)
	A	*																	
	A										OVRPMT		8	Y	2	O	5	70	TEXT('EXCESO EN EL PAGO')
	A																	8	EDTCDE(1)
	A			55														9	DSPATR(BL)
	A		N	56															DSPATR(ND)
	A	*																	
	A										STSCDE		1	A		H			TEXT('CODIGO DE ESTADO')

Figura 62. Especificaciones de Descripción de Datos para un Formato de Registro de Subarchivo

Las especificaciones de descripción de datos (DDS) para un formato de registro de subarchivo describe los registros en un subarchivo:

1 La palabra clave SFL identifica el formato de registro como un subarchivo.

2 Las entradas de línea y posición definen la ubicación de los campos en la pantalla.

3 La palabra clave VALUES especifica que el usuario sólo puede especificar *YES o *NO como valores para el campo ACPPMT.

4 Las entradas de utilización definen si el campo denominado debe ser un campo de salida (O), entrada (I), salida/entrada (B) u oculto (H).

5 La entrada CHECK(FE) especifica que el usuario no puede saltar al siguiente campo de entrada sin pulsar una de las teclas de salida de campo.

6 La entrada AUTO(RAB) especifica que los datos introducidos en el campo AMPAID van a ser justificados a la derecha automáticamente, y los caracteres iniciales van a rellenarse con espacios en blanco.

7 La entrada CMP(GT 0) especifica que los datos introducidos para el campo AMPAID deben compararse con cero para asegurar que el valor es mayor que cero.

8 La palabra clave EDTCDE especifica la edición deseada para el campo de salida OVRPMT. EDTCDE(1) indica que debe imprimirse el campo OVRPMT con puntos, comas decimales y sin signos. Asimismo, se imprimirá un saldo de cero y se suprimirán los ceros iniciales.

9 La palabra clave DSPATR se utiliza para especificar los atributos de pantalla para el campo nombrado cuando el estado del indicador correspondiente es cierto. Los atributos especificados son:

- BL (parpadeo)
- RI (contraste invertido)
- PR (protección)
- MDT (establece indicador de datos modificados)
- ND (no visualizar)

IBM International Business Machines

AS/400 ESPECIFICACIONES DE DESCRIPCION DE DATOS

GX21-8081-0 UM/0000-
Impreso en España.
*El número de hojas por área de relleno puede variar ligeramente.

Archivo		Instrucciones de Grabación	Signo								Descripción	Página de
Programador	Fecha		Tecla									

A

Número de Secuencia	Tipo de Formulario	And/Or/Coment. (A/O/*)	Condicionamiento: No (N)	Condicionamiento: Nombre Condición: Indicador	Condicionamiento: Nombre Condición: No (N)	Condicionamiento: Nombre Condición: Indicador	Condicionamiento: Nombre Condición: No (N)	Condicionamiento: Nombre Condición: Indicador	Tipo Nombre o Base (B/N/H/J/K/S/O)	Reservado	Nombre	Referencia (R)	Longitud	Tipo Dato/Desplazamiento Teclado	Posiciones Decimales	Utilización (B/O/I/E/H/M/N/P)	Ubicación: Línea	Ubicación: Pos	Funciones
	A								R		CONTROL1								TEXT('CONTROL SUBARCHIVO')
	A																		SFLCTL(SUBARCH1) 1
	A																		SFLSIZ(17) 2
	A																		SFLPAG(17) 3
	A			61															SFLCLR 4
	A			62															SFLDSP 5
	A			62															SFLDSPCTL 6
	A																		RECUBRIMIENTO
	A																		BLOQU 7
	A	*																	
	A																		HELP(99 'TECLA DE AYUDA') 8
	A																		CA12(98 'FIN DE ACTUALIZACION PAGO')
	A																		CA11(97 'IGNORAR ENTRADA')
	A	*																	
	A			99														9	SFLMSG('F11-IGNORAR ENTR NO VALIDA+
	A																		F12 - FIN DE ACTUALI+
	A																		ZACION DE PAGO')
	A	*																	
	A																1	2	'SOLICITUD ACTUALIZ. PAGO CLIENTE'
	A																1	65	'FECHA'
	A																1	78	FECHA EDTCDE(Y)
	A			63													3	2	'ACEPTAR'
	A			63													4	2	'PAGO'
	A																3	14	'CLIENTE'
	A																3	26	'PAGO'
	A			64													3	37	'MENSAJE DE EXCEPCION'
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		

Figura 63. Especificaciones de Descripción de Datos para un Formato de Registro de Control de Subarchivo

El formato de registro de control del subarchivo define los atributos del subarchivo, el campo de entrada de búsqueda, las constantes y las teclas de mandatos. Las palabras clave utilizadas indican lo siguiente:

1 SFLCTL identifica este registro como un registro de control de subarchivo y nombra los registros de subarchivo asociados (SUBFILE1).

2 SFLSIZ indica el número total de registros a incluir en el subarchivo (17).

3 SFLPAG indica el número total de registros en una pagina (17).

4 SFLCLR indica cuándo debe borrarse el subarchivo (cuando el indicador 61 está activo).

5 SFLDSP indica cuándo se visualiza el subarchivo (cuando el indicador 62 está activo).

6 SFLDSPCTL indica cuándo se visualiza el registro de control de subarchivo (cuando el indicador 62 está activo).

7 La palabra clave LOCK impide que el usuario de la estación de trabajo utilice el teclado cuando el formato de registro CONTROL1 se visualiza inicialmente.

8 HELP permite que el usuario pulse la tecla Ayuda y active el indicador 99.

9 SFLMSG identifica la constante como un mensaje que se visualiza si se activa el indicador 99.

Además de la información de control, el formato de registro de control de subarchivo define las constantes a utilizar como encabezamientos de columna para el formato de registro de subarchivo. Consulte la Figura 63 en la página 169 para ver un ejemplo de formato de registro de control de subarchivo.

Archivos de Múltiples Dispositivos y Archivos de Dispositivo Único

Un **archivo de múltiples dispositivos** es un archivo de pantalla o un archivo de función de comunicaciones entre sistemas (ICF). Un archivo de múltiples dispositivos puede adquirir más de un dispositivo de programa. Para obtener un ejemplo de la utilización de archivos de múltiples dispositivos, consulte la Figura 64 en la página 172.

Un **archivo de dispositivo único** es un archivo de dispositivo creado con un sólo dispositivo de programa definido para él. Los archivos de impresora, de diskette y de cinta son archivos de dispositivo único. Los archivos de pantalla y los archivos de función de comunicaciones entre sistemas (ICF) creados con un número máximo de un dispositivo de programa también son archivos de dispositivo único.

Un archivo de pantalla puede tener múltiples dispositivos de programa cuando el parámetro MAXDEV del mandato CRTDSPF es mayor que 1. Si se especifica *NONE para el parámetro DEV de este mandato, se debe suministrar el nombre de un dispositivo de pantalla *antes* de utilizar cualquier campo relacionado con el archivo.

Para más información acerca de la creación y utilización de un archivo de pantalla, consulte la publicación *Guía para la Gestión de Datos*.

Los archivos ICF pueden tener dispositivos múltiples de programa cuando el parámetro MAXPGMDEV del mandato CRTICFF sea mayor que 1. Para más información acerca de la creación y utilización de archivos ICF, consulte la *ICF Programmer's Guide*.

COBOL determina en tiempo de ejecución si un archivo es un archivo de dispositivo único o un archivo de múltiples dispositivos, basándose en si el archivo *es capaz* de tener varios dispositivos. El número real de dispositivos adquiridos no afecta a si un archivo se considera de único dispositivo o de múltiples dispositivos. Si un archivo es un archivo de dispositivo único o múltiple *no* se determina en tiempo de compilación; esta determinación se basa en la descripción actual de la pantalla o del archivo ICF.

Para archivos de múltiples dispositivos, si se ha de utilizar un dispositivo de programa particular en una instrucción de E/S, tal dispositivo se especifica mediante la frase TERMINAL. La frase TERMINAL también puede especificarse para un archivo de dispositivo único.

Las páginas siguientes contienen un ejemplo que ilustra la utilización de archivos de múltiples dispositivos. El programa utiliza un archivo de pantalla y está pensado para ejecutarse en modo de proceso por lotes. El programa adquiere terminales e invita a esos terminales mediante una pantalla de inicio de sesión. Después de invitar a los terminales, éstos se sondean. Si nadie inicia la sesión antes de que el tiempo de espera expire, el programa finaliza. Si se entra una contraseña válida, el usuario está autorizado a actualizar un archivo de empleados llamando a otro programa COBOL. Una vez completada la actualización, se invita al dispositivo de nuevo y los terminales se sondean otra vez.

IBM International Business Machines

AS/400 ESPECIFICACIONES DE DESCRIPCION DE DATOS

GX21-9891-0 UM/050*
Impreso en España
*El número de hojas por área de relleno puede variar ligeramente.

Archivo		Instrucciones de Grabación	Signo								Descripción	Página de
Programador	Fecha		Tecla									

A

Número de Secuencia	Tipo de Formulario	Condicionamiento: And/Or/Coment. (A/O/*)	Condicionamiento: No (N)	Condicionamiento: Nombre Condición: Indicador	Condicionamiento: Nombre Condición: No (N)	Condicionamiento: Nombre Condición: Indicador	Condicionamiento: Nombre Condición: No (N)	Condicionamiento: Nombre Condición: Indicador	Tipo Nombre o Espac (B/R/H/J/K/S/O)	Reservado	Nombre	Referencia (R)	Longitud	Tipo Dato/Desplazamiento Teclado	Posiciones Decimales	Utilización (b/O/I/B/H/M/N/P)	Ubicación: Línea	Ubicación: Pos	Funciones
1 2 3 4 5	6	7	8	9 10	11 12	13 14	15 16		17	18	19 20 21 22 23 24 25 26 27 28	29	30 31 32 33 34	35	36 37	38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
	A	*																	
	A	*	DDS PARA EL ARCHIVO DE PANTALLA DE MULTIPLES DISPOSITIVOS																
	A	*																	
	A								R		SIGNON								INVITE 1
	A															O	5	20	'bbbbbbbbbbbbbbbbbbbb'
	A																		DSPATR(RI)
	A															O	6	20	'bb'
	A																		DSPATR(RI)
																O	6	38	'bb'
																			DSPATR(RI)
																O	7	20	'bb'
																			DSPATR(RI)
																O	7	27	'M D F'
																			DSPATR(HI BL)
																O	7	38	'bb'
																			DSPATR(RI)
																O	8	20	'bb'
																			DSPATR(RI)
																O	8	38	'bb'
	A																		DSPATR(RI)
	A															O	9	20	'bbbbbbbbbbbbbbbbbbbb'
	A																		DSPATR(RI)
	A															O	20	20	'INICIE LA SESION'
	A																		DSPATR(HI)
	A				PASSWORD								10A			I	20	43	DSPATR(PC ND)
	A				WRONG								20A			O	21	43	
	A			R	UPDATE														
	A															O	3	5	'ACTUALIZACION ARCHIVO DE PERSONAL'
	A																		DSPATR(BL)
	A															O	7	5	'TECLEE NUMERO EMPLEADO +
	A																		A ACTUALIZAR'
	A				NUM								7A			I	7	44	DSPATR(RI PC)
	A			R	EMPLOYEE														
	A															O	3	5	'NUMERO EMPLEADO'
	A				NUM								7A			B	3	25	DSPATR(PC)
	A															O	5	5	'NOMBRE EMPLEADO'
	A				NAME								30A			B	5	25	DSPATR(PC)
	A															O	7	5	'DIRECCION EMPLEADO'
	A															O	9	5	'CALLE'
	A				STREET								30A			B	9	25	DSPATR(PC)
	A															O	11	5	'NUMERO APARTAMENTO'
	A				APTNO								5A			B	11	25	'DSPATR(PC)
	A															O	13	5	'CIUDAD'
	A				CITY								20A			B	13	25	DSPATR(PC)
	A															O	15	5	'PROVINCIA'
	A				PROV								20A			B	15	25	DSPATR(PC)
	A			R	RECOVERY														
	A															O	3	5	'EL NUMERO DE EMPLEADO QUE HA +
	A																		DADO NO ES VALIDO'
	A															O	6	5	'TECLEE S PARA REINTENTAR'
	A															O	8	5	'TECLEE N PARA SALIR'
	A				ANSWER								1X			I	10	5	DSPATR(RI PC)
	A																		VALORES('S' 'N')
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		

1 El formato SIGNON tiene la palabra clave INVITE asociada. Así, si se utiliza el formato SIGNON en una sentencia WRITE, se invitará al dispositivo en el que se está escribiendo.

Figura 64 (Parte 1 de 3). Ejemplo de Utilización de Archivos de Múltiples Dispositivos

IBM International Business Machines

AS/400 ESPECIFICACIONES DE DESCRIPCION DE DATOS

GX21-9891-0 UM/080*
impreso en España
*El número de hojas por área de relleno puede variar ligeramente.

Archivo		Instrucciones de Grabación	Signo								Descripción	Página de
Programador	Fecha		Tecla									

A

Número de Secuencia	Tipo de Formulario	And/Or/Coment. (A/O/*)	Condicionamiento: No (N)	Nombre Condición: Indicador	No (N)	Indicador	No (N)	Indicador	Tipo Nombre o Espac (b/R/H/J/K/S/O)	Reservado	Nombre	Referencia (R)	Longitud	Tipo Datos/Desplazamiento Teclado	Posiciones Decimales	Utilización (b/O/I/B/H/M/N/P)	Ubicación: Línea	Pos	Funciones
	A	*																	
	A	*		DDS PARA LA CONTRASEÑA DE ARCHIVO FISICO															
	A	*																	
	A																		UNIQUE
	A								R		PASSWORDS								
	A										PASSKEY		10						
	A										PASSWORD		10						
	A								K		PASSKEY								
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		

Figura 64 (Parte 2 de 3). Ejemplo de Utilización de Archivos de Múltiples Dispositivos

```
     A*
     A* DDS PARA EL TERMINO DE ARCHIVO FISICO
     A* QUE CONTIENE LA LISTA DE TERMINALES
     A*
     A          R TERM
     A            TERM          10
     A
     A
     A
     A
     A
     A
     A
     A
     A
     A
     A
     A
     A
     A
```

Figura 64 (Parte 3 de 3). Ejemplo de Utilización de Archivos de Múltiples Dispositivos

```
5763CB1 V3R0M5  001000            Fuente COBOL AS/400              TESTER/SAMPMDF         AS400SYS  03/31/94 13:58:05     Página  2
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
    1  000010 IDENTIFICATION DIVISION.
    2  000020 PROGRAM-ID.     SAMPMDF.
    3  000030   AUTHOR.       PROGRAMMER NAME.
       000040
       000050*************************************************************
       000060* EL SIGUIENTE PROGRAMA MUESTRA ALGUNAS FUNCIONES DISPONIBLES*
       000070* CON EL SOPORTE DEL ARCHIVO DE MÚLTIPLES DISPOSITIVOS.      *
       000080*************************************************************
       000090
    4  000100   INSTALLATION. COBOL DEVELOPMENT CENTRE.
    5  000110   DATE-WRITTEN. 02/02/87.
    6  000120   DATE-COMPILED.  03/31/94 13:58:05   .
    7  000130 ENVIRONMENT DIVISION.
    8  000140 CONFIGURATION SECTION.
    9  000150 SOURCE-COMPUTER. IBM-AS400.
   10  000160 OBJECT-COMPUTER. IBM-AS400.
   11  000170 SPECIAL-NAMES.  ATTRIBUTE-DATA IS ATTR.  1
   12  000180 INPUT-OUTPUT SECTION.
   13  000190 FILE-CONTROL.
   14  000200     SELECT MULTIPLE-FILE
   15  000210     ASSIGN TO WORKSTATION-MULT
   16  000220     ORGANIZATION IS TRANSACTION  2
   17  000230     ACCESS MODE IS SEQUENTIAL
   18  000240     FILE STATUS IS MULTIPLE-FS1, MULTIPLE-FS2  3
   19  000250     CONTROL-AREA IS MULTIPLE-CONTROL-AREA.
       000260                          4
   20  000270     SELECT TERMINAL-FILE
   21  000280     ASSIGN TO DATABASE-TERM
   22  000290     ORGANIZATION IS SEQUENTIAL
   23  000300     ACCESS IS SEQUENTIAL
   24  000310     FILE STATUS IS TERMINAL-FS1.
       000320
   25  000330     SELECT PASSWORD-FILE
   26  000340     ASSIGN TO DATABASE-PASSWORD
   27  000350     ORGANIZATION IS INDEXED
   28  000360     RECORD KEY IS EXTERNALLY-DESCRIBED-KEY
   29  000370     ACCESS MODE IS RANDOM
   30  000380     FILE STATUS IS PASSWORD-FS1.
       000390
   31  000400     SELECT PRINTER-FILE
   32  000410     ASSIGN TO PRINTER-QPRINT.
   33  000420 DATA DIVISION.
   34  000430 FILE SECTION.
   35  000440 FD  MULTIPLE-FILE.
   36  000450 01  MULTIPLE-REC. COPY DDS-SIGNON OF MULT.  5
   37 +000001       05  MULT-RECORD PIC X(20).                                  SIGNON
      +000002* FORMATO ENTRADA:SIGNON   DESDE ARCH MULT    DE BIBLIOTECA TESTER  SIGNON
      +000003*                                                                  SIGNON
   38 +000004       05  SIGNON-I      REDEFINES MULT-RECORD.                    SIGNON
   39 +000005           06 PASSWORD              PIC X(10).  6                  SIGNON
      +000006* FORMATO SALIDA:SIGNON    DESDE ARCH MULT    DE BIBLIOTECA TESTER  SIGNON
      +000007*                                                                  SIGNON
   40 +000008       05  SIGNON-O      REDEFINES MULT-RECORD.                    SIGNON
   41 +000009           06 WRONG                 PIC X(20).                     SIGNON
       000460
   42  000470 FD  TERMINAL-FILE.
   43  000480 01  TERMINAL-REC. COPY DDS-ALL-FORMATS OF TERM.
   44 +000001       05  TERM-RECORD PIC X(10).                                  <-ALL-FMTS
      +000002*   FORMATO E-S:TERM  DESDE ARCHIVO TERM    DE BIBLIOTECA TESTER  <-ALL-FMTS
      +000003*                                                                  <-ALL-FMTS
   45 +000004       05  TERM          REDEFINES TERM-RECORD.                    <-ALL-FMTS
   46 +000005           06 TERM                  PIC X(10).                     <-ALL-FMTS
       000490
   47  000500 FD  PASSWORD-FILE.
   48  000510 01  PASSWORD-REC. COPY DDS-ALL-FORMATS OF PASSWORD.
   49 +000001       05  PASSWORD-RECORD PIC X(20).                              <-ALL-FMTS
      +000002*    FORMATO E-S:PASSWORDS    DESDE ARCHIVO PASSWORD   DE BIBLO TESTER  <-ALL-FMTS
      +000003*                                                                  <-ALL-FMTS
      +000004*  DEFINICIONES CLAVE PARA FORMATO DE REGISTRO PASSWORDS            <-ALL-FMTS
      +000005*  NÚMERO              NOMBRE           RECUPERACIÓN    TIPO  ALTSEQ  <-ALL-FMTS
      +000006*   0001   PASSKEY                        ASCENDING      AN      NO  <-ALL-FMTS
   50 +000007       05  PASSWORDS     REDEFINES PASSWORD-RECORD.                <-ALL-FMTS
   51 +000008           06 PASSKEY               PIC X(10).                     <-ALL-FMTS
   52 +000009           06 PASSWORD              PIC X(10).                     <-ALL-FMTS
       000520
```

Figura 65 (Parte 1 de 4). Listado Fuente COBOL para el Soporte de Archivos de Múltiples Dispositivos

```
5763CB1 V3R0M5  001000            Fuente COBOL AS/400              TESTER/SAMPMDF         AS400SYS  03/31/94 13:58:05     Página  2
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
   53  000530 FD  PRINTER-FILE.
   54  000540 01  PRINTER-REC.
   55  000550    05 PRINTER-RECORD        PIC X(132).
       000560
   56  000570 WORKING-STORAGE SECTION.
       000580
       000590*****************************************************************
       000600*     DECLARAR EL ESTADO DE ARCHIVO PARA CADA ARCHIVO           *
       000610*****************************************************************
       000620
   57  000630 01  MULTIPLE-FS1            PIC X(2)    VALUE SPACES.
   58  000640 01  MULTIPLE-FS2.  7
   59  000650    05  MULTIPLE-MAJOR       PIC X(2)    VALUE SPACES.
   60  000660    05  MULTIPLE-MINOR       PIC X(2)    VALUE SPACES.
   61  000670 01  TERMINAL-FS1            PIC X(2)    VALUE SPACES.
   62  000680 01  PASSWORD-FS1            PIC X(2)    VALUE SPACES.
       000690
       000700*****************************************************************
       000710* DECLARAR ESTRUCTURA PARA ATRIBUTOS DE ARCHIVO DE MANTENIM. *
       000720*****************************************************************
       000730
   63  000740 01  STATION-ATTR.
   64  000750    05 STATION-TYPE          PIC X(1).  8
   65  000760    05 STATION-SIZE          PIC X(1).
   66  000770    05 STATION-LOC           PIC X(1).
   67  000780    05 FILLER                PIC X(1).
   68  000790    05 STATION-ACQUIRE       PIC X(1).
   69  000800    05 STATION-INVITE        PIC X(1).
   70  000810    05 STATION-DATA          PIC X(1).
   71  000820    05 STATION-STATUS        PIC X(1).
   72  000830    05 STATION-DISPLAY       PIC X(1).
   73  000840    05 STATION-KEYBOARD      PIC X(1).
   74  000850    05 STATION-SIGNON        PIC X(1).
   75  000860    05 FILLER                PIC X(5).
       000870
       000880*****************************************************************
       000890*     DECLARAR ÁREA DE CONTROL PARA VARIOS ARCHIVOS             *
       000900*****************************************************************
       000910
   76  000920 01  MULTIPLE-CONTROL-AREA.
   77  000930    05 MULTIPLE-KEY-FEEDBACK PIC X(2)    VALUE SPACES.
   78  000940    05 MULTIPLE-DEVICE-NAME  PIC X(10)   VALUE SPACES.
   79  000950    05 MULTIPLE-FORMAT-NAME  PIC X(10)   VALUE SPACES.
       000960
       000970*****************************************************************
       000980*         DECLARAR VARIABLES DE INFORME DE ERROR                *
       000990*****************************************************************
       001000
   80  001010 01  HEADER-LINE.
   81  001020    05 FILLER                PIC X(60)   VALUE SPACES.
   82  001030    05 FILLER                PIC X(72)
   83  001040                             VALUE "MDF ERROR REPORT".
   84  001050 01  DETAIL-LINE.
   85  001060    05 FILLER                PIC X(15)   VALUE SPACES.
   86  001070    05 DESCRIPTION           PIC X(25)   VALUE SPACES.
   87  001080    05 DETAIL-VALUE          PIC X(92)   VALUE SPACES.
       001090
       001100*****************************************************************
       001110*  DECLARAR CONTADORES, DISTINTIVOS Y VARIABLES DE ALMACENAM.*
       001120*****************************************************************
       001130
   88  001140 01  CURRENT-TERMINAL        PIC X(10)   VALUE SPACES.
   89  001150 01  TERMINAL-ARRAY.
   90  001160    05 LIST-OF-TERMINALS OCCURS 250 TIMES.
   91  001170      07 DEVICE-NAME         PIC X(10).
   92  001180 01  COUNTER                 PIC 9(3)    VALUE IS 1.
   93  001190 01  NO-OF-TERMINALS         PIC 9(3)    VALUE IS 1.
   94  001200 01  TERMINAL-LIST-FLAG      PIC 1.
   95  001210    88 END-OF-TERMINAL-LIST              VALUE IS B"1".
   96  001220    88 NOT-END-OF-TERMINAL-LIST          VALUE IS B"0".
   97  001230 01  NO-DATA-FLAG            PIC 1.
   98  001240    88 NO-DATA-AVAILABLE                 VALUE IS B"1".
   99  001250    88 DATA-AVAILABLE                    VALUE IS B"0".
       001260
```

Figura 65 (Parte 2 de 4). Listado Fuente COBOL para el Soporte de Archivos de Múltiples Dispositivos

```
5763CB1 V3R0M5  001000            Fuente COBOL AS/400              TESTER/SAMPMDF         AS400SYS  03/31/94 13:58:05     Página  2
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
  100  001270 PROCEDURE DIVISION.
       001280
       001290 DECLARATIVES.
       001300
       001310 MULTIPLE-SECTION SECTION.
       001320     USE AFTER STANDARD EXCEPTION PROCEDURE ON MULTIPLE-FILE.
       001330
       001340 MULTIPLE-PARAGRAPH.
  101  001350     WRITE PRINTER-REC FROM HEADER-LINE AFTER ADVANCING PAGE.
  102  001360     MOVE "FILE NAME IS:" TO DESCRIPTION OF DETAIL-LINE.
  103  001370     MOVE "MULTIPLE FILE" TO DETAIL-VALUE OF DETAIL-LINE.
  104  001380     WRITE PRINTER-REC FROM DETAIL-LINE AFTER ADVANCING 5 LINES.
  105  001390     MOVE "FILE STATUS IS:" TO DESCRIPTION OF DETAIL-LINE.
  106  001400     MOVE MULTIPLE-FS1 TO DETAIL-VALUE OF DETAIL-LINE.
  107  001410     WRITE PRINTER-REC FROM DETAIL-LINE AFTER ADVANCING 2 LINES.
  108  001420     MOVE "EXTENDED STATUS IS:" TO DESCRIPTION OF DETAIL-LINE. 9
  109  001430     MOVE MULTIPLE-FS2 TO DETAIL-VALUE OF DETAIL-LINE.
  110  001440     WRITE PRINTER-REC FROM DETAIL-LINE AFTER ADVANCING 2 LINES.
  111  001450     ACCEPT STATION-ATTR FROM ATTR. 9A
  112  001460     MOVE "FILE ATTRIBUTES ARE:" TO DESCRIPTION OF DETAIL-LINE.
  113  001470     MOVE STATION-ATTR TO DETAIL-VALUE OF DETAIL-LINE.
  114  001480     WRITE PRINTER-REC FROM DETAIL-LINE AFTER ADVANCING 2 LINES.
  115  001490     STOP RUN.
       001500
       001510 TERMINAL-SECTION SECTION.
       001520     USE AFTER STANDARD EXCEPTION PROCEDURE ON TERMINAL-FILE.
       001530 TERMINAL-PARAGRAPH.
  116  001540     WRITE PRINTER-REC FROM HEADER-LINE AFTER ADVANCING PAGE.
  117  001550     MOVE "FILE NAME IS:" TO DESCRIPTION OF DETAIL-LINE.
  118  001560     MOVE "TERMINAL FILE" TO DETAIL-VALUE OF DETAIL-LINE.
  119  001570     WRITE PRINTER-REC FROM DETAIL-LINE AFTER ADVANCING 5 LINES.
  120  001580     MOVE "FILE STATUS IS:" TO DESCRIPTION OF DETAIL-LINE.
  121  001590     MOVE TERMINAL-FS1 TO DETAIL-VALUE OF DETAIL-LINE.
  122  001600     WRITE PRINTER-REC FROM DETAIL-LINE AFTER ADVANCING 2 LINES.
  123  001610     STOP RUN.
       001620
       001630 PASSWORD-SECTION SECTION.
       001640     USE AFTER STANDARD EXCEPTION PROCEDURE ON PASSWORD-FILE.
       001650 PASSWORD-PARAGRAPH.
  124  001660     WRITE PRINTER-REC FROM HEADER-LINE AFTER ADVANCING PAGE.
  125  001670     MOVE "FILE NAME IS:" TO DESCRIPTION OF DETAIL-LINE.
  126  001680     MOVE "PASSWORD FILE" TO DETAIL-VALUE OF DETAIL-LINE.
  127  001690     WRITE PRINTER-REC FROM DETAIL-LINE AFTER ADVANCING 5 LINES.
  128  001700     MOVE "FILE STATUS IS:" TO DESCRIPTION OF DETAIL-LINE.
  129  001710     MOVE PASSWORD-FS1 TO DETAIL-VALUE OF DETAIL-LINE.
  130  001720     WRITE PRINTER-REC FROM DETAIL-LINE AFTER ADVANCING 2 LINES.
  131  001730     STOP RUN.
       001740
       001750 END DECLARATIVES.
       001760
       001770*****************************************************************
       001780*       LÓGICA DE PROGRAMA PRINCIPAL EMPIEZA AQUÍ           *
       001790*****************************************************************
       001800
       001810 MAIN-LINE SECTION.
       001820 MAIN-LINE-PARAGRAPH.
  132  001830     OPEN I-O    MULTIPLE-FILE 10
       001840          INPUT  TERMINAL-FILE
       001850          I-O    PASSWORD-FILE
       001860          OUTPUT PRINTER-FILE.
       001870
  133  001880       MOVE 1 TO COUNTER.
  134  001890       SET NOT-END-OF-TERMINAL-LIST TO TRUE.
       001900       PERFORM
  135  001910           FILL-TERMINAL-LIST UNTIL END-OF-TERMINAL-LIST.
       001920       PERFORM
  136  001930           ACQUIRE-AND-INVITE-TERMINALS
       001940               VARYING COUNTER FROM 1 BY 1
       001950               UNTIL COUNTER GREATER THAN NO-OF-TERMINALS.
  137  001960       MOVE 1 TO COUNTER.
  138  001970       SET DATA-AVAILABLE TO TRUE.
       001980       PERFORM
  139  001990           POLL-TERMINALS UNTIL NO-DATA-AVAILABLE.
       002000       PERFORM
  140  002010           DROP-TERMINALS
       002020               VARYING COUNTER FROM 1 BY 1
       002030               UNTIL COUNTER GREATER THAN NO-OF-TERMINALS.
```

Figura 65 (Parte 3 de 4). Listado Fuente COBOL para el Soporte de Archivos de Múltiples Dispositivos

```
5763CB1 V3R0M5  001000           Fuente COBOL AS/400             TESTER/SAMPMDF         AS400SYS  03/31/94 13:58:05     Página  2
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
  141  002040       CLOSE   MULTIPLE-FILE
       002050               TERMINAL-FILE
       002060               PASSWORD-FILE
       002070               PRINTER-FILE.
  142  002080       STOP RUN.
       002090
       002100*****************************************************************
       002110*                PROCEDIMIENTOS                                 *
       002120*****************************************************************
       002130
       002140 PROCEDURE-SECTION SECTION.
       002150 FILL-TERMINAL-LIST.
  143  002160     READ TERMINAL-FILE RECORD INTO LIST-OF-TERMINALS(COUNTER)
       002170         AT END
  144  002180             SET END-OF-TERMINAL-LIST TO TRUE
  145  002190             SUBTRACT 1 FROM COUNTER
  146  002200             MOVE COUNTER TO NO-OF-TERMINALS.
  147  002210     ADD 1 TO COUNTER.
       002220
       002230 ACQUIRE-AND-INVITE-TERMINALS.
  148  002240     ACQUIRE LIST-OF-TERMINALS(COUNTER) FOR MULTIPLE-FILE. 11
  149  002250     WRITE MULTIPLE-REC 12
       002260         FORMAT IS "SIGNON"
       002270         TERMINAL IS LIST-OF-TERMINALS(COUNTER).
       002280
       002290 POLL-TERMINALS.
  150  002300     READ MULTIPLE-FILE RECORD. 13
  151  002310     IF MULTIPLE-FS2 EQUAL "310" THEN
  152  002320         SET NO-DATA-AVAILABLE TO TRUE. 14
  153  002330     IF DATA-AVAILABLE THEN
  154  002340         MOVE MULTIPLE-DEVICE-NAME TO CURRENT-TERMINAL
  155  002350         PERFORM PASSWORD-VALIDATION. 15
       002360
       002370 PASSWORD-VALIDATION.
  156  002380     MOVE CURRENT-TERMINAL TO PASSKEY OF PASSWORD-REC.
  157  002390     READ PASSWORD-FILE RECORD.
  158  002400     IF PASSWORD OF SIGNON-I EQUAL PASSWORD OF PASSWORD-REC THEN
  159  002410         CALL "UPDT" USING CURRENT-TERMINAL
  160  002420         MOVE SPACES TO WRONG OF SIGNON-O
       002430     ELSE
  161  002440         MOVE "INVALID PASSWORD" TO WRONG OF SIGNON-O.
  162  002450     WRITE MULTIPLE-REC
       002460         FORMAT IS "SIGNON"
       002470         TERMINAL IS CURRENT-TERMINAL.
       002480
       002490 DROP-TERMINALS.
  163  002500     DROP LIST-OF-TERMINALS(COUNTER) FROM MULTIPLE-FILE. 16
                        * * * * *  F I N  D E  F U E N T E  * * * * *
```

Figura 65 (Parte 4 de 4). Listado Fuente COBOL para el Soporte de Archivos de Múltiples Dispositivos

Atributos del Archivo de Dispositivo

1 ATTR es el nombre mnemotécnico asociado con el nombre de función ATTRIBUTE-DATA. ATTR se utiliza en la instrucción ACCEPT para obtener los datos de atributos para el archivo TRANSACTION, llamado MULTIPLE-FILE. Vea el ítem **9A** .

2 El archivo MULT debe haberse creado utilizando el mandato CRTDSPF, donde el parámetro DEV tiene el valor *NONE y el parámetro MAXDEV tiene un valor mayor que 1. El parámetro WAITRCD especifica el tiempo de espera para las operaciones READ del archivo. El parámetro WAITRCD debe tener un valor mayor que 0.

3 MULTIPLE-FS2 es el estado del archivo ampliado para el archivo TRANSACTION, llamado MULTIPLE-FILE. Esta variable se declaró en la sección WORKING-STORAGE del programa. Vea el ítem **7** .

4 MULTIPLE-CONTROL-AREA es el área de control para el archivo TRANSACTION, llamado MULTIPLE-FILE. Esta variable se utiliza para determinar qué dispositivo de programa se ha utilizado para iniciar la sesión. Vea el ítem **15** .

5 La descripción de datos para MULTIPLE-REC se ha definido utilizando la instrucción COPY DDS.

Nota: Sólo se nombran los campos que se copian. Consulte las DDS de este ejemplo para comentarios acerca de las DDS utilizadas.

6 El formato SIGNON es el formato con la palabra clave INVITE. Este es el formato que se utilizará para invitar a los dispositivos mediante la instrucción WRITE.

7 Esta es la declaración para el estado del archivo ampliado MULTIPLE-FS2. Es un campo de 4 bytes que se subdivide en un código de retorno principal (los 2 primeros bytes) y un código de retorno secundario (los 2 últimos bytes).

8 STATION-ATTR es el lugar en el que la instrucción ACCEPT contiene los datos del atributo para el archivo TRANSACTION, llamado MULTIPLE-FILE. Vea el ítem **9A** .

9 En esta instrucción, está grabándose el estado de archivo ampliado MULTIPLE-FS2.

9A Este es un ejemplo de aceptación de datos de atributo para el archivo TRANSACTION, llamado MULTIPLE-FILE. Dado que no interesa un dispositivo de programa específico, sino más bien el último dispositivo de programa utilizado, las frases FOR no se utilizan con la instrucción ACCEPT.

10 Esta instrucción abre el archivo TRANSACTION llamado MULTIPLE-FILE. Dado que el parámetro ACQPGMDEV del mandato CRTDSPF tiene el valor *NONE, no se adquiere implícitamente ningún dispositivo de programa cuando se abre el archivo.

11 Esta instrucción adquiere el dispositivo de programa contenido en la variable LIST-OF-TERMINALS (COUNTER), para el archivo TRANSACTION, llamado MULTIPLE-FILE.

12 Esta instrucción WRITE invita al dispositivo de programa especificado en la frase TERMINAL. El formato SIGNON tiene la palabra clave DDS INVITE asociada con él. Consulte el ítem **13** .

13 Esta instrucción READ leerá desde cualquier dispositivo de programa invitado. Vea el ítem **12** . Si el tiempo de espera expira antes de que alguien introduzca datos en los dispositivos invitados, el estado ampliado del archivo se colocará en “0310” y el proceso continuará. Vea el ítem **14** .

14 En esta instrucción, el estado ampliado del archivo para MULTIPLE-FILE está comprobándose para ver si ha expirado el tiempo de espera.

15 El nombre del dispositivo de programa almacenado en el área de control se utiliza para determinar qué dispositivo de programa se utilizó para iniciar la sesión. Vea el ítem **4** .

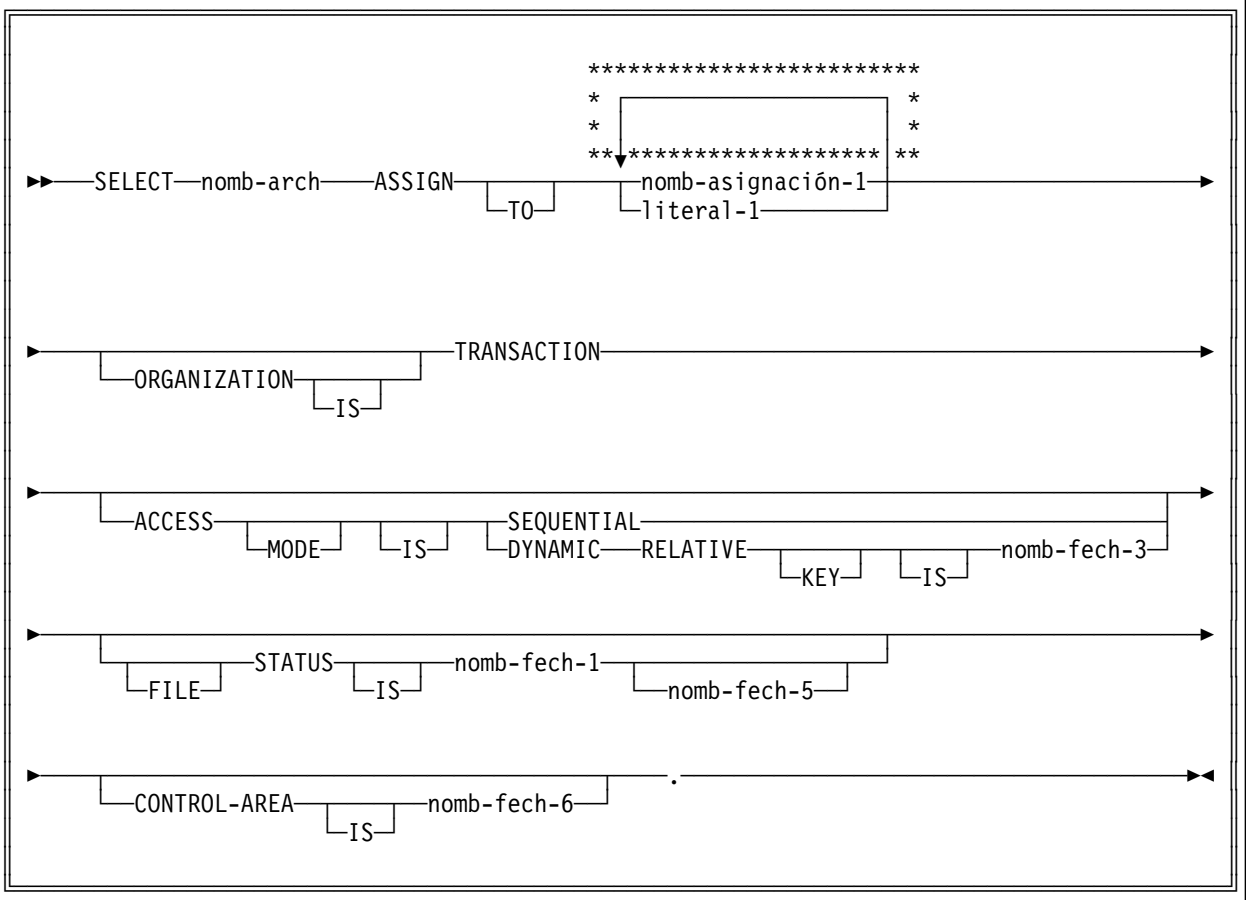
16 Esta instrucción DROP libera el dispositivo de programa contenido en la variable LIST-OF-TERMINALS del archivo TRANSACTION, llamado MULTIPLE-FILE.

División de Entorno

Entrada de Control de Archivo

El archivo TRANSACTION debe designarse mediante una entrada de control de archivo en el párrafo FILE-CONTROL. Esta entrada también especifica otra información relacionada con el archivo.

Formato

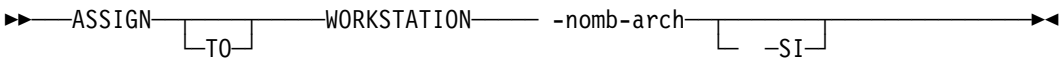


Cláusula ASSIGN

La cláusula ASSIGN asocia el archivo TRANSACTION con un archivo de pantalla o un archivo ICF mediante la utilización de nombre de asignación-1.

El nombre de asignación-1 tiene la estructura siguiente:

Formato



El dispositivo especifica el tipo de dispositivo asociado con el archivo. El valor debe ser WORKSTATION.

El nombre de archivo de AS/400 es un nombre externo de uno a diez caracteres de un archivo de pantalla o un archivo ICF especificado en los mandatos de crear archivo de dispositivo, CRTDSPF o CRTICFF.

El atributo -SI se utiliza para especificar la opción a nivel de archivo para un área de indicadores separada. Consulte el apartado “Utilización de Indicadores con Archivos Transaction” en la página 149 para más detalles.

El segundo y siguientes nombres de asignación se comprueban sintácticamente, pero se tratan como documentación.

Cláusula ORGANIZATION

La cláusula ORGANIZATION especifica la estructura lógica de un archivo. La organización TRANSACTION significa una interacción entre el programa y un usuario de estación de trabajo u otro sistema.

Organización TRANSACTION: El proceso TRANSACTION se define como la llegada al azar de un registro desde una de las múltiples fuentes posibles mediante el proceso apropiado, y finalmente, mediante la salida de los resultados o la información de realimentación de algún tipo al fuente del registro.

En algunos casos, todos los registros son homogéneos; es decir, una transacción lógica se completa con un intercambio de registros. En otras situaciones, una serie de registros se transfiere hacia atrás y hacia adelante en progresión lógica con varios tipos de registro seleccionados por el iniciador o como parte del proceso basándose en los valores de datos de entrada.

Cada transacción puede procesarse mediante un programa distinto, o varias transacciones pueden procesarse mediante el mismo programa, según el entorno del sistema.

La iniciación de una transacción puede provocar que se planifique un programa para procesar la transacción.

Una transacción puede consistir en una serie de peticiones y respuestas alternativas (un diálogo). Cada petición y respuesta puede consistir en varios registros lógicos.

Cláusula ACCESS MODE

Para archivos con organización TRANSACTION, la modalidad de acceso puede ser SEQUENTIAL o DYNAMIC.

Nota: El **proceso dinámico** es un método de lectura o grabación a un archivo en orden no secuencial y de lectura desde un archivo en orden secuencial con la misma instrucción OPEN.

Cuando se especifica o implica ACCESS IS SEQUENTIAL, el nombre del formato contenido en el campo de nombre de formato del área de control especifica a qué registro se ha accedido. Cuando se especifica ACCESS IS SEQUENTIAL para un archivo TRANSACTION, no especifique el ítem de datos RELATIVE KEY.

Cuando se especifique ACCESS IS DYNAMIC, los registros del archivo pueden tener acceso secuencial o al azar, según la forma de la petición de entrada/salida específica. El acceso al azar a un archivo TRANSACTION sólo es válido si se realiza el proceso del subarchivo. Para el proceso del subarchivo, *deberá* especificar ACCESS IS DYNAMIC.

Cláusula RELATIVE KEY

La cláusula RELATIVE KEY especifica el número relativo de registro para un registro específico de un subarchivo. El ítem de datos RELATIVE KEY, nombre de datos 3, debe definirse como entero sin signo y no puede ser escalado. Asimismo, el ítem de datos no debe definirse en una entrada de descripción de registros asociada con el archivo TRANSACTION.

Cláusula FILE STATUS

El nombre de datos 5 identifica el ítem de datos del estado ampliado del archivo, que contiene códigos de retorno principal y secundario. Estos códigos de retorno principal y secundario pueden, en algunos casos, indicar errores de E/S cuando no lo hace el código de estado del archivo. Una vez se realiza la operación de E/S en un archivo no abierto, el estado ampliado del archivo tendrá un valor de cero.

Para más información acerca de la cláusula FILE STATUS, consulte el el apartado “Estado de Archivos y Áreas de Realimentación” en la página 108. Las consideraciones generales acerca de la cláusula FILE STATUS y el nombre de datos 1 se describen en la segunda parte del manual *COBOL/400 Reference*, en la sección “Cláusula FILE STATUS”.

Para información acerca del papel que desempeña el estado del archivo en el manejo de errores, consulte el Capítulo 6, “Manejo de Errores y Excepciones COBOL/400” en la página 71.

El nombre de datos 5 debe estar definido en la División de Datos como un ítem de datos alfanuméricos de 4 bytes; *no debe* estar definido en la Sección de Archivos. Los dos primeros bytes del ítem de datos del estado del archivo ampliado contienen el código de retorno principal, y los dos bytes siguientes contienen el código retorno secundario. Los códigos de retorno se mueven al nombre de datos 5 después de cada operación de entrada o salida (excepto la instrucción ACCEPT o CLOSE) del archivo TRANSACTION. Los valores colocados en el nombre de datos 5 también puede tener acceso mediante la instrucción ACCEPT utilizando el nombre de función I-O-FEEDBACK. Para más información acerca de los códigos de retorno principal y secundario, consulte las publicaciones *Guía para la Gestión de Datos* y *ICF Programmer's Guide*.

Cláusula CONTROL-AREA

La cláusula CONTROL-AREA especifica la información que depende del dispositivo y del sistema y que se utiliza para controlar las operaciones de entrada/salida para archivos TRANSACTION.

El nombre de datos 6 es un ítem de datos CONTROL-AREA que debe definirse en LINKAGE SECTION o WORKING-STORAGE SECTION. Se asume que el número de datos 6 tiene el formato siguiente:

```
01   nombre datos 6.
     02   tecla-función PIC X(2).
               (Campo de realimentación de tecla de función).
     02   nombre-dispositivo PIC X(10).
               (Nombre de dispositivo de programa).
     02   formato-registro PIC X(10).
               (Formato de registro)
```

El nombre de datos 6 debe tener una longitud de 2, 12 ó 22 caracteres. Basándose en la longitud del nombre de datos 6, el compilador asume la disponibilidad de los bytes de realimentación de tecla, el nombre del dispositivo de programa y el formato de registro.

Nota acerca de la Programación: Para un archivo ICF, el nombre real de un dispositivo debe ser diferente del nombre de dispositivo de programa (nombre de datos 11).

La información se mueve al nombre de datos 6 para cada operación READ desde un archivo que se asignó al tipo de dispositivo WORKSTATION. La información sólo es válida si la operación READ se completa satisfactoriamente (siempre que no haya expirado el tiempo de espera). La información se encuentra en el formato fijo tal como se muestra en el ejemplo siguiente:

```
 FILE-CONTROL.
 SELECT SCREEN-FILE
     ASSIGN TO WORKSTATION-MYFMTS
     ORGANIZATION IS TRANSACTION
     CONTROL-AREA IS
      TRANSACTION-CONTROL-AREA.
⋮
 WORKING-STORAGE SECTION.
 01  TRANSACTION-CONTROL-AREA.
*    ÍTEM DE REALIMENTACIÓN
     02  FUNCTION-KEY  PIC XX.
     02  TERMINAL-ID   PIC X(10).
     02  FORMAT-NAME   PIC X(10).
```

Cada campo en el ítem de datos TRANSACTION-CONTROL-AREA del ejemplo se describe de la forma siguiente:

- FUNCTION-KEY: Un número de dos dígitos insertado en el campo por la interfaz de la estación de trabajo que identifica la tecla de función pulsada por el operador para iniciar la transacción. Los códigos son los siguientes:

00	Tecla Intro
01-24	Teclas de función 1 a 24
90	Tecla de Giro Arriba
91	Tecla de Giro Abajo
92	Tecla Impr
93	Tecla Ayuda
94	Tecla Borra
95	Tecla Inicio
99	No definido

  Cualquier tecla de función para la que se precisa información de realimentación debe definirse para el archivo de pantalla utilizando las DDS.

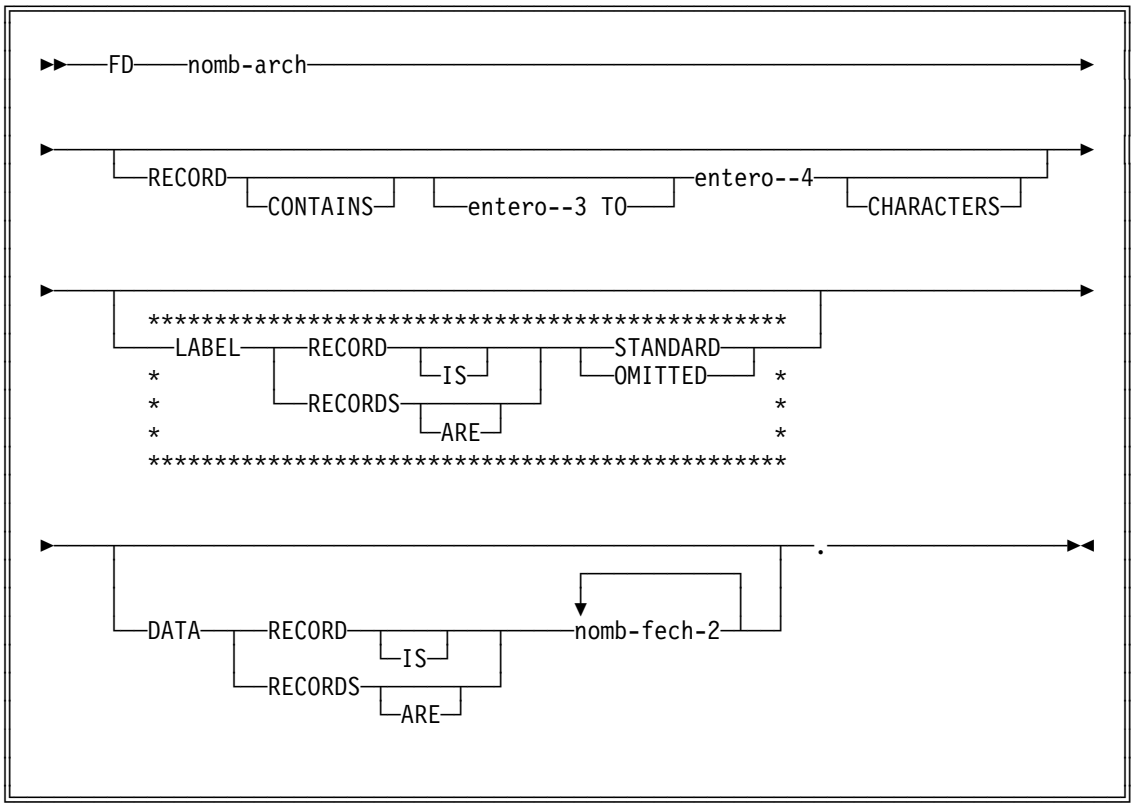
- TERMINAL-ID: El *nombre de dispositivo del programa*
- FORMAT-NAME: El nombre de formato de registro de las DDS que se ha referenciado, mediante la última instrucción de E/S ejecutada.

División de Datos

Entrada de Descripción de Archivos

Una entrada de descripción de archivos consta de un indicador de nivel (FD), un nombre de archivo y una serie de cláusulas independientes. Para un archivo TRANSACTION, las cláusulas independientes permitidas son la cláusula RECORD CONTAINS, la cláusula LABEL RECORDS y la cláusula DATA RECORDS.

Formato



La cláusula LABEL RECORDS especifica si se presentan las etiquetas o no. Esta cláusula es necesaria en cada entrada de descripción de archivos. Se comprueba sintácticamente, pero se trata como documentación.

Ítems de Datos Booleanos

La utilización de datos booleanos y de indicadores se describe en el apartado “Entrada de Descripción de Datos–Datos Booleanos” en la página 151.

División de Procedimiento

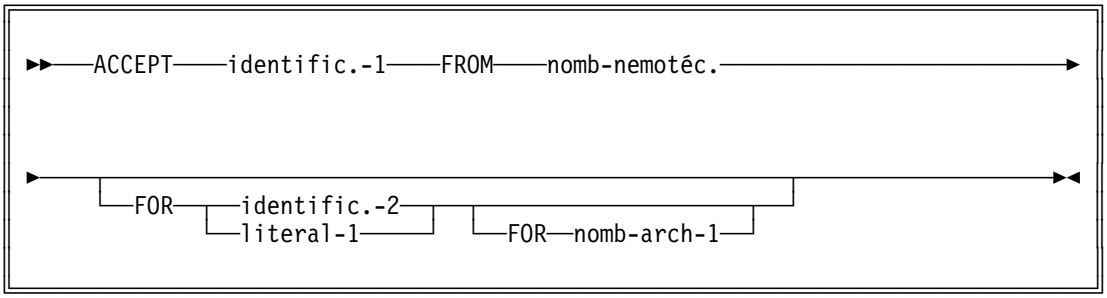
Conceptos de División de Procedimiento

El lenguaje COBOL/400 proporciona una serie de ampliaciones a las instrucciones PROCEDURE DIVISION para dar soporte al proceso TRANSACTION. Las secciones siguientes describen las instrucciones implicadas y su uso.

Instrucción ACCEPT

La instrucción ACCEPT recupera la información (datos de atributo) sobre un dispositivo de programa particular asociado con un archivo TRANSACTION.

Instrucción ACCEPT – Formato 6 – Datos de Atributo



Este formato de la instrucción ACCEPT sólo puede utilizarse para archivos con una organización de TRANSACTION. El nombre mnemotécnico debe asociarse al nombre de función ATTRIBUTE-DATA en el párrafo SPECIAL-NAMES.

Si no se especifica el nombre de archivo, el archivo por omisión para la instrucción ACCEPT es el primer archivo TRANSACTION especificado en una cláusula SELECT del párrafo FILE-CONTROL.

Si se especifica el literal-1 o el contenido del identificador-2, indica el nombre del dispositivo de programa para el que están disponibles los datos de atributo. Este dispositivo debe definirse mediante un mandato CL: CRTDSPF, ADDICFDEVE o OVRICFDEVE. No es obligatorio que se adquiera realmente el dispositivo. Si se especifica el literal-1, debe ser no numérico con una longitud de 10 caracteres o menos. Si se especifica el contenido del identificador-2, debe ser un ítem de datos alfanumérico con una longitud de 10 caracteres o menos. Si se especifica un nombre incorrecto de dispositivo de programa, o si el archivo no está abierto cuando se procesa la instrucción ACCEPT, se emite el mensaje LBE7205

```
Instrucción ACCEPT ATTRIBUTE-DATA anómala (C D F).
```

y termina el proceso.

Si se omiten ambas frases FOR (indicando que se está utilizando el archivo por omisión TRANSACTION), la instrucción ACCEPT utiliza el dispositivo de programa desde el que se realizó más recientemente una operación READ, WRITE, REWRITE o ACCEPT (Datos de Atributo). Si la única operación anterior del archivo fue una OPEN, la instrucción ACCEPT utiliza el dispositivo de programa adquirido implícitamente por el archivo cuando éste se abrió. Cuando se omiten ambas frases FOR, ha de haberse adquirido un dispositivo de programa con el fin de utilizar este formato particular de la instrucción ACCEPT.

Los atributos del dispositivo de programa se mueven al identificador-1 desde el formato de datos de atributos apropiado, de acuerdo con las normas para una instrucción MOVE de grupo sin la frase CORRESPONDING.

El usuario podrá hacer uso de varios archivos de pantalla junto con archivos ordinarios en un programa que incluya una instrucción ACCEPT o DISPLAY ampliadas. Vea la publicación *COBOL/400 Reference* para más información.

Formatos de Datos Atributo

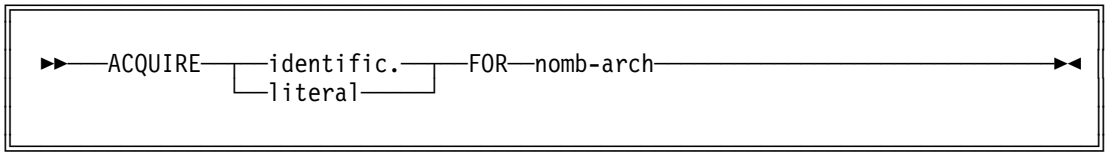
Los datos atributo recuperados mediante la instrucción ACCEPT tienen dos formatos distintos, según sean los datos para una estación de trabajo o para un dispositivo de comunicaciones.

El nombre mnemotécnico ATTRIBUTE-DATA *sólo* puede utilizarse para obtener información acerca de un dispositivo de programa para un archivo TRANSACTION. Los datos atributo *no* proporcionan información acerca del estado de una operación completa de E/S o un intento en la misma operación. Para obtener información acerca de las operaciones de E/S, utilice la instrucción ACCEPT con Formato 3 con los nombres mnemotécnicos I-O-FEEDBACK u OPEN-FEEDBACK. Para más información acerca de estos nombres mnemotécnicos, consulte la sección “Párrafo SPECIAL NAMES” de la publicación *COBOL/400 Reference*.

Instrucción ACQUIRE

La instrucción ACQUIRE adquiere un dispositivo de programa para un archivo TRANSACTION.

Instrucción ACQUIRE – Archivo TRANSACTION



El literal o los contenidos del identificador indican el nombre del dispositivo de programa que debe adquirir el archivo especificado. Si se especifica el literal, debe ser no numérico con una longitud de 10 caracteres o menos. Si se especifica el identificador, debe hacer referencia a un ítem de datos alfanuméricos con una longitud de 10 caracteres o menor.

El nombre de archivo debe ser el nombre de un archivo con una organización de TRANSACTION, y el archivo debe estar abierto cuando se ejecuta la instrucción ACQUIRE. Se emite un mensaje de error de compilación si la organización no es TRANSACTION.

Para una descripción de las condiciones que deben cumplirse antes de que se pueda adquirir un dispositivo de comunicaciones, consulte la publicación *ICF Programmer's Guide*. Para más información acerca de los requisitos para pantallas, consulte la publicación *Guía para la Gestión de Datos*.

La realización satisfactoria de la operación ACQUIRE hace disponible para operaciones de entrada y salida el dispositivo de programa.

Si la operación ACQUIRE no es satisfactoria, el valor de estado del archivo se establece en 9H y se llama al procedimiento USE AFTER EXCEPTION/ERROR (si se especifica). Para más información, consulte el Capítulo 6, “Manejo de Errores y Excepciones COBOL/400”.

Al abrir un archivo, sólo se puede adquirir implícitamente un dispositivo de programa. Si un archivo es ICF, el parámetro ACQPGMDEV del mandato CRTICFF determina el único dispositivo de programa adquirido implícitamente. Si el archivo es un archivo de pantalla, la primera entrada en el parámetro DEV del mandato CRTDSPF determina el único dispositivo de programa adquirido implícitamente. Los dispositivos de programa adicionales *deben* adquirirse explícitamente.

Un dispositivo de programa se adquiere explícitamente utilizando la instrucción ACQUIRE. Para un archivo ICF, dicho dispositivo debe haber sido definido con el mandato CL ADDICFDEVE o OVRICFDEVE antes de abrir el archivo. Para los archivos de pantalla no existe tal requisito. Es decir, el dispositivo nombrado en la instrucción ACQUIRE no tiene que estar especificado ni en el parámetro DEV del mandato CRTDSPF, ni en el CHGDSPF ni en el OVRDSPF. En un archivo de pantalla, el nombre de dispositivo de programa debe coincidir con el de dispositivo de pantalla.

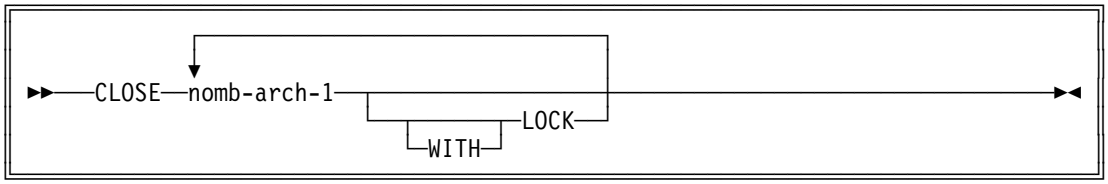
La instrucción ACQUIRE también puede utilizarse como ayuda para la recuperación de errores I/O. Para más información, consulte la sección “Instrucción ACQUIRE” de la publicación *COBOL/400 Reference*.

Para más información acerca de dichos mandatos, consulte la publicación *CL Reference*.

Instrucción CLOSE

La instrucción CLOSE termina el proceso de volúmenes y archivos, con bloqueo opcional donde sea procedente.

Instrucción CLOSE – Formato 3 – Archivo TRANSACTION

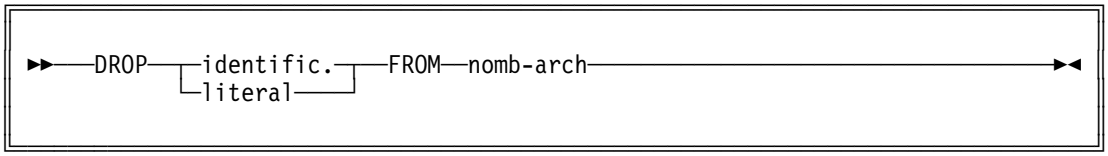


Para un análisis detallado de la instrucción CLOSE, consulte la sección “Instrucción CLOSE” de la publicación *COBOL/400 Reference*.

Instrucción DROP

La instrucción DROP libera un dispositivo de programa adquirido por un archivo TRANSACTION.

Instrucción DROP



El literal o el contenido del identificador indican el nombre del dispositivo de programa del dispositivo a liberar. Si se especifica el literal, debe ser no numérico y con una longitud de 10 caracteres o menor. Si se especifica el identificador, debe referirse a un ítem de datos alfanumérico con una longitud de 10 caracteres o menos.

El nombre del archivo ha de referirse a un archivo con una organización de TRANSACTION, y el archivo debe estar abierto para ser utilizado en la instrucción DROP. Si no se emite ninguna instrucción DROP, los dispositivos de programa conectados a un archivo TRANSACTION se liberan implícitamente cuando el archivo se cierra finalmente.

Los dispositivos de programa especificados en una instrucción DROP han de haber sido adquiridos por el archivo TRANSACTION, mediante una instrucción ACQUIRE explícita o mediante una instrucción ACQUIRE implícita en tiempo OPEN.

Después de la ejecución satisfactoria de la instrucción DROP, el dispositivo de programa no está ya disponible para las operaciones de entrada y salida a través del archivo TRANSACTION. El dispositivo puede volverse a adquirir si es necesario. El contenido del área de registros asociada con un dispositivo de programa liberado no está ya disponible, incluso si se vuelve a adquirir el dispositivo.

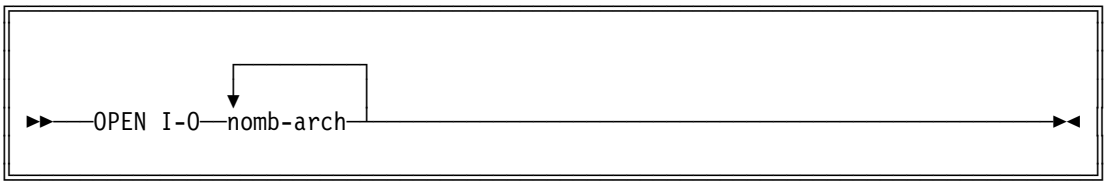
Si la operación DROP no es satisfactoria, se procesa el procedimiento USE AFTER EXCEPTION/ERROR (si se especifica). Para más información, consulte el Capítulo 6, “Manejo de Errores y Excepciones COBOL/400”.

La instrucción DROP también puede utilizarse como ayuda para la recuperación de errores de E/S. Para más información, consulte la sección “Instrucción DROP” de la publicación *COBOL/400 Reference*.

Instrucción OPEN

La instrucción OPEN inicia el proceso de archivos.

Instrucción OPEN – Formato 3 – Archivos TRANSACTION



Un archivo TRANSACTION debe abrirse en modalidad de E/S. Para un tratamiento adicional de la instrucción OPEN, consulte la *COBOL/400 Reference*.

La instrucción OPEN puede provocar que se adquiera implícitamente un dispositivo de programa para un archivo TRANSACTION. Para un análisis adicional acerca de la adquisición de dispositivos de programa, consulte el apartado “Instrucción ACQUIRE” en la página 187.

Recursos Comunes de Proceso

El comentario siguiente sobre las frases FORMAT, INDICATORS, SUBFILE y TERMINAL está relacionado con las instrucciones READ, REWRITE y WRITE.

Frase FORMAT

El literal o identificador especificado debe ser una serie de caracteres con una longitud de 10 caracteres o menos.

Varios registros de datos, cada uno con un formato distinto, pueden estar activos de forma simultánea para un archivo TRANSACTION. Si se especifica la frase FORMAT, debe especificar un nombre de formato válido que se define para el sistema, y debe realizarse la operación de E/S en un registro de datos del mismo formato. Si el formato es un nombre no válido o si no existe, el ítem de datos FILE STATUS, si se especifica, se establece con un valor 9K y el contenido del área de registro es indefinido.

Registro Especial DB-FORMAT-NAME: Después de la ejecución de una instrucción de entrada/salida para un archivo TRANSACTION, el registro especial DB-FORMAT-NAME se modifica de acuerdo con las normas siguientes:

- Si la operación de entrada/salida es satisfactoria, el nombre del formato de registro se mueve implícitamente al registro especial después de la finalización de la operación de entrada/salida.
- Si la operación de entrada/salida no es satisfactoria, DB-FORMAT-NAME contiene el nombre del formato de registro utilizado en la última operación de entrada/salida satisfactoria.

Si no se especifica la frase FORMAT, puede utilizarse DB-FORMAT-NAME si el archivo contiene un nombre de formato de registro por omisión. El valor por omisión siempre se mueve al registro especial DB-FORMAT-NAME.

DB-FORMAT-NAME se define implícitamente como PICTURE X(10).

Frase INDICATORS

El identificador especificado en la frase INDICATORS debe ser un ítem de datos booleanos elemental especificado sin la cláusula OCCURS o un ítem de grupo que tiene ítems de datos booleanos subordinados a él.

Cuando se graba o se regraba un registro de datos, los indicadores pueden grabarse o regrabarse con él. Los indicadores pueden controlar el modo de visualización del registro y las diversas funciones de gestión de datos.

Cuando se lee un registro de datos, los indicadores pueden leerse con él. Los indicadores pueden utilizarse para pasar información acerca de los registros de datos y y de su entrada en el programa.

Al definir un formato utilizando DDS, el usuario determina qué funciones deben controlarse mediante los indicadores y qué indicadores controlan una determinada función.

Para una información detallada de la frase INDICATORS, consulte el apartado “Utilización de Indicadores con Archivos Transaction” en la página 149.

Frase SUBFILE

Si se especifica la frase SUBFILE, ésta indica que todos los formatos referenciados por la instrucción son subarchivos. Cuando no se especifica SUBFILE en una instrucción TRANSACTION I/O, indica que ninguno de los formatos referenciados por la instrucción son subarchivos. Esta información no se verifica en tiempo de compilación. Si se especifica de manera incorrecta, el archivo se procesa como una serie de operaciones de entrada/salida directamente al dispositivo de pantalla. Cuando el nombre de formato especificado exista como un formato de archivo de pantalla, las operaciones READ/WRITE se realizan de modo satisfactorio.

Cuando no se especifica SUBFILE, el ítem de datos RELATIVE KEY asociado con el archivo, si se especifica, no se referencia o cambia mediante la operación de E/S.

Cuando se especifica SUBFILE, el ítem de datos RELATIVE KEY debe definirse para el archivo. La operación de E/S referencia su valor, y a veces lo cambia. Vea cada una de las instrucciones asociadas con las operaciones SUBFILE para una descripción detallada de cuándo y cómo se cambia el ítem de datos RELATIVE KEY.

La frase SUBFILE sólo puede especificarse para los archivos de pantalla.

Frase TERMINAL

Cuando se especifica la frase TERMINAL, indica que un dispositivo de programa específico se va a utilizar para una operación READ, WRITE o REWRITE en un archivo TRANSACTION.

La frase TERMINAL puede omitirse para las operaciones de E/S en archivos de dispositivo único, porque ese dispositivo es el que se utiliza siempre.

Si se omite la frase TERMINAL para una operación de E/S en un archivo TRANSACTION que adquirió varios dispositivos de programa, se utiliza el dispositivo de programa que intentó por última vez una operación READ, WRITE, REWRITE, ACQUIRE, DROP o ACCEPT (Datos de Atributo) en el archivo. Si la

única de las operaciones anteriores del archivo fue una OPEN, el dispositivo de programa por omisión utilizado es el dispositivo de programa adquirido implícitamente por el archivo TRANSACTION cuando se abrió el archivo. Se produce un mensaje de error en tiempo de ejecución si no se ha adquirido ningún dispositivo de programa al abrir el archivo.

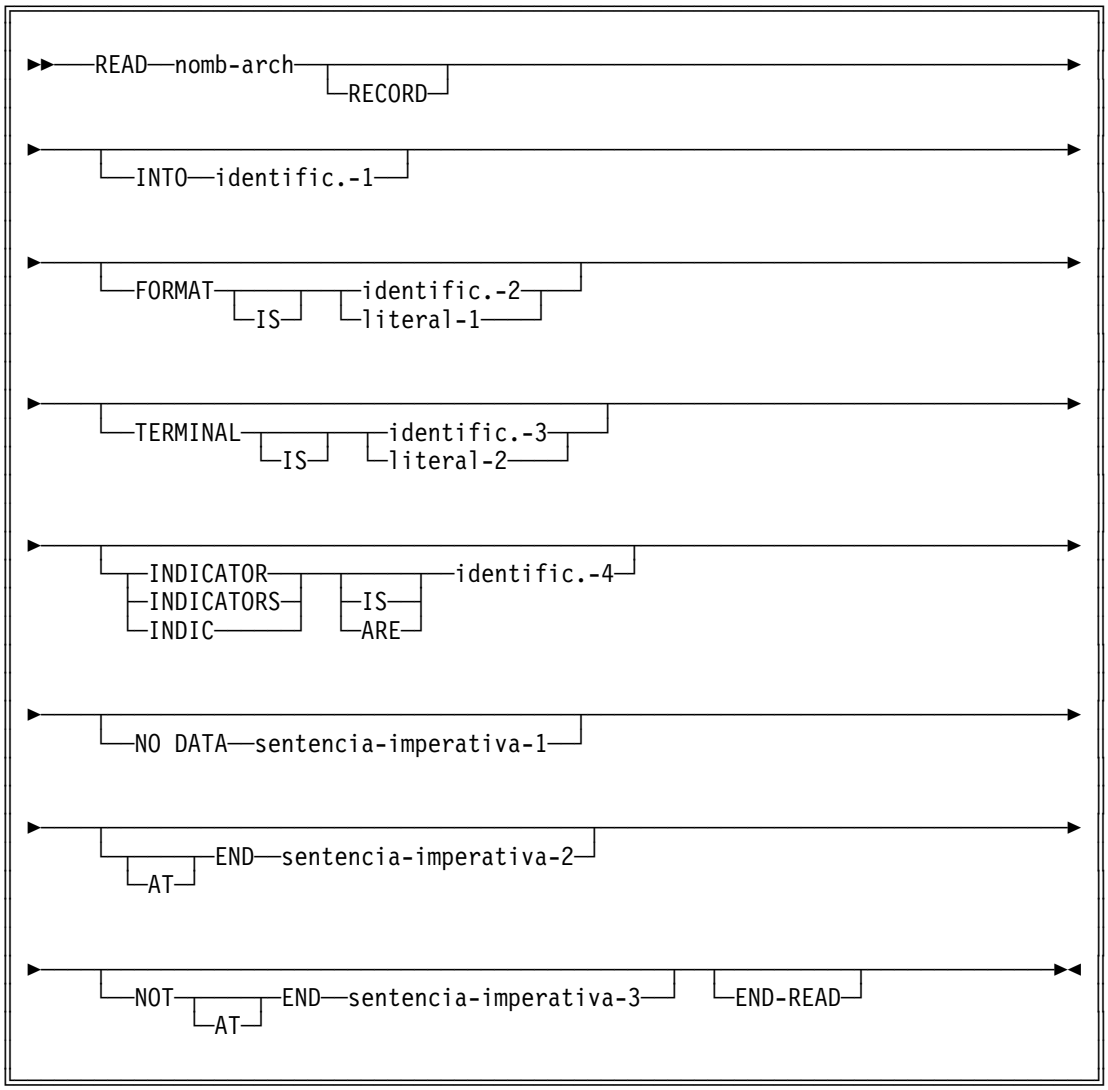
Para una instrucción READ con la frase TERMINAL y la frase NO DATA especificadas, se ejecuta la instrucción imperativa en la frase NO DATA sólo si los datos no están inmediatamente disponibles desde el dispositivo de programa especificado mediante la frase TERMINAL.

Si se especifica la frase TERMINAL y el ítem de datos o el literal está en blanco, la frase se trata en tiempo de ejecución como si no se hubiera especificado.

Instrucción READ

La instrucción READ hace que un registro esté disponible desde un dispositivo, utilizando un formato designado. Si el formato es un subarchivo, la instrucción READ hace que un registro específico esté disponible desde dicho subarchivo.

Instrucción READ – Formato 4 – Archivo TRANSACTION (No subarchivo)



El Formato 4 sólo se utiliza para leer un formato que no sea un subarchivo. El ítem de datos RELATIVE KEY, si se especifica en la entrada FILE-CONTROL, no se utiliza. La instrucción READ con Formato 4 no es válida para un registro de subarchivo. Sin embargo, debe utilizarse una instrucción READ de Formato 4 para el formato de registro de control de subarchivo para colocar dichos registros de subarchivo que se actualizaron en una pantalla en el subarchivo.

Si los datos solicitados están disponibles, se devuelven al área de registros. Los nombres del formato de registro y del dispositivo de programa se devuelven al área I-O-FEEDBACK en el CONTROL-AREA.

La instrucción READ sólo es válida cuando hay dispositivos adquiridos para el archivo. Si se procesa una instrucción READ y no hay dispositivos adquiridos, el estado del archivo se establece en 92 (error lógico).

El modo de funcionamiento de la instrucción READ con Formato 4 depende de:

- Si la instrucción READ es para un archivo de un solo dispositivo o para archivos de múltiples dispositivos
- Si se ha solicitado un dispositivo de programa específico mediante la frase TERMINAL
- Si se ha solicitado un formato de registro específico mediante la frase FORMAT
- Si se ha especificado la frase NO DATA.

En las secciones siguientes, las referencias a *datos disponibles* o *devueltos* incluyen la situación en la que sólo se establecen los indicadores de respuesta. Esto se aplica también incluso cuando se utiliza un área de indicadores separada y dichos indicadores no se devuelven al área de registros del archivo.

La tabla siguiente muestra las combinaciones posibles de frases y la función realizada para un archivo de dispositivo único o para un archivo de múltiples dispositivos. Por ejemplo, si TERMINAL es N, FORMAT es N y NO DATA es N, el dispositivo único es D y el dispositivo múltiple es A.

Función	**Frase**	**Y=Sí N=No**
Comprobado en Compilación	TERMINAL[2] FORMAT[2] NO DATA	N N N N Y Y Y Y N N Y Y N N Y Y N Y N Y N Y N Y
Determinado en Tiempo de Ejecución	Un Solo Dispositivo Dispositivos Múltiples	D C D B D C D B A A D B D C D B

A continuación se explican los códigos A a D:

Código A–Lectura desde Dispositivo de Programa Invitado (solamente Archivos de Dispositivos Múltiples)

Este tipo de READ recibe datos del primer dispositivo de programa invitado que tenga datos disponibles. Los dispositivos de programa invitados son estaciones de trabajo u otros dispositivos de comunicaciones que han sido invitados para enviar entradas. La invitación se realiza escribiendo el dispositivo de programa con un formato que especifique la palabra clave INVITE de DDS. Una vez que se ha leído un dispositivo de programa invitado, ya no queda invitado por más tiempo. Este dispositivo de programa no se utilizará para la entradas mediante otra instrucción READ a menos que se vuelva a invitar, o a menos que se le dirija una instrucción READ especificando la frase TERMINAL o la frase FORMAT.

El formato de registro devuelto desde el dispositivo de programa se determina por medio del sistema. Consulte el capítulo sobre soporte de dispositivo de pantalla en la publicación *Guía para la Gestión de Datos* para obtener información acerca de cómo se determina el formato de registro para estaciones de trabajo. Consulte la

[2] Si se especifica la frase y el ítem de datos o literal está en blanco, la frase se trata en tiempo de ejecución como si no se hubiera especificado.

publicación *ICF Programmer's Guide* para obtener información acerca del parámetro FMTSLT de los mandatos ADDICFDEVE y OVRICFDEVE.

Esta instrucción READ puede completarse sin devolver ningún dato en los casos siguientes:

- Si no hay dispositivos invitados.
- Si se produce una cancelación controlada del trabajo. Esto da como resultado un valor de estado de archivo de 9A y un valor de código de retorno principal/secundario de 0309.
- Si se omite la frase NO DATA y expira el tiempo de espera especificado. Esto da como resultado un valor de estado de archivo de 00 y un valor de código de retorno principal/secundario de 0310.
- Si el tiempo de espera especificado es el valor entrado en el parámetro WAITRCD para el archivo.
- Si se especifica la frase NO DATA y no hay datos inmediatamente disponibles cuando se procesa la READ.

Si hay datos disponibles, se devuelven al área de registros. El formato de registro se devuelve en el área I-O-FEEDBACK y en el CONTROL-AREA. Para más información acerca de "Lectura desde Dispositivos de Programa Invitados," consulte la publicación *ICF Programmer's Guide*.

Código B–Lectura Desde Un Dispositivo de Programa (Combinación no Permitida)

Se emite un mensaje en tiempo de compilación, y se ignora la frase NO DATA. Consulte la entrada de tabla para la misma combinación de frases con la frase NO DATA omitida.

Código C–Lectura Desde Un Dispositivo de Programa (con la frase NO DATA)

Esta función de la instrucción READ nunca hace que un programa se detenga y espere hasta que los datos estén disponibles. Se procesan los datos que estén inmediatamente disponibles o bien la instrucción imperativa NO DATA.

Esta función READ puede utilizarse para comprobar periódicamente si los datos están disponibles desde un dispositivo de programa particular (o bien el dispositivo de programa por omisión o uno especificado mediante la frase TERMINAL). Esta comprobación de los datos se realiza de la manera siguiente:

1. El dispositivo de programa se determina de la manera siguiente:
   a. Si se ha omitido la frase TERMINAL o contiene espacios en blanco, se utiliza el dispositivo de programa por omisión. El dispositivo de programa por omisión es el que se utiliza por la instrucción READ, WRITE, REWRITE, ACQUIRE o DROP que se intentó por última vez. Si no se emitió previamente ninguna de estas operaciones de E/S, el dispositivo de programa por omisión es el primer dispositivo de programa adquirido.
   b. Si se especificó la frase TERMINAL, se utiliza el dispositivo de programa indicado.
2. Se efectúa una comprobación para determinar si los datos están disponibles y si el dispositivo de programa está invitado.

3. Si los datos están disponibles, esos datos se devuelven al área de registro y el dispositivo de programa deja de estar invitado. Si no hay datos inmediatamente disponibles, se ejecuta la instrucción imperativa NO DATA y el dispositivo de programa permanece invitado.
4. Si el dispositivo de programa no está invitado, se dará la condición AT END y el estado del archivo se establece en 10.

Código D–Lectura Desde Un Dispositivo de Programa (sin la frase NO DATA)

Esta instrucción READ siempre espera a que los datos estén disponibles. Incluso si el trabajo recibe una cancelación controlada o se especifica un tiempo WAITRCD para el archivo, el programa nunca recuperará el control de la instrucción READ. Esta operación READ se realiza de la manera siguiente:

1. El dispositivo de programa se determina así:
   a. Si se omite la frase TERMINAL o contiene un valor en blanco, se utiliza el dispositivo de programa por omisión. El dispositivo de programa por omisión es el dispositivo de programa que utiliza la instrucción READ, WRITE, REWRITE, ACQUIRE, DROP o ACCEPT (Datos de Atributo) que se intentó por última vez. Si no se realizó ninguna de estas operaciones, se utiliza el dispositivo de programa adquirido implícitamente al abrir el archivo. Si no hay dispositivos adquiridos, se dará la condición AT END.
   b. Si se especifica la frase TERMINAL, se utiliza el dispositivo de programa indicado.
2. El formato de registro se determina de la manera siguiente:
   a. Si se omite la frase FORMAT o contiene espacios en blanco, el formato de registro devuelto lo determina el sistema. Para obtener información acerca de cómo se calcula el formato de registro para los dispositivos de estación de trabajo, consulte la publicación *Guía para la Gestión de Datos*. Para obtener información acerca de cómo se determina el formato de registro para las comunicaciones, consulte la sección del parámetro FMTSLT de los mandatos ADDICFDEVE y OVRICFDEVE en la publicación *ICF Programmer's Guide*.
   b. Si se especifica la frase FORMAT, se devuelve el formato de registro indicado. Si los datos disponibles no coinciden con el formato de registro solicitado, se establece un estado de archivo de 9G.
3. El proceso del programa se detiene hasta que los datos están disponibles. Los datos se devuelven al área de registro después de que se ejecute la instrucción READ. Si el dispositivo de programa se invitó anteriormente, no permanecerá invitado por más tiempo después de esta instrucción READ.

Frase INTO

La frase INTO puede especificarse si:

- Sólo hay una descripción de registro subordinada a la entrada de descripción del archivo,

o bien

- Todos los nombres de registros asociados al nombre de archivo y el ítem de datos al que hace referencia el identificador-1 describen un ítem de grupo o un ítem alfanumérico elemental.

Frase FORMAT

El literal-1 o el identificador-2 especifican el nombre del formato de registro a leer. Si se especifica el literal-1, debe ser no numérico con una longitud de 10 caracteres o menos. Si se especifica el identificador-2, debe referirse a un ítem de datos alfanumérico con una longitud de 10 caracteres o menos. Si el identificador-2 contiene espacios en blanco, se ejecuta la instrucción READ como si se omitiera la frase FORMAT.

Frase NO DATA

Cuando se especifica la frase NO DATA, la instrucción READ determina si los datos están inmediatamente disponibles. Si hay datos disponibles, se devuelven al área de registros. Si no hay datos inmediatamente disponibles, se procesa la instrucción imperativa NO DATA. La frase NO DATA evita que la instrucción READ espere a que los datos estén disponibles.

Frase TERMINAL

El literal-2 o el identificador-3 especifican el nombre del dispositivo de programa. Si se especifica el literal-2, debe ser no numérico con una longitud de 10 caracteres o menos. Si se especifica el identificador-3, debe referirse a un ítem de datos alfanuméricos con una longitud de 10 caracteres o menos. El dispositivo de programa debe haberse adquirido antes de que se procese la instrucción READ. Si el identificador-3 contiene espacios en blanco, se procesa la instrucción READ como si se omitiera la frase TERMINAL. Para un archivo de un solo dispositivo, puede omitirse la frase TERMINAL. Se asume que el dispositivo de programa es aquel dispositivo único.

Si se omite la frase TERMINAL para una instrucción READ de un archivo TRANSACTION que ha adquirido múltiples dispositivos de programa, se utiliza el dispositivo de programa por omisión. Consulte el comentario de la frase TERMINAL en la página 191, para ver cómo se determina el dispositivo de programa por omisión.

Frase AT END

La instrucción imperativa-2 se realiza cuando se detecta la condición AT END.

Nota: Una condición AT END se produce en los casos siguientes:

- Durante una instrucción READ para un archivo con acceso secuencial cuando no hay ningún registro lógico siguiente en el archivo, o cuando el número de dígitos significativos en el número relativo de registro es mayor que el tamaño del ítem de datos de clave relativa, o cuando no se presenta ningún archivo de entrada opcional.
- Durante una instrucción RETURN cuando no hay registros lógicos para la clasificación asociada o el archivo de fusión.
- Durante una instrucción SEARCH cuando la operación de búsqueda finaliza sin satisfacer la condición especificada en cualquier frase WHEN asociada.

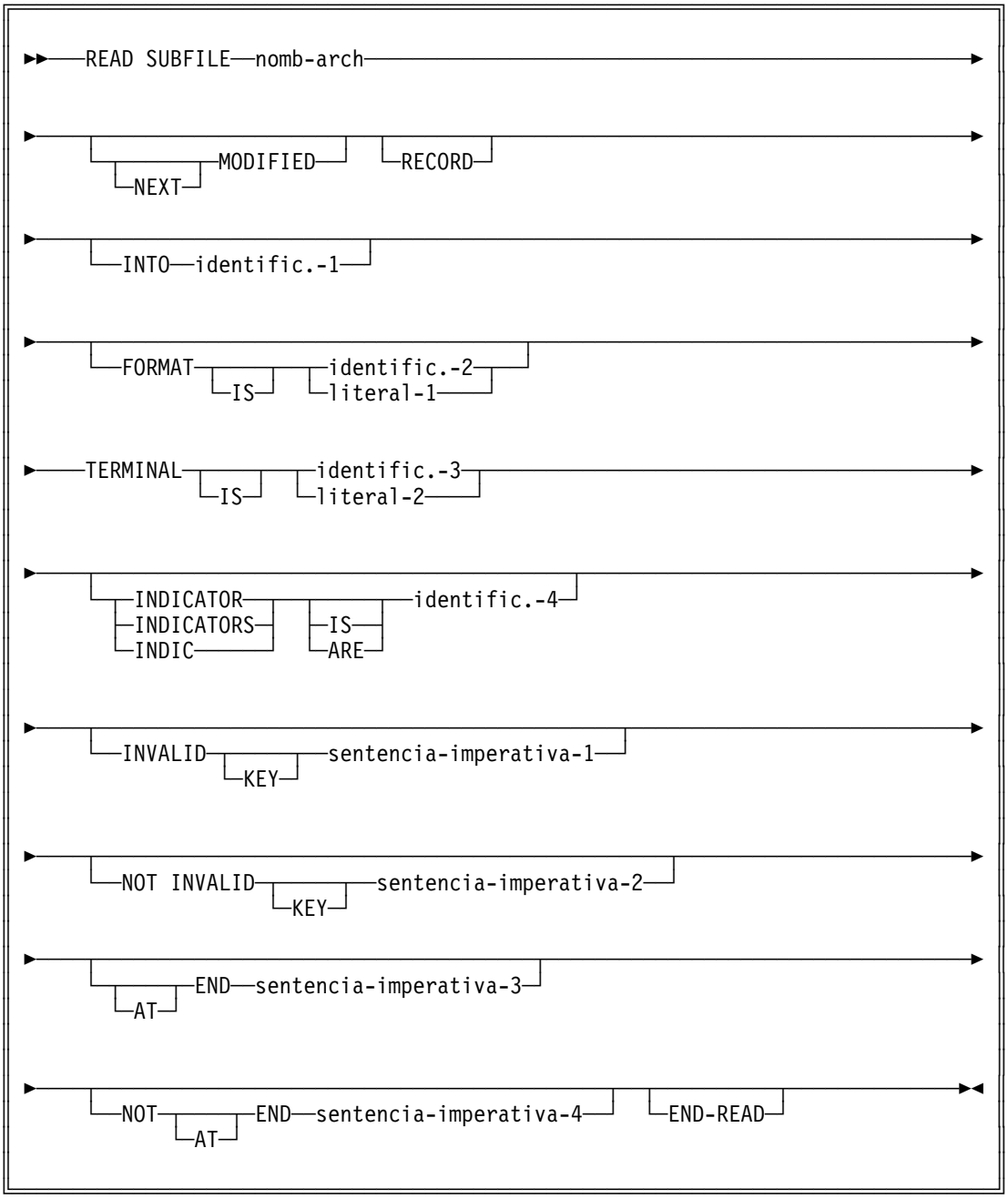
Frase NOT AT END

Esta frase sirve para especificar procedimientos a efectuar cuando la operación READ es satisfactoria.

Frase END-READ

La frase END-READ sirve para delimitar explícitamente el ámbito de la instrucción.

Instrucción READ – Formato 5 – Archivo TRANSACTION (Subarchivo)



El formato 5 sólo se utiliza para leer un formato que sea un registro de subarchivo. La frase AT END sólo puede utilizarse cuando se especifica la frase NEXT MODIFIED. La frase INVALID KEY no debe utilizarse cuando se especifica la frase NEXT MODIFIED.

El formato 5 no puede utilizarse para dispositivos de comunicaciones. Si el formato de subarchivo de la instrucción READ se utiliza para un dispositivo de comunicaciones, la instrucción READ falla y se establece un estado de valor de 90.

Acceso al Azar de Registros de Subarchivo: La frase NEXT MODIFIED no debe utilizarse para registros de acceso al azar en un subarchivo. La frase INVALID KEY sólo puede utilizarse para el acceso al azar de registros de subarchivo.

Acceso Secuencial de Registros de Subarchivo: La frase NEXT MODIFIED debe especificarse para acceder a registros de subarchivo de forma secuencial. La frase AT END sólo puede especificarse con la frase NEXT MODIFIED.

Frase NEXT MODIFIED

Cuando no se especifica NEXT MODIFIED, el registro de datos disponible es el registro en el subarchivo con un número relativo de registro que corresponde al valor del ítem de datos RELATIVE KEY.

Cuando no se especifica la frase NEXT MODIFIED, y si el ítem de datos RELATIVE KEY contiene un valor que no sea el número de registro relativo de un registro en un subarchivo, se dará la condición INVALID KEY y la ejecución de la instrucción READ no será satisfactoria.

Cuando se especifica la frase NEXT MODIFIED, el registro disponible es el registro siguiente modificado que sigue a la posición del puntero actual en el archivo. Para obtener información acerca de la activación del Distintivo de Datos Modificados, consulte la publicación *Guía para la Gestión de Datos*.

La búsqueda del siguiente registro modificado comienza:

- Al principio del subarchivo si:
  - Se ha realizado una operación de E/S para el registro de control de subarchivo.
  - La operación de E/S ha borrado, inicializado o visualizado el subarchivo.
- Para todos los otros casos, comienza en el registro que sigue al registro que se leyó mediante una operación anterior de lectura.

El valor del ítem de datos RELATIVE KEY se actualiza para reflejar el número relativo de registro del registro disponible para el programa.

Si se especifica NEXT MODIFIED y no hay más registros modificados por el usuario en el subarchivo, existe la condición AT END. Entonces se ejecuta la instrucción imperativa 2 o un procedimiento USE AFTER ERROR/EXCEPTION procedente, si lo hay.

Frase FORMAT

Cuando no se especifica un nombre de formato, el formato utilizado es el último formato de registro grabado en el dispositivo de pantalla que contiene campos de entrada, campos de entrada/salida o campos ocultos. Si no existe tal formato para el archivo de pantalla, el formato utilizado es el formato de registro de la última operación WRITE para el dispositivo de pantalla.

Nota: Un **campo de entrada** es un campo especificado en un archivo de pantalla o en un archivo de base de datos que se reserva para la información suministrada por el usuario

Si se especifica la frase FORMAT, el literal-1 o el contenido del identificador-2 deben especificar un formato que esté activo para el dispositivo de programa apropiado. La instrucción READ lee un registro de datos del formato especificado.

Para asegurarse de que los resultados sean correctos, especifique siempre la frase FORMAT para archivos con múltiples formatos. Para más información acerca de la frase FORMAT, consulte la División de Procedimientos, en “Recursos Comunes de Proceso” en la página 190.

Frase TERMINAL

Consulte el Formato 4 de la instrucción READ para obtener las consideraciones generales concernientes a la frase TERMINAL.

Para una instrucción READ de Formato 5, si se omite la frase TERMINAL de un archivo que tenga dispositivos múltiples adquiridos, se lee un registro desde el subarchivo asociado al dispositivo de programa por omisión. Consulte el análisis de la frase TERMINAL en la página 191, para ver cómo se determina el dispositivo de programa por omisión.

Frase INVALID KEY

Si en el momento de ejecución de la instrucción, el ítem de datos RELATIVE KEY contiene un valor que no corresponde a un número relativo de registro para el subarchivo, la condición INVALID KEY existe y la ejecución de la instrucción no es satisfactoria. Para ver lo que ocurre a continuación, consulte los diagramas de la página 80 a la página 82.

| Para una instrucción READ con Formato 5, especifique la frase INVALID KEY si no se especifica la frase NEXT MODIFIED y si no hay el procedimiento USE pertinente especificado para el nombre de archivo.

Frase NOT INVALID KEY

Esta frase le permite especificar los procedimientos a realizar cuando la operación READ es satisfactoria.

Frase AT END

Si se especifica la frase NEXT MODIFIED y no hay ningún registro modificado por el usuario en el subarchivo, existe la condición AT END y la operación READ no es satisfactoria.

Especifique la frase AT END cuando se utiliza la frase NEXT MODIFIED y no se especifica ningún procedimiento USE aplicable para el nombre del archivo. Si se especifica la frase AT END y el procedimiento USE para un archivo, y surge la condición AT END, el control se transfiere a la instrucción imperativa AT END y el procedimiento USE no se ejecuta.

Frase NOT AT END

Esta frase le permite especificar los procedimientos a realizar cuando la operación READ es satisfactoria.

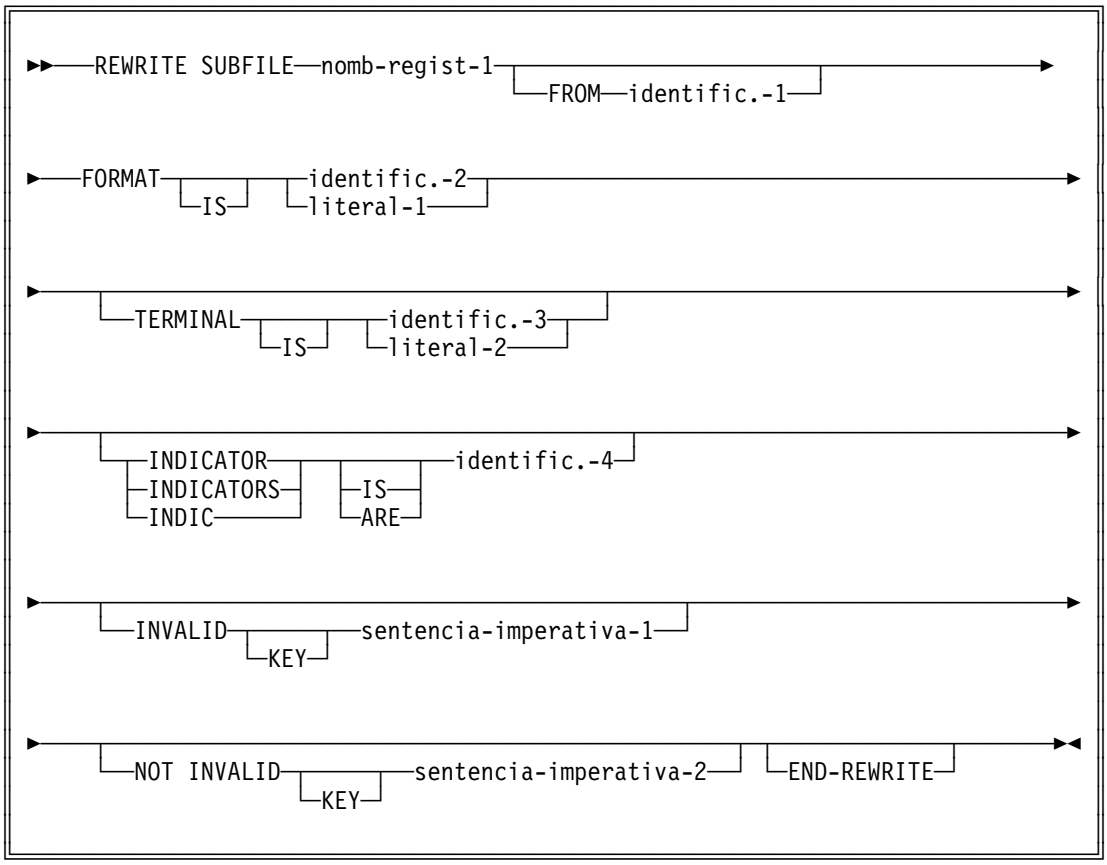
Frase END-READ

La frase END-READ sirve para delimitar explícitamente el ámbito de la instrucción.

Instrucción REWRITE

La instrucción REWRITE se utiliza para sustituir un registro de subarchivo que ya existe en el subarchivo.

Instrucción REWRITE – Formato 2 – Archivo TRANSACTION (Subarchivo)



El número de posiciones de caracteres en el registro referenciado por el nombre de registro debe ser igual al número de posiciones de caracteres en el registro a sustituir. Una operación READ satisfactoria en el registro debe ser anterior a la operación REWRITE. El registro sustituido en el subarchivo es el registro en el subarchivo accedido mediante la anterior operación READ.

Frase FORMAT

El formato de registro especificado en la frase FORMAT debe ser el formato de registro accedido en la operación READ anterior. El literal-1 o el contenido del identificador-2 debe ser el nombre del formato del subarchivo accedido en la anterior instrucción READ. Para más información acerca de la frase FORMAT, consulte el apartado “Recursos Comunes de Proceso” en la página 190.

Frase TERMINAL

La frase TERMINAL indica en qué subarchivo de dispositivo de programa va a regrabarse un registro. Si se especifica la frase TERMINAL, el literal-2 o el identificador-3 deben hacer referencia a una estación de trabajo adquirida mediante el archivo TRANSACTION. Si el literal-2 o el identificador-3 contienen espacios en blanco, la frase TERMINAL no tiene efecto. Debe haberse adquirido el dispositivo de programa especificado mediante la frase TERMINAL, explícita o implícitamente, y debe tener un subarchivo asociado con el dispositivo.

El literal-2 o el identificador-3 deben ser nombres válidos de dispositivo de programa. Si se especifica el literal-2, debe ser no numérico con una longitud de 10 caracteres o menos. Si se especifica el Identificador-3, debe referirse a un ítem de datos alfanumérico, de 10 caracteres o menos.

Si se omite la frase TERMINAL de un archivo TRANSACTION que adquirió varios dispositivos de programa, el subarchivo utilizado es el subarchivo asociado con el último dispositivo de programa desde el que se ha intentado una instrucción READ del archivo TRANSACTION.

La instrucción REWRITE no puede utilizarse para dispositivos de comunicaciones. Si se utiliza la instrucción REWRITE para un dispositivo de comunicaciones, la operación fracasa y el estado del archivo se establece en 90.

Frase INVALID KEY

Si en el momento de ejecución de la instrucción, el ítem de datos RELATIVE KEY contiene un valor que no corresponde a un número relativo de registro para el subarchivo, la condición INVALID KEY existe y la ejecución de la instrucción no es satisfactoria. Para ver lo que ocurre a continuación, consulte los diagramas de la página 80 a la página 82.

Frase NOT INVALID KEY

Esta frase le permite especificar los procedimientos a realizar cuando la operación REWRITE es satisfactoria.

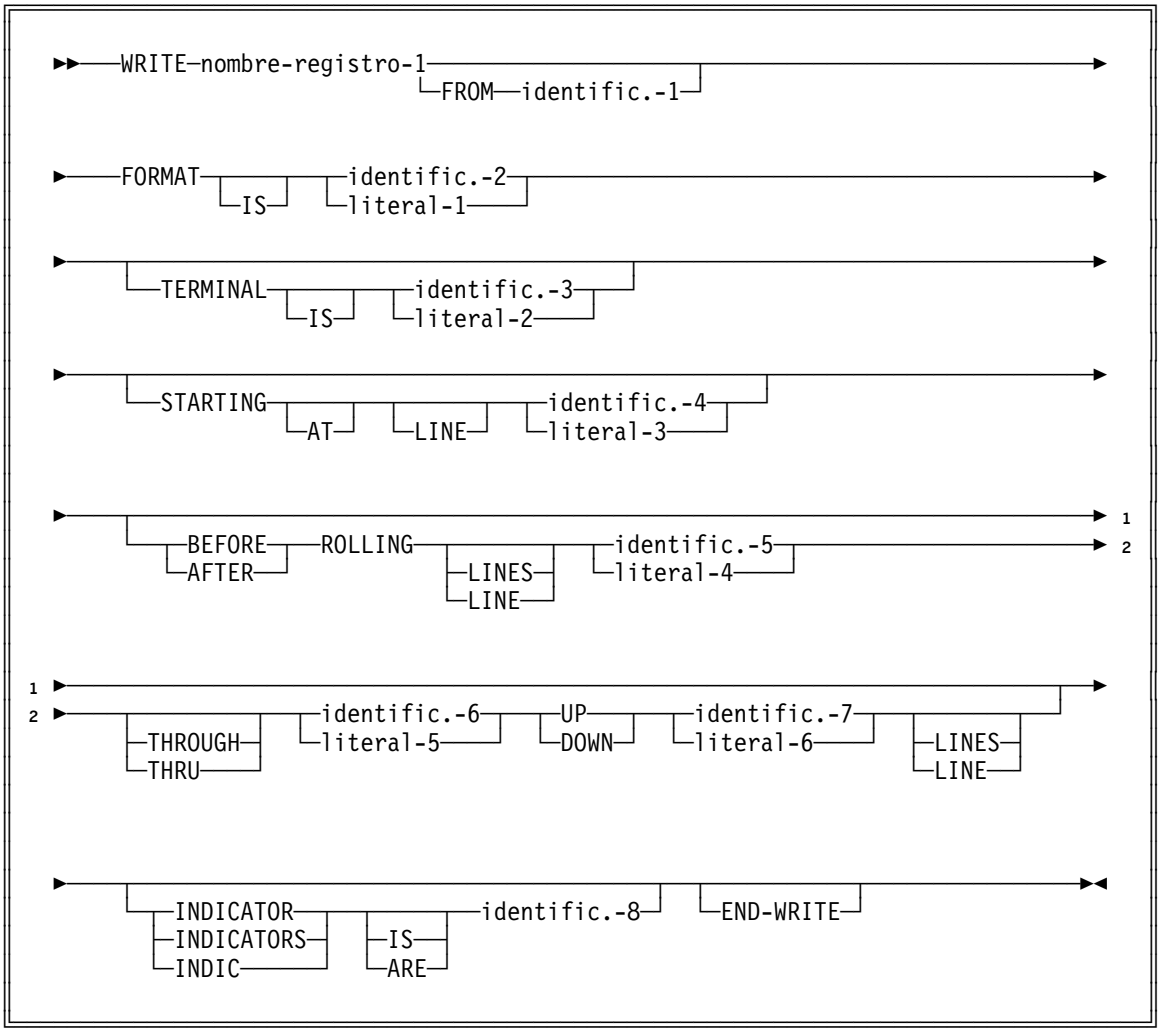
Frase END-REWRITE

La frase END-REWRITE sirve para delimitar explícitamente el ámbito de la instrucción.

Instrucción WRITE

La instrucción WRITE libera un registro lógico para el archivo.

Instrucción WRITE – Formato 4 – Archivo TRANSACTION (No subarchivo)



Frase TERMINAL

La frase TERMINAL especifica los dispositivos de programa a los que va a enviar el registro de salida.

El contenido del literal-2 o el identificador-3 debe ser el nombre de un dispositivo de programa anteriormente adquirido, implícita o explícitamente, por el archivo. Si se especifica el literal-2, debe ser no numérico con una longitud de 10 caracteres o menos. Si se especifica el identificador-3, debe referirse a un ítem de datos alfanuméricos con una longitud de 10 caracteres o menos. Un valor de espacios en blanco se trata como si se omitiera la frase TERMINAL.

Si el archivo TRANSACTION sólo ha adquirido un único dispositivo de programa, la frase TERMINAL puede omitirse. Dicho dispositivo de programa siempre se utiliza para la instrucción WRITE.

Si la frase TERMINAL se omite para una operación WRITE en un archivo TRANSACTION que ha adquirido múltiples dispositivos de programa, se utiliza el dispositivo de programa por omisión. Consulte el análisis de la frase TERMINAL en la página 191, para ver cómo se determina el dispositivo de programa por omisión.

Frase STARTING

La frase STARTING especifica el número inicial de línea para los formatos de registro que utilicen la palabra clave línea de comienzo variable. Esta frase sólo es válida para dispositivos de pantalla.

El número real de línea en el que comienza un campo puede determinarse con la ecuación siguiente:

```
Línea-real = Línea inicial + Línea inicial de las DDS – 1
```

Figura 66. Ecuación del Número de Línea para la Frase STARTING

Donde:

Línea real es el número real de línea
Línea inicial es el número de línea inicial especificado en el programa
Línea inicial de las DDS es el número de línea especificado en las posiciones 39 a 41 del formulario Especificaciones de Descripción de Datos.

La operación WRITE es satisfactoria si:

- El resultado de la ecuación anterior es positivo y menor o igual al número de líneas en la pantalla.
- El valor especificado para la frase STARTING es 0. En este caso, se asume un valor de 1.

La operación WRITE no es satisfactoria, y el programa finaliza, si:

- El resultado de la ecuación anterior es mayor que el número de líneas en la pantalla.
- El valor especificado para la frase STARTING es negativo.

Si el valor especificado para la frase STARTING está dentro del área de pantalla, se ignora cualquier campo fuera del área de pantalla.

El literal-3 de la frase STARTING debe ser un literal numérico. El identificador-4 debe ser un ítem numérico elemental.

Para utilizar la frase STARTING, debe especificarse la palabra clave a nivel de registro DDS SLNO(*VAR) para el formato a grabar. Si el formato de registro no especifica esta palabra clave, se ignora la frase STARTING en tiempo de ejecución.

La palabra clave CLRL de las DDS también afecta a la frase STARTING. CLRL controla la cantidad de pantalla que se borra cuando se procesa la instrucción WRITE.

Vea la publicación *DDS Reference* para obtener más información acerca de las palabras clave SLNO(*VAR) y CLRL.

Frase ROLLING

La frase ROLLING le permite mover líneas visualizadas en la pantalla de la estación de trabajo. Todas o algunas líneas de la pantalla pueden girarse hacia arriba o hacia abajo. Se borran las líneas que dejan desocupadas las líneas giradas, y pueden tener otro formato de pantalla escritos en ellas. Esta frase sólo es válida para dispositivos de pantalla.

ROLLING se especifica en la instrucción WRITE que esté grabando un nuevo formato en la pantalla. El usuario deberá especificar si la grabación es antes o después de girar, así como el rango de líneas que se desea girar, la cantidad de líneas que se desea girar dichas líneas y si la operación de girar es hacia arriba o hacia abajo.

Una vez que las líneas se han girado, los campos en estas líneas retienen sus atributos de pantalla de las DDS (por ejemplo, el subrayado), pero pierden sus atributos de utilización de las DDS (por ejemplo, la posibilidad de entrada). Los campos en líneas que se graban y después se giran (frase BEFORE ROLLING) también pierden sus atributos de utilización.

Si se gira cualquier parte de un formato, todo el formato pierde sus atributos de utilización. Si hay más de un formato, sólo los formatos girados pierden sus atributos de utilización.

Cuando especifique la frase ROLLING, se aplican las normas generales siguientes.

- La palabra clave ALWROL a nivel de registro de las DDS debe especificarse para cada formato de registro grabado en una instrucción WRITE que contiene la frase ROLLING.
- No han de utilizarse otras palabras clave DDS que se excluyan mutuamente con la palabra clave ALWROL.
- Ambas palabras clave DDS, CLRL u OVERLAY, deben especificarse para un formato de registro que se va a grabar y girar para evitar que la pantalla de visualización se borre cuando se graba el formato de registro. Consulte el manual *DDS Reference* para más información acerca de las palabras clave DDS.
- Todos los identificadores y literales deben representar valores enteros positivos.
- El número de la línea inicial del giro (identificador-5 o literal-4) no debe sobrepasar el número de línea final (identificador-6 o literal-5).
- Desaparece el contenido de las líneas que al girar salen fuera de la ventana especificada por los números de línea inicial y final.

La Figura 67 en la página 207 muestra un ejemplo de operación de giro. Se escribe un formato de pantalla inicial, FMT1, en la pantalla. El programa procesa este formato de pantalla y ahora está listo para escribir el siguiente formato de

pantalla, FMT2, en la pantalla de la estación de trabajo. Una parte del FMT1 se gira dos líneas hacia abajo antes de que se escriba FMT2 en la pantalla.

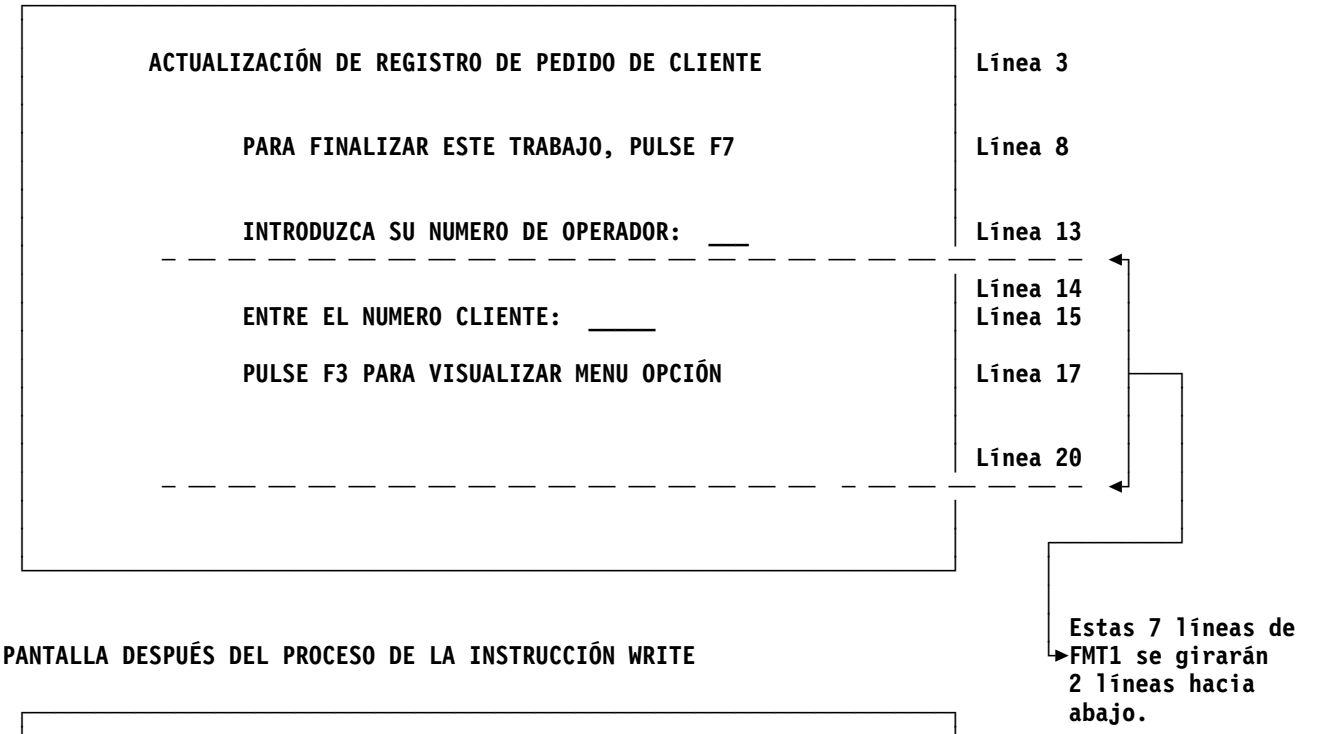
El proceso de la instrucción WRITE siguiente provoca que una parte de FMT1 se gire dos líneas hacia abajo, y que FMT2 se escriba en la pantalla de la estación de trabajo:

```
WRITE SCREENREC FORMAT "FMT2"
 AFTER ROLLING LINES 14 THROUGH 20
 DOWN 2 LINES
```

Cuando se ejecuta la instrucción WRITE, se realizan los pasos siguientes:

1. El contenido de las líneas 14 a 20 se giran hacia abajo dos líneas.
   a. El contenido de las líneas 14 a 18 aparecen ahora en las líneas 16 a 20.
   b. El contenido de las líneas 14 y 15 está vacío y borrado.
   c. El contenido de las líneas 19 y 20 al girar se traslada fuera de la ventana y desaparece.
2. Una vez que se lleva a cabo la operación de giro, se graba FMT2 en la pantalla.
   a. Se escribe parte de FMT2 en el área desocupada por la operación de giro.
   b. Se escribe parte de FMT2 encima de los datos que quedan de FMT1.
3. Cuando una instrucción READ devuelve el contenido de la pantalla al programa, sólo se devuelven los campos de FMT2 con capacidad de entrada.

PANTALLA ANTES DEL PROCESO DE LA INSTRUCCIÓN WRITE



PANTALLA DESPUÉS DEL PROCESO DE LA INSTRUCCIÓN WRITE

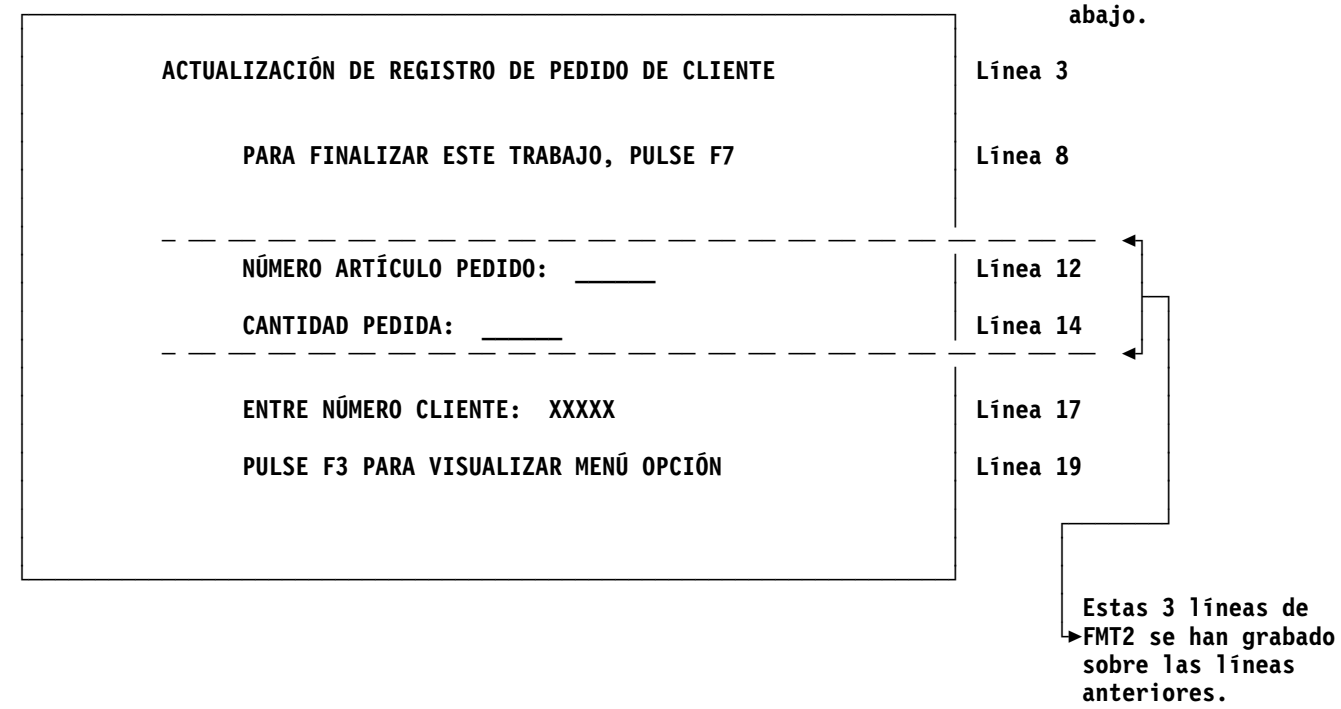
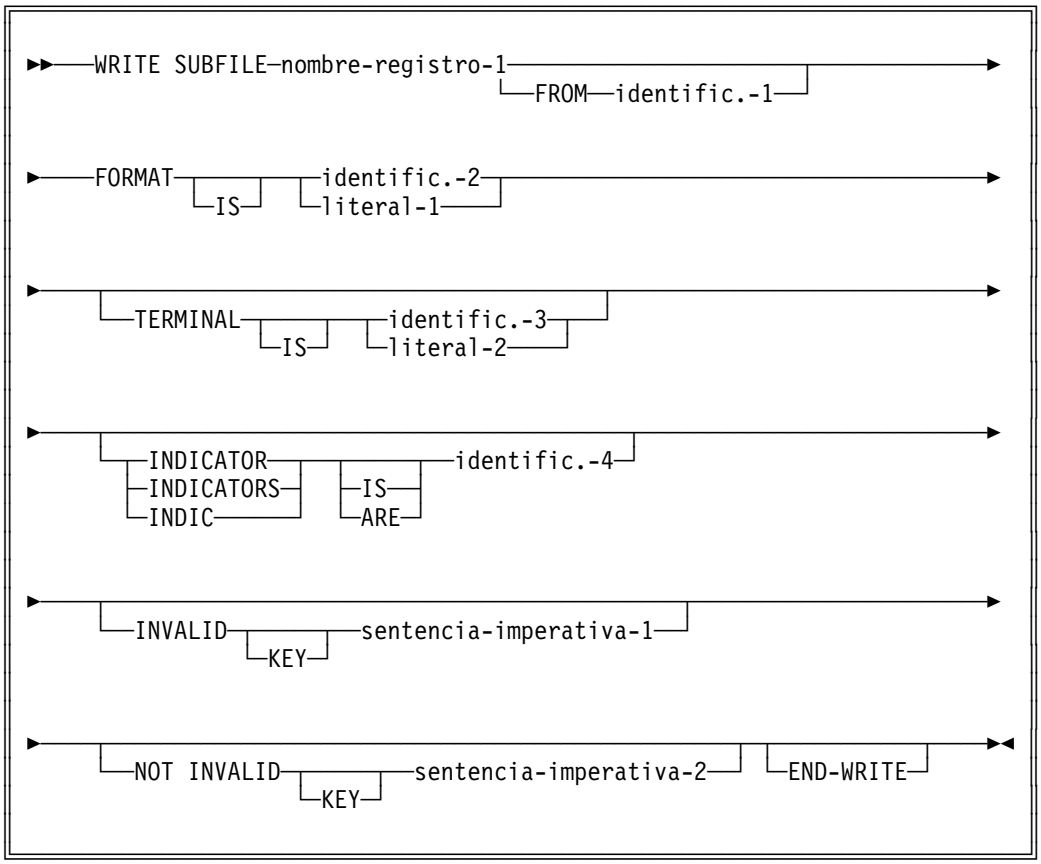


Figura 67. Ejemplo de Operación ROLLING

Instrucción WRITE – Formato 5 – Archivo TRANSACTION (Subarchivo)



El formato 5 sólo se puede utilizar para dispositivos de pantalla. Si se utiliza el formato de subarchivo de la instrucción WRITE para cualquier otro tipo de dispositivo, falla la operación WRITE y el estado del archivo se establece en 90.

Si el formato es un registro de subarchivo, y se especifica SUBFILE, tiene que haberse especificado la cláusula RELATIVE KEY en la cláusula SELECT para el archivo a grabar. El registro grabado en el subarchivo es el registro del subarchivo identificado mediante el nombre de formato que tiene un número relativo de registro igual al valor del ítem de datos RELATIVE KEY. Vea la publicación *Guía para la Gestión de Datos* para más información acerca de los subarchivos.

Frase TERMINAL

Consulte la explicación que sigue al Formato 4 para ver las consideraciones generales concernientes a la frase TERMINAL.

La frase TERMINAL especifica en qué subarchivo de dispositivo de programa va a grabarse un registro. Si se especifica la frase TERMINAL, el literal-2 o el identificador-3 deben hacer referencia a una estación de trabajo asociada con el archivo TRANSACTION. Si el literal-2 o el identificador-3 contienen un valor en blanco, la frase TERMINAL se trata como si no se hubiera especificado. La

estación de trabajo estación de trabajo especificada mediante la frase TERMINAL debe adquirirse explícita o implícitamente.

Si se omite la frase TERMINAL, el subarchivo utilizado es el subarchivo asociado con el dispositivo de programa por omisión. Consulte el análisis de la frase TERMINAL en la página 191, para ver cómo se determina el dispositivo de programa por omisión.

Frase INVALID KEY

La condición INVALID KEY existe si un registro ya está en el subarchivo con ese número de registro, o si el número relativo de registro especificado es mayor que el máximo número de registro de subarchivo permitido. La frase INVALID KEY debe especificarse en la instrucción WRITE SUBFILE para todos los archivos en los que no se especifique un procedimiento USE adecuado.

Para más información acerca de lo que ocurre cuando aparece la condición INVALID KEY, consulte los diagramas en las páginas 80 a 82.

Frase NOT INVALID KEY

Esta frase le permitirá especificar los procedimientos a realizar cuando la operación WRITE sea satisfactoria.

Frase END-WRITE

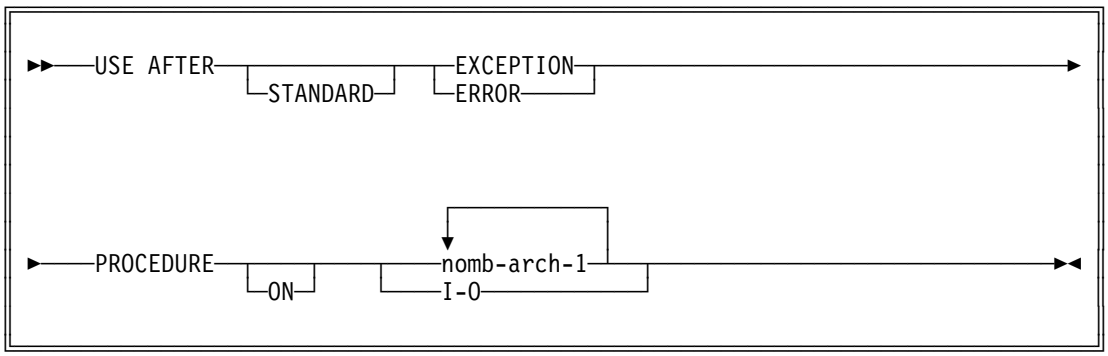
La frase END-WRITE sirve para delimitar explícitamente el ámbito de la instrucción.

Para un análisis más detallado de la instrucción WRITE, de la frase FROM y de la frase INVALID KEY, vea el manual *COBOL/400 Reference*. Para más información acerca de la frase FORMAT, consulte la División de Procedimientos, en el apartado “Recursos Comunes de Proceso” en la página 190.

Instrucción USE

La instrucción USE especifica los procedimientos para el manejo de errores de entrada/salida que existen junto con los procedimientos estándar proporcionados por el sistema de control de entrada/salida.

Formato



Consulte la sección “Instrucción USE” de la publicación *COBOL/400 Reference* para un análisis más detallado de la instrucción USE.

Programas Ejemplo de Estación de Trabajo

Esta sección contiene ejemplos de programas COBOL que ilustran las aplicaciones de estaciones de trabajo en el sistema AS/400.

Programa Básico de Consulta

La Figura 68 muestra las DDS asociadas a un programa básico de consulta que utiliza el archivo TRANSACTION de COBOL

IBM International Business Machines

AS/400 ESPECIFICACIONES DE DESCRIPCION DE DATOS

GX21-8891-0 UM/050* Impreso en España
*El número de hojas por área de relleno puede variar ligeramente.

Archivo		Instrucciones de Grabación	Signo								Descripción	Página de
Programador	Fecha		Teclа									

A

Número de Secuencia	Tipo de Formulario	Anid/Or/Comment. (A/O/*)	Condicionamiento: No (N)	Nombre Condición: Indicador	No (N)	Indicador	No (N)	Indicador	Tipo Nombre o Espac (B/R/N/S/I/K/S/O)	Reservado	Nombre	Referencia (R)	Longitud	Tipo Datos/Desplazamiento Teclado	Posiciones Decimales	Utilización (B/O/I/I/B/H/M/N/P)	Ubicación: Línea	Pos	Funciones
	A	*	ARCHIVO MAESTRO DE CONSULTA DE CLIENTE -- CUSMINQ																
	A	*																	
	A																		REF(CUSMSTP)
	A								R		CUSPMT								TEXT('SOLICITUD DE CLIENTE')
	A																		CA03(15 'FIN DE PROGRAMA')
	A																1	3	'CONSULTA MAESTRA CLIENTES'
	A																3	3	'NUMERO CLIENTE'
	A										CUST	R				I	3	26	
	A		99																ERRMSG('NUMERO DE CLIENTE NO ENCON +
	A																		TRADO PULSE RESTAURAR, LUEGO INTRO +
	A																		DUCIR NUMERO VALIDO' 99)
	A																5	3	'UTILICE F3 PARA FINALIZAR PROGRAMA, +
	A																		UTILICE LA TECLA INTRO PARA VOLVER +
	A																		A SOLICITAR PANTALLA'
	A								R		CUSFLDS								TEXT('PANTALLA DE CLIENTE')
	A																		CA03(15 'FIN DE PROGRAMA')
	A																		OVERLAY
	A																8	3	'NOMBRE'
	A										NAME	R					8	11	
	A																9	3	'DIRECCION'
	A										ADDR	R					9	11	
	A																10	3	'CIUDAD'
	A										CITY	R					10	11	
	A																11	3	'PROVINCIA'
	A										STATE	R					11	11	
	A																11	21	'CODIGO POSTAL'
	A										ZIP	R					11	31	
	A																12	3	'BALANCE A/R'
	A										ARBAL	R					12	17	
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		

Figura 68. Ejemplo de Programa de Consulta TRANSACTION que Utiliza un Solo Dispositivo de Pantalla

Las especificaciones de descripción de datos (DDS) para el archivo de dispositivo de pantalla (CUSMINQ) que utiliza este programa describen dos formatos de registro: CUSPMT y CUSFLDS.

El formato de registro CUSPMT contiene la constante 'Consulta Muestra Clientes', que identifica a la pantalla. Contiene, asimismo, el indicador 'Número Cliente' y el campo de entrada (CUST) en el que puede entrar el número de cliente. En la pantalla en la que se va a entrar el número de cliente, aparecerán cinco caracteres de subrayado debajo del campo de entrada CUST. El mensaje de error:

```
Número de cliente no encontrado
```

también se incluye en este formato de registro. Este mensaje se visualiza si el programa **ACTIVA** el indicador 99. Además, este formato de registro define una tecla de función que se puede pulsar para finalizar el programa. Cuando se pulsa la tecla de función F3, el indicador 15 se **ACTIVA** en el programa COBOL. Este indicador se utiliza para finalizar el programa.

El formato de registro CUSFLDS contiene las constantes siguientes:

- Nombre
- Dirección
- Ciudad
- Provincia
- Código Postal
- Saldo C/C.

Estas constantes identifican los campos a grabar fuera del programa. Este formato de registro también describe los campos que corresponden a estas constantes. Todos estos campos se describen como campos de salida (espacio en blanco en la posición 38) debido a que los rellena el programa; no entre ningún dato en estos campos. Para entrar otro nombre de cliente, pulse Intro en respuesta a este registro. Observe que el registro CUSFLDS recubre el registro CUSPMT. Por lo tanto, cuando el registro CUSFLDS se graba en la pantalla, el registro CUSPMT permanece en pantalla.

Además de la descripción de constantes, campos y atributos para la pantalla, los formatos de registro también definen los números de líneas y las posiciones horizontales en las que se van a visualizar las constantes y campos.

Nota: Los atributos de campo se definen en un archivo físico (CUSMSTP) utilizado a efectos de referencia de campos, en lugar de hacerlo en las DDS para el archivo de pantalla. Por ejemplo, EDTCDE(J) se define en CUSMSTP para el campo ARBAL.

IBM International Business Machines

AS/400 ESPECIFICACIONES DE DESCRIPCION DE DATOS

GX21-9801-0 UM/050*
Impreso en España
*El número de hojas por área de relleno puede variar ligeramente.

Archivo		Instrucciones de Grabación	Signo								Descripción	Página de
Programador	Fecha		Teclа									

A

Número de Secuencia	Tipo de Formulario	And/Or/Coment. (A/O/*)	Condicionamiento: No (N)	Nombre Condición: Indicador	No (N)	Indicador	No (N)	Indicador	Tipo Nombre o Campo (B/N/H/J/K/S/O)	Reservado	Nombre	Referencia (R)	Longitud	Tipo Dato/Desplazamiento Teclado	Posiciones Decimales	Utilización (B/O/I/E/H/M/N/P)	Ubicación: Línea	Pos	Funciones
A**			FISICO						CUSMSTP		ARCHIVO MAESTRO DE CLIENTES								
A								R	CUSMST										TEXT('REGISTRO DE CABECERA DE PEDIDOS')
A									CUST				5						TEXT('NUMERO DE CLIENTE')
A									NAME				25						TEXT('NOMBRE DE CLIENTE')
A									ADDR				20	/					TEXT('DIRECCION DE CLIENTE')
A									CITY				20	/					TEXT('CIUDAD DE CLIENTE')
A									STATE				2						TEXT('PROVINCIA')
A									ZIP				5	00	/				TEXT('CODIGO POSTAL')
A									SRHCOD				6						TEXT('CODIGO BUSQUEDA NUM CLIENTE')
A									CUSTYP				1	00	/				TEXT('TIPO CLIENTE 1=GOB 2=PLAN +
A																			3=BUS 4=PVT 5=OT')
A									ARBAL				8	02	/				TEXT('SALDO CUENTAS POR COBRAR')
A									ORDBAL				8	02	/				TEXT('IMPORTE A/R EN ARCH PEDIDOS')
A									LSTAMT				8	02	/				TEXT('ULTIMO IMPORTE PAGADO EN A/R')
A									LSTDAT				6	00	/				TEXT('ULTIMA FECHA PAGADA EN A/R')
A									CRDLMT				8	02	/				TEXT('LIMITE CREDITO CLIENTE')
A									SLSYR				10	02	/				TEXT('VENTAS CLIENTE ESTE AÑO')
A									SLSLYR				10	02	/				TEXT('VENTAS CLIENTE AÑO PASADO')
A								K	CUST										
A																			

Figura 69. Especificaciones de Descripción de Datos para un Formato de Registro CUSMST

Las especificaciones de descripción de datos (DDS) para el archivo de base de datos que utiliza este programa describen un formato de registro: CUSMST. Se describe cada campo en el formato de registro, y el campo CUST se identifica como el campo de clave para el formato de registro.

```
5763CB1 V3R0M5                  Listado Fuente AS/400 COBOL
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
    1  000100 IDENTIFICATION DIVISION.                                                          01/22/94
    2  000200 PROGRAM-ID.     XMPLE766.                                                         03/22/94
       000300*    EJEMPLO PROG. DE CONSULTA TRANSACTION QUE UTILIZA 1 DISP. DE PANT.            01/22/94
    3  000400 AUTHOR.       PROGRAMMER NAME.                                                    01/22/94
    4  000500 INSTALLATION. TORONTO COBOL DEVELOPMENT CENTRE.                                   01/22/94
    5  000600 DATE-WRITTEN. 12/21/88.                                                           01/22/94
    6  000070 DATE-COMPILED.  05/24/94 13:42:50   .
    7  000800 ENVIRONMENT DIVISION.                                                             01/22/94
    8  000900 CONFIGURATION SECTION.                                                            01/22/94
    9  001000 SOURCE-COMPUTER. IBM-AS400.                                                       01/22/94
   10  001100 OBJECT-COMPUTER. IBM-AS400.                                                       01/22/94
   11  001200 INPUT-OUTPUT SECTION.                                                             01/22/94
   12  001300 FILE-CONTROL.                                                                     01/22/94
   13  001400     SELECT CUST-DISPLAY                                                           01/22/94
   14  001500         ASSIGN TO WORKSTATION-CUSMINQ                                             01/22/94
   15  001600         ORGANIZATION IS TRANSACTION                                               01/22/94
   16  001700         CONTROL-AREA IS WS-CONTROL.                                               01/22/94
   17  001800     SELECT CUST-MASTER                                                            01/22/94
   18  001900         ASSIGN TO DATABASE-CUSMSTP                                                01/22/94
   19  002000         ORGANIZATION IS INDEXED                                                   01/22/94
   20  002100         ACCESS IS RANDOM                                                          01/22/94
   21  002200             RECORD KEY IS CUST OF CUSMST                                          01/22/94
   22  002300         FILE STATUS IS CM-STATUS.                                                 01/22/94
   23  002400 DATA DIVISION.                                                                    01/22/94
   24  002500 FILE SECTION.                                                                     01/22/94
   25  002600 FD  CUST-DISPLAY                                                                  01/22/94
   26  002700     LABEL RECORDS ARE OMITTED.                                                    01/22/94
   27  002800 01  DISP-REC.                                                                    01/22/94
   28  002900     COPY DDS-ALL-FORMATS OF CUSMINQ.                                              01/22/94
   29 +000001       05  CUSMINQ-RECORD PIC X(80).                                 <-ALL-FMTS
      +000002*  FORMATO ENTRADA:CUSPMT    DESDE ARCHIVO CUSMINQ   DE BIBLIOTECA XMPLIB     <-ALL-FMTS
      +000003*                            SOLICITUD CLIENTE                               <-ALL-FMTS
   30 +000004       05  CUSPMT-I      REDEFINES CUSMINQ-RECORD.                          <-ALL-FMTS
   31 +000005           06 CUSPMT-I-INDIC.                                                <-ALL-FMTS
   32 +000006                07 IN15         PIC 1  INDIC 15.                           <-ALL-FMTS
      +000007*                              FIN DE PROGRAMA                               <-ALL-FMTS
   33 +000008                07 IN99         PIC 1  INDIC 99.                           <-ALL-FMTS
      +000009*                             NÚMERO CLIENTE NO HALLADO, PULSAR REST, EL     <-ALL-FMTS
   34 +000010           06 CUST              PIC X(5).                                  <-ALL-FMTS
      +000011*                              NÚMERO CLIENTE                                <-ALL-FMTS
      +000012*FORMATO SALIDA:CUSPMT    DESDE ARCHIVO CUSMINQ    DE BIBLIOTECA XMPLIB     <-ALL-FMTS
      +000013*                            SOLICITUD CLIENTE                               <-ALL-FMTS
   35 +000014       05  CUSPMT-O      REDEFINES CUSMINQ-RECORD.                          <-ALL-FMTS
   36 +000015           06 CUSPMT-O-INDIC.                                                <-ALL-FMTS
   37 +000016                07 IN99         PIC 1  INDIC 99.                           <-ALL-FMTS
      +000017*                             NÚMERO CLIENTE NO HALLADO, PULSAR REST, EL     <-ALL-FMTS
      +000018*  FORMATO ENTRADA:CUSFLDS    DESDE ARCHIVO CUSMINQ   DE BIBLIOTECA XMPLIB    <-ALL-FMTS
      +000019*                              PANTALLA CLIENTE                              <-ALL-FMTS
   38 +000020       05  CUSFLDS-I     REDEFINES CUSMINQ-RECORD.                          <-ALL-FMTS
   39 +000021           06 CUSFLDS-I-INDIC.                                               <-ALL-FMTS
   40 +000022                07 IN15         PIC 1  INDIC 15.                           <-ALL-FMTS
      +000023*                              FIN DE PROGRAMA                               <-ALL-FMTS
      +000024*FORMATO SALIDA:CUSFLDS    DESDE ARCHIVO CUSMINQ    DE BIBLIOTECA XMPLIB     <-ALL-FMTS
      +000025*                              PANTALLA CLIENTE                              <-ALL-FMTS
   41 +000026       05  CUSFLDS-O     REDEFINES CUSMINQ-RECORD.                          <-ALL-FMTS
   42 +000027           06 NAME              PIC X(25).                                 <-ALL-FMTS
      +000028*                              NOMBRE CLIENTE                                <-ALL-FMTS
   43 +000029           06 ADDR              PIC X(20).                                 <-ALL-FMTS
      +000030*                              DIRECCIÓN CLIENTE                             <-ALL-FMTS
   44 +000031           06 CITY              PIC X(20).                                 <-ALL-FMTS
      +000032*                              CIUDAD CLIENTE                                <-ALL-FMTS
   45 +000033           06 STATE             PIC X(2).                                  <-ALL-FMTS
      +000034*                              PROVINCIA                                     <-ALL-FMTS
   46 +000035           06 ZIP               PIC S9(5).                                 <-ALL-FMTS
      +000036*                              CÓDIGO POSTAL                                 <-ALL-FMTS
   47 +000037           06 ARBAL             PIC S9(6)V9(2).                            <-ALL-FMTS
      +000038*                              SALDO CUENTAS PEND                            <-ALL-FMTS
       003000
   48  003100 FD  CUST-MASTER
   49  003200     LABEL RECORDS ARE STANDARD.
   50  003300 01  CUST-REC.
   51  003400     COPY DDS-CUSMST OF CUSMSTP.
      +000001*   FORMATO E-S:CUSMST   DESDE CUSMSTP  DE BIBLIOTECA XMPLIB           CUSMST
```

Figura 70 (Parte 1 de 2). Listado Fuente de un Programa de Consulta TRANSACTION que Utiliza un Solo Dispositivo de Pantalla

```
5763CB1 V3R0M5                  Listado Fuente AS/400 COBOL
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
      +000002*                          REGISTRO MAESTRO CLIENTE                          CUSMST
      +000003*DEFINICIONES CLAVE PARA FORMATO DE REGISTRO CUSMST                           CUSMST
      +000004*  NÚMERO              NOMBRE            RECUPERACIÓN  TIPO    SECALT       CUSMST
      +000005*   0001   CUST                          ASCENDENTE     AN      NO         CUSMST
   52 +000006       05  CUSMST.                                                          CUSMST
   53 +000007           06 CUST             PIC X(5).                                    CUSMST
      +000008*                          NÚMERO CLIENTE                                    CUSMST
   54 +000009           06 NAME             PIC X(25).                                   CUSMST
      +000010*                          NOMBRE CLIENTE                                    CUSMST
   55 +000011           06 ADDR             PIC X(20).                                   CUSMST
      +000012*                          DIRECCIÓN CLIENTE                                 CUSMST
   56 +000013           06 CITY             PIC X(20).                                   CUSMST
      +000014*                          CIUDAD CLIENTE                                    CUSMST
   57 +000015           06 STATE            PIC X(2).                                    CUSMST
      +000016*                          PROVINCIA                                         CUSMST
   58 +000017           06 ZIP              PIC S9(5)        COMP-3.                     CUSMST
      +000018*                          CÓDIGO POSTAL                                     CUSMST
   59 +000019           06 SRHCOD           PIC X(6).                                    CUSMST
      +000020*                          CÓDIGO BÚSQUEDA NÚMERO CLIENTE                    CUSMST
   60 +000021           06 CUSTYP           PIC S9(1)        COMP-3.                     CUSMST
      +000022*                          TIPO CLIENTE  1=GOB 2=PLA 3=NEG 4=FAB 5=OTR       CUSMST
   61 +000023           06 ARBAL            PIC S9(6)V9(2)   COMP-3.                     CUSMST
      +000024*                          SALDO CUENTAS PEND                                CUSMST
   62 +000025           06 ORDBAL           PIC S9(6)V9(2)   COMP-3.                     CUSMST
      +000026*                          CANTIDAD C/C EN ARCH. PEDIDOS                     CUSMST
   63 +000027           06 LSTAMT           PIC S9(6)V9(2)   COMP-3.                     CUSMST
      +000028*                          ULTIMA CANTIDAD PAGADA EN C/C                     CUSMST
   64 +000029           06 LSTDAT           PIC S9(6)        COMP-3.                     CUSMST
      +000030*                          ULTIMA FECHA PAGADA EN C/C                        CUSMST
   65 +000031           06 CRDLMT           PIC S9(6)V9(2)   COMP-3.                     CUSMST
      +000032*                          LÍMITE CRÉDITO CLIENTE                            CUSMST
   66 +000033           06 SLSYR            PIC S9(8)V9(2)   COMP-3.                     CUSMST
      +000034*                          VENTAS CLIENTE ESTE AÑO                           CUSMST
   67 +000035           06 SLSLYR           PIC S9(8)V9(2)   COMP-3.                     CUSMST
      +000036*                          VENTAS CLIENTE AÑO PASADO                         CUSMST
       003500
   68  003600 WORKING-STORAGE SECTION.
   69  003700 01  ONE                              PIC 1 VALUE B"1".
   70  003800 01  CM-STATUS                       PIC X(2).
   71  003900 01  WS-CONTROL.
   72  004000     02  WS-IND                       PIC X(2).
   73  004100     02  WS-FORMAT                    PIC X(10).
   74  004200 PROCEDURE DIVISION.
       004300 BEGIN.
   75  004400     OPEN I-O CUST-DISPLAY, INPUT CUST-MASTER.
   76  004500     MOVE ZERO TO IN99 OF CUSPMT-O.
       004600 LOOP.
   77  004700     WRITE DISP-REC FORMAT IS "CUSPMT".
   78  004800     READ CUST-DISPLAY RECORD.
   79  004900     IF IN15 OF CUSPMT-I
       005000       IS EQUAL TO ONE
   80  005100        THEN GO TO FINIS.
   81  005200     MOVE CUST OF CUSPMT-I TO CUST OF CUSMST.
   82  005300     READ CUST-MASTER RECORD.
   83  005400     IF CM-STATUS IS NOT EQUAL "00" THEN
   84  005500         MOVE ONE TO IN99 OF CUSPMT-O,  GO TO LOOP.
   86  005600     MOVE CORRESPONDING CUSMST TO CUSFLDS-O.
   87  005700     WRITE DISP-REC FORMAT IS "CUSFLDS".
   88  005800     READ CUST-DISPLAY RECORD.
   89  005900     IF IN15 OF CUSFLDS-I
       006000     IS EQUAL TO ONE
   90  006100     THEN GO TO FINIS.
   91  006200     MOVE ZERO TO IN99 OF CUSPMT-O.
   92  006300     GO TO LOOP.
       006400 FINIS.
   93  006500     CLOSE CUST-DISPLAY, CUST-MASTER.
       006600 RETURN-TO-CALLER.
   94  006700     EXIT PROGRAM.
                        * * * * *  F I N  D E  F U E N T E  * * * * *
```

Figura 70 (Parte 2 de 2). Listado Fuente de un Programa de Consulta TRANSACTION que Utiliza un Solo Dispositivo de Pantalla

El listado fuente para este programa de ejemplo se muestra aquí en su totalidad. En especial, observe las entradas FILE-CONTROL y FD y las estructuras de datos generadas por las instrucciones COPY de Formato 2.

La operación WRITE en la instrucción 77 escribe el formato CUSPMT en la pantalla. Este registro solicita que el usuario entre un número de cliente. Si entra un número de cliente y pulsa Intro, entonces la operación READ lee el registro de nuevo en el programa.

La operación READ en la instrucción 82 utiliza el campo de número de cliente (CUST) para recuperar el registro CUSMST correspondiente del archivo CUSMSTP. Si no se encuentra ningún registro en el archivo CUSMSTP, el indicador 99 se activa. La operación GO TO en la instrucción 84, que se ejecuta cuando se activa el indicador 99, provoca que el programa se bifurque hacia el comienzo. El mensaje:

```
Número cliente no hallado
```

se visualiza cuando se escribe el formato, porque está condicionado por el indicador 99 en las DDS para el archivo. Al recibir este mensaje, el teclado se bloquea. Debe pulsar la tecla Reset en respuesta a ese mensaje para desbloquear el teclado. El usuario puede entonces entrar otro número de cliente.

Si la operación READ recupera un registro del archivo CUSMSTP, la operación WRITE escribe el registro CUSFLDS en la estación de trabajo de la pantalla. Este registro contiene el nombre de cliente, la dirección y el saldo de cuentas por cobrar.

Puede entonces pulsar Intro, y el programa se bifurca hacia el comienzo. Puede introducir otro número de cliente o finalizar el programa. Para finalizar el programa, pulse F3, que activa el indicador 15 en el programa.

Cuando se activa el indicador 15, el programa cierra todos los archivos y procesa la instrucción EXIT PROGRAM. El programa devuelve el control a la persona que llamó al programa COBOL.

Esta es la pantalla inicial grabada por la operación WRITE en la instrucción 77:

```
Consulta Maestro de Clientes

Número Cliente   ________

Utilice F3 para finalizar programa, utilice la tecla Intro para regresar a
pantalla de solicitud
```

Esta pantalla aparece si se encuentra un registro en el archivo CUSMSTP para el número de cliente entrado en respuesta a la primera pantalla:

```
Consulta Maestro de Clientes

Número Cliente    1000

Utilice F3 para finalizar programa, utilice la tecla Intro para regresar a
pantalla de solicitud


Nombre       EXAMPLE WHOLESALERS LTD.
Dirección    ANYWHERE STREET
Ciudad       ACITY
Provincia    IL        Cód. Postal 12345
Saldo C/C              137.02
```

Esta pantalla aparece si el archivo CUSMSTP no contiene un registro para el número de cliente entrado en respuesta a la primera pantalla:

```
Consulta Maestro de Clientes

Número Cliente

Utilice F3 para finalizar programa, utilice la tecla Intro para regresar a
pantalla de solicitud




No encontrado número cliente, pulsar Reset, luego entrar número válido
```

Programas de Consulta de Pedidos que utilizan Subarchivos

La Figura 72 en la página 220 muestra un ejemplo de un programa de consulta de pedidos, XMPLE773, que utiliza subarchivos. Se muestran asimismo las DDS asociadas, excepto las DDS del archivo maestro de clientes, CUSMSTP. Consulte la Figura 69 en la página 212 para las DDS de CUSMSTP.

XMPLE773 visualiza todos los registros de detalle del pedido para el número de pedido solicitado. El programa solicita al usuario que entre el número de pedido que debe revisarse. El número de pedido se comprueba con el archivo de cabecera de pedido, ORDHDRP. Si existe el número de pedido, el número de cliente accedido desde el archivo de cabecera de pedido se comprueba con el archivo maestro de clientes, CUSMSTP. Se leen y se graban en el subarchivo todos los registros de detalle de pedido en ORDDTLP para el pedido solicitado. Se procesa una grabación para el formato de registro de control de subarchivo, y los registros de detalle de pedido en el subarchivo se visualizan en la pantalla para que los revise el usuario. El programa finaliza pulsando F12.

IBM International Business Machines

AS/400 ESPECIFICACIONES DE DESCRIPCION DE DATOS

GX21-9861-0 UM/060* Impreso en España
*El número de hojas por área de relleno puede variar ligeramente.

Archivo		Instrucciones de Grabación	Signo								Descripción	Página	de
Programador	Fecha		Tecla										

A

Número de Secuencia	Tipo de Formulario	And/Or/Comment. (A/O/*)	Condicionamiento: No (N)	Nombre Condición: Indicador	No (N)	Indicador	No (N)	Indicador	Tipo Nombre o Base (R/E/I/J/S/K/blanco)	Reservado	Nombre	Referencia (R)	Longitud	Tipo Datos/Desplazamiento Teclado	Posiciones Decimales	Utilización (B/O/I/H/M/P)	Ubicación: Línea	Pos	Funciones
1 2 3 4 5	6	7	8	9 10 11	12 13	14 15	16 17	18	19	20	21 22 23 24 25 26 27 28	29	30 31 32 33	34	35 36	37 38	39 40 41	42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
	A	*	*	PHYSICAL							ORDDTLP		ARCHIVO DE DETALLE DE PEDIDO						
	A																		TEXT('UBICACION ALMACEN')
	A	*																	
	A								R		ORDDTL								TEXT('REGISTRO DETALLE PEDIDO')
	A	*																	
	A										CUST		5						CHECK(MF)
	A																		COLHDG('CLIENTE' 'NUMERO')
	A	*																	
	A										ORDERN		5		0				COLHDG('PEDIDO' 'NUMERO')
	A	*																	
	A										LINNUM		3		0				
	A																		COLHDG('LINEA' 'NO')
	A																		TEXT('NUM LINEA DE LINEA EN PEDIDO'+
	A																		)
	A	*																	
	A										ITEM		5		0				CHECK(M10)
	A																		COLHDG('ITEM' 'NUMERO')
	A										QTYORD		3		0				
	A																		COLHDG('CANTIDAD' 'PEDIDA')
	A																		TEXT('CANTIDAD PEDIDA')
	A	*																	
	A										DESCRP		30						COLHDG('DESCRIPCION ITEM')
	A	*																	
	A										PRICE		6		2				CMP(GT 0)
	A																		COLHDG('PRECIO')
	A																		TEXT('PRECIO VENTA')
	A																		EDTCDE(J)
	A										EXTENS		8		2				COLHDG('EXTENSION')
	A																		TEXT('CALCULO IMPORTE DE PRECIO X+
	A																		QTYORD')
	A	*																	
	A										WHSLOC		3						CHECK(MF)
	A																		COLHDG('BIN' 'NO.')
	A	*																	
	A										ORDDAT		6		0				TEXT('INTRODUCIDO EL PEDIDO+
	A																		DE FECHA')
	A										CUSTYP		1		0				RANGE(1 5)
	A																		COLHDG('CLIEN' 'TIPO')
	A																		TEXT('TIPO CLIENTE 1=GOB 2=PLAN +
	A																		3=BUS 4=PVT 5=OT')
	A	*																	
	A										STATE		2						CHECK(MF)
	A																		COLHDG('PROVINCIA')
	A	*																	
	A	*																	
	A										ACTMTH		2		0				COLHDG('MES' 'CONT')
	A																		TEXT('MES CONTABLE DE VENTA')
	A	*																	
	A										ACTYR		2		0				COLHDG('AÑO' 'CONT')
	A																		TEXT('AÑO CONTABLE DE VENTA')
	A								K		ORDERN								
	A								K		LINNUM								
	A																		

Figura 71 (Parte 1 de 3). Especificaciones de Descripción de Datos para un Programa de Consulta de Pedidos

IBM International Business Machines

AS/400 ESPECIFICACIONES DE DESCRIPCION DE DATOS

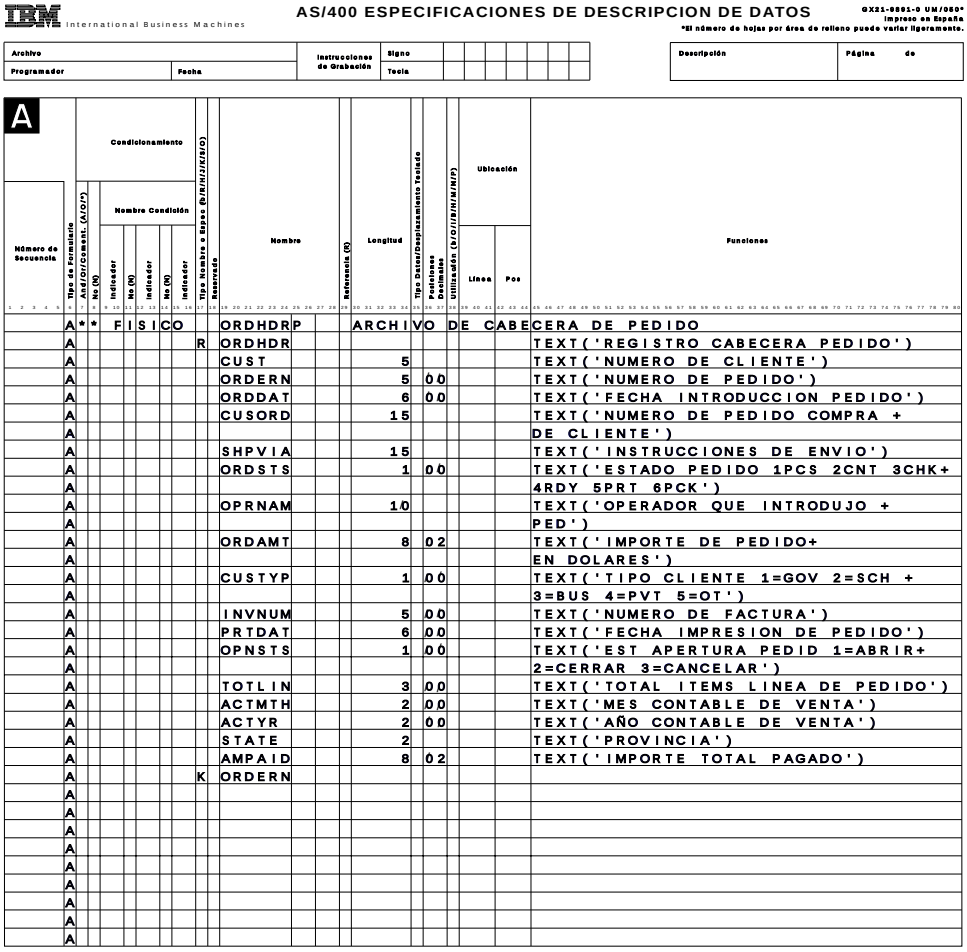
GX21-9091-0 UM/050*
Impreso en España
*El número de hojas por área de relleno puede variar ligeramente.

Archivo		Instrucciones de Grabación	Signo								Descripción	Página	de
Programador	Fecha		Teclа										

A

Número de Secuencia	Tipo de Formulario	Condicionamiento: Add/O/C/Consent. (A/O/*)	Condicionamiento: Nombre Condición No (N)	Condicionamiento: Nombre Condición Indicador	Condicionamiento: Nombre Condición No (N)	Condicionamiento: Nombre Condición Indicador	Condicionamiento: Nombre Condición No (N)	Condicionamiento: Nombre Condición Indicador	Tipo Nombre o Espac. (B/I/N/H/J/K/R/I/O)	Reservado	Nombre	Referencia (R)	Longitud	Tipo Datos/Desplazamiento Teclado	Posiciones Decimales	Utilización (B/O/I/U/B/N/H/N/P)	Ubicación: Línea	Ubicación: Pos	Funciones
	A	*	ORDINQD						R		REVISION DE PEDIDO EXISTENTE								
	A								R		SUB1								SFL
	A										ITEM		5		0		10	2	TEXT('NUMERO ARTICULO')
	A										QTYORD		3		0		10	9	TEXT('CANTIDAD PEDIDA')
	A										DESCRP		30				10	14	TEXT('DESCRIPCION ITEM')
	A										PRICE		6		2		10	46	TEXT('PRECIO VENTA')
	A										EXTENS		8		2		10	56	EDTCDE(J)
	A																		TEXT('CALCULO VALOR DE+
	A																		PEDIDO QTYORD X')
	A								R		SUBCTL1								SFLCTL(SUB1)
	A		58																SFLCLR
	A		57																SFLDSP
	A		N58																SFLDSPCTL
	A																		SFLSIZ(57)
	A																		SFLPAG(14)
	A		57																SFLEND
	A																		OVERLAY
	A																		LOCK
	A		N45																
	A		ON47																ROLLUP(97 'CONTINUAR VISUALIZ.')
	A																		CA12(98 'FIN DE PROGRAMA')
	A																		SETOFF(57 'SUBARCHIVO DE PANTALLA')
	A																		SETOFF(58 'DESAC=VISU. SUBCTL1 AC=+
	A																		BORRAR SUBARCHIVO')
	A																1	2	'CONSULTA PEDIDO EXISTENTE'
	A																3	2	'PEDIDO'
	A										ORDERN		5	Y	0	B	3	8	TEXT('NUMERO PEDIDO')
	A		61																ERRMSG('NUM PEDIDO NO ENCONTRADO'61)
	A		47																ERRMSG('SIN LINEA PARA PEDIDO'47)
	A		62																ERRMSG('NO SE ENCONTRO REGISTRO DE +
	A																		CLIENTE PARA ESTE PEDIDO' 62)
	A																4	2	'FECHA'
	A										ORDDAT		6		0		4	7	TEXT('INTRODUCIDO PEDIDO DE FECHA')
	A																5	2	'CLI #'
	A										CUST		5				5	9	TEXT('NUMERO DE CLIENTE')
	A										NAME		25				3	16	TEXT('NOMBRE DE CLIENTE')
	A										ADDR		20				4	16	TEXT('DIRECCION DE CLIENTE')
	A										CITY		20				5	16	TEXT('CIUDAD DE CLIENTE')
	A										STATE		2				6	16	TEXT('PROVINCIA DE CLIENTE')
	A										ZIP		5		0		6	31	TEXT('CODIGO POSTAL')
	A																1	44	'TOTAL'
	A										ORDAMT		8		2		1	51	TEXT('CANTIDAD TOTAL DEL PEDIDO EN+
	A																		DOLARES')
	A																2	44	'PROVINCIA'
	A										STSORD		12				2	51	
	A																3	44	'ABIERTO'
	A										STSOPN		12				3	51	
	A																4	44	'PEDIDO DE CLIENTE'
	A										CUSORD		15				4	59	TEXT('NUMERO DE PEDIDO DE COM +
	A																		PRA DE CLIENTE')
	A																5	44	'MEDIO ENVIO'
	A										SHPVIA		15				5	59	TEXT('INSTRUCCIONES DE ENVIO')
	A																6	44	'FECHA IMPRESO'
	A										PRTDAT		6		0		6	57	TEXT('IMPRESO PEDIDO DE FECHA')
	A																7	29	'FACTURA'
	A										INVNUM		5		0		7	38	TEXT('NUMERO DE FACTURA')
	A																7	64	'MES'
	A										ACTMTH		2		0		7	68	TEXT('MES CONTABLE DE VENTA')
	A																7	72	'AÑO'
	A										ACTYR		2		0		7	77	TEXT('AÑO CONTABLE DE VENTA')
	A																8	2	'ITEM'
	A																8	8	'CANT'
	A																8	14	'DESCRIPCION ITEM'
	A																8	46	'PRECIO'
	A																8	55	'CALCULO'

Figura 71 (Parte 2 de 3). Especificaciones de Descripción de Datos para un Programa de Consulta de Pedidos



AS/400 ESPECIFICACIONES DE DESCRIPCION DE DATOS

GX21-9801-0 UM/060* Impreso en España
*El número de hojas por área de relleno puede variar ligeramente.

International Business Machines

Archivo		Instrucciones de Grabación	Signo								Descripción	Página de
Programador	Fecha		Tecla									

A

Número de Secuencia	Tipo de Formato	Condicionamiento: And/Or/Comment. (A/O/*)	Condicionamiento: Nombre Condición: No (N)	Condicionamiento: Nombre Condición: Indicador	Condicionamiento: Nombre Condición: No (N)	Condicionamiento: Nombre Condición: Indicador	Condicionamiento: Nombre Condición: No (N)	Condicionamiento: Nombre Condición: Indicador	Tipo Nombre o Espec. (R/K/E/J/S/C/M/O)	Reservado	Nombre	Referencia (R)	Longitud	Tipo Datos/Desplazamiento Tasado	Posiciones Decimales	Utilización (B/O/I/P/N/H/M/KP)	Ubicación: Línea	Ubicación: Pos	Funciones
	A	*	F	I	S	I	C	O			ORDHDRP		ARCHIVO		DE		CABE		CERA DE PEDIDO
	A								R		ORDHDR								TEXT('REGISTRO CABECERA PEDIDO')
	A										CUST		5						TEXT('NUMERO DE CLIENTE')
	A										ORDERN		5		00				TEXT('NUMERO DE PEDIDO')
	A										ORDDAT		6		00				TEXT('FECHA INTRODUCCION PEDIDO')
	A										CUSORD		15						TEXT('NUMERO DE PEDIDO COMPRA +
	A																		DE CLIENTE')
	A										SHPVIA		15						TEXT('INSTRUCCIONES DE ENVIO')
	A										ORDSTS		1		00				TEXT('ESTADO PEDIDO 1PCS 2CNT 3CHK+
	A																		4RDY 5PRT 6PCK')
	A										OPRNAM		10						TEXT('OPERADOR QUE INTRODUJO +
	A																		PED')
	A										ORDAMT		8		02				TEXT('IMPORTE DE PEDIDO+
	A																		EN DOLARES')
	A										CUSTYP		1		00				TEXT('TIPO CLIENTE 1=GOV 2=SCH +
	A																		3=BUS 4=PVT 5=OT')
	A										INVNUM		5		00				TEXT('NUMERO DE FACTURA')
	A										PRTDAT		6		00				TEXT('FECHA IMPRESION DE PEDIDO')
	A										OPNSTS		1		00				TEXT('EST APERTURA PEDID 1=ABRIR+
	A																		2=CERRAR 3=CANCELAR')
	A										TOTLIN		3		00				TEXT('TOTAL ITEMS LINEA DE PEDIDO')
	A										ACTMTH		2		00				TEXT('MES CONTABLE DE VENTA')
	A										ACTYR		2		00				TEXT('AÑO CONTABLE DE VENTA')
	A										STATE		2						TEXT('PROVINCIA')
	A										AMPAID		8		02				TEXT('IMPORTE TOTAL PAGADO')
	A								K		ORDERN								
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		

Figura 71 (Parte 3 de 3). Especificaciones de Descripción de Datos para un Programa de Consulta de Pedidos

```
5763CB1 V3R0M5                  Listado Fuente AS/400 COBOL
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
    1  000100 IDENTIFICATION DIVISION.                                                              01/25/94
    2  000200 PROGRAM-ID.     XMPLE773.                                                             03/22/94
       000300*    EJEMPLO PROG. DE CONSULTA DE PEDIDOS                                              03/22/94
    3  000400 AUTHOR.       PROGRAMMER NAME.                                                        01/25/94
    4  000500 INSTALLATION. TORONTO COBOL DEVELOPMENT CENTRE.                                       01/25/94
    5  000600 DATE-WRITTEN. 12/22/88.                                                               01/25/94
    8  000080 DATE-COMPILED.  05/24/94 13:29:54   .                                                03/01/94
    7  000800 ENVIRONMENT DIVISION.                                                                 01/25/94
    8  000900 CONFIGURATION SECTION.                                                                01/25/94
    9  001000 SOURCE-COMPUTER. IBM-AS400.                                                           01/25/94
   10  001100 OBJECT-COMPUTER. IBM-AS400.                                                           01/25/94
   11  001200 INPUT-OUTPUT SECTION.                                                                 01/25/94
   12  001300 FILE-CONTROL.                                                                         01/25/94
   13  001400     SELECT ORDER-HEADER-FILE                                                          01/25/94
   14  001500         ASSIGN TO DATABASE-ORDHDRP                                                    03/21/94
   15  001600         ORGANIZATION IS INDEXED                                                       01/25/94
   16  001700         ACCESS MODE IS RANDOM                                                         01/26/94
   17  001800         RECORD KEY IS ORDERN OF ORDER-HEADER-RECORD.                                  01/26/94
   18  001900     SELECT ORDER-DETAIL-FILE                                                          01/25/94
   19  002000         ASSIGN TO DATABASE-ORDDTLP                                                    03/21/94
   20  002100         ORGANIZATION IS INDEXED                                                       01/25/94
   21  002200         ACCESS IS DYNAMIC                                                             01/25/94
   22  002300         RECORD KEY IS ORDER-DETAIL-RECORD-KEY.                                        01/27/94
   23  002400     SELECT CUSTOMER-MASTER-FILE                                                       01/25/94
   24  002500         ASSIGN TO DATABASE-CUSMSTP                                                    01/25/94
   25  002600         ORGANIZATION IS INDEXED                                                       01/25/94
   26  002700         ACCESS IS RANDOM                                                              01/25/94
   27  002800         RECORD KEY IS CUST OF CUSTOMER-MASTER-RECORD.                                 01/26/94
   28  002900     SELECT EXISTING-ORDER-DISPLAY-FILE                                                01/25/94
   29  003000         ASSIGN TO WORKSTATION-ORDINQD                                                 03/23/94
   30  003100         ORGANIZATION IS TRANSACTION                                                   01/25/94
   31  003200         ACCESS IS DYNAMIC                                                             01/25/94
   32  003300         RELATIVE KEY IS SUBFILE-RECORD-NUMBER                                         01/25/94
   33  003400         FILE STATUS IS STATUS-CODE-ONE.                                               01/25/94
   34  003500 DATA DIVISION.                                                                        01/25/94
   35  003600 FILE SECTION.                                                                         01/25/94
   36  003700 FD  ORDER-HEADER-FILE                                                                 01/25/94
   37  003800     LABEL RECORDS ARE STANDARD.                                                       01/25/94
   38  003900 01  ORDER-HEADER-RECORD.                                                            01/25/94
   39  004000     COPY DDS-ORDHDR OF ORDHDRP.                                                       03/21/94
      +000001*   FORMATO E-S:ORDHDR   DESDE ORDHDRP  DE BIBLIOTECA XMPLIB                 ORDHDR
      +000002*                         REGISTRO CABECERA PEDIDO                           ORDHDR
      +000003*DEFINICIONES CLAVE PARA FORMATO DE REGISTRO ORDHDR                         ORDHDR
      +000004*  NÚMERO              NOMBRE            RECUPERACIÓN  TIPO    SECALT      ORDHDR
      +000005*   0001   ORDERN                         ASCENDENTE  SIGNO      NO        ORDHDR
   40 +000006       05  ORDHDR.                                                          ORDHDR
   41 +000007           06 CUST            PIC X(5).                                     ORDHDR
      +000008*                         NÚMERO CLIENTE                                     ORDHDR
   42 +000009           06 ORDERN          PIC S9(5)        COMP-3.                      ORDHDR
      +000010*                         NÚMERO PEDIDO                                      ORDHDR
   43 +000011           06 ORDDAT          PIC S9(6)        COMP-3.                      ORDHDR
      +000012*                         SE ENTRÓ FECHA PEDIDO                              ORDHDR
   44 +000013           06 CUSORD          PIC X(15).                                    ORDHDR
      +000014*                         NÚMERO PEDIDO COMPRA CLIENTE                       ORDHDR
   45 +000015           06 SHPVIA          PIC X(15).                                    ORDHDR
      +000016*                         INSTRUCCIONES ENVÍO                                ORDHDR
   46 +000017           06 ORDSTS          PIC S9(1)        COMP-3.                      ORDHDR
      +000018*                         ESTADO PEDIDO 1PEND 2CONT  3COM 4LIS 5IMP 6ENP     ORDHDR
   47 +000019           06 OPRNAM          PIC X(10).                                    ORDHDR
      +000020*                         NOMBRE OPERADOR QUE ENTRÓ EL PEDIDO                ORDHDR
   48 +000021           06 ORDAMT          PIC S9(6)V9(2)   COMP-3.                      ORDHDR
      +000022*                         TOTAL SUMA PESETAS DEL PEDIDO                      ORDHDR
   49 +000023           06 CUSTYP          PIC S9(1)        COMP-3.                      ORDHDR
      +000024*                         TIPO CLIENTE  1=GOB 2=PLA 3=NEG 4=FAB 5=OTR        ORDHDR
   50 +000025           06 INVNUM          PIC S9(5)        COMP-3.                      ORDHDR
      +000026*                         NÚMERO FACTURA                                     ORDHDR
   51 +000027           06 PRTDAT          PIC S9(6)        COMP-3.                      ORDHDR
      +000028*                         FECHA IMPRESIÓN PEDIDO                             ORDHDR
   52 +000029           06 OPNSTS          PIC S9(1)        COMP-3.                      ORDHDR
      +000030*                         EST.PEDIDO ABIERTO 1=ABIERTO 2= CERRADO 3=CANCELAD.ORDHDR
   53 +000031           06 TOTLIN          PIC S9(3)        COMP-3.                      ORDHDR
      +000032*                         TOTAL LÍNEAS ÍTEMS EN PEDIDO                       ORDHDR
   54 +000033           06 ACTMTH          PIC S9(2)        COMP-3.                      ORDHDR
      +000034*                         MES CONTABLE DE VENTA                              ORDHDR
   55 +000035           06 ACTYR           PIC S9(2)        COMP-3.                      ORDHDR
      +000036*                         AÑO CONTABLE DE VENTA                              ORDHDR
```

Figura 72 (Parte 1 de 7). Ejemplo de un Programa de Consulta de Pedidos

```
5763CB1 V3R0M5                  Listado Fuente AS/400 COBOL
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
   56 +000037           06 STATE            PIC X(2).                                    ORDHDR
      +000038*                            PROVINCIA                                    ORDHDR
   57 +000039           06 AMPAID           PIC S9(6)V9(2)   COMP-3.                     ORDHDR
      +000040*                            TOTAL SUMA PESETAS PAGADAS                   ORDHDR
       004100
   58  004200 FD  ORDER-DETAIL-FILE
   59  004300     LABEL RECORDS ARE STANDARD.
   60  004400 01  ORDER-DETAIL-RECORD.
   61  004500     COPY DDS-ORDDTL OF ORDDTLP.
      +000001*   FORMATO E-S:ORDDTL   DESDE ORDDTLP  DE BIBLIOTECA XMPLIB                 ORDDTL
      +000002*                          REGISTRO DETALLE PEDIDO                        ORDDTL
      +000003*DEFINICIONES CLAVE PARA FORMATO DE REGISTRO ORDDTL                        ORDDTL
      +000004*  NÚMERO              NOMBRE             RECUPERACIÓN  TIPO    SECALT    ORDDTL
      +000005*   0001   ORDERN                          ASCENDENTE  SIGNO      NO      ORDDTL
      +000006*   0002   LINNUM                          ASCENDENTE  SIGNO      NO      ORDDTL
   62 +000007       05  ORDDTL.                                                        ORDDTL
   63 +000008           06 CUST             PIC X(5).                                  ORDDTL
      +000009*                            NÚMERO CLIENTE                               ORDDTL
   64 +000010           06 ORDERN           PIC S9(5)        COMP-3.                   ORDDTL
      +000011*                            NÚMERO PEDIDO                                ORDDTL
   65 +000012           06 LINNUM           PIC S9(3)        COMP-3.                   ORDDTL
      +000013*                            NÚMERO LÍNEA DE LÍNEA DE PEDIDO              ORDDTL
   66 +000014           06 ITEM             PIC S9(5)        COMP-3.                   ORDDTL
      +000015*                            NÚMERO ÍTEM                                  ORDDTL
   67 +000016           06 QTYORD           PIC S9(3)        COMP-3.                   ORDDTL
      +000017*                            CANTIDAD PEDIDA                              ORDDTL
   68 +000018           06 DESCRP           PIC X(30).                                 ORDDTL
      +000019*                            DESCRIPCIÓN ÍTEM                             ORDDTL
   69 +000020           06 PRICE            PIC S9(4)V9(2)   COMP-3.                   ORDDTL
      +000021*                            PRECIO DE VENTA                              ORDDTL
   70 +000022           06 EXTENS           PIC S9(6)V9(2)   COMP-3.                   ORDDTL
      +000023*                            VALOR CALCULADO: CANTIDAD PEDIDA X PRECIO    ORDDTL
   71 +000024           06 WHSLOC           PIC X(3).                                  ORDDTL
      +000025*                            BIN NÚM.                                     ORDDTL
   72 +000026           06 ORDDAT           PIC S9(6)        COMP-3.                   ORDDTL
      +000027*                            SE ENTRÓ FECHA PEDIDO                        ORDDTL
   73 +000028           06 CUSTYP           PIC S9(1)        COMP-3.                   ORDDTL
      +000029*                            TIPO CLIENTE  1=GOB 2=PLA 3=NEG 4=FAB 5=OTR  ORDDTL
   74 +000030           06 STATE            PIC X(2).                                  ORDDTL
      +000031*                            PROVINCIA                                    ORDDTL
   75 +000032           06 ACTMTH           PIC S9(2)        COMP-3.                   ORDDTL
      +000033*                            MES CONTABLE DE VENTA                        ORDDTL
   76 +000034           06 ACTYR            PIC S9(2)        COMP-3.                   ORDDTL
      +000035*                            AÑO CONTABLE DE VENTA                        ORDDTL
   77  004600 66  ORDER-DETAIL-RECORD-KEY RENAMES ORDERN THRU LINNUM.
       004700
   78  004800 FD  CUSTOMER-MASTER-FILE
   79  004900     LABEL RECORDS ARE STANDARD.
   80  005000 01  CUSTOMER-MASTER-RECORD.
   81  005100     COPY DDS-CUSMST OF CUSMSTP.
      +000001*   FORMATO E-S:CUSMST   DESDE CUSMSTP  DE BIBLIOTECA XMPLIB                 CUSMST
      +000002*                          REGISTRO MAESTRO CLIENTES                      CUSMST
      +000003*DEFINICIONES CLAVE PARA FORMATO DE REGISTRO CUSMST                        CUSMST
      +000004*  NÚMERO              NOMBRE             RECUPERACIÓN  TIPO    SECALT    CUSMST
      +000005*   0001   CUST                           ASCENDENTE     AN      NO      CUSMST
   82 +000006       05  CUSMST.                                                        CUSMST
   83 +000007           06 CUST             PIC X(5).                                  CUSMST
      +000008*                            NÚMERO CLIENTE                               CUSMST
   84 +000009           06 NAME             PIC X(25).                                 CUSMST
      +000010*                            NOMBRE CLIENTE                               CUSMST
   85 +000011           06 ADDR             PIC X(20).                                 CUSMST
      +000012*                            DIRECCIÓN CLIENTE                            CUSMST
   86 +000013           06 CITY             PIC X(20).                                 CUSMST
      +000014*                            CIUDAD CLIENTE                               CUSMST
   87 +000015           06 STATE            PIC X(2).                                  CUSMST
      +000016*                            PROVINCIA                                    CUSMST
   88 +000017           06 ZIP              PIC S9(5)        COMP-3.                   CUSMST
      +000018*                            CÓDIGO POSTAL                                CUSMST
   89 +000019           06 SRHCOD           PIC X(6).                                  CUSMST
      +000020*                            CÓDIGO BÚSQUEDA NÚMERO CLIENTE               CUSMST
   90 +000021           06 CUSTYP           PIC S9(1)        COMP-3.                   CUSMST
      +000022*                            TIPO CLIENTE  1=GOB 2=PLA 3=NEG 4=FAB 5=OTR  CUSMST
   91 +000023           06 ARBAL            PIC S9(6)V9(2)   COMP-3.                   CUSMST
      +000024*                            SALDO CUENTAS PEND.                          CUSMST
   92 +000025           06 ORDBAL           PIC S9(6)V9(2)   COMP-3.                   CUSMST
```

Figura 72 (Parte 2 de 7). Ejemplo de un Programa de Consulta de Pedidos

```
5763CB1 V3R0M5                  Listado Fuente AS/400 COBOL
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
     +000026*                          CANTIDAD C/C EN ARCH. PEDIDOS                      CUSMST
   93 +000027           06 LSTAMT            PIC S9(6)V9(2)   COMP-3.                    CUSMST
     +000028*                          ULTIMA CANTIDAD PAGADA EN C/C                      CUSMST
   94 +000029           06 LSTDAT            PIC S9(6)        COMP-3.                    CUSMST
     +000030*                          ULTIMA FECHA PAGADA EN C/C                         CUSMST
   95 +000031           06 CRDLMT            PIC S9(6)V9(2)   COMP-3.                    CUSMST
     +000032*                          LÍMITE CRÉDITO CLIENTE                             CUSMST
   96 +000033           06 SLSYR             PIC S9(8)V9(2)   COMP-3.                    CUSMST
     +000034*                          VENTAS CLIENTE ESTE AÑO                            CUSMST
   97 +000035           06 SLSLYR            PIC S9(8)V9(2)   COMP-3.                    CUSMST
     +000036*                          VENTAS CLIENTE AÑO PASADO                          CUSMST
      005200
   98  005300 FD  EXISTING-ORDER-DISPLAY-FILE
   99  005400     LABEL RECORDS ARE OMITTED.
  100  005500 01  EXISTING-ORDER-DISPLAY-RECORD.
  101  005600     COPY DDS-ALL-FORMATS OF ORDINQD.
  102 +000001       05  ORDINQD-RECORD PIC X(171).                                       <-ALL-FMTS
     +000002*   FORMATO E-S:SUB1  DESDE ARCHIVO ORDINQD    DE BIBLIOTECA XMPLIB          <-ALL-FMTS
     +000003*                                                                           <-ALL-FMTS
  103 +000004       05  SUB1          REDEFINES ORDINQD-RECORD.                        <-ALL-FMTS
  104 +000005           06 ITEM              PIC S9(5).                                 <-ALL-FMTS
     +000006*                           NÚMERO ÍTEM                                    <-ALL-FMTS
  105 +000007           06 QTYORD            PIC S9(3).                                 <-ALL-FMTS
     +000008*                           CANTIDAD PEDIDA                                <-ALL-FMTS
  106 +000009           06 DESCRP            PIC X(30).                                 <-ALL-FMTS
     +000010*                           DESCRIPCIÓN ÍTEM                               <-ALL-FMTS
  107 +000011           06 PRICE             PIC S9(4)V9(2).                            <-ALL-FMTS
     +000012*                           PRECIO DE VENTA                                <-ALL-FMTS
  108 +000013           06 EXTENS            PIC S9(6)V9(2).                            <-ALL-FMTS
     +000014*                           VALOR CALCULADO: CANTIDAD PEDIDA X PRECIO      <-ALL-FMTS
     +000015*  FORMATO ENTRADA:SUBCTL1    DESDE ARCHIVO ORDINQD   DE BIBLIOTECA XMPLIB  <-ALL-FMTS
     +000016*                                                                           <-ALL-FMTS
  109 +000017       05  SUBCTL1-I     REDEFINES ORDINQD-RECORD.                        <-ALL-FMTS
  110 +000018           06 SUBCTL1-I-INDIC.                                            <-ALL-FMTS
  111 +000019                07 IN97         PIC 1  INDIC 97.                          <-ALL-FMTS
     +000020*                           CONTINUAR VISUALIZACIÓN                        <-ALL-FMTS
  112 +000021                07 IN98         PIC 1  INDIC 98.                          <-ALL-FMTS
     +000022*                           FIN DE PROGRAMA                                <-ALL-FMTS
  113 +000023                07 IN57         PIC 1  INDIC 57.                          <-ALL-FMTS
     +000024*                           VISUALIZAR SUBARCHIVO                          <-ALL-FMTS
  114 +000025                07 IN58         PIC 1  INDIC 58.                          <-ALL-FMTS
     +000026*             DESACTIVADO=VISUALIZAR SUBCTLQ ACTIVADO=BORRAR SUBARCHIVO     <-ALL-FMTS
  115 +000027                07 IN61         PIC 1  INDIC 61.                          <-ALL-FMTS
     +000028*                           NO ENCONTRADO NÚMERO PEDIDO                    <-ALL-FMTS
  116 +000029                07 IN47         PIC 1  INDIC 47.                          <-ALL-FMTS
     +000030*                           NO HAY LÍNEA PARA ESTE PEDIDO                  <-ALL-FMTS
  117 +000031                07 IN62         PIC 1  INDIC 62.                          <-ALL-FMTS
     +000032*                           NO HAY REGISTRO CLIENTE                        <-ALL-FMTS
  118 +000033           06 ORDERN            PIC S9(5).                                 <-ALL-FMTS
     +000034*                           NÚMERO PEDIDO                                  <-ALL-FMTS
     +000035*FORMATO SALIDA:SUBCTL1    DESDE ARCHIVO ORDINQD    DE BIBLIOTECA XMPLIB    <-ALL-FMTS
     +000036*                                                                           <-ALL-FMTS
  119 +000037       05  SUBCTL1-O     REDEFINES ORDINQD-RECORD.                        <-ALL-FMTS
  120 +000038           06 SUBCTL1-O-INDIC.                                            <-ALL-FMTS
  121 +000039                07 IN58         PIC 1  INDIC 58.                          <-ALL-FMTS
     +000040*             DESACTIVADO=VISUALIZAR SUBCTL1 ON=BORRAR SUBARCHIVO           <-ALL-FMTS
  122 +000041                07 IN57         PIC 1  INDIC 57.                          <-ALL-FMTS
     +000042*                           VISUALIZAR SUBARCHIVO                          <-ALL-FMTS
  123 +000043                07 IN45         PIC 1  INDIC 45.                          <-ALL-FMTS
  124 +000044                07 IN47         PIC 1  INDIC 47.                          <-ALL-FMTS
     +000045*                           NO HAY LÍNEA PARA ESTE PEDIDO                  <-ALL-FMTS
  125 +000046                07 IN61         PIC 1  INDIC 61.                          <-ALL-FMTS
     +000047*                           NO ENCONTRADO NÚMERO PEDIDO                    <-ALL-FMTS
  126 +000048                07 IN62         PIC 1  INDIC 62.                          <-ALL-FMTS
     +000049*                           NO HAY REGISTRO CLIENTE                        <-ALL-FMTS
  127 +000050           06 ORDERN            PIC S9(5).                                 <-ALL-FMTS
     +000051*                           NÚMERO PEDIDO                                  <-ALL-FMTS
  128 +000052           06 ORDDAT            PIC S9(6).                                 <-ALL-FMTS
     +000053*                           SE ENTRÓ FECHA PEDIDO                          <-ALL-FMTS
  129 +000054           06 CUST              PIC X(5).                                  <-ALL-FMTS
     +000055*                           NÚMERO CLIENTE                                 <-ALL-FMTS
  130 +000056           06 NAME              PIC X(25).                                 <-ALL-FMTS
     +000057*                           NOMBRE CLIENTE                                 <-ALL-FMTS
  131 +000058           06 ADDR              PIC X(20).                                 <-ALL-FMTS
     +000059*                           DIRECCIÓN CLIENTE                              <-ALL-FMTS
  132 +000060           06 CITY              PIC X(20).                                 <-ALL-FMTS
```

Figura 72 (Parte 3 de 7). Ejemplo de un Programa de Consulta de Pedidos

```
5763CB1 V3R0M5                  Listado Fuente AS/400 COBOL
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
     +000061*                          CIUDAD CLIENTE                                    <-ALL-FMTS
  133 +000062           06 STATE             PIC X(2).                                   <-ALL-FMTS
     +000063*                          PROVINCIA CLIENTE                                 <-ALL-FMTS
  134 +000064           06 ZIP               PIC S9(5).                                  <-ALL-FMTS
     +000065*                          CÓDIGO POSTAL                                     <-ALL-FMTS
  135 +000066           06 ORDAMT             PIC S9(6)V9(2).                             <-ALL-FMTS
     +000067*                          IMPORTE TOTAL DEL PEDIDO                          <-ALL-FMTS
  136 +000068           06 STSORD            PIC X(12).                                  <-ALL-FMTS
  137 +000069           06 STSOPN            PIC X(12).                                  <-ALL-FMTS
  138 +000070           06 CUSORD            PIC X(15).                                  <-ALL-FMTS
     +000071*                          NÚMERO PEDIDO COMPRA CLIENTE                      <-ALL-FMTS
  139 +000072           06 SHPVIA            PIC X(15).                                  <-ALL-FMTS
     +000073*                          INSTRUCCIONES DE ENVÍO                            <-ALL-FMTS
  140 +000074           06 PRTDAT            PIC S9(6).                                  <-ALL-FMTS
     +000075*                          FECHA IMPRESIÓN PEDIDO                            <-ALL-FMTS
  141 +000076           06 INVNUM            PIC S9(5).                                  <-ALL-FMTS
     +000077*                          NÚMERO FACTURA                                    <-ALL-FMTS
  142 +000078           06 ACTMTH            PIC S9(2).                                  <-ALL-FMTS
     +000079*                          MES CONTABLE DE VENTA                             <-ALL-FMTS
  143 +000080           06 ACTYR             PIC S9(2).                                  <-ALL-FMTS
     +000081*                          AÑO CONTABLE DE VENTA                             <-ALL-FMTS
      005700
  144  005800 WORKING-STORAGE SECTION.
  145  005900 01  EXISTING-ORDER-DISPLAY-KEY.
  146  006000    05  SUBFILE-RECORD-NUMBER              PIC 9(2)
  147  006100                                        VALUE ZERO.
      006200
  148  006300 01  ORDER-STATUS-COMMENT-VALUES.
  149  006400    05  FILLER                             PIC X(12)
  150  006500                                        VALUE "1-IN PROCESS".
  151  006600    05  FILLER                             PIC X(12)
  152  006700                                        VALUE "2-CONTINUED ".
  153  006800    05  FILLER                             PIC X(12)
  154  006900                                        VALUE "3-CREDIT CHK".
  155  007000    05  FILLER                             PIC X(12)
  156  007100                                        VALUE "4-READY PRT ".
  157  007200    05  FILLER                             PIC X(12)
  158  007300                                        VALUE "5-PRINTED   ".
  159  007400    05  FILLER                             PIC X(12)
  160  007500                                        VALUE "6-PICKED    ".
  161  007600    05  FILLER                             PIC X(12)
  162  007700                                        VALUE "7-INVOICED  ".
  163  007800    05  FILLER                             PIC X(12)
  164  007900                                        VALUE "8-INVALID   ".
  165  008000    05  FILLER                             PIC X(12)
  166  008100                                        VALUE "9-CANCELED  ".
      008200
  167  008300 01  ORDER-STATUS-COMMENT-TABLE
  168  008400         REDEFINES ORDER-STATUS-COMMENT-VALUES.
  169  008500    05  ORDER-STATUS OCCURS 9 TIMES.
  170  008600       10 ORDER-STATUS-COMMENT              PIC X(12).
      008700
  171  008800 01  OPEN-STATUS-COMMENT-VALUES.
  172  008900    05  FILLER                             PIC X(12)
  173  009000                                        VALUE "1-OPEN      ".
  174  009100    05  FILLER                             PIC X(12)
  175  009200                                        VALUE "2-CLOSED    ".
  176  009300    05  FILLER                             PIC X(12)
  177  009400                                        VALUE "3-CANCELED  ".
      009500
  178  009600 01  OPEN-STATUS-COMMENT-TABLE
  179  009700         REDEFINES OPEN-STATUS-COMMENT-VALUES.
  180  009800    05  OPEN-STATUS OCCURS 3 TIMES.
  181  009900       10  OPEN-STATUS-COMMENT              PIC X(12).
      010000
  182  010100 01  ERRHDL-PARAMETERS.
  183  010200    05  STATUS-CODE-ONE                    PIC X(2).
  184  010300       88  SUBFILE-IS-FULL                VALUE "9M".
      010400
  185  010500 01  ERRPGM-PARAMETERS.
  186  010600    05  DISPLAY-PARAMETER                  PIC X(8)
  187  010700                                        VALUE "ORD220D ".
  188  010800    05  DUMMY-ONE                          PIC X(6)
  189  010900                                        VALUE SPACES.
  190  011000    05  DUMMY-TWO                          PIC X(8)
  191  011100                                        VALUE SPACES.
```

Figura 72 (Parte 4 de 7). Ejemplo de un Programa de Consulta de Pedidos

```
5763CB1 V3R0M5                  Listado Fuente AS/400 COBOL
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
  192  011200    05  STATUS-CODE-TWO.
  193  011300       10  PRIMARY                          PIC X(1).
  194  011400       10  SECONDARY                        PIC X(1).
  195  011500       10  FILLER                           PIC X(5)
  196  011600                                        VALUE SPACES.
       011700
  197  011800 01  SWITCH-AREA.
  198  011900    05  SW01                                PIC 1.
  199  012000       88  NO-MORE-DETAIL-LINE-ITEMS      VALUE B"1".
  200  012100       88  MORE-DETAIL-LINE-ITEMS-EXIST   VALUE B"0".
  201  012200    05  SW02                                PIC 1.
  202  012300       88  WRITE-DISPLAY                  VALUE B"1".
  203  012400       88  READ-DISPLAY                   VALUE B"0".
  204  012500    05  SW03                                PIC 1.
  205  012600       88  SUBCTL1-FORMAT                 VALUE B"1".
  206  012700       88  NOT-SUBCTL1-FORMAT             VALUE B"0".
  207  012800    05  SW04                                PIC 1.
  208  012900       88  SUB1-FORMAT                    VALUE B"1".
  209  013000       88  NOT-SUB1-FORMAT                VALUE B"0".
       013100
  210  013200 01  INDICATOR-AREA.
  211  013300    05  IN98                                PIC 1 INDIC 98.
  212  013400       88  END-OF-EXISTING-ORDER-INQUIRY  VALUE B"1".
  213  013500    05  IN97                                PIC 1 INDIC 97.
  214  013600       88  CONTINUE-DETAIL-LINES-DISPLAY  VALUE B"1".
  215  013700    05  IN62                                PIC 1 INDIC 62.
  216  013800       88  CUSTOMER-NOT-FOUND              VALUE B"1".
  217  013900       88  CUSTOMER-EXIST                  VALUE B"0".
  218  014000    05  IN61                                PIC 1 INDIC 61.
  219  014100       88  ORDER-NOT-FOUND                 VALUE B"1".
  220  014200       88  ORDER-EXIST                     VALUE B"0".
  221  014300    05  IN58                                PIC 1 INDIC 58.
  222  014400       88  CLEAR-SUBFILE                   VALUE B"1".
  223  014500       88  DISPLAY-SUBFILE-CONTROL         VALUE B"0".
  224  014600    05  IN57                                PIC 1 INDIC 57.
  225  014700       88  DISPLAY-SUBFILE                 VALUE B"1".
  226  014800    05  IN47                                PIC 1 INDIC 47.
  227  014900       88  NO-DETAIL-LINES-FOR-ORDER       VALUE B"1".
  228  015000       88  DETAIL-LINES-FOR-ORDER-EXIST    VALUE B"0".
  229  015100    05  IN45                                PIC 1 INDIC 45.
  230  015200       88  END-OF-ORDER                    VALUE B"1".
       015300
  231  015400 PROCEDURE DIVISION.
       015500
       015600 DECLARATIVES.
       015700 TRANSACTION-ERROR SECTION.
       015800     USE AFTER STANDARD ERROR PROCEDURE
       015900         EXISTING-ORDER-DISPLAY-FILE.
       016000 WORK-STATION-ERROR-HANDLER.
  232  016100     IF SUBFILE-IS-FULL THEN
       016200         NEXT SENTENCE
       016300     ELSE
  233  016400         DISPLAY "WORK-STATION ERROR"   STATUS-CODE-ONE.
       016500 END DECLARATIVES.
       016600
       016700 INQUIRY-INTO-EXISTING-ORDER SECTION.
       016800 MAINLINE-ROUTINE.
  234  016900     PERFORM SET-UP-ROUTINE.
  235  017000     PERFORM EXISTING-ORDER-INQUIRY
       017100         UNTIL END-OF-EXISTING-ORDER-INQUIRY.
  236  017200     PERFORM CLEAN-UP-ROUTINE.
       017300
       017400 SET-UP-ROUTINE.
  237  017500     OPEN INPUT  ORDER-HEADER-FILE
       017600                 ORDER-DETAIL-FILE
       017700                 CUSTOMER-MASTER-FILE
       017800          I-O    EXISTING-ORDER-DISPLAY-FILE.
  238  017900     MOVE SPACES TO   CUST   OF SUBCTL1-O
       018000                      NAME   OF SUBCTL1-O
       018100                      ADDR   OF SUBCTL1-O
       018200                      CITY   OF SUBCTL1-O
       018300                      STATE  OF SUBCTL1-O
       018400                      STSORD OF SUBCTL1-O
       018500                      STSOPN OF SUBCTL1-O
       018600                      CUSORD OF SUBCTL1-O.
```

Figura 72 (Parte 5 de 7). Ejemplo de un Programa de Consulta de Pedidos

```
5763CB1 V3R0M5                   Listado Fuente AS/400 COBOL
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
  239  018700     MOVE ZEROS TO    ORDERN OF SUBCTL1-O
       018800                      ORDDAT OF SUBCTL1-O
       018900                      ZIP    OF SUBCTL1-O
       019000                      ORDAMT OF SUBCTL1-O
       019100                      PRTDAT OF SUBCTL1-O
       019200                      INVNUM OF SUBCTL1-O
       019300                      ACTMTH OF SUBCTL1-O
       019400                      ACTYR  OF SUBCTL1-O.
  240  019500     MOVE ALL B'0' TO INDICATOR-AREA.
  241  019600     SET READ-DISPLAY
       019700         NOT-SUBCTL1-FORMAT
       019800         NOT-SUB1-FORMAT TO TRUE.
  242  019900     MOVE CORR INDICATOR-AREA TO SUBCTL1-O-INDIC.
  243  020000     WRITE EXISTING-ORDER-DISPLAY-RECORD FORMAT IS "SUBCTL1".
  244  020100     READ EXISTING-ORDER-DISPLAY-FILE RECORD.
  245  020200     MOVE CORR SUBCTL1-I-INDIC TO INDICATOR-AREA.
       020300
       020400 EXISTING-ORDER-INQUIRY.
  246  020500     IF CONTINUE-DETAIL-LINES-DISPLAY THEN
  247  020600         PERFORM READ-NEXT-ORDER-DETAIL-RECORD
  248  020700         IF MORE-DETAIL-LINE-ITEMS-EXIST THEN
  249  020800             IF ORDERN OF ORDER-DETAIL-RECORD IS NOT EQUAL TO
       020900                ORDERN OF ORDER-HEADER-RECORD THEN
  250  021000                 SET DISPLAY-SUBFILE TO TRUE
  251  021100                 SET NO-DETAIL-LINES-FOR-ORDER TO TRUE
       021200             ELSE
  252  021300                 PERFORM SUBFILE-SET-UP
       021400         ELSE
  253  021500             SET DISPLAY-SUBFILE TO TRUE
  254  021600             SET NO-DETAIL-LINES-FOR-ORDER TO TRUE
       021700     ELSE
  255  021800         PERFORM ORDER-NUMBER-VALIDATION.
  256  021900     MOVE CORR INDICATOR-AREA TO SUBCTL1-O-INDIC.
  257  022000     SET WRITE-DISPLAY TO TRUE.
  258  022100     SET SUBCTL1-FORMAT TO TRUE.
  259  022200     WRITE EXISTING-ORDER-DISPLAY-RECORD FORMAT IS "SUBCTL1".
  260  022300     READ EXISTING-ORDER-DISPLAY-FILE RECORD.
  261  022400     MOVE CORR SUBCTL1-I-INDIC TO INDICATOR-AREA.
       022500 ORDER-NUMBER-VALIDATION.
  262  022600     PERFORM READ-ORDER-HEADER-FILE.
  263  022700     IF ORDER-EXIST THEN
  264  022800         PERFORM READ-CUSTOMER-MASTER-FILE
  265  022900         IF CUSTOMER-EXIST THEN
  266  023000             PERFORM READ-FIRST-ORDER-DETAIL-RECORD
  267  023100             IF DETAIL-LINES-FOR-ORDER-EXIST THEN
  268  023200                 PERFORM SUBFILE-SET-UP
       023300             ELSE
       023400                 NEXT SENTENCE
       023500         ELSE
       023600             NEXT SENTENCE
       023700     ELSE
       023800         NEXT SENTENCE.
       023900 READ-ORDER-HEADER-FILE.
  269  024000     MOVE ORDERN OF SUBCTL1-I OF EXISTING-ORDER-DISPLAY-RECORD
       024100         TO ORDERN OF ORDER-HEADER-RECORD.
  270  024200     READ ORDER-HEADER-FILE
  271  024300         INVALID KEY SET ORDER-NOT-FOUND TO TRUE.
       024400 READ-CUSTOMER-MASTER-FILE.
  272  024500     MOVE CUST OF ORDER-HEADER-RECORD
       024600         TO CUST OF CUSTOMER-MASTER-RECORD.
  273  024700     READ CUSTOMER-MASTER-FILE
  274  024800         INVALID KEY SET CUSTOMER-NOT-FOUND TO TRUE.
       024900 READ-FIRST-ORDER-DETAIL-RECORD.
  275  025000     MOVE ORDERN OF ORDER-HEADER-RECORD
       025100         TO ORDERN OF ORDER-DETAIL-RECORD.
  276  025200     MOVE 1 TO LINNUM OF ORDER-DETAIL-RECORD.
  277  025300     READ ORDER-DETAIL-FILE
  278  025400         INVALID KEY SET NO-DETAIL-LINES-FOR-ORDER TO TRUE.
       025500 SUBFILE-SET-UP.
  279  025600     SET CLEAR-SUBFILE TO TRUE.
  280  025700     MOVE CORR INDICATOR-AREA TO SUBCTL1-O-INDIC.
  281  025800     SET WRITE-DISPLAY TO TRUE.
  282  025900     SET SUBCTL1-FORMAT TO TRUE.
  283  026000     WRITE EXISTING-ORDER-DISPLAY-RECORD FORMAT IS "SUBCTL1".
  284  026100     SET DISPLAY-SUBFILE-CONTROL TO TRUE.
```

Figura 72 (Parte 6 de 7). Ejemplo de un Programa de Consulta de Pedidos

```
5763CB1 V3R0M5                  Listado Fuente AS/400 COBOL
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
  285  026200     PERFORM BUILD-DISPLAY-SUBFILE
       026300         UNTIL NO-MORE-DETAIL-LINE-ITEMS
       026400           OR SUBFILE-IS-FULL.
  286  026500     MOVE CORR ORDHDR OF ORDER-HEADER-RECORD
       026600         TO SUBCTL1-O OF EXISTING-ORDER-DISPLAY-RECORD.
  287  026700     MOVE CORR CUSMST OF CUSTOMER-MASTER-RECORD
       026800         TO SUBCTL1-O OF EXISTING-ORDER-DISPLAY-RECORD.
  288  026900     MOVE ORDER-STATUS(ORDSTS) TO STSORD.
  289  027000     MOVE OPEN-STATUS(OPNSTS) TO STSOPN.
  290  027100     SET MORE-DETAIL-LINE-ITEMS-EXIST TO TRUE.
  291  027200     MOVE ZEROS TO SUBFILE-RECORD-NUMBER.
       027300 BUILD-DISPLAY-SUBFILE.
  292  027400     MOVE CORR ORDDTL OF ORDER-DETAIL-RECORD
       027500         TO SUB1 OF EXISTING-ORDER-DISPLAY-RECORD.
  293  027600     SET WRITE-DISPLAY TO TRUE.
  294  027700     SET SUB1-FORMAT TO TRUE.
  295  027800     ADD 1 TO SUBFILE-RECORD-NUMBER.
  296  027900     WRITE SUBFILE EXISTING-ORDER-DISPLAY-RECORD FORMAT IS "SUB1".
  297  028000     IF SUBFILE-IS-FULL THEN
  298  028100         SET DISPLAY-SUBFILE TO TRUE
       028200     ELSE
  299  028300         PERFORM READ-NEXT-ORDER-DETAIL-RECORD
  300  028400         IF NO-MORE-DETAIL-LINE-ITEMS THEN
       028500             NEXT SENTENCE
       028600         ELSE
  301  028700             IF ORDERN OF ORDER-DETAIL-RECORD IS NOT EQUAL TO
       028800                ORDERN OF ORDER-HEADER-RECORD THEN
  302  028900                 SET DISPLAY-SUBFILE TO TRUE
  303  029000                 SET NO-MORE-DETAIL-LINE-ITEMS TO TRUE
       029100             ELSE
       029200                 NEXT SENTENCE.
       029300 READ-NEXT-ORDER-DETAIL-RECORD.
  304  029400     READ ORDER-DETAIL-FILE NEXT RECORD
  305  029500         AT END SET DISPLAY-SUBFILE TO TRUE
  306  029600                SET NO-MORE-DETAIL-LINE-ITEMS TO TRUE.
       029700 CLEAN-UP-ROUTINE.
  307  029800      CLOSE     ORDER-HEADER-FILE
       029900                ORDER-DETAIL-FILE
       030000                CUSTOMER-MASTER-FILE
       030100                EXISTING-ORDER-DISPLAY-FILE.
  308  030200      STOP RUN.
                        * * * * *  F I N  D E  F U E N T E  * * * * *
```

Figura 72 (Parte 7 de 7). Ejemplo de un Programa de Consulta de Pedidos

Esta es la pantalla inicial de solicitud de entrada de pedidos grabada para la estación de trabajo:

```
Entrada Pedido Existente                Total  000000000
                                        Estado
Pedido 12400                            Abierto
Fecha  000000                           Pedido cliente
Cli  #                                  Medio envío
                                00000   Fecha Impreso 000000
                              Factura 00000                  Mes 00  Año 00
Item  Cant  Descripción Item               Precio  Cálculo
```

Esta pantalla aparece si existen registros de detalle del pedido para el cliente cuyo número de pedido se ha entrado en la primera pantalla:

```
Entrada Pedido Existente                Total  007426656
                                        Estado   7-INVOICED
Pedido  17924   ABC HARDWARE LTD.       Abierto  2-CLOSED
Fecha  110587   123 ANYWHERE AVE.       Pedido cliente TESTCS17933001I
Cli  # 11200    TORONTO                 Medio envío    TRUCKCO
                ONT            M4K 0A0  Fecha impreso  082788
                           Factura 17924                    Mes 12  Año 88
Item  Cant  Descripción Item              Precio  Cálculo
33001  003  TORQUE WRENCH 75LB 14 INCH    009115       273.45
33100  001  TORQUE WRENCH W/GAUGE 200 LB  015777       650.95
44529  004  WOOD CHISEL - 3 1/4           006840        56.87
44958  002  POWER DRILL 1/2 REV           008200       797.50
46102  001  WROUGHT IRON RAILING 4FTX6FT  007930       237.75
46201  001  WROUGHT IRON HAND RAIL 6FT    007178        77.35
47902  002  ESCUTCHEON BRASS 15X4 INCHES  044488       213.00
```

Esta pantalla aparece si el archivo ORDHDRP no contiene ningún registro para el número de pedido entrado en la primera pantalla:

```
Entrada Pedido Existente                Total  000000000
                                        Estado
Pedido 12400                            Abierto
Fecha  000000                           Pedido cliente
Cli  #                                  Medio envío
                             00000      Fecha Impreso 000000
                           Factura 00000                    Mes 00  Año 00
Item  Cant  Descripción Item              Precio  Cálculo









No encontrado número de pedido
```

Un Programa de Actualización de Pagos

La Figura 74 en la página 231 muestra un ejemplo de un programa de actualización de pagos, PAYUPDT. Para las DDS relacionadas, consulte la Figura 73 en la página 228. Para los ejemplos de pantallas asociadas, consulte la página 238. Para las DDS correspondientes al archivo maestro de clientes, CUSMSTP, consulte la Figura 69 en la página 212.

En este ejemplo, se registran los pagos de los clientes. Se solicita al empleado que entre uno o más números de clientes y la cantidad de dinero a abonar a cada cuenta de los clientes. El programa comprueba el número de cliente y acepta incondicionalmente cualquier pago para un cliente dado que tenga facturas pendientes de pago. Si la cantidad pagada por un cliente resulta excesiva, el empleado tiene la opción de aceptar o rechazar el pago. Si no existe ningún registro del cliente, se emite un mensaje de error. Los pagos pueden entrarse hasta que el empleado finalice el programa pulsando F12.

IBM International Business Machines

AS/400 ESPECIFICACIONES DE DESCRIPCION DE DATOS

GX21-9801-0 UM/050*
Impreso en España
*El número de hojas por área de relleno puede variar ligeramente.

Archivo		Instrucciones de Grabación	Signo							
Programador	Fecha		Tecla							

Descripción	Página de

A

Número de Secuencia	Tipo de Formulario	Condicionamiento: And/Or/Coment. (A/O/*)	Condicionamiento: No (N)	Condicionamiento: Nombre Condición: Indicador	Condicionamiento: Nombre Condición: No (N)	Condicionamiento: Nombre Condición: Indicador	Condicionamiento: Nombre Condición: No (N)	Condicionamiento: Nombre Condición: Indicador	Tipo Nombre o Espac. (B/R/H/J/K/S/O)	Reservado	Nombre	Referencia (R)	Longitud	Tipo Datos/Desplazamiento Teclado	Posiciones Decimales	Utilización (B/O/I/B/H/N/N/P)	Ubicación: Línea	Ubicación: Pos	Funciones
	A	*	*	L	O	G	I	C	O					A	R	C	H	I	V	O D E P E D I D O S D E O R D H D R
	A								R		O R D H D R								P F I L E (O R D H D R P)
	A	*																	
	A										C U S T								
	A										I N V N U M								
	A										O R D E R N								
	A										O R D D A T								
	A										C U S O R D								
	A										S H P V I A								
	A										O R D S T S								
	A										O P R N A M								
	A										O R D A M T								
	A										C U S T Y P								
	A										P R T D A T								
	A										O P N S T S								
	A										T O T L I N								
	A										A C T M T H								
	A										A C T Y R								
	A										S T A T E								
	A										A M P A I D								
	A								K		C U S T								
	A								K		I N V N U M								
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		

Figura 73 (Parte 1 de 3). Ejemplo de una Especificación de Descripción de Datos para un Programa de Actualización de Pagos

IBM International Business Machines

AS/400 ESPECIFICACIONES DE DESCRIPCION DE DATOS

GX21-9801-0 UM/050* Impreso en España

*El número de hojas por área de relleno puede variar ligeramente.

Archivo		Instrucciones de Grabación	Signo								Descripción	Página de
Programador	Fecha		Tecla									

A

Número de Secuencia	Tipo de Formulario	And/Or/Coment. (A/O/*)	No (N)	Condicionamiento: Nombre Condición: Indicador	No (N)	Indicador	No (N)	Indicador	Tipo Nombre o Espac (B/R/H/J/K/S/O)	Reservado	Nombre	Referencia (R)	Longitud	Tipo Dato/Desplazamiento Teclado	Posiciones Decimales	Utilización (b/O/I/B/M/N/P)	Ubicación: Línea	Pos	Funciones
	A	*									DDS PARA ARCHIVO DISPOSITIVO PANTALLA/PAYUPDTD								
	A	*									ACTUALIZACION DE PAGO INTERACTIVO DE CUENTAS POR COBRAR								
	A	*																	
	A								R		SUBFILE1								SFL
	A																		TEXT('SUBARCH PARA PAGO DE CLIENTE')
	A	*																	
	A										ACPPMT		4	A		I	5	4	TEXT('ACEPTAR PAGO')
	A																		VALUES('*YES' '*NO')
	A			51															DSPATR(RI MDT)
	A		N	51															DSPATR(ND PR)
	A	*																	
	A										CUST		5			B	5	15	TEXT('NUMERO DE CLIENTE')
	A			52															DSPATR(RI)
	A			53															DSPATR(ND)
	A			54															DSPATR(PR)
	A	*																	
	A										AMPAID		8	02		B	5	24	TEXT('IMPORTE PAGADO')
	A																		CHECK(FE)
	A																		AUTO(RAB)
	A																		CMP(GT 0)
	A			52															DSPATR(RI)
	A			53															DSPATR(ND)
	A			54															DSPATR(PR)
	A	*																	
	A										ECPMSG		31	A		O	5	37	TEXT('MENSAJE DE EXCEPCION')
	A			52															DSPATR(RI)
	A			53															DSPATR(ND)
	A			54															DSPATR(PR)
	A	*																	
	A										OVRPMT		8	Y	2	O	5	70	TEXT('PAGO EXCESIVO')
	A																		EDTCDE(1)
	A			55															DSPATR(BL)
	A		N	56															DSPATR(ND)
	A	*																	
	A										STSCDE		1	A		H			TEXT('CODIGO ESTADO')

Figura 73 (Parte 2 de 3). Ejemplo de una Especificación de Descripción de Datos para un Programa de Actualización de Pagos

IBM International Business Machines

AS/400 ESPECIFICACIONES DE DESCRIPCION DE DATOS

GX21-9891-0 UM/050* Impreso en España

*El número de hojas por área de relleno puede variar ligeramente.

Archivo		Instrucciones de Grabación	Signo								Descripción	Página de
Programador	Fecha		Tecla									

A

Número de Secuencia	Tipo de Formulario	Condicionamiento: And/Or/Coment. (A/O/*)	Condicionamiento: No (N)	Condicionamiento: Nombre Condición: Indicador	Condicionamiento: Nombre Condición: No (N)	Condicionamiento: Nombre Condición: Indicador	Condicionamiento: Nombre Condición: No (N)	Condicionamiento: Nombre Condición: Indicador	Tipo Nombre o Espac (B/R/H/J/K/E/O)	Reservado	Nombre	Referencia (R)	Longitud	Tipo Datos/Desplazamiento Teclado	Posiciones Decimales	Utilización (b/O/I/B/H/M/N/P)	Ubicación: Línea	Ubicación: Pos	Funciones
	A	*							R		CONTROL1								TEXT('SUBARCHIVO DE CONTROL')
	A																		SFLCTL(SUBARCHIVO1)
	A																		SFLSIZ(17)
	A																		SFLPAG(17)
	A			61															SFLCLR
	A			62															SFLDSP
	A			62															SFLDSPCTL
	A																		OVERLAY
	A																		LOCK
	A	*																	
	A																		HELP(99 'TECLA DE AYUDA')
	A																		CA12(98 'FIN DE ACTUALIZACION PAGO')
	A																		CA11(97 'IGNORAR ENTRADA')
	A	*																	
	A			99															SFLMSG('F11-IGNORAR ENTR NO VALIDA+
	A																		F12 - FIN DE ACTUA+
	A																		LIZACION DE PAGO')
	A	*																	
	A																1	2	'SOLICITUD ACTUALIZ PAGOS DE CLIENTE'
	A																1	65	'FECHA'
	A																1	78	FECHA EDTCDE(Y)
	A			63													3	2	'ACEPTAR'
	A			63													4	2	'PAGO'
	A																3	14	'CLIENTE'
	A																3	26	'PAGO'
	A			64													3	37	'MENSAJE DE EXCEPCION'
	A	*																	
	A								R		MESSAGE1								TEXT('REGISTRO DE MENSAJE')
	A																		OVERLAY
	A																		LOCK
	A	*																	
	A			71													24	2	'ACEPTAR VALORES PAGO: (*NO *SI)'
	A																		DSPATR(RI)
	A																		
	A																		

Figura 73 (Parte 3 de 3). Ejemplo de una Especificación de Descripción de Datos para un Programa de Actualización de Pagos

```
5763CB1 V3R0M5                  Listado Fuente AS/400 COBOL
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
    1  000100 IDENTIFICATION DIVISION.                                                                  02/01/94
    2  000200 PROGRAM-ID.  PAYUPDT.                                                                     03/22/94
    3  000300 ENVIRONMENT DIVISION.                                                                     02/01/94
    4  000400 CONFIGURATION SECTION.                                                                    02/01/94
    5  000500 SOURCE-COMPUTER. IBM-AS400.                                                               02/02/94
    6  000600 OBJECT-COMPUTER. IBM-AS400.                                                               02/02/94
    7  000700 INPUT-OUTPUT SECTION.                                                                     02/01/94
    8  000800 FILE-CONTROL.                                                                             02/01/94
    9  000900     SELECT CUSTOMER-INVOICE-FILE                                                          02/01/94
   10  001000         ASSIGN TO DATABASE-ORDHDRL                                                        02/01/94
   11  001100         ORGANIZATION IS INDEXED                                                           02/01/94
   12  001200         ACCESS MODE IS SEQUENTIAL                                                         02/01/94
   13  001300         RECORD KEY IS COMP-KEY                                                            02/01/94
   14  001400         FILE STATUS IS STATUS-CODE-ONE.                                                   02/01/94
   15  001500     SELECT CUSTOMER-MASTER-FILE                                                           02/01/94
   16  001600         ASSIGN TO DATABASE-CUSMSTP                                                        02/01/94
   17  001700         ORGANIZATION IS INDEXED                                                           02/01/94
   18  001800         ACCESS IS RANDOM                                                                  02/01/94
   19  001900         RECORD KEY IS CUST OF CUSTOMER-MASTER-RECORD.                                     02/01/94
   20  002000     SELECT PAYMENT-UPDATE-DISPLAY-FILE                                                    02/01/94
   21  002100         ASSIGN TO WORKSTATION-PAYUPDTD                                                    03/22/94
   22  002200         ORGANIZATION IS TRANSACTION                                                       02/01/94
   23  002300         ACCESS IS DYNAMIC                                                                 02/01/94
   24  002400         RELATIVE KEY IS REL-NUMBER                                                        02/01/94
   25  002500         FILE STATUS IS STATUS-CODE-ONE                                                    02/01/94
   26  002600         CONTROL-AREA IS WS-CONTROL.                                                       02/01/94
       002700                                                                                           02/01/94
   27  002800 DATA DIVISION.                                                                            02/01/94
   28  002900 FILE SECTION.                                                                             02/01/94
   29  003000 FD  CUSTOMER-INVOICE-FILE                                                                 02/01/94
   30  003100     LABEL RECORDS ARE STANDARD.                                                           02/01/94
   31  003200 01  CUSTOMER-INVOICE-RECORD.                                                             02/01/94
   32  003300     COPY DDS-ORDHDR OF ORDHDRL.                                                           03/14/94
      +000001*   FORMATO E-S:ORDHDR     DESDE ARCHIVO ORDHDRL    DE BIBLIOTECA XMPLIB      ORDHDR
      +000002*                                                                                        ORDHDR
      +000003*DEFINICIONES CLAVE PARA FORMATO DE REGISTRO ORDHDR                                       ORDHDR
      +000004*  NÚMERO              NOMBRE             RECUPERACIÓN  TIPO    SECALT       ORDHDR
      +000005*   0001   CUST                           ASCENDENTE     AN      NO         ORDHDR
      +000006*   0002   INVNUM                         ASCENDENTE   SIGNO     NO         ORDHDR
   33 +000007       05  ORDHDR.                                                                        ORDHDR
   34 +000008           06 CUST             PIC X(5).                                                 ORDHDR
      +000009*                            NÚMERO CLIENTE                                             ORDHDR
   35 +000010           06 INVNUM           PIC S9(5)        COMP-3.                                  ORDHDR
      +000011*                            NÚMERO FACTURA                                             ORDHDR
   36 +000012           06 ORDERN           PIC S9(5)        COMP-3.                                  ORDHDR
      +000013*                            NÚMERO PEDIDO                                              ORDHDR
   37 +000014           06 ORDDAT           PIC S9(6)        COMP-3.                                  ORDHDR
      +000015*                            SE ENTRÓ FECHA PEDIDO                                      ORDHDR
   38 +000016           06 CUSORD           PIC X(15).                                                ORDHDR
      +000017*                            NÚMERO PEDIDO COMPRA CLIENTE                               ORDHDR
   39 +000018           06 SHPVIA           PIC X(15).                                                ORDHDR
      +000019*                            INSTRUCCIONES ENVÍO                                        ORDHDR
   40 +000020           06 ORDSTS           PIC S9(1)        COMP-3.                                  ORDHDR
      +000021*                            ESTADO PEDIDO 1PEND 2CONT  3COM 4LIS 5IMP 6ENP             ORDHDR
   41 +000022           06 OPRNAM           PIC X(10).                                                ORDHDR
      +000023*                            NOMBRE OPERADOR QUE ENTRÓ EL PEDIDO                        ORDHDR
   42 +000024           06 ORDAMT           PIC S9(6)V9(2)   COMP-3.                                 ORDHDR
      +000025*                            TOTAL SUMA PESETAS DEL PEDIDO                              ORDHDR
   43 +000026           06 CUSTYP           PIC S9(1)        COMP-3.                                  ORDHDR
      +000027*                            TIPO CLIENTE  1=GOB 2=PLA 3=NEG 4=FAB 5=OTR                ORDHDR
   44 +000028           06 PRTDAT           PIC S9(6)        COMP-3.                                  ORDHDR
      +000029*                            FECHA IMPRESIÓN PEDIDO                                     ORDHDR
   45 +000030           06 OPNSTS           PIC S9(1)        COMP-3.                                  ORDHDR
      +000031*                            EST.PEDIDO ABIERTO 1=ABIERTO 2= CERRADO 3=CANCELAD.ORDHDR
   46 +000032           06 TOTLIN           PIC S9(3)        COMP-3.                                  ORDHDR
      +000033*                            TOTAL LÍNEAS ÍTEMS EN PEDIDO                               ORDHDR
   47 +000034           06 ACTMTH           PIC S9(2)        COMP-3.                                  ORDHDR
      +000035*                            MES CONTABLE DE VENTA                                      ORDHDR
   48 +000036           06 ACTYR            PIC S9(2)        COMP-3.                                  ORDHDR
      +000037*                            AÑO CONTABLE DE VENTA                                      ORDHDR
   49 +000038           06 STATE            PIC X(2).                                                 ORDHDR
      +000039*                            PROVINCIA                                                  ORDHDR
   50 +000040           06 AMPAID           PIC S9(6)V9(2)   COMP-3.                                 ORDHDR
      +000041*                            TOTAL SUMA PESETAS PAGADAS                                 ORDHDR
   51  003400 66  COMP-KEY RENAMES CUST THRU INVNUM.
       003500
```

Figura 74 (Parte 1 de 8). Listado Fuente de un Ejemplo de Programa de Actualización de Pagos

```
5763CB1 V3R0M5                  Listado Fuente AS/400 COBOL
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
   52  003600 FD  CUSTOMER-MASTER-FILE
   53  003700     LABEL RECORDS ARE STANDARD.
   54  003800 01  CUSTOMER-MASTER-RECORD.
   55  003900     COPY DDS-CUSMST OF CUSMSTP.
      +000001*   FORMATO E-S:CUSMST   DESDE CUSMSTP  DE BIBLIOTECA XMPLIB                 CUSMST
      +000002*                         REGISTRO MAESTRO CLIENTES                         CUSMST
      +000003*DEFINICIONES CLAVE PARA FORMATO DE REGISTRO CUSMST                         CUSMST
      +000004*  NÚMERO              NOMBRE             RECUPERACIÓN  TIPO    SECALT      CUSMST
      +000005*   0001   CUST                           ASCENDENTE     AN      NO        CUSMST
   56 +000006       05  CUSMST.                                                         CUSMST
   57 +000007           06 CUST             PIC X(5).                                   CUSMST
      +000008*                           NÚMERO CLIENTE                                CUSMST
   58 +000009           06 NAME             PIC X(25).                                  CUSMST
      +000010*                           NOMBRE CLIENTE                                CUSMST
   59 +000011           06 ADDR             PIC X(20).                                  CUSMST
      +000012*                           DIRECCIÓN CLIENTE                             CUSMST
   60 +000013           06 CITY             PIC X(20).                                  CUSMST
      +000014*                           CIUDAD CLIENTE                                CUSMST
   61 +000015           06 STATE            PIC X(2).                                   CUSMST
      +000016*                           PROVINCIA                                     CUSMST
   62 +000017           06 ZIP              PIC S9(5)        COMP-3.                    CUSMST
      +000018*                           CÓDIGO POSTAL                                 CUSMST
   63 +000019           06 SRHCOD           PIC X(6).                                   CUSMST
      +000020*                           CÓDIGO BÚSQUEDA NÚMERO CLIENTE                CUSMST
   64 +000021           06 CUSTYP           PIC S9(1)        COMP-3.                    CUSMST
      +000022*                           TIPO CLIENTE  1=GOB 2=PLA 3=NEG 4=FAB 5=OTR   CUSMST
   65 +000023           06 ARBAL            PIC S9(6)V9(2)   COMP-3.                    CUSMST
      +000024*                           SALDO CUENTAS PEND.                           CUSMST
   66 +000025           06 ORDBAL           PIC S9(6)V9(2)   COMP-3.                    CUSMST
      +000026*                           CANTIDAD C/C EN ARCH. PEDIDOS                 CUSMST
   67 +000027           06 LSTAMT           PIC S9(6)V9(2)   COMP-3.                    CUSMST
      +000028*                           ULTIMA CANTIDAD PAGADA EN C/C                 CUSMST
   68 +000029           06 LSTDAT           PIC S9(6)        COMP-3.                    CUSMST
      +000030*                           ULTIMA FECHA PAGADA EN C/C                    CUSMST
   69 +000031           06 CRDLMT           PIC S9(6)V9(2)   COMP-3.                    CUSMST
      +000032*                           LÍMITE CRÉDITO CLIENTE                        CUSMST
   70 +000033           06 SLSYR            PIC S9(8)V9(2)   COMP-3.                    CUSMST
      +000034*                           VENTAS CLIENTE ESTE AÑO                       CUSMST
   71 +000035           06 SLSLYR           PIC S9(8)V9(2)   COMP-3.                    CUSMST
      +000036*                           VENTAS CLIENTE AÑO PASADO                     CUSMST
       004000
   72  004100 FD  PAYMENT-UPDATE-DISPLAY-FILE
   73  004200     LABEL RECORDS ARE OMITTED.
   74  004300 01  PAYMENT-UPDATE-DISPLAY-RECORD.
   75  004400     COPY DDS-ALL-FORMATS OF PAYUPDTD.
   76 +000001       05  PAYUPDTD-RECORD PIC X(59).                                     <-ALL-FMTS
      +000002*  FORMATO ENTRADA:SUBFILE1    DESDE ARCHIVO PAYUPDTD   DE BIBLIOTECA XMPLIB  <-ALL-FMTS
      +000003*                           SUBARCHIVO PARA PAGOS CLIENTE                 <-ALL-FMTS
   77 +000004       05  SUBFILE1-I    REDEFINES PAYUPDTD-RECORD.                      <-ALL-FMTS
   78 +000005           06 ACPPMT           PIC X(4).                                   <-ALL-FMTS
      +000006*                           ACEPTAR PAGO                                  <-ALL-FMTS
   79 +000007           06 CUST             PIC X(5).                                   <-ALL-FMTS
      +000008*                           CUSTOMER NUMBER                               <-ALL-FMTS
   80 +000009           06 AMPAID           PIC S9(6)V9(2).                             <-ALL-FMTS
      +000010*                           CANTIDAD PAGADA                               <-ALL-FMTS
   81 +000011           06 ECPMSG           PIC X(31).                                  <-ALL-FMTS
      +000012*                           MENSAJE DE EXCEPCIÓN                          <-ALL-FMTS
   82 +000013           06 OVRPMT           PIC S9(6)V9(2).                             <-ALL-FMTS
      +000014*                           PAGO EXCESIVO                                 <-ALL-FMTS
   83 +000015           06 STSCDE           PIC X(1).                                   <-ALL-FMTS
      +000016*                           CÓDIGO DE ESTADO                              <-ALL-FMTS
      +000017*FORMATO SALIDA:SUBFILE1    DESDE ARCHIVO PAYUPDTD    DE BIBLIOTECA XMPLIB    <-ALL-FMTS
      +000018*                           SUBARCHIVO PARA PAGOS CLIENTE                 <-ALL-FMTS
   84 +000019       05  SUBFILE1-O    REDEFINES PAYUPDTD-RECORD.                      <-ALL-FMTS
   85 +000020           06 SUBFILE1-O-INDIC.                                            <-ALL-FMTS
   86 +000021                07 IN51        PIC 1  INDIC 51.                          <-ALL-FMTS
   87 +000022                07 IN52        PIC 1  INDIC 52.                          <-ALL-FMTS
   88 +000023                07 IN53        PIC 1  INDIC 53.                          <-ALL-FMTS
   89 +000024                07 IN54        PIC 1  INDIC 54.                          <-ALL-FMTS
   90 +000025                07 IN55        PIC 1  INDIC 55.                          <-ALL-FMTS
   91 +000026                07 IN56        PIC 1  INDIC 56.                          <-ALL-FMTS
   92 +000027           06 CUST             PIC X(5).                                   <-ALL-FMTS
      +000028*                           NÚMERO CLIENTE                                <-ALL-FMTS
   93 +000029           06 AMPAID           PIC S9(6)V9(2).                             <-ALL-FMTS
      +000030*                           CANTIDAD PAGADA                               <-ALL-FMTS
   94 +000031           06 ECPMSG           PIC X(31).                                  <-ALL-FMTS
      +000032*                           MENSAJE DE EXCEPCIÓN                          <-ALL-FMTS
```

Figura 74 (Parte 2 de 8). Listado Fuente de un Ejemplo de Programa de Actualización de Pagos

```
5763CB1 V3R0M5                  Listado Fuente AS/400 COBOL
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
   95 +000033           06 OVRPMT            PIC S9(6)V9(2).                            <-ALL-FMTS
      +000034*                              PAGO EXCESIVO                              <-ALL-FMTS
   96 +000035           06 STSCDE            PIC X(1).                                  <-ALL-FMTS
      +000036*                              CÓDIGO DE ESTADO                           <-ALL-FMTS
      +000037*  FORMATO ENTRADA:CONTROL1    DESDE ARCHIVO PAYUPDTD   DE BIBLIOTECA XMPLIB  <-ALL-FMTS
      +000038*                              CONTROL SUBARCHIVO                         <-ALL-FMTS
   97 +000039       05  CONTROL1-I    REDEFINES PAYUPDTD-RECORD.                      <-ALL-FMTS
   98 +000040           06 CONTROL1-I-INDIC.                                          <-ALL-FMTS
   99 +000041                07 IN99         PIC 1  INDIC 99.                        <-ALL-FMTS
      +000042*                              TECLA AYUDA                                <-ALL-FMTS
  100 +000043                07 IN98         PIC 1  INDIC 98.                        <-ALL-FMTS
      +000044*                              FIN ACTUALIZACIÓN PAGOS                    <-ALL-FMTS
  101 +000045                07 IN97         PIC 1  INDIC 97.                        <-ALL-FMTS
      +000046*                              IGNORAR ENTRADA                            <-ALL-FMTS
      +000047*FORMATO SALIDA:CONTROL1    DESDE ARCHIVO PAYUPDTD    DE BIBLIOTECA XMPLIB    <-ALL-FMTS
      +000048*                              CONTROL SUBARCHIVO                         <-ALL-FMTS
  102 +000049       05  CONTROL1-O    REDEFINES PAYUPDTD-RECORD.                      <-ALL-FMTS
  103 +000050           06 CONTROL1-O-INDIC.                                          <-ALL-FMTS
  104 +000051                07 IN61         PIC 1  INDIC 61.                        <-ALL-FMTS
  105 +000052                07 IN62         PIC 1  INDIC 62.                        <-ALL-FMTS
  106 +000053                07 IN99         PIC 1  INDIC 99.                        <-ALL-FMTS
      +000054*                              TECLA AYUDA                                <-ALL-FMTS
  107 +000055                07 IN63         PIC 1  INDIC 63.                        <-ALL-FMTS
  108 +000056                07 IN64         PIC 1  INDIC 64.                        <-ALL-FMTS
      +000057*  FORMATO ENTRADA:MESSAGE1    DESDE ARCHIVO PAYUPDTD   DE BIBLIOTECA XMPLIB  <-ALL-FMTS
      +000058*                              REGISTRO MENSAJES                          <-ALL-FMTS
      +000059*      05  MESSAGE1-I    REDEFINES PAYUPDTD-RECORD.                      <-ALL-FMTS
      +000060*FORMATO SALIDA:MESSAGE1    DESDE ARCHIVO PAYUPDTD    DE BIBLIOTECA XMPLIB    <-ALL-FMTS
      +000061*                              REGISTRO MENSAJES                          <-ALL-FMTS
  109 +000062       05  MESSAGE1-O    REDEFINES PAYUPDTD-RECORD.                      <-ALL-FMTS
  110 +000063           06 MESSAGE1-O-INDIC.                                          <-ALL-FMTS
  111 +000064                07 IN71         PIC 1  INDIC 71.                        <-ALL-FMTS
       004500
  112  004600 WORKING-STORAGE SECTION.
       004700
  113  004800 01  REL-NUMBER                  PIC 9(05)
  114  004900                                    VALUE ZEROS.
       005000
  115  005100 01  WS-CONTROL.
  116  005200     05  WS-IND                   PIC X(02).
  117  005300     05  WS-FORMAT                PIC X(10).
  118  005400 01  SYSTEM-DATE.
  119  005500     05  SYSTEM-YEAR              PIC 99.
  120  005600     05  SYSTEM-MONTH             PIC 99.
  121  005700     05  SYSTEM-DAY               PIC 99.
  122  005800 01  PROGRAM-DATE.
  123  005900     05  PROGRAM-MONTH            PIC 99.
  124  006000     05  PROGRAM-DAY              PIC 99.
  125  006100     05  PROGRAM-YEAR             PIC 99.
  126  006200 01  FILE-DATE REDEFINES PROGRAM-DATE
  127  006300                                  PIC S9(6).
  128  006400 01  EXCEPTION-STATUS.
  129  006500     05  STATUS-CODE-ONE          PIC XX.
  130  006600         88  SUBFILE-IS-FULL      VALUE '9M'.
  131  006700 01  EXCEPTION-MESSAGES.
  132  006800     05  MESSAGE-ONE              PIC X(31)
  133  006900         VALUE 'CUSTOMER DOES NOT EXIST        '.
  134  007000     05  MESSAGE-TWO              PIC X(31)
  135  007100         VALUE 'NO INVOICES EXIST FOR CUSTOMER '.
  136  007200     05  MESSAGE-THREE            PIC X(31)
  137  007300         VALUE 'CUSTOMER HAS AN OVER PAYMENT OF'.
  138  007400 01  PROGRAM-VARIABLES.
  139  007500     05  AMOUNT-OWED              PIC S9(6)V99.
  140  007600     05  AMOUNT-PAID              PIC S9(6)V99.
  141  007700     05  INVOICE-BALANCE          PIC S9(6)V99.
  142  007800 01  ERRPGM-PARAMETERS.
  143  007900     05  DISPLAY-PARAMETER        PIC X(8)
  144  008000                                    VALUE 'PAYUPDTD'.
  145  008100     05  DUMMY-ONE                PIC X(6)
  146  008200                                    VALUE SPACES.
  147  008300     05  DUMMY-TWO                PIC X(6)
  148  008400                                    VALUE SPACES.
  149  008500     05  STATUS-CODE-TWO.
  150  008600        10  PRIMARY               PIC X(1).
  151  008700        10  SECONDARY             PIC X(1).
```

Figura 74 (Parte 3 de 8). Listado Fuente de un Ejemplo de Programa de Actualización de Pagos

```
5763CB1 V3R0M5                  Listado Fuente AS/400 COBOL
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
  152  008800       10  FILLER                 PIC X(5)
  153  008900                                    VALUE SPACES.
  154  009000    05  DUMMY-THREE               PIC X(10)
  155  009100                                    VALUE SPACES.
       009200
  156  009300 01  SWITCH-AREA.
  157  009400    05  SW01                      PIC 1.
  158  009500       88  WRITE-DISPLAY               VALUE B'1'.
  159  009600       88  READ-DISPLAY                VALUE B'0'.
  160  009700    05  SW02                      PIC 1.
  161  009800       88  SUBFILE1-FORMAT             VALUE B'1'.
  162  009900       88  NOT-SUBFILE1-FORMAT         VALUE B'0'.
  163  010000    05  SW03                      PIC 1.
  164  010100       88  CONTROL1-FORMAT             VALUE B'1'.
  165  010200       88  NOT-CONTROL1-FORMAT         VALUE B'1'.
  166  010300    05  SW04                      PIC 1.
  167  010400       88  NO-MORE-TRANSACTIONS-EXIST   VALUE B'1'.
  168  010500       88  TRANSACTIONS-EXIST           VALUE B'0'.
  169  010600    05  SW05                      PIC 1.
  170  010700       88  CUSTOMER-NOT-FOUND           VALUE B'1'.
  171  010800       88  CUSTOMER-EXIST               VALUE B'0'.
  172  010900    05  SW06                      PIC 1.
  173  011000       88  NO-MORE-INVOICES-EXIST       VALUE B'1'.
  174  011100       88  CUSTOMER-INVOICE-EXIST       VALUE B'0'.
  175  011200    05  SW07                      PIC 1.
  176  011300       88  NO-MORE-PAYMENT-EXIST        VALUE B'0'.
  177  011400       88  PAYMENT-EXIST                VALUE B'0'.
  178  011500    05  SW08                      PIC 1.
  179  011600       88  INPUT-ERRORS-EXIST           VALUE B'1'.
  180  011700       88  NO-INPUT-ERRORS-EXIST        VALUE B'0'.
  181  011800    05  SW09                      PIC 1.
  182  011900       88  OVER-PAYMENT-DISPLAYED-ONCE  VALUE B'1'.
  183  012000       88  OVER-PAYMENT-NOT-DISPLAYED   VALUE B'0'.
       012100
  184  012200 01  INDICATOR-AREA.
  185  012300    05  IN99                      PIC 1 INDIC 99.
  186  012400       88 HELP-IS-NEEDED              VALUE B'1'.
  187  012500       88 HELP-IS-NOT-NEEDED          VALUE B'0'.
  188  012600    05  IN98                      PIC 1 INDIC 98.
  189  012700       88 END-OF-PAYMENT-UPDATE       VALUE B'1'.
  190  012800    05  IN97                      PIC 1 INDIC 97.
  191  012900       88 IGNORE-INPUT                VALUE B'1'.
  192  013000    05  IN51                      PIC 1 INDIC 51.
  193  013100       88 DISPLAY-ACCEPT-PAYMENT      VALUE B'1'.
  194  013200       88 DO-NOT-DISPLAY-ACCEPT-PAYMENT  VALUE B'0'.
  195  013300    05  IN52                      PIC 1 INDIC 52.
  196  013400       88 REVERSE-FIELD-IMAGE         VALUE B'1'.
  197  013500       88 DO-NOT-REVERSE-FIELD-IMAGE  VALUE B'0'.
  198  013600    05  IN53                      PIC 1 INDIC 53.
  199  013700       88 DO-NOT-DISPLAY-FIELD        VALUE B'1'.
  200  013800       88 DISPLAY-FIELD               VALUE B'0'.
  201  013900    05  IN54                      PIC 1 INDIC 54.
  202  014000       88 PROTECT-INPUT-FIELD         VALUE B'1'.
  203  014100       88 DO-NOT-PROTECT-INPUT-FIELD  VALUE B'0'.
  204  014200    05  IN55                      PIC 1 INDIC 55.
  205  014300       88 MAKE-FIELD-BLINK            VALUE B'1'.
  206  014400       88 DO-NOT-MAKE-FIELD-BLINK     VALUE B'0'.
  207  014500    05  IN56                      PIC 1 INDIC 56.
  208  014600       88 DISPLAY-OVER-PAYMENT        VALUE B'1'.
  209  014700       88 DO-NOT-DISPLAY-OVER-PAYMENT VALUE B'0'.
  210  014800    05  IN61                      PIC 1 INDIC 61.
  211  014900       88 CLEAR-SUBFILE               VALUE B'1'.
  212  015000       88 DO-NOT-CLEAR-SUBFILE        VALUE B'0'.
  213  015100    05  IN62                      PIC 1 INDIC 62.
  214  015200       88 DISPLAY-SCREEN              VALUE B'1'.
  215  015300       88 DO-NOT-DISPLAY-SCREEN       VALUE B'0'.
  216  015400    05  IN63                      PIC 1 INDIC 63.
  217  015500       88 DISPLAY-ACCEPT-HEADING      VALUE B'1'.
  218  015600       88 DO-NOT-DISPLAY-ACCEPT-HEADING  VALUE B'0'.
  219  015700    05  IN64                      PIC 1 INDIC 64.
  220  015800       88 DISPLAY-EXCEPTION           VALUE B'1'.
  221  015900       88 DO-NOT-DISPLAY-EXCEPTION    VALUE B'0'.
  222  016000    05  IN71                      PIC 1 INDIC 71.
  223  016100       88 DISPLAY-ACCEPT-MESSAGE      VALUE B'1'.
  224  016200       88 DO-NOT-DISPLAY-ACCEPT-MESSAGE  VALUE B'0'.
       016300
```

Figura 74 (Parte 4 de 8). Listado Fuente de un Ejemplo de Programa de Actualización de Pagos

```
5763CB1 V3R0M5                  Listado Fuente AS/400 COBOL
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
  225  016400 PROCEDURE DIVISION.
       016500
       016600 DECLARATIVES.
       016700
       016800 TRANSACTION-ERROR SECTION.
       016900     USE AFTER STANDARD ERROR PROCEDURE
       017000         PAYMENT-UPDATE-DISPLAY-FILE.
       017100 WORK-STATION-ERROR-HANDLER.
  226  017200     IF SUBFILE-IS-FULL THEN
       017300       NEXT SENTENCE
       017400     ELSE
  227  017500       DISPLAY 'ERROR IN PAYMENT-UPDATE' STATUS-CODE-ONE.
       017600 END DECLARATIVES.
       017700
       017800 CUSTOMER-PAYMENT-UPDATE SECTION.
       017900 MAINLINE-ROUTINE.
  228  018000     PERFORM SET-UP-ROUTINE.
  229  018100     PERFORM PROCESS-TRANSACTION-FILE
       018200       UNTIL END-OF-PAYMENT-UPDATE.
  230  018300     PERFORM CLEAN-UP-ROUTINE.
       018400
       018500 SET-UP-ROUTINE.
  231  018600     OPEN I-O      CUSTOMER-INVOICE-FILE
       018700                   CUSTOMER-MASTER-FILE
       018800                   PAYMENT-UPDATE-DISPLAY-FILE.
  232  018900     MOVE ALL B'0' TO INDICATOR-AREA
       019000                    SWITCH-AREA.
  233  019100     ACCEPT SYSTEM-DATE FROM DATE.
  234  019200     MOVE SYSTEM-YEAR  TO PROGRAM-YEAR.
  235  019300     MOVE SYSTEM-MONTH TO PROGRAM-MONTH.
  236  019400     MOVE SYSTEM-DATE  TO PROGRAM-DAY.
  237  019500     SET WRITE-DISPLAY
       019600         CONTROL1-FORMAT
       019700         DO-NOT-DISPLAY-OVER-PAYMENT
       019800         DO-NOT-PROTECT-INPUT-FIELD
       019900         DO-NOT-REVERSE-FIELD-IMAGE
       020000         DO-NOT-MAKE-FIELD-BLINK
       020100         CLEAR-SUBFILE TO TRUE.
  238  020200     MOVE CORR INDICATOR-AREA TO CONTROL1-O-INDIC.
  239  020300     WRITE PAYMENT-UPDATE-DISPLAY-RECORD
       020400         FORMAT IS 'CONTROL1'.
  240  020500     SET DO-NOT-CLEAR-SUBFILE TO TRUE.
  241  020600     PERFORM INITIALIZE-SUBFILE-RECORD 17 TIMES.
  242  020700     SET DISPLAY-SCREEN TO TRUE.
  243  020800     MOVE CORR INDICATOR-AREA TO CONTROL1-O-INDIC.
  244  020900     WRITE PAYMENT-UPDATE-DISPLAY-RECORD
       021000         FORMAT IS 'CONTROL1'.
  245  021100     READ PAYMENT-UPDATE-DISPLAY-FILE RECORD
       021200         FORMAT IS 'CONTROL1'.
  246  021300     MOVE CORR CONTROL1-I-INDIC TO INDICATOR-AREA.
       021400 PROCESS-TRANSACTION-FILE.
  247  021500     IF HELP-IS-NOT-NEEDED THEN
  248  021600        IF  IGNORE-INPUT THEN
  249  021700            SET WRITE-DISPLAY
       021800                CONTROL1-FORMAT
       021900                CLEAR-SUBFILE
       022000                DISPLAY-FIELD
       022100                DO-NOT-DISPLAY-OVER-PAYMENT
       022200                DO-NOT-PROTECT-INPUT-FIELD
       022300                DO-NOT-REVERSE-FIELD-IMAGE
       022400                DO-NOT-DISPLAY-ACCEPT-PAYMENT
       022500                DO-NOT-DISPLAY-ACCEPT-HEADING
       022600                DO-NOT-DISPLAY-ACCEPT-MESSAGE
       022700                DO-NOT-MAKE-FIELD-BLINK TO TRUE
  250  022800            MOVE CORR INDICATOR-AREA TO CONTROL1-O-INDIC
  251  022900            WRITE PAYMENT-UPDATE-DISPLAY-RECORD
       023000                FORMAT IS 'CONTROL1'
  252  023100            SET DO-NOT-CLEAR-SUBFILE TO TRUE
  253  023200            MOVE 0 TO REL-NUMBER
  254  023300            PERFORM INITIALIZE-SUBFILE-RECORD 17 TIMES
       023400         ELSE
  255  023500            SET TRANSACTIONS-EXIST
       023600                DO-NOT-DISPLAY-ACCEPT-HEADING
       023700                DO-NOT-DISPLAY-ACCEPT-MESSAGE
       023800                DO-NOT-DISPLAY-EXCEPTION TO TRUE
  256  023900            PERFORM READ-MODIFIED-SUBFILE-RECORD
```

Figura 74 (Parte 5 de 8). Listado Fuente de un Ejemplo de Programa de Actualización de Pagos

```
5763CB1 V3R0M5                   Listado Fuente AS/400 COBOL
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
  257  024000            PERFORM TRANSACTION-VALIDATION
       024100                UNTIL NO-MORE-TRANSACTIONS-EXIST
  258  024200            SET NO-INPUT-ERRORS-EXIST TO TRUE
  259  024300            PERFORM TEST-FOR-RECORD-INPUT-ERRORS
       024400                VARYING REL-NUMBER
       024500                FROM    1
       024600                BY      1
       024700                UNTIL   REL-NUMBER IS GREATER THAN 17
       024800                OR      INPUT-ERRORS-EXIST
  260  024900            IF  NO-INPUT-ERRORS-EXIST THEN
  261  025000                IF  OVER-PAYMENT-DISPLAYED-ONCE THEN
  262  025100                    SET WRITE-DISPLAY
       025200                        CONTROL1-FORMAT
       025300                        DO-NOT-DISPLAY-OVER-PAYMENT
       025400                        DO-NOT-PROTECT-INPUT-FIELD
       025500                        DO-NOT-REVERSE-FIELD-IMAGE
       025600                        DO-NOT-MAKE-FIELD-BLINK
       025700                        DO-NOT-DISPLAY-ACCEPT-PAYMENT
       025800                        DO-NOT-DISPLAY-ACCEPT-HEADING
       025900                        DO-NOT-DISPLAY-ACCEPT-MESSAGE
       026000                        DO-NOT-DISPLAY-EXCEPTION
       026100                        CLEAR-SUBFILE
       026200                        DISPLAY-FIELD
       026300                          TO TRUE
  263  026400                    MOVE CORR INDICATOR-AREA TO CONTROL1-O-INDIC
  264  026500                    WRITE PAYMENT-UPDATE-DISPLAY-RECORD
       026600                        FORMAT IS 'CONTROL1'
  265  026700                    SET DO-NOT-CLEAR-SUBFILE TO TRUE
  266  026800                    MOVE 0 TO REL-NUMBER
  267  026900                    PERFORM INITIALIZE-SUBFILE-RECORD 17 TIMES
       027000                ELSE
  268  027100                    SET OVER-PAYMENT-DISPLAYED-ONCE TO TRUE
       027200            ELSE
       027300                NEXT SENTENCE
       027400     ELSE
       027500         NEXT SENTENCE.
  269  027600     SET WRITE-DISPLAY, DISPLAY-SCREEN TO TRUE.
  270  027700     MOVE CORR INDICATOR-AREA TO MESSAGE1-O-INDIC.
  271  027800     WRITE PAYMENT-UPDATE-DISPLAY-RECORD
       027900         FORMAT IS 'MESSAGE1'.
  272  028000     SET WRITE-DISPLAY, CONTROL1-FORMAT TO TRUE.
  273  028100     MOVE CORR INDICATOR-AREA TO CONTROL1-O-INDIC.
  274  028200     WRITE PAYMENT-UPDATE-DISPLAY-RECORD
       028300         FORMAT IS 'CONTROL1'.
  275  028400     READ PAYMENT-UPDATE-DISPLAY-FILE RECORD
       028500         FORMAT IS 'CONTROL1'.
  276  028600     MOVE CORR CONTROL1-I-INDIC TO INDICATOR-AREA.
       028700 READ-MODIFIED-SUBFILE-RECORD.
  277  028800     READ SUBFILE PAYMENT-UPDATE-DISPLAY-FILE
       028900         NEXT MODIFIED RECORD FORMAT IS 'SUBFILE1'
  278  029000         AT END SET NO-MORE-TRANSACTIONS-EXIST TO TRUE.
       029100 TEST-FOR-RECORD-INPUT-ERRORS.
  279  029200     READ SUBFILE PAYMENT-UPDATE-DISPLAY-FILE RECORD
       029300         FORMAT IS 'SUBFILE1'.
  280  029400     IF  STSCDE OF SUBFILE1-I IS EQUAL TO '1' THEN
  281  029500         SET INPUT-ERRORS-EXIST TO TRUE
       029600     ELSE
       029700         NEXT SENTENCE.
       029800 TRANSACTION-VALIDATION.
  282  029900     MOVE CUST OF SUBFILE1-I OF PAYMENT-UPDATE-DISPLAY-RECORD
       030000         TO CUST OF CUSTOMER-MASTER-RECORD.
  283  030100     SET CUSTOMER-EXIST TO TRUE.
  284  030200     READ CUSTOMER-MASTER-FILE
  285  030300         INVALID KEY SET CUSTOMER-NOT-FOUND TO TRUE.
  286  030400     IF  CUSTOMER-EXIST THEN
  287  030500         MOVE CUST OF CUSMST TO CUST OF ORDHDR
  288  030600         MOVE ZEROES TO INVNUM
  289  030700         SET CUSTOMER-INVOICE-EXIST TO TRUE
  290  030800         PERFORM START-ON-CUSTOMER-INVOICE-FILE
  291  030900         IF  CUSTOMER-INVOICE-EXIST THEN
  292  031000             PERFORM READ-CUSTOMER-INVOICE-RECORD
  293  031100             IF CUSTOMER-INVOICE-EXIST THEN
  294  031200                 PERFORM CUSTOMER-MASTER-FILE-UPDATE
  295  031300                 MOVE AMPAID OF SUBFILE1-I TO AMOUNT-PAID
  296  031400                 SET PAYMENT-EXIST TO TRUE
```

Figura 74 (Parte 6 de 8). Listado Fuente de un Ejemplo de Programa de Actualización de Pagos

```
5763CB1 V3R0M5                  Listado Fuente AS/400 COBOL
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
  297  031500                 PERFORM PAYMENT-UPDATE
       031600                     UNTIL NO-MORE-INVOICES-EXIST
       031700                     OR NO-MORE-PAYMENT-EXIST
  298  031800                 IF  ARBAL OF CUSTOMER-MASTER-RECORD IS NEGATIVE
  299  031900                     SET MAKE-FIELD-BLINK
       032000                         DISPLAY-FIELD
       032100                         DO-NOT-REVERSE-FIELD-IMAGE
       032200                         OVER-PAYMENT-NOT-DISPLAYED
       032300                         DISPLAY-OVER-PAYMENT
       032400                         DISPLAY-EXCEPTION
       032500                         DO-NOT-DISPLAY-ACCEPT-PAYMENT
       032600                         PROTECT-INPUT-FIELD TO TRUE
  300  032700                     MOVE ARBAL TO OVRPMT OF SUBFILE1-O
  301  032800                     MOVE MESSAGE-THREE TO ECPMSG OF SUBFILE1-O
  302  032900                     MOVE '0' TO STSCDE OF SUBFILE1-O
  303  033000                     PERFORM REWRITE-DISPLAY-SUBFILE-RECORD
       033100                 ELSE
  304  033200                     SET DO-NOT-DISPLAY-FIELD
       033300                         DO-NOT-DISPLAY-OVER-PAYMENT
       033400                         DO-NOT-REVERSE-FIELD-IMAGE
       033500                         DO-NOT-MAKE-FIELD-BLINK
       033600                         DO-NOT-DISPLAY-ACCEPT-PAYMENT
       033700                         PROTECT-INPUT-FIELD TO TRUE
  305  033800                     MOVE SPACES TO ECPMSG OF SUBFILE1-O
  306  033900                     MOVE ZEROES TO OVRPMT OF SUBFILE1-O
  307  034000                     MOVE '0' TO STSCDE OF SUBFILE1-O
  308  034100                     PERFORM REWRITE-DISPLAY-SUBFILE-RECORD
       034200             ELSE
  309  034300                 PERFORM NO-CUSTOMER-INVOICE-ROUTINE
       034400         ELSE
  310  034500             PERFORM NO-CUSTOMER-INVOICE-ROUTINE
       034600     ELSE
  311  034700         SET REVERSE-FIELD-IMAGE
       034800             DO-NOT-PROTECT-INPUT-FIELD
       034900             DISPLAY-FIELD
       035000             DO-NOT-DISPLAY-OVER-PAYMENT
       035100             DO-NOT-MAKE-FIELD-BLINK
       035200             DISPLAY-EXCEPTION
       035300             DO-NOT-DISPLAY-ACCEPT-PAYMENT
       035400             DO-NOT-PROTECT-INPUT-FIELD TO TRUE
  312  035500         MOVE ZEROES TO OVRPMT OF SUBFILE1-O
  313  035600         MOVE MESSAGE-ONE TO ECPMSG OF SUBFILE1-O
  314  035700         MOVE '1' TO STSCDE OF SUBFILE1-O
  315  035800         PERFORM REWRITE-DISPLAY-SUBFILE-RECORD.
  316  035900     PERFORM READ-MODIFIED-SUBFILE-RECORD.
       036000 START-ON-CUSTOMER-INVOICE-FILE.
  317  036100     START CUSTOMER-INVOICE-FILE
       036200         KEY IS GREATER THAN COMP-KEY
  318  036300         INVALID KEY SET NO-MORE-INVOICES-EXIST TO TRUE.
       036400 READ-CUSTOMER-INVOICE-RECORD.
  319  036500     READ CUSTOMER-INVOICE-FILE NEXT RECORD
  320  036600         AT END SET NO-MORE-INVOICES-EXIST TO TRUE.
  321  036700     IF  CUST OF CUSTOMER-MASTER-RECORD
       036800         IS NOT EQUAL TO CUST OF CUSTOMER-INVOICE-RECORD THEN
  322  036900         SET NO-MORE-INVOICES-EXIST TO TRUE
       037000     ELSE
       037100         NEXT SENTENCE.
       037200 CUSTOMER-MASTER-FILE-UPDATE.
  323  037300     MOVE FILE-DATE TO LSTDAT OF CUSTOMER-MASTER-RECORD.
  324  037400     MOVE AMPAID OF SUBFILE1-I
       037500         TO LSTAMT OF CUSTOMER-MASTER-RECORD.
  325  037600     SUBTRACT AMPAID OF SUBFILE1-I
       037700         FROM ARBAL OF CUSTOMER-MASTER-RECORD.
  326  037800     REWRITE CUSTOMER-MASTER-RECORD.
       037900  REWRITE-DISPLAY-SUBFILE-RECORD.
  327  038000     MOVE AMPAID OF SUBFILE1-I TO AMPAID OF SUBFILE1-O.
  328  038100     MOVE CUST OF SUBFILE1-I TO CUST OF SUBFILE1-O.
  329  038200     SET WRITE-DISPLAY TO TRUE.
  330  038300     SET SUBFILE1-FORMAT TO TRUE.
  331  038400     MOVE CORR INDICATOR-AREA TO SUBFILE1-O-INDIC.
  332  038500     REWRITE SUBFILE PAYMENT-UPDATE-DISPLAY-RECORD
       038600         FORMAT IS 'SUBFILE1'.
       038700 NO-CUSTOMER-INVOICE-ROUTINE.
  333  038800     IF  STSCDE OF SUBFILE1-I IS EQUAL TO '1' THEN
  334  038900        IF  ACPPMT OF SUBFILE1-I IS EQUAL TO '*NO' THEN
  335  039000           SET DO-NOT-DISPLAY-FIELD
```

Figura 74 (Parte 7 de 8). Listado Fuente de un Ejemplo de Programa de Actualización de Pagos

```
5763CB1 V3R0M5                  Listado Fuente AS/400 COBOL
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
      039100               DO-NOT-DISPLAY-OVER-PAYMENT
      039200               DO-NOT-REVERSE-FIELD-IMAGE
      039300               DO-NOT-MAKE-FIELD-BLINK
      039400               DO-NOT-DISPLAY-ACCEPT-PAYMENT
      039500               PROTECT-INPUT-FIELD
      039600                 TO TRUE
  336 039700            MOVE SPACES TO ECPMSG OF SUBFILE1-O
  337 039800            MOVE ZEROES TO OVRPMT OF SUBFILE1-O
  338 039900            MOVE '0' TO STSCDE OF SUBFILE1-O
  339 040000            PERFORM REWRITE-DISPLAY-SUBFILE-RECORD
      040100        ELSE
  340 040200            PERFORM CUSTOMER-MASTER-FILE-UPDATE
  341 040300            SET MAKE-FIELD-BLINK
      040400                DISPLAY-FIELD
      040500                DO-NOT-REVERSE-FIELD-IMAGE
      040600                OVER-PAYMENT-NOT-DISPLAYED
      040700                DISPLAY-OVER-PAYMENT
      040800                DISPLAY-EXCEPTION
      040900                DO-NOT-DISPLAY-ACCEPT-PAYMENT
      041000                PROTECT-INPUT-FIELD
      041100                  TO TRUE
  342 041200            MOVE ARBAL TO OVRPMT OF SUBFILE1-O
  343 041300            MOVE MESSAGE-THREE TO ECPMSG OF SUBFILE1-O
  344 041400            MOVE '0' TO STSCDE OF SUBFILE1-O
  345 041500            PERFORM REWRITE-DISPLAY-SUBFILE-RECORD
      041600     ELSE
  346 041700         SET REVERSE-FIELD-IMAGE
      041800             DISPLAY-FIELD
      041900             DO-NOT-PROTECT-INPUT-FIELD
      042000             DO-NOT-DISPLAY-OVER-PAYMENT
      042100             DISPLAY-EXCEPTION
      042200             DISPLAY-ACCEPT-PAYMENT
      042300             DISPLAY-ACCEPT-HEADING
      042400             DISPLAY-ACCEPT-MESSAGE
      042500             DO-NOT-MAKE-FIELD-BLINK
      042600               TO TRUE
  347 042700         MOVE ZEROS TO OVRPMT OF SUBFILE1-O
  348 042800         MOVE MESSAGE-TWO TO ECPMSG OF SUBFILE1-O
  349 042900         MOVE '1' TO STSCDE OF SUBFILE1-O
  350 043000         PERFORM REWRITE-DISPLAY-SUBFILE-RECORD.
      043100 PAYMENT-UPDATE.
  351 043200     SUBTRACT AMPAID OF CUSTOMER-INVOICE-RECORD
      043300         FROM ORDAMT OF CUSTOMER-INVOICE-RECORD
      043400         GIVING AMOUNT-OWED.
  352 043500     SUBTRACT AMOUNT-PAID
      043600         FROM AMOUNT-OWED
      043700         GIVING INVOICE-BALANCE.
  353 043800     IF  INVOICE-BALANCE IS LESS THAN .01 THEN
  354 043900         MOVE 2 TO OPNSTS OF CUSTOMER-INVOICE-RECORD
  355 044000         MOVE ORDAMT OF CUSTOMER-INVOICE-RECORD
      044100           TO AMPAID OF CUSTOMER-INVOICE-RECORD
  356 044200         SUBTRACT AMOUNT-OWED
      044300             FROM AMOUNT-PAID
      044400     ELSE
  357 044500         ADD AMOUNT-PAID TO AMPAID OF CUSTOMER-INVOICE-RECORD
  358 044600         SET NO-MORE-PAYMENT-EXIST TO TRUE.
  359 044700     REWRITE CUSTOMER-INVOICE-RECORD.
  360 044800     IF  NO-MORE-PAYMENT-EXIST THEN
      044900         NEXT SENTENCE
      045000     ELSE
  361 045100         PERFORM READ-CUSTOMER-INVOICE-RECORD.
      045200 INITIALIZE-SUBFILE-RECORD.
  362 045300     MOVE SPACES TO CUST OF SUBFILE1-O.
  363 045400     MOVE SPACES TO ECPMSG OF SUBFILE1-O.
  364 045500     MOVE '0' TO STSCDE OF SUBFILE1-O.
  365 045600     MOVE ZEROS TO AMPAID OF SUBFILE1-O.
  366 045700     MOVE ZEROS TO OVRPMT OF SUBFILE1-O.
  367 045800     ADD 1 TO REL-NUMBER.
  368 045900     MOVE CORR INDICATOR-AREA TO SUBFILE1-O-INDIC.
  369 046000     WRITE SUBFILE PAYMENT-UPDATE-DISPLAY-RECORD
      046100         FORMAT IS 'SUBFILE1'.
      046200 CLEAN-UP-ROUTINE.
  370 046300     CLOSE CUSTOMER-INVOICE-FILE
      046400           CUSTOMER-MASTER-FILE
      046500           PAYMENT-UPDATE-DISPLAY-FILE.
  371 046600     STOP RUN.
                       * * * * *  F I N  D E  F U E N T E  * * * * *
```

Figura 74 (Parte 8 de 8). Listado Fuente de un Ejemplo de Programa de Actualización de Pagos

Esta es la pantalla incial que se graba en la estación de trabajo para solicitar al usuario la entrada del número de cliente y pago:

```
Solicitud de Actualización Pagos de Cliente                    Fecha  05/24/94

            Cliente     Pago


                _____   __________
                _____   __________
                _____   __________
                _____   __________
                _____   __________
                _____   __________
                _____   __________
                _____   __________
                _____   __________
                _____   __________

                _____   __________
                _____   __________
                _____   __________
                _____   __________
                _____   __________
```

El usuario entra los números de cliente y pagos:

```
Solicitud de Actualización Pagos de Cliente                    Fecha  05/24/94

            Cliente     Pago


             34500      2000
             40500     30000
             36000      2500
             12500       200
             22799      4500
             41900      7500
             10001      5000
             49500      2500
             13300      3500
             56900      4000
```

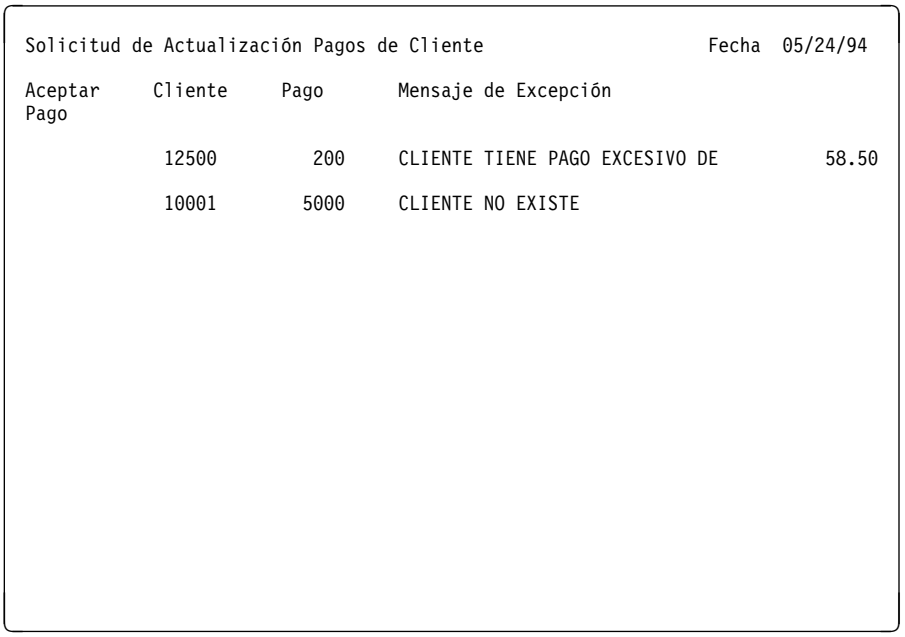
Los pagos excesivos o que tengan números incorrectos de cliente se dejan en la pantalla y se añaden mensajes adecuados:

```
Solicitud de Actualización Pagos de Cliente                    Fecha  05/24/94

Aceptar     Cliente     Pago       Mensaje de Excepción
Pago

_____        40500       30000     NO EXISTEN FACTURAS PARA CLIENTES

_____        12500         200     NO EXISTEN FACTURAS PARA CLIENTES

_____        41900        7500     NO EXISTEN FACTURAS PARA CLIENTES
             10001        5000     CLIENTE NO EXISTE

_____        13300        3500     NO EXISTEN FACTURAS PARA CLIENTES









Aceptar valores de pago: (*NO *YES)
```

Indique qué pagos hay que aceptar:

```
Solicitud de Actualización Pagos de Cliente                    Fecha  05/24/94

Aceptar     Cliente     Pago       Mensaje de Excepción
Pago

  *NO        40500       30000     NO EXISTEN FACTURAS PARA CLIENTES

  *YES       12500         200     NO EXISTEN FACTURAS PARA CLIENTES

  *NO        41900       7500      NO EXISTEN FACTURAS PARA CLIENTES
             10001        5000     CLIENTE NO EXISTE

  *NO        13300        3500     NO EXISTEN FACTURAS PARA CLIENTES









Aceptar valores de pago: (*NO *YES)
```

Se procesan los pagos aceptados y se visualiza la información de pagos excesivos:



Fin de Ampliación de IBM

Capítulo 9. Archivos de Impresora

En este capítulo se describe cómo COBOL/400 interactúa con los distintos tipos de archivos de impresora.

Es posible obtener salida impresa de un programa COBOL emitiendo instrucciones WRITE a uno o más archivos de impresora. Cada archivo de impresora debe tener un nombre único y debe estar asignado a un dispositivo PRINTER o FORMATFILE en la cláusula ASSIGN de esa entrada FILE-CONTROL del archivo.

Debe utilizarse un dispositivo PRINTER (impresora) para los archivos descritos por programa, y debe utilizarse un dispositivo FORMATFILE para archivos de impresora descritos externamente. El mandato Crear Archivo de Impresión (CRTPRTF) puede utilizarse para crear un archivo de impresora (consulte el manual *CL Reference* para obtener más información sobre el mandato CRTPRTF) o un archivo de dispositivo de impresora suministrado por IBM, como por ejemplo QPRINT.

Las operaciones de archivo válidas para un archivo de impresora son WRITE, OPEN y CLOSE. Para obtener una descripción completa de estas operaciones, consulte el manual *COBOL/400 Reference*.

Consulte la publicación *DDS Reference* para obtener más información sobre DDS para archivos de impresora descritos externamente. Para obtener más información acerca de los archivos FORMATFILE, consulte el apartado “Archivos FORMATFILE” en la página 244.

Párrafo SPECIAL-NAMES y Frase ADVANCING

Cuando se especifica el nombre mnemotécnico asociado con el nombre de función CSP en la frase ADVANCING de una instrucción WRITE para un archivo de impresora, se obtiene el mismo resultado que si se especifica ADVANCING 0 LINES.

Cuando se especifica el nombre mnemotécnico asociado con el nombre de función C01 en la frase ADVANCING de una instrucción WRITE para un archivo de impresora, se obtiene el mismo resultado que si se especifica ADVANCING PAGE.

La frase ADVANCING no puede especificarse en instrucciones WRITE para archivos asignados a FORMATFILE.

Cláusula LINAGE

Cuando se especifica la cláusula LINAGE para un archivo asignado a PRINTER, todos los controles de espaciado y paginación se manejan internamente por el código generado por el compilador.

La colocación del papel sólo se realiza cuando se ejecuta la primera instrucción WRITE. El papel de la impresora se coloca en una nueva página física y LINAGE-COUNTER se establece en 1. Cuando se comparte el archivo de impresora y otros programas poseen registros escritos en el archivo, la instrucción COBOL WRITE aún se considera la primera instrucción WRITE. El compilador COBOL/400 maneja la colocación del papel aunque no sea la primera instrucción WRITE para ese archivo.

Todo el espaciado y la paginación de las instrucciones WRITE se controla internamente. El tamaño físico de la página se ignora cuando la posición del papel no se define correctamente para el compilador COBOL/400. Para un archivo con una cláusula LINAGE y que esté asignado a PRINTER, la paginación incluye el espaciado hasta el final de la página lógica (cuerpo de la página) y en el espaciado hasta los márgenes inferiores y superiores.

La utilización de la cláusula LINAGE degrada el rendimiento. La cláusula LINAGE sólo debe utilizarse cuando sea necesaria. Si la paginación física es aceptable, no se necesita la cláusula LINAGE.

La claúsula LINAGE no debe utilizarse para los archivos asignados a FORMATFILE.

Archivos FORMATFILE

Los archivos de impresora descritos externamente deben asignarse a un dispositivo de FORMATFILE. El término FORMATFILE se utiliza porque la frase FORMAT es válida en instrucciones WRITE para el archivo, y el formato de los datos se especifica en las DDS para el archivo.

Cuando se ha especificado un dispositivo FORMATFILE, se puede obtener el formato de la salida impresa de dos maneras:

1. Elija los formatos que han de imprimirse y en qué orden utilizando valores apropiados en las frases FORMAT especificadas en las instrucciones WRITE. Por ejemplo, utilice un formato una sola vez por página para producir un encabezamiento, y utilice otro formato para producir las líneas de detalle en la página.
2. Elija las opciones apropiadas que han de tomarse cuando se imprime cada formato estableciendo los valores de los indicadores y transfiriendo estos indicadores a la frase INDICATOR para la instrucción WRITE. Por ejemplo, los campos pueden subrayarse, las líneas en blanco pueden producirse antes o después de imprimir el formato o puede saltarse la impresión de ciertos campos.

El uso de descripciones externas para archivos de impresora disfruta de las siguientes ventajas con respecto a las descripciones de programa:

- Pueden imprimirse varias líneas mediante una instrucción WRITE. Cuando una instrucción WRITE escribe varias líneas y se alcanza la condición END-OF-PAGE, la instrucción imperativa END-OF-PAGE se procesa cuando se han imprimido todas las líneas. Es posible imprimir líneas en el área de desbordamiento, y en la página siguiente antes de que se procese la instrucción imperativa END-OF-PAGE.

  La Figura 75 en la página 245 muestra un ejemplo de una aparición de la condición END-OF-PAGE en FORMATFILE.
- Se pueden imprimir opcionalmente campos basándose en valores del indicador.
- La edición de los valores de campos se define fácilmente.
- Es más fácil el mantenimiento de formatos de impresión, especialmente los utilizados por múltiples programas.

La utilización de la frase ADVANCING para archivos FORMATFILE origina la emisión de un mensaje de error. El avance de líneas se controla en un archivo FORMATFILE mediante palabras clave DDS, como por ejemplo SKIPA y SKIPB, así como mediante el uso de números de línea.

Para los archivos FORMATFILE, la cláusula LINAGE no es válida.

```
5763CB1 V3R0M5                  Fuente COBOL AS/400
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
    1  000100 IDENTIFICATION DIVISION.                                                              02/01/94
    2  000200 PROGRAM-ID.     FRMTFILE.                                                             03/22/94
    3  000300   AUTHOR.       PROGRAMMER NAME.                                                      01/27/94
    4  000400   INSTALLATION. TORONTO COBOL DEVELOPMENT CENTRE.                                     01/27/94
    5  000500   DATE-WRITTEN. 02/02/89.                                                             02/04/94
    8  000080   DATE-COMPILED.  05/24/94 14:29:31   .                                             03/01/94
    7  000700 ENVIRONMENT DIVISION.                                                                01/27/94
    8  000800 CONFIGURATION SECTION.                                                               01/27/94
    9  000900 SOURCE-COMPUTER. IBM-AS400.                                                          01/27/94
   10  001000 OBJECT-COMPUTER. IBM-AS400.                                                          01/27/94
   11  001100 INPUT-OUTPUT SECTION.                                                                01/27/94
   12  001200 FILE-CONTROL.                                                                        01/27/94
   13  001300     SELECT PERSREPT ASSIGN TO FORMATFILE-PERSREPT-SI  1                               02/04/94
   14  001400         ORGANIZATION IS SEQUENTIAL.                                                  02/04/94
   15  001500     SELECT PERSFILE ASSIGN TO DATABASE-PERSFILE                                      02/04/94
   16  001600         ORGANIZATION IS INDEXED                                                      02/04/94
   17  001700         ACCESS MODE IS SEQUENTIAL                                                    02/04/94
   18  001800         RECORD IS EXTERNALLY-DESCRIBED-KEY.                                          02/04/94
   19  001900 DATA DIVISION.                                                                       01/27/94
   20  002000 FILE SECTION.                                                                        01/27/94
   21  002100 FD  PERSREPT                                                                         02/04/94
   22  002200     LABEL RECORDS ARE STANDARD.                                                      02/04/94
   23  002300 01  PERSREPT-REC.                                                                   02/04/94
   24  002400     COPY DDS-ALL-FORMATS-O OF PERSREPT.  2
   25 +000001       05  PERSREPT-RECORD PIC X(130).                                 <-ALL-FMTS
      +000002* FORMATO SALIDA:HEADING    DESDE ARCHIVO PERSREPT   DE BIBLIOT. XMPLIB      <-ALL-FMTS
      +000003*                                                                      <-ALL-FMTS
   26 +000004       05  HEADING-O     REDEFINES PERSREPT-RECORD.                    <-ALL-FMTS
   27 +000005           06 ORDERTYPE         PIC X(15).                            <-ALL-FMTS
      +000006* FORMATO SALIDA:DETAIL    DESDE ARCHIVO PERSREPT   DE BIBLIOT. XMPLIB       <-ALL-FMTS
      +000007*                                                                      <-ALL-FMTS
   28 +000008       05  DETAIL-O      REDEFINES PERSREPT-RECORD.  3                 <-ALL-FMTS
   29 +000009           06 NAME              PIC X(30).                            <-ALL-FMTS
   30 +000010           06 EMPLNO            PIC S9(6).                            <-ALL-FMTS
   31 +000011           06 BIRTHDATE         PIC X(6).                             <-ALL-FMTS
   32 +000012           06 ADDRESS1          PIC X(35).                            <-ALL-FMTS
   33 +000013           06 MARSTAT           PIC X(1).                             <-ALL-FMTS
   34 +000014           06 SPOUSENAME        PIC X(30).                            <-ALL-FMTS
   35 +000015           06 ADDRESS2          PIC X(20).                            <-ALL-FMTS
   36 +000016           06 NUMCHILD          PIC S9(2).                            <-ALL-FMTS
   37  002500 FD  PERSFILE
   38  002600     LABEL RECORDS ARE STANDARD.
   39  002700 01  PERSFILE-REC.
   40  002800     COPY DDS-ALL-FORMATS-O OF PERSFILE.
   41 +000001       05  PERSFILE-RECORD PIC X(130).                                 <-ALL-FMTS
      +000002*    FORMATO E-S:PERSREC    DESDE ARCHIVO PERSFILE   DE BIBLOTE. XMPLIB      <-ALL-FMTS
      +000003*                                                                      <-ALL-FMTS
      +000004*DEFINICIONES CLAVE PARA FORMATO DE REGISTRO PERSREC                    <-ALL-FMTS
      +000005*  NÚMERO              NOMBRE          RECUPERACIÓN    TIPO  ALTSEQ      <-ALL-FMTS
      +000006*   0001   EMPLNO                        ASCENDENTE    SIGNO     NO      <-ALL-FMTS
   42 +000007       05  PERSREC       REDEFINES PERSFILE-RECORD.                    <-ALL-FMTS
   43 +000008           06 EMPLNO            PIC S9(6).                            <-ALL-FMTS
   44 +000009           06 NAME              PIC X(30).                            <-ALL-FMTS
   45 +000010           06 ADDRESS1          PIC X(35).                            <-ALL-FMTS
   46 +000011           06 ADDRESS2          PIC X(20).                            <-ALL-FMTS
   47 +000012           06 BIRTHDATE         PIC X(6).                             <-ALL-FMTS
   48 +000013           06 MARSTAT           PIC X(1).                             <-ALL-FMTS
   49 +000014           06 SPOUSENAME        PIC X(30).                            <-ALL-FMTS
   50 +000015           06 NUMCHILD          PIC S9(2).                            <-ALL-FMTS
   51  002900 WORKING-STORAGE SECTION.
   52  003000 77  HEAD-ORDER                  PIC X(15)
   53  003100                                  VALUE "EMPLOYEE NUMBER".
   54  003200 01  PERSREPT-INDICS.
   55  003300     COPY DDS-ALL-FORMATS-O-INDIC OF PERSREPT.  4
```

Figura 75 (Parte 1 de 2). Ejemplo de la Condición END-OF-PAGE

```
5763CB1 V3R0M5                  Fuente COBOL AS/400
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
   56 +000001       05  PERSREPT-RECORD.                                                  <-ALL-FMTS
      +000002* FORMATO SALIDA:HEADING    DESDE ARCHIVO PERSREPT   DE BIBLIOT. XMPLIB      <-ALL-FMTS
      +000003*                                                                           <-ALL-FMTS
      +000004*          06 HEADING-O-INDIC.                                              <-ALL-FMTS
      +000005* FORMATO SALIDA:DETAIL    DESDE ARCHIVO PERSREPT   DE BIBLIOT. XMPLIB      <-ALL-FMTS
      +000006*                                                                           <-ALL-FMTS
   57 +000007           06 DETAIL-O-INDIC.                                               <-ALL-FMTS
   58 +000008                07 IN01         PIC 1  INDIC 01.                          <-ALL-FMTS
       003400
   59  003500 77  EOF-FLAG                     PIC X(1)
   60  003600                                  VALUE "0".
   61  003700    88  NOT-END-OF-FILE           VALUE "0".
   62  003800    88  END-OF-FILE               VALUE "1".
   63  003900 77  MARRIED                      PIC X(1)
   64  004000                                  VALUE "M".
       004100
   65  004200 PROCEDURE DIVISION.
       004300 FIRST-SECT SECTION.
       004400 FIRST-PARA.
   66  004500     OPEN INPUT PERSFILE
       004600          OUTPUT PERSREPT.
   67  004700     PERFORM HEADING-LINE.
   68  004800     PERFORM PROCESS-RECORD UNTIL END-OF-FILE.
   69  004900     CLOSE  PERSFILE
       005000            PERSREPT.
   70  005100     STOP RUN.
       005200
       005300 PROCESS-RECORD.
   71  005400     READ PERSFILE AT END SET END-OF-FILE TO TRUE.
   73  005500     IF NOT-END-OF-FILE THEN
   74  005600         PERFORM PRINT-RECORD. 5
       005700
       005800 PRINT-RECORD.
   75  005900     MOVE CORR PERSREC TO DETAIL-O. 6
   76  006000     IF MARSTAT IN PERSFILE-REC IS EQUAL MARRIED THEN 7
   77  006100         MOVE B"1" TO IN01 IN DETAIL-O-INDIC
       006200     ELSE
   78  006300         MOVE B"0" TO IN01 IN DETAIL-O-INDIC.
   79  006400     WRITE PERSREPT-REC FORMAT IS "DETAIL" 8
       006500         INDICATORS ARE DETAIL-O-INDIC
   80  006600           AT EOP PERFORM HEADING-LINE. 9
       006700 HEADING-LINE.
   81  006800         MOVE HEAD-ORDER TO ORDERTYPE
   82  006900         WRITE PERSREPT-REC FORMAT IS "HEADING".
       007000
                        * * * * *  F I N  D E  F U E N T E  * * * * *
```

Figura 75 (Parte 2 de 2). Ejemplo de la Condición END-OF-PAGE

1 El archivo de impresora descrito externamente se asigna al dispositivo FORMATFILE.

2 La instrucción COPY de Formato 2 se utiliza para copiar los campos para el archivo de impresora en el programa.

3 Tenga en cuenta que, a pesar de que los campos del formato DETAIL se imprimirán en tres líneas separadas, están definidos en un registro.

4 COPY-DDS se utiliza para copiar los indicadores utilizados en el archivo de impresora del programa.

5 El párrafo PROCESS-RECORD procesa PRINT-RECORD para cada registro de empleado.

6 Todos los campos en el registro de empleado se trasladan al registro para el formato DETAIL.

7 Si el empleado está casado, se activa el indicador 01; si no, el indicador se desactiva, evitando que se imprima el campo de nombre de la esposa en DETAIL.

8 Se imprime el formato DETAIL con el indicador 01, transferido para controlar la impresión.

9 Si se ha excedido el número de líneas por página, se produce END-OF-PAGE. El formato HEADING se imprime en la página siguiente.

IBM International Business Machines

AS/400 ESPECIFICACIONES DE DESCRIPCION DE DATOS

GX21-9891-0 UM/050*
Impreso en España
*El número de hojas por área de relleno puede variar ligeramente.

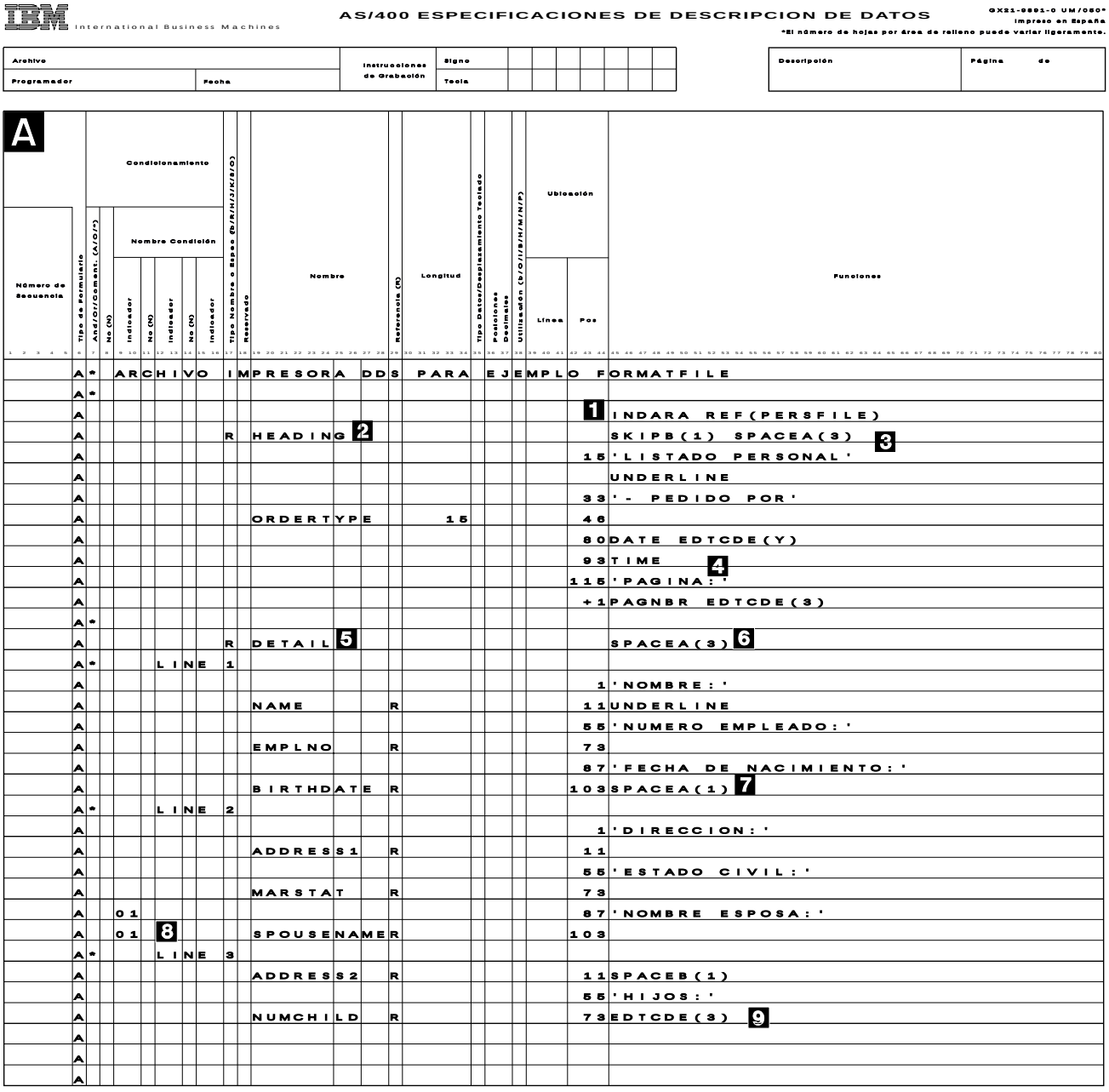
Archivo		Instrucciones de Grabación	Signo
Programador	Fecha		Tecla

Descripción	Página de

A

Número de Secuencia (1-5)	Tipo de Formulario (6)	And/Or/Coment. (A/O/*) (7)	No (N) (8)	Indicador (9-10)	No (N) (11)	Indicador (12-13)	No (N) (14)	Indicador (15-16)	Tipo Nombre o Espec (b/R/H/J/K/S/O) (17)	Reservado (18)	Nombre (19-28)	Referencia (R) (29)	Longitud (30-34)	Tipo Datos/Desplazamiento Teclado (35)	Posiciones Decimales (36-37)	Utilización (b/O/I/B/H/M/N/P) (38)	Ubicación Línea (39-41)	Ubicación Pos (42-44)	Funciones (45-80)
	A	*		ARCHIVO FISICO DDS PARA ARCHIVO PERSONAL EN EJEMPLO FORMATFILE															
	A																		
	A	*							R		PERSREC								
	A										EMPLNO		6	S					
	A										NAME		30						
	A										ADDRESS1		35						
	A										ADDRESS2		20						
	A										BIRTHDATE		6						
	A										MARSTAT		1						
	A										SPOUSENAME		30						
	A										NUMCHILD		2	S					
	A								K		EMPLNO								
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		

Figura 76 (Parte 1 de 2). Ejemplo de DDS de la Utilización de Archivos de Impresora Descritos Externamente Asignados a un Dispositivo de FORMATFILE



AS/400 ESPECIFICACIONES DE DESCRIPCION DE DATOS

GX21-9891-0 UM/080* Impreso en España

*El número de hojas por área de relleno puede variar ligeramente.

Archivo | Instrucciones de Grabación | Signo | Descripción | Página de

Programador | Fecha | Teclа

A

Número de Secuencia	Tipo de Formulario	And/Or/Comment: (A/O/*)	Condicionamiento: No (N)	Nombre Condición: Indicador	No (N)	Indicador	No (N)	Indicador	Tipo Nombre o Espac (b/R/H/3/K/8/O)	Reservado	Nombre	Referencia (R)	Longitud	Tipo Dato/Desplazamiento Teclado	Posiciones Decimales	Utilización (b/O/I/B/H/M/N/P)	Ubicación: Línea	Pos	Funciones
	A	*									ARCHIVO IMPRESORA DDS PARA EJEMPLO FORMATFILE								
	A	*																	
	A																1		INDARA REF(PERSFILE)
	A								R		HEADING 2								SKIPB(1) SPACEA(3) 3
	A																15		'LISTADO PERSONAL'
	A																		UNDERLINE
	A																33		'- PEDIDO POR'
	A										ORDERTYPE		15				46		
	A																80		DATE EDTCDE(Y)
	A																93		TIME 4
	A																115		'PAGINA:'
	A																+1		PAGNBR EDTCDE(3)
	A	*																	
	A								R		DETAIL 5								SPACEA(3) 6
	A	*				LINE			1										
	A																1		'NOMBRE:'
	A										NAME	R					11		UNDERLINE
	A																55		'NUMERO EMPLEADO:'
	A										EMPLNO	R					73		
	A																87		'FECHA DE NACIMIENTO:'
	A										BIRTHDATE	R					103		SPACEA(1) 7
	A	*				LINE			2										
	A																1		'DIRECCION:'
	A										ADDRESS1	R					11		
	A																55		'ESTADO CIVIL:'
	A										MARSTAT	R					73		
	A			01													87		'NOMBRE ESPOSA:'
	A			01		8					SPOUSENAME	R					103		
	A	*				LINE			3										
	A										ADDRESS2	R					11		SPACEB(1)
	A																55		'HIJOS:'
	A										NUMCHILD	R					73		EDTCDE(3) 9
	A																		
	A																		
	A																		

Figura 76 (Parte 2 de 2). Ejemplo de DDS de la Utilización de Archivos de Impresora Descritos Externamente Asignados a un Dispositivo de FORMATFILE

1 INDARA especifica que el archivo utilizará un área de indicadores separada.

2 HEADING es el nombre de formato que proporciona encabezamientos para cada página.

3 SKIPB(1) y SPACEA(3) se utilizan para:

1. Saltar a la línea 1 de la página siguiente antes de que se imprima el formato HEADING.
2. Dejar 3 líneas en blanco después de que imprima el formato HEADING.

4 DATE, TIME y PAGNBR se utilizan para imprimir automáticamente la fecha, hora y número de página actual cuando se imprime el formato HEADING.

5 DETAIL es el nombre de formato que se utiliza para imprimir la línea de detalle para cada empleado en el archivo de personal.

6 SPACEA(3) hace que se dejen tres líneas en blanco a la izquierda después de cada línea de detalle de empleado.

7 SPACEA(1) hace que se imprima una línea en blanco después de que imprima el campo BIRTHDATE. Como resultado, los campos subsiguientes en el mismo formato se imprimen en una línea nueva.

8 01 significa que estos campos sólo se imprimen si el programa COBOL activa el indicador 01 y lo transfiere cuando se imprime el formato DETAIL.

9 EDTCDE(3) se utiliza para eliminar los ceros a la izquierda cuando se imprime este campo numérico.

Capítulo 10. Archivos DISK y DATABASE

Los archivos de base de datos, que se asocian con los dispositivos COBOL de DATABASE y DISK, pueden ser:

- Archivos descritos externamente, cuyos campos se describen a OS/400 mediante las DDS
- Archivos descritos por el programa, cuyos campos se describen en el programa que utiliza el archivo.

Todos los archivos de base de datos se crean mediante los mandatos Crear Archivo de OS/400. Consulte el manual *Guía para la Base de Datos* para obtener una descripción sobre el mandato Crear Archivo para archivos de bases de datos.

Archivos DATABASE frente a Archivo DISK

La asignación de un archivo a DISK en COBOL limita al usuario al proceso tradicional DISK. La utilización de DATABASE como dispositivo permite que el usuario utilice las características especiales de COBOL/400, como por ejemplo formatos y claves de registro duplicadas.

Métodos de Proceso para Archivos DISK y DATABASE

Archivos Indexados COBOL

Un archivo indexado es un archivo cuya vía de acceso se ha creado mediante valores de claves. El usuario debe crear una vía de acceso por clave para un archivo indexado utilizando las DDS.

Para escribir programas en ANSI X3.23-1985 COBOL estándar que acceden a un archivo indexado, es preciso crear un archivo con una serie de características determinadas. La tabla siguiente lista estas características así como los elementos que las controlan:

Característica	**Control**
El archivo debe ser un archivo físico.	El mandato CL CRTPF
El archivo no puede tener registros con valores de clave duplicados.	La palabra clave UNIQUE de las DDS
El archivo no puede ser un archivo compartido	El mandato CL CRTPF
Debe definirse una clave para el archivo.	DDS
Las claves deben estar en secuencia ascendente.	DDS
Las claves han de ser contiguas dentro del registro.	DDS
Los campos de clave deben ser alfanuméricos. No pueden ser sólo numéricos.	DDS
El valor de la clave utilizada para poner en secuencia debe incluir 8 bits de cada byte.	DDS
No puede especificarse una posición inicial para recuperar registros.	El mandato CL OVRDBF

Característica	Control
No pueden utilizarse para el archivo palabras clave a nivel de seleccionar/omitir.	DDS

Un archivo indexado se identifica mediante la cláusula ORGANIZATION IS INDEXED de la instrucción SELECT.

Los campos de clave identifican los registros en un archivo indexado. El usuario especifica el campo de clave en la cláusula RECORD KEY de la instrucción SELECT. El ítem de datos RECORD KEY debe definirse dentro de una descripción de registro para el archivo indexado. Si hay múltiples descripciones de registro para el archivo, sólo una necesita contener el nombre de datos RECORD KEY. No obstante, se accede a las mismas posiciones en la descripción de registro que contienen el ítem de datos RECORD KEY en las otras descripciones de registro como el valor KEY para cualquier referencia a otras descripciones para ese archivo.

Se puede accederse a un archivo indexado de forma secuencial dinámicamente o al azar mediante una clave.

RECORD KEYS Válidas

Las DDS para el archivo especifican los campos a utilizar como campo de clave. Si el archivo posee campos de clave múltiples, el campo de clave debe ser contiguo en cada registro a no ser que se especifique RECORD KEY IS EXTERNALLY-DESCRIBED-KEY.

Cuando las DDS especifican sólo un campo clave para el archivo, RECORD KEY debe ser un sólo campo con la misma longitud que el campo clave definido en las DDS.

Si una instrucción COPY de Formato 2 se especifica para el archivo, la cláusula RECORD KEY debe especificar una de las siguientes condiciones:

- El nombre utilizado en las DDS para el campo de clave si el nombre no es una palabra reservada de COBOL.
- El nombre utilizado en las DDS para el campo de clave añadiendo -DDS al final si el nombre es una palabra reservada de COBOL.
- El nombre de datos definido en una descripción de registro descrita por programa para el archivo, con la longitud apropiada y en la ubicación correcta.
- EXTERNALLY-DESCRIBED-KEY. Esta palabra clave especifica que las claves definidas en DDS para cada formato de registro deben utilizarse para acceder al archivo. No es preciso que estas claves sean contiguas. Pueden definirse en distintas posiciones dentro del formato de registro.

Cuando las DDS especifican múltiples campos clave contiguos, el nombre de clave RECORD KEY debe ser un campo único cuya longitud sea igual a la suma de las longitudes de múltiples campos de clave en las DDS. Si se especifica para el archivo una instrucción COPY de Formato 2, también debe haber una descripción de registro descrita por programa para el archivo que define el nombre de datos RECORD KEY con la longitud y ubicación adecuada en el registro.

Los **Ítems contiguos** son ítems de grupo o elementales consecutivos en la División de Datos que forman parte de una jerarquía de datos única.

Referencia a una Clave Parcial

Una instrucción START genérica permite utilizar una clave parcial. Se necesita la frase KEY IS.

Consulte el apartado “Instrucción START” de la publicación *COBOL/400 Reference* para obtener más información acerca de las normas necesarias para especificar un argumento de búsqueda que haga referencia a una clave parcial.

La Figura 77 en la página 254 muestra un ejemplo de instrucciones START genéricas que utilizan un archivo descrito por el programa.

```
5763CB1 V3R0M5                  Fuente COBOL AS/400
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
    7  000700 FILE-CONTROL.
    8  000800     SELECT FILE-1 ASSIGN TO DISK-FILE1
    9  000900     ACCESS IS DYNAMIC RECORD KEY IS FULL-NAME IN FILE-1
   10  001000     ORGANIZATION IS INDEXED.
   11  001100 DATA DIVISION.
   12  001200 FILE SECTION.
   13  001300 FD  FILE-1 LABEL RECORDS ARE STANDARD.
   14  001400 01  RECORD-DESCRIPTION.
   15  001500     03 FULL-NAME.
   16  001600        05 LAST-AND-FIRST-NAMES.
   17  001700           07 LAST-NAME             PIC X(20).
   18  001800           07 FIRST-NAME            PIC X(20).
   19  001900        05 MIDDLE-NAME               PIC X(20).
   20  002000     03 LAST-FIRST-MIDDLE-INITIAL-NAME REDEFINES FULL-NAME
   21  002100                                     PIC X(41).
   22  002200     03 REST-OF-RECORD
       002300/
   23  002400 PROCEDURE DIVISION.
       002500 START-PROGRAM.
   24  002600     OPEN INPUT FILE-1.
       002700*
       002800* SITÚA EL ARCHIVO COMENZANDO CON REGISTROS QUE TENGAN EL APELLIDO
       002900* "SMITH"
   25  003000     MOVE "SMITH" TO LAST-NAME.
   26  003100     START FILE-1 KEY IS EQUAL TO LAST-NAME
   27  003200           INVALID KEY DISPLAY "NO DATA IN SYSTEM FOR " LAST-NAME
   28  003300                       GO-TO ERROR ROUTINE.
       003400*
       003500*
       003600*
       003700*
       003800* SITÚA EL ARCHIVO COMENZANDO CON REGISTROS QUE TENGAN EL APELLIDO
       003900* "SMITH" Y EL NOMBRE DE "ROBERT"
   29  004000     MOVE "SMITH" TO LAST-NAME.
   30  004100     MOVE "ROBERT" TO FIRST-NAME.
   31  004200     START FILE-1 KEY IS EQUAL TO LAST-AND-FIRST-NAMES
   32  004300           INVALID KEY DISPLAY "NO DATA IN SYSTEM FOR "
       004400                               LAST-AND-FIRST-NAMES
   33  004500                       GO-TO ERROR ROUTINE.
       004600*
       004700*
       004800*
       004900*
       005000* SITÚA EL ARCHIVO COMENZANDO CON REGISTROS QUE TENGAN EL APELLIDO
       005100* "SMITH", EL NOMBRE "ROBERT", Y LA INICIAL DEL SEGUNDO NOMBRE SEA "M"
   34  005200     MOVE "SMITH" TO LAST-NAME.
   35  005300     MOVE "ROBERT" TO FIRST-NAME.
   36  005400     MOVE "M" TO MIDDLE-NAME.
   37  005500     START FILE-1 KEY IS EQUAL TO LAST-AND-FIRST-MIDDLE-INITIAL-NAME
   38  005600           INVALID KEY DISPLAY "NO DATA IN SYSTEM FOR "
       005700                               LAST-FIRST-MIDDLE-INITIAL-NAME
   39  005800                       GO-TO ERROR ROUTINE.
       005900
       006000
       006100 ERROR-ROUTINE.
   40  006200     STOP-RUN.
```

Figura 77. Instrucciones START que utilizan un archivo descrito por el programa

La Figura 78 y la Figura 79 muestran un ejemplo de instrucciones START genéricas que utilizan un archivo descrito externamente.

```
                                FUENTE DE DESCRIPCION DE DATOS
     NUMSEC *... ... 1 ... ... 2 ... ... 3 ... ... 4 ... ... 5 ... ... 6 ... ... 7 ... ... 8  FECH
        100      A                                    UNIQUE
        200      A          R RDE                     TEXT('DESCRIPCIÓN DE REGISTRO')
        300      A            FNAME         20         TEXT('NOMBRE')
        400      A            MINAME         1         TEXT('INICIAL/SEGUNDO NOMBRE')
        500      A            MNAME         19         TEXT('RESTO DEL SEGUNDO NOMBRE')
        600      A            LNAME         20         TEXT('APELLIDO')
        700      A            PHONE         10  0      TEXT('NÚMERO DE TELÉFONO')
        800      A            DATA          40         TEXT('DATOS RESTANTES')
        900      A          K LNAME
       1000      A          K FNAME
       1100      A          K MINAME
       1200      A          K MNAME
```

Figura 78. Instrucciones START Genéricas que Utilizan un archivo Descrito Externamente -- DDS

```
5763CB1 V3R0M5                  Fuente COBOL AS/400
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
    7  000700 FILE-CONTROL.
    8  000800     SELECT FILE-1 ASSIGN TO DATABASE-NAMES
    9  000900     ACCESS IS DYNAMIC RECORD KEY IS EXTERNALLY-DESCRIBED-KEY
   10  001000     ORGANIZATION IS INDEXED.
   11  001100 DATA DIVISION.
   12  001200 FILE SECTION.
   13  001300 FD  FILE-1 LABEL RECORDS ARE STANDARD.
   14  001400 01  RECORD-DESCRIPTION
   15  001500     COPY DDS-RDE IN NAMES-PUBS.
   17 +000001                                                                 RDE
      +000002*    DESDE FILE NAMES      DE BIBLIOTECA XMPLIB                   RDE
      +000003*    DESCRIPCIÓN DE REGISTRO                                      RDE
   18 +000004       05  RDE.                                                  RDE
      +000005*    CLAVE DE REG. PARA ARCH. INDEX. CLAVE'0002 NOM. CAMPO CLAVE FNAME   .    RDE
   19 +000006           06 FNAME             PIC X(20).                       RDE
      +000007*    NOMBRE                                                       RDE
      +000008*    CLAVE ARCH. PARA ARCH. INDEX., CLAVE'0003 NOMB. CAMPO CLAVE MINAME  .    RDE
   20 +000009           06 MINAME            PIC X(1).                        RDE
      +000010*    INICIAL SEGUNDO NOMBRE                                       RDE
      +000011*    CLAVE ARCH. PARA ARCH. INDEX., CLAVE'0004 NOMB. CAMPO CLAVE MNAME   .    RDE
   21 +000012           06 MNAME             PIC X(19).                       RDE
      +000013*    RESTO SEGUNDO NOMBRE                                         RDE
      +000014*    CLAVE ARCH. PARA ARCH. INDEX., CLAVE'0001 NOMB. CAMPO CLAVE LNAME   .    RDE
   22 +000015           06 LNAME             PIC X(20).                       RDE
      +000016*    APELLIDO                                                     RDE
   23 +000017           06 PHONE             PIC S9(10).      COMP-3          RDE
      +000018*    NÚMERO TELÉFONO                                              RDE
   24 +000019           06 DATA-DDS          PIC X(40).                       RDE
      +000020*    DATOS RESTANTES                                              RDE
   25  001600 66  MIDDLE-NAME RENAMES MINAME THRU MNAME.
       001700/
   26  001800 PROCEDURE DIVISION.
       001900 START PROGRAM.
   27  002000     OPEN INPUT FILE-1.
       002100*
       002200* SITÚA EL ARCHIVO COMENZANDO CON LOS REGISTROS QUE TENGAN EL APELLIDO
       002300*  "SMITH"
   28  002400     MOVE "SMITH" TO LNAME.
   29  002500     START FILE-1 KEY IS EQUAL TO LNAME
   30  002600           INVALID KEY DISPLAY "NO DATA IN SYSTEM FOR " LNAME
   31  002700                       GO TO ERROR-ROUTINE.
       002800*          .
       002900*          .
       003000*          .
       003100*
       003200* SITÚA EL ARCHIVO COMENZANDO CON LOS REGISTROS QUE TENGAN EL APELLIDO
       003300* "SMITH" Y EL NOMBRE "ROBERT"
   32  003400     MOVE "SMITH" TO LNAME.
   33  003500     MOVE "ROBERT" TO FNAME.
   34  003600     START FILE-1 KEY IS EQUAL TO LNAME, FNAME
   35  003700           INVALID KEY DISPLAY "NO DATA IN SYSTEM FOR "
       003800                               LNAME " " FNAME
   36  003900                       GO TO ERROR-ROUTINE.
       004000*          .
       004100*          .
       004200*          .
       004300*
       004400* SITÚA EL ARCHIVO COMENZANDO CON REGISTROS QUE TENGAN EL APELLIDO
       004500* "SMITH", EL NOMBRE "ROBERT", Y LA INICIAL DEL SEGUNDO NOMBRE SEA "M"
   32  004600     MOVE "SMITH" TO LNAME.
   33  004700     MOVE "ROBERT" TO FNAME.
   33  004800     MOVE "M" TO MINAME.
   34  004900     START FILE-1 KEY IS EQUAL TO LNAME, FNAME, MINAME
   35  005000           INVALID KEY DISPLAY "NO DATA IN SYSTEM FOR "
       005100                               LNAME SPACE FNAME SPACE MINAME
   42  005200                       GO TO ERROR-ROUTINE.
       005300
       005400
       005500 ERROR-ROUTINE.
       005600     STOP-RUN.
```

Figura 79. Instrucciones START Genéricas que Utilizan un Archivo Descrito Externamente

Consideraciones Sobre Archivos Lógicos

Cuando un archivo lógico con múltiples formatos de registro, cada uno con campos de clave asociados, se procesa como archivo indexado en COBOL, se aplican las restricciones y consideraciones siguientes:

- La frase FORMAT debe especificarse en todas las instrucciones WRITE para el archivo a menos que exista el Programa Selector de Formato de Registro y se haya especificado en el parámetro FMTSLR del mandato Crear Archivo Lógico (CRTLF), el mandato Cambiar Archivo Lógico (CHGLF) o el mandato Alterar Temporalmente Archivo de Base de Datos (OVRDBF). Para obtener más información acerca de la utilización de programas selectores de formato, consulte la publicación *Guía para la Base de Datos*.
- Si la modalidad de acceso es RANDOM o DYNAMIC, y no se especifica para el archivo la frase DUPLICATES, la frase FORMAT debe especificarse en todas las instrucciones DELETE y REWRITE.
- Cuando no se especifica la frase FORMAT, el sistema utiliza solamente la parte del ítem de datos RECORD KEY que es común a todos los formatos de registro del archivo como la clave para la instrucción de E/S. Cuando se especifica la frase FORMAT, el sistema utiliza como clave sólo la parte del ítem de datos RECORD KEY que se define para el formato de registro especificado. Consulte la publicación *Guía para la Base de Datos* para obtener más información sobre el proceso de archivos lógicos.
- Cuando se especifica *NONE como el primer campo de clave para cualquier formato en un archivo, sólo se puede acceder a los registros secuencialmente. Cuando un archivo se lee al azar:
  - Si se especifica un nombre de formato, se devuelve el primer registro con el formato especificado.
  - Si no se especifica un nombre de formato, se devuelve el primer registro del archivo.

  En ambos casos, se hace caso omiso del ítem de datos RECORD KEY.
- Para un campo de clave definido por programa:
  - Los campos de clave dentro de cada formato de registro deben ser contiguos.
  - El primer campo de clave para cada formato de registro debe comenzar en la misma posición relativa dentro de cada registro.
  - La longitud del ítem de datos RECORD KEY debe ser igual a la longitud de la clave más larga de cualquier formato del archivo.
- Para una EXTERNALLY-DESCRIBED-KEY:
  - Los campos de clave dentro de cada formato de registro pueden no ser contiguos.
  - Los campos de clave pueden comenzar en posiciones distintas en cada formato de registro.

La Figura 80 en la página 258 y la Figura 81 en la página 259 muestran ejemplos de cómo utilizar las DDS para describir la vía de acceso para archivos indexados.

IBM International Business Machines

AS/400 ESPECIFICACIONES DE DESCRIPCION DE DATOS

GX21-9891-0 UM/050*
Impreso en España
*El número de hojas por área de relleno puede variar ligeramente.

Archivo		Instrucciones de Grabación	Signo	Descripción	Página de
Programador	Fecha		Tecla		

A

Número de Secuencia (1-5)	Tipo de Formulario (6)	And/Or/Coment. (A/O/*) (7)	No (N) (8)	Indicador (9-10)	No (N) (11)	Indicador (12-13)	No (N) (14)	Indicador (15-16)	Tipo Nombre o Espec (b/R/H/J/K/S/O) (17)	Reservado (18)	Nombre (19-28)	Referencia (R) (29)	Longitud (30-34)	Tipo Datos/Desplazamiento Teclado (35)	Posiciones Decimales (36-37)	Utilización (b/O/I/B/H/M/N/P) (38)	Ubicación Línea (39-41)	Ubicación Pos (42-44)	Funciones (45-80)
	A								R		FORMATA								PFILE(ORDDTLP)
	A																		TEXT('VIA ACCESO PARA ARC INDEXADO')
	A										FLDA		14						
	A										ORDERN		5	S	0				
	A										FLDB		101						
	A								K		ORDERN								
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		

Figura 80. Utilización de Especificaciones de Descripción de Datos para Definir la Vía de Acceso de un Archivo

Las especificaciones de descripción de datos deben utilizarse para crear la vía de acceso para un archivo indexado descrito por programa.

En las DDS para el formato de registro FORMATA del archivo lógico ORDDTLL, el campo ORDERN, que tiene una longitud de cinco dígitos, se define como el campo de clave. La definición de ORDERN como campo de clave establece el acceso por clave para este archivo. Otros dos campos, FLDA y FLDB, describen las posiciones restantes en este registro como campos de tipo carácter.

El campo de entrada descrito por programa ORDDTLL se describe en la sección FILE-CONTROL de la cláusula SELECT como un archivo indexado.

Las descripciones COBOL de cada campo en la entrada FD deben coincidir con la descripción correspondiente en el archivo DDS. El ítem de datos RECORD KEY debe definirse como entero numérico de cinco dígitos empezando en la posición 15 del registro.

IBM International Business Machines

AS/400 ESPECIFICACIONES DE DESCRIPCION DE DATOS

GX21-9891-0 UM/050*
Impreso en España
*El número de hojas por área de relleno puede variar ligeramente.

Archivo		Instrucciones de Grabación	Signo	Descripción	Página de
Programador	Fecha		Tecla		

A

Número de Secuencia (1-5)	Tipo de Formulario (6)	And/Or/Coment. (A/O/*) (7)	No (N) (8)	Indicador (9-10)	No (N) (11)	Indicador (12-13)	No (N) (14)	Indicador (15-16)	Tipo Nombre o Espec (b/R/H/J/K/S/O) (17)	Reservado (18)	Nombre (19-28)	Referencia (R) (29)	Longitud (30-34)	Tipo Datos/Desplazamiento Teclado (35)	Posiciones Decimales (36-37)	Utilización (b/O/I/B/H/M/N/P) (38)	Ubicación Línea (39-41)	Ubicación Pos (42-44)	Funciones (45-80)
	A								R		FORMAT								PFILE(ORDDTLP)
	A																		TEXT('VIA ACCESO PARA ARC INDEXADO')
	A										FLDA		14						
	A										ORDERN		5	S	0				
	A										ITEM		5						
	A										FLDB		96						
	A								K		ORDERN								
	A								K		ITEM								
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		
	A																		

Figura 81. Especificaciones de Descripción de Datos para la Definición de la Vía de acceso (una Clave Compuesta) de un Archivo Indexado

En este ejemplo, las especificaciones de descripción de datos definen dos campos de clave para el formato de registro FORMAT en el archivo lógico ORDDTLL. Para los dos campos a utilizar como clave compuesta para un archivo indexado descrito por programa, los campos de clave deben ser contiguos en el registro.

La descripción COBOL de cada campo debe coincidir con la descripción correspondiente en el archivo DDS. En la cláusula RECORD-KEY de la entrada de

control de archivo debe definirse un ítem de diez caracteres que empiece en la posición 15 del registro. Las descripciones COBOL de los campos DDS ORDERN e ITEM deberían subordinarse al ítem de 10 caracteres en la cláusula RECORD KEY.

Archivos Relativos COBOL

Un archivo relativo COBOL es un archivo que se va a procesar mediante un número relativo de registro. Para procesar un archivo mediante un número relativo de registro, se debe especificar ORGANIZATION IS RELATIVE en la instrucción SELECT para el archivo. Se puede acceder a un archivo relativo de forma secuencial, al azar mediante un número de registro o dinámicamente.

Para escribir programas en COBOL ANSI X3.23-1985 estándar que accedan a un archivo relativo, es preciso crear el archivo con unas características determinadas. En la siguiente tabla se listan estas características así como los elementos que las controlan.

Característica	**Control**
El archivo debe ser un archivo físico.	El mandato CL CRTPF
El archivo no puede ser un archivo compartido	El mandato CL CRTPF
No puede especificarse ninguna clave en el archivo.	DDS
No puede especificarse una posición inicial para recuperar registros.	El mandato CL OVRDBF
No pueden utilizarse para el archivo palabras clave a nivel de seleccionar/omitir.	DDS
No se pueden volver a utilizar los registros del archivo.	El mandato CL CRTPF

Para un archivo COBOL con una organización RELATIVE, el mandato CL Reorganizar Miembro de Archivo Físico (RGZPFM) puede:

- Eliminar todos los registros suprimidos del archivo. Debido a que COBOL inicializa todos los registros del archivo relativo a registros suprimidos, se eliminará del archivo cualquier registro que no se haya grabado explícitamente. Los números relativos de registro de todos los registros que estén después del primer registro suprimido en el archivo se cambiarán.
- Cambiar los números relativos de registro si el archivo tiene una clave y se modifica la secuencia de llegada para que coincida con una secuencia de clave (con el parámetro KEYFILE).

Además, el mandato CL Cambiar Archivo Físico (CHGPF) que da soporte la opción REUSEDLT, puede modificar el orden de los registros recuperados o grabados cuando el archivo trabaja de forma secuencial, ya que permite volver a utilizar los registros eliminados.

Archivos Secuenciales COBOL

Un archivo secuencial COBOL es un archivo en el que se procesan registros en el mismo orden en que han sido colocados en el archivo, es decir, en secuencia de llegada. Por ejemplo, el décimo registro colocado en el archivo ocupa la décima posición de registro y será el décimo registro que se procese. Para procesar un archivo como secuencial, es preciso especificar ORGANIZATION IS SEQUENTIAL en la cláusula SELECT o hacer caso omiso de la cláusula ORGANIZATION. A un archivo secuencial sólo puede accederse secuencialmente.

Para escribir programas en COBOL ANSI X3.23-1985 estándar que accedan a un archivo secuencial, es preciso crear dicho archivo a partir de unas características determinadas. En la siguiente tabla se listan estas características así como los elementos que las controlan.

Característica	**Control**
El archivo debe ser un archivo físico.	El mandato CL CRTPF
El archivo no puede ser un archivo compartido	El mandato CL CRTPF
No puede especificarse ninguna clave en el archivo.	DDS
El archivo debe tener un tipo de archivo de DATA.	El mandato CL CRTPF
No puede utilizarse la edición de campo.	DDS
No puede especificarse información de línea ni posición.	DDS
No pueden especificarse palabras clave de espaciar ni saltar.	DDS
No pueden utilizarse los indicadores.	DDS
No pueden utilizarse funciones suministradas por el sistema, tales como fecha, hora y número de página.	DDS
No pueden utilizarse para el archivo palabras clave a nivel de seleccionar/omitir.	DDS
No se pueden volver a utilizar los registros del archivo.	El mandato CL CRTPF

Para conservar la secuencia de registros de un archivo que se abre en modalidad de E/S (actualización), no modifique el archivo, ya que de lo contrario no podrá volver a utilizar los registros que contenga. Es decir, no utilice el parámetro CL Cambiar Archivo Físico (CHGPF) que da soporte a la opción REUSEDLT.

Nota: El compilador COBOL/400 no comprueba si el dispositivo asociado con el archivo externo es del tipo especificado en la parte de dispositivo del nombre de asignación. El dispositivo especificado en el nombre de asignación debe coincidir con el dispositivo real al que se asigna el archivo. Consulte la sección “Cláusula ASSIGN” de la publicación *COBOL/400 Reference* para obtener más información.

Consideraciones sobre la Organización de Archivos COBOL y la Vía de Acceso del Archivo AS/400

EN COBOL, puede procesarse un archivo con una vía de acceso por secuencia de claves como un archivo con una organización INDEXED, RELATIVE o SEQUENTIAL.

Para procesar un archivo en secuencia por claves como un archivo relativo COBOL, debe ser un archivo físico, o un archivo lógico cuyos miembros se basen en un miembro del archivo físico. Para procesar un archivo en secuencia de claves como un archivo secuencial en COBOL, debe ser un archivo físico, o un archivo lógico que se base en un miembro del archivo físico y que no contenga lógica de selección/omisión.

Una archivo con una vía de acceso en secuencia de llegada puede procesarse en COBOL como un archivo con organización RELATIVE o SEQUENTIAL. El archivo debe ser un archivo físico o lógico en el que cada miembro del archivo lógico únicamente esté basado en un solo miembro del archivo físico.

Cuando se especifica el acceso secuencial para un archivo lógico, se accede a los registros del archivo mediante la vía de acceso creada con las opciones de creación de archivos.

Métodos de Proceso de Archivos

La Figura 82 en la página 264 muestra los métodos de proceso válidos, así como la operación esperada para las combinaciones de organización, modalidad de acceso, estado de apertura, verbo de E/S y modificadores de verbo de E/S.

Se borran todos los archivos de base de datos físicos que se abren para OUTPUT. Los archivos de bases de datos con organización RELATIVE y con modalidad de acceso dinámica o al azar, también se inicializan con registros suprimidos.

Los archivos relativos nuevos que se han abierto para OUTPUT en modalidad de acceso secuencial se tratan de forma diferente. La Tabla 4 en la página 263 resume las condiciones que los afectan.

Tabla 4. Inicialización de Archivos Relativos de Salida			
Acceso a Archivos y Especificaciones CL	**Condiciones durante la Apertura**	**Condiciones durante el Cierre**	**Límite de Archivo**
Secuencial *INZDLT		Los registros no escritos se inicializan	Todos los incrementos
Secuencial *INZDLT Tamaño *NOMAX		CLOSE satisfactorio El estado de archivo es 0Q	Hasta el límite de registros escritos
Secuencial *NOINZDLT			Hasta el límite de registros escritos
Al azar o dinámicamente	Los registros se inicializan Se abre el archivo		Todos los incrementos
Al azar o dinámicamente Tamaño *NOMAX	OPEN no satisfactorio Estado de archivo 9Q		Archivo vacío

Para extender el límite de archivo más allá del número actual de registros, pero permaneciendo dentro del tamaño del archivo, utilice el mandato INZPFM para añadir registros eliminados antes de procesar el archivo. Es preciso efectuar esta operación si recibe un estado de archivo de 0Q y todavía quiere añadir más registros al archivo.

Cualquier intento de ampliar un archivo relativo más allá de su tamaño actual da como resultado una violación de límite.

Para recuperar el estado de archivo de 9Q, utilice el mandato CHGPF tal como se describe en el texto de mensaje en tiempo de ejecución asociado.

Las demoras de larga duración son normales cuando hay gran cantidad de registros (más de un 1 000 000) que se deben inicializar para registros suprimidos cuando se ejecuta la instrucción CLOSE.

Cuando la primera instrucción OPEN para el archivo no es OPEN OUTPUT, los archivos relativos deben borrarse e inicializarse con registros suprimidos antes de utilizarlos. Consulte la descripción sobre los mandatos CLRPFM y INZPFM en el manual *CL Reference* para obtener más información.

El parámetro RECORDS del mandato INZPFM debe especificar *DLT. Las alteraciones temporales deben aplicarse cuando COBOL procesa las operaciones de borrado e inicialización, pero no cuando se procesan con mandatos CL.

Las demoras de larga duración son normales en el proceso OPEN OUTPUT para archivos sumamente grandes (más de 1 000 000 registros) a los que se accede en modalidad dinámica o al azar.

ORG	ACC	DEV	OPEN	READ	WRITE	START	REWRITE	DELETE	CLOSE	FORMAT	SELECT CLAUSE KEY IS
S	S	ANY	INPUT	X					X		
S	S	ANY	OUTPUT		X(F1)				X	A1	
S	S	ANY	I-O	X			X		X		
S	S	ANY	EXTEND		X				X		
I	S	D/DB	INPUT	X		X			X	B1	C1
I	S	D/DB	OUTPUT		X(F1)				X	B1	C1
I	S	D/DB	I-O	X		X	X	X	X	B1	C1
I	R	D/DB	INPUT	X					X	B1	D1
I	R	D/DB	OUTPUT		X(F1)				X	B1	D1
I	R	D/DB	I-O	X	X		X	X	X	B1	D1
I	D	D/DB	INPUT	X		X			X	B1	D1
I	D	D/DB	OUTPUT		X(F1)				X	B1	D1
I	D	D/DB	I-O	X	X	X	X	X	X	B1	D1
R	S	D/DB	INPUT	X		X			X		C1
R	S	D/DB	OUTPUT		X(G1)				X		C1
R	S	D/DB	I-O	X		X	X	X	X		C1
R	R	D/DB	INPUT	X					X		E1
R	R	D/DB	OUTPUT		X(G1)				X		E1
R	R	D/DB	I-O	X	X		X	X	X		E1
R	D	D/DB	INPUT	X		X			X		E1
R	D	D/DB	OUTPUT		X(G1)				X		E1
R	D	D/DB	I-O	X	X	X	X	X	X		E1
T	S	W	I-O	X	X				X	H1	
T	D	W	I-O	X(K1)	X(K1)		X		X	I1	J1

ORG:

S = Secuencial
R = Relativo
I = Indexado
T = TRANSACTION

ACC:

S = Secuencial
R = Al azar
D = Dinámico

DEV:

ANY = Cualquier dispositivo
D = DISK
DB = DATABASE
W = WORKSTATION

Figura 82. Tabla Resumen de Métodos de Proceso

Los párrafos siguientes explican las claves utilizadas en la Figura 82.

X Se permite la combinación.

A1 Se requiere la frase FORMAT para archivos FORMATFILE con formatos múltiples y no se permite para todos los otros archivos de dispositivo.

B1 La frase FORMAT es opcional para archivos DATABASE y no se permite para archivos DISK. Si no se especifica la frase FORMAT, se utiliza el nombre de formato por omisión del archivo. El nombre de formato por omisión del archivo es el primer nombre de formato definido en el archivo.

Puede utilizarse el registro especial DB-FORMAT-NAME para recuperar el nombre de formato que se ha utilizado en la última operación satisfactoria de E/S.

C1 Se hace caso omiso de la frase KEY en la cláusula SELECT excepto en las instrucciones START. Si no se especifica la frase KEY en la instrucción START, la frase RECORD KEY o RELATIVE KEY de la cláusula SELECT se utiliza y entonces se asume KEY =.

D1 Se utiliza la frase KEY con la cláusula SELECT excepto para la instrucción START. Si no se especifica la frase KEY en la instrucción START, se utiliza la frase RECORD KEY en la cláusula SELECT y se asume KEY =.

NEXT, PRIOR, FIRST, o LAST sólo pueden especificarse en la instrucción READ para archivos DATABASE con acceso DYNAMIC. Si se especifica NEXT, PRIOR, FIRST o LAST, se hace caso omiso de la frase KEY de la cláusula SELECT.

E1 Se utiliza la frase RELATIVE KEY en la cláusula SELECT.

La frase NEXT sólo puede especificarse en la instrucción READ para un archivo con modalidad de acceso DYNAMIC. Si se especifica NEXT, se hace caso omiso de la frase KEY en la cláusula SELECT.

El ítem de datos RELATIVE KEY se actualiza con el número relativo de registro para archivos con acceso secuencial en operaciones READ.

F1 Se borra un archivo físico abierto para salida.

G1 Se borra un archivo físico abierto para salida y se inicializa a registros suprimidos. Existe una serie de excepciones según el tamaño del archivo y las opciones especificadas. Para obtener más información, consulte la Tabla 4 en la página 263.

H1 La frase FORMAT es necesaria para la instrucción WRITE.

I1 La frase FORMAT es necesaria para distinguir entre los registros del subarchivo y el registro de control del subarchivo. El nombre del formato de registro de control WRITE FORMAT IS visualiza el subarchivo, pero se necesita un nombre de formato de registro de control READ FORMAT IS para que los datos puedan entrarse y al mismo tiempo para que la entrada del operador de los registros del subarchivo se coloque en el subarchivo.

J1 Se utiliza la frase RELATIVE KEY en la cláusula SELECT para las instrucciones READ, WRITE y REWRITE que utilizan la frase SUBFILE, excepto cuando READ SUBFILE NEXT MODIFIED utilice el número relativo de registro actual del sistema en lugar del ítem de datos RELATIVE KEY. El ítem de datos RELATIVE KEY se actualiza con el número relativo de registro para registros del subarchivo en las instrucciones READ con la cláusula NEXT MODIFIED.

K1 Se requiere la frase SUBFILE cuando una operación de E/S trata con un registro particular en lugar de con un archivo completo.

Consideraciones sobre el Archivo Descendente

Los archivos creados con una secuencia en clave descendente (en DDS) provocan que las frases NEXT, PRIOR, FIRST y LAST de la instrucción READ se comporten de forma exactamente opuesta a la de un archivo con una secuencia de clave ascendente. En la **secuencia de clave descendente**, los datos se ubican a partir del valor más alto del campo de clave hasta el valor más bajo de dicho campo.

Por ejemplo, READ FIRST recupera el registro con el valor de campo más alto, mientras que READ LAST recupera el registro del valor de campo más bajo. Los archivos que tienen una secuencia de clave descendente también provocan que los calificadores START se comporten de forma contraria. Por ejemplo, START GREATER THAN coloca el puntero del registro actual en un registro con una clave menor que la clave actual.

Capítulo 11. Consideraciones de Programación en COBOL/400

Este capítulo describe:

- La emisión de un mandato CL desde un programa COBOL
- La frase CORRESPONDING
- La cláusula LIKE
- La modificación de referencias
- Cómo deshacer edición
- Consideraciones sobre el rendimiento.

Interfaz de Programación de Uso General

Emisión de un Mandato CL desde un Programa COBOL

El usuario podrá emitir un mandato CL desde un programa COBOL mediante una llamada (CALL) a QCMDEXC.

En el programa ejemplo siguiente, la instrucción CALL a QCMDEXC (en el número de secuencia 001600) da como resultado el proceso del mandato CL Añadir Entrada de Lista de Bibliotecas (ADDLIBLE) (en el número de secuencia 001100). La terminación satisfactoria del mandato CL da como resultado la adición de la biblioteca, COBOLTEST, a la lista de bibliotecas.

```
           -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..
000100      IDENTIFICATION DIVISION.
000200      PROGRAM-ID. CMDXMPLE.
000300      ENVIRONMENT DIVISION.
000400      CONFIGURATION SECTION.
000500        SOURCE-COMPUTER. IBM-AS400.
000600        OBJECT-COMPUTER. IBM-AS400.
000700      DATA DIVISION.
000800      WORKING-STORAGE SECTION.
000900      01  PROGRAM-VARIABLES.
001000          05  CL-CMD       PIC X(33)
001100                            VALUE "ADDLIBLE COBOLTEST".
001200          05  PACK-VAL     PIC 9(10)V9(5) COMP-3
001300                            VALUE 18.
001400      PROCEDURE DIVISION.
001500      MAINLINE.
001600          CALL "QCMDEXC" USING CL-CMD PACK-VAL.
001700          STOP RUN.
```

Nota: No utilice el mandato Reclamar Recurso (RCLRSC) en esta situación, ya que cancela todos los programas superiores en la pila de programas de forma que la instrucción STOP RUN en el programa causaría una excepción en tiempo de ejecución.

Para más información acerca de QCMDEXC, consulte la publicación *CL Guía del Programador*.

Fin de Interfaz de Programación de Uso General

Utilización de la Frase CORRESPONDING

En el programa ejemplo siguiente, la instrucción ADD CORRESPONDING en el número de secuencia 000270 añade GROUP1 ITEM1 al GROUP2 ITEM1 y GROUP1 ITEM2 al GROUP2 ITEM2. La instrucción MOVE CORRESPONDING en el número de secuencia 000290 mueve GROUP1 ITEM1, ITEM2, ITEM3 e ITEM4 al GROUP2 ITEM1, ITEM2, ITEM3 e ITEM4.

La instrucción MOVE CORRESPONDING en el número de secuencia 000300 no se procesa porque no hay ítems correspondientes que mover y se genera un mensaje de error.

La Figura 83 en la página 269 se ha producido con la opción PRTCORR activada.

```
5763CB1 V3R0M5                   Fuente COBOL AS/400             XMPLIB/CORR
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
    1  000010 IDENTIFICATION DIVISION.
    2  000020 PROGRAM-ID.     CORRPHRASE.
    3  000030   AUTHOR.       PROGRAMMER NAME.
    4  000040   INSTALLATION. TORONTO COBOL DEVELOPMENT CENTRE.
    5  000050   DATE-WRITTEN. 05/24/91.
    6  000060   DATE-COMPILED.  05/24/94 11:09:11   .
    7  000070 ENVIRONMENT DIVISION.
    8  000080 CONFIGURATION SECTION.
    9  000090 SOURCE-COMPUTER. IBM-AS400.
   10  000100 OBJECT-COMPUTER. IBM-AS400.
   11  000110 DATA DIVISION.
   12  000120 WORKING-STORAGE SECTION.
   13  000130 01  GROUP1.
   14  000140     05  ITEM1    PIC 99        VALUE 1.
   15  000150     05  ITEM2    PIC 99        VALUE 2.
   16  000160     05  ITEM3    PIC X(10)     VALUE "GREEN".
   17  000170     05  ITEM4    PIC X(10)     VALUE "BLUE".
   18  000180 01  GROUP2.
   19  000190     05  ITEM1    PIC 99        VALUE 8.
   20  000200     05  ITEM2    PIC 99        VALUE 9.
   21  000210     05  ITEM3    PIC XXBX(8)   VALUE SPACES.
   22  000220     05  ITEM4    PIC X(10)     VALUE SPACES.
   23  000230 01  GROUP3.
   24  000240     05  SPECIAL  PIC XX.
   25  000250 PROCEDURE DIVISION.
       000260 MAINLINE.
   26  000270     ADD CORRESPONDING GROUP1 TO GROUP2.
               *       ** Ítems CORRESPONDING para instrucción 26:
               *       **     ITEM1
               *       **     ITEM2
               *       ** Fin de ítems CORRESPONDING para instrucción 26
   27  000280     SUBTRACT CORRESPONDING GROUP1 FROM GROUP2.
               *       ** Ítems CORRESPONDING para instrucción 27:
               *       **     ITEM1
               *       **     ITEM2
               *       ** Fin de ítems CORRESPONDING para instrucción 27
   28  000290     MOVE CORRESPONDING GROUP1 TO GROUP2.
               *       ** Ítems CORRESPONDING para instrucción 28:
               *       **     ITEM1
               *       **     ITEM2
               *       **     ITEM3
               *       **     ITEM4
               *       ** Fin de ítems CORRESPONDING para instrucción 28
   29  000300     MOVE CORRESPONDING GROUP3 TO GROUP2.
               *       ** Ítems CORRESPONDING para instrucción 29:
               *       **     No se han encontrado ítems CORRESPONDING
               *       ** Fin de ítems CORRESPONDING para instrucción 29
   30  000310     STOP RUN.
                          * * * * *   F I N   D E   F U E N T E   * * * * *
5763CB1 V3R0M5                 Mensajes COBOL AS/400           XMPLIB/CORR
 INST
*   29  MSGID: LBL0336  GRAVEDAD: 10  NUMSEC:  000300
       Mensaje . . . . :   No se han encontrado ítems CORRESPONDING. Sentencia
         ignorada.
              * * * * *   F I N   D E   M E N S A J E S   * * * * *
                                    Resumen Mensajes
 Total    Info(0-4)    Aviso(5-19)      Error(20-29)    Grave(30-39)    Terminal(40-99)
     1          0            1               0               0               0
Registros fuente leídos . . . . . :   31
Registros de copia leídos . . . . :   0
Miembros de copia procesados  . . :   0
Errores de secuencia  . . . . . . :   0
Mensaje de gravedad más alta emitido:   10
 LBL0901 00  Programa CORR creado en biblioteca XMPLIB.
          * * * * *   F I N   D E   C O M P I L A C I O N   * * * * *
```

Figura 83. Ejemplo de la Frase CORRESPONDING

Cláusula LIKE

La cláusula LIKE sirve para definir las características PICTURE, USAGE, SIGN y BLANK WHEN ZERO de un nombre de datos copiándolas de un nombre de datos definido con anterioridad. LIKE sólo puede referirse a un nombre de datos o un nombre de índice, y tales nombres deben calificarse únicamente si se han definido anteriormente. También sirve para cambiar la longitud de los nombres de datos definidos por el usuario.

Esta cláusula es particularmente útil, ya que el usuario puede utilizarla para definir los identificadores en la Sección de Almacenamiento del programa que tengan los mismos atributos como variables definidas utilizando la instrucción COPY.

Para crear el nombre de datos `DEPTH` con los mismos atributos que el nombre de datos `HEIGHT`, escriba:

```
DEPTH LIKE HEIGHT
```

Para crear el nombre de datos `PROVINCE` con los mismos atributos que el nombre de datos `STATE`, sólo que con un byte más, escriba:

```
PROVINCE LIKE STATE (+1)
```

Este ejemplo le muestra cómo puede crear el ítem de datos `WS-KEY3` con los mismos atributos que el ítem de datos `KEY3` en la Sección de Almacenamiento de Trabajo:

```
5763CB1 V3R0M5                  Fuente AS/400 COBOL
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S
       001400   FILE SECTION.
       001500   FD FILE1.
       001600   01  FILE1-REC.
       001700   COPY DDS-ALL-FORMATS OF COPYDDS2.
      +000001     05 COPYDDS2-RECORD PIC X(20).
      +000002*   FORMATO E-S: RECORD1 DESDE ARCHIVO COPYDDS2 DE BIBLIOTECA COPYLIB
      +000003*
      +000004*LAS DEFINICIONES DE CLAVE PARA FORMATO DE REGISTRO RECORD1
      +000005*  NUMERO              NOMBRE          RECUPERAC.   TIPO
      +000006* 0001   KEY1-DDS                        ASCENDENTE
      +000007*       NOMBRE CLAVE ORIGINAL DEL ARCHIVO FÍSICO
      +000008     05 RECORD1      REDEFINES COPYDDS2-RECORD.
      +000009        06 KEY3                 PIC X(8).
      +000010        06 FILLER REDEFINES KEY3.
      +000011           07 KEY1-DDS          PIC X(4).
      +000012           07 FILLER            PIC X(4).
      +000013        06 DATA1                PIC X(12).
       001800  WORKING-STORAGE SECTION.
       001900  01  WS-KEY3 LIKE KEY3.
             * PICTURE IS X(8)
```

Figura 84. COPY DDS con la Cláusula LIKE.

La cláusula LIKE no puede utilizarse conjuntamente con las cláusulas REDEFINES, SIGN, USAGE o PICTURE. Si utiliza alguna de estas cláusulas con la cláusula LIKE, se producirá un error de duplicación. Igualmente, sólo puede especificarse BLANK WHEN ZERO conjuntamente con la cláusula LIKE si el atributo BLANK WHEN ZERO no ha sido heredado por la cláusula LIKE.

Una cláusula LIKE válida tiene uno de los formatos siguientes:

nombre-datos-1 cláusula LIKE xxxxx.

nombre-datos-1 xxxxx cláusula LIKE.

nombre-datos-1 xxxxx cláusula LIKE xxxxx.

Las xxxxx son una cláusula o una combinación de las cláusulas siguientes: JUSTIFIED, SYNCHRONIZED, BLANK WHEN ZERO, VALUE, OCCURS.

El ejemplo siguiente muestra lo que la cláusula LIKE puede hacer:

```
 01  INCOME.
     05  ANNUAL-WAGES   PIC 9(6)V9(2) COMP-3.
 01  YTD-WAGES LIKE ANNUAL-WAGES.
* PICTURE IS 9(6)V9(2)
* USAGE IS PACKED-DECIMAL

 01  RATES.
     05  MONTHLY-RATE   PIC 9(3).
 66  GROSS-RATE RENAMES MONTHLY-RATE.
 01  NET-RATE LIKE GROSS-RATE.
* PICTURE IS 9(3)

 01  FAMILY-NAME        PIC X(20)   VALUE "JONES".
 01  GIVEN-NAME LIKE FAMILY-NAME.
* PICTURE IS X(20)

 01  EMPLOYEE-NUMBER    PIC X(6).
 01  DEPARTMENT-MEMBERS.
     05  DEPT-EMPLOYEE-NUMBER LIKE EMPLOYEE-NUMBER
         OCCURS 10 TIMES.
* PICTURE IS X(6)
```

Nota: DEPARTMENT-MEMBERS, en el ejemplo anterior, tiene una longitud de 60 bytes.

```
     05  TENANT-NAME    PIC X(20)  OCCURS 10 TIMES.
 01  RENEWAL-RECORD.
     05  RENEWAL-MONTH  PIC X(3).
     05  RENEWAL-NAME LIKE TENANT-NAME.
* PICTURE IS X(20)
```

Nota: RENEWAL-RECORD, en el ejemplo anterior, sólo tiene una longitud de 23 bytes.

La parte de PICTURE del comentario generado se muestra en un formato conciso.

Nota: Un campo numérico con el atributo BLANK WHEN ZERO se considera un campo numérico editado.

```
 01  ORDER-DETAILS.
     05  ORDER-TYPE     PIC XX.
     05  ORDER-CODE LIKE ORDER-TYPE.
* PICTURE IS X(2)

 01  FASTENINGS.
     05  NAILS          PIC 9V99  BLANK WHEN ZERO.
     05  RIVETS LIKE NAILS.
* PICTURE IS 9V9(2)
* BLANK WHEN ZERO

 01  MORTGAGE-PAYMENT.
     05  MORTGAGE-TOTAL PIC S999V99 SIGN IS LEADING SEPARATE.
     05  MORTGAGE-INTEREST LIKE MORTGAGE-TOTAL.
* PICTURE IS S9(3)V9(2)
* SIGN IS LEADING SEPARATE

 01  PROFIT.
     05  GROSS-PROFIT   PIC 999(3)PP(5).
     05  NET-PROFIT LIKE GROSS-PROFIT.
* PICTURE IS 9(5)P(6)
```

Puede utilizar un entero para aumentar o disminuir la longitud del campo. El ejemplo siguiente muestra cómo aumentar la longitud de campo de WEEKLY-AMOUNT:

```
 01  WEEKLY-AMOUNT      PIC 9(3).
 01  ANNUAL-AMOUNT LIKE WEEKLY-AMOUNT (+3).
* PICTURE IS 9(6)
```

Asimismo, tenga en cuenta que:

- Cualquier campo que tenga atributos de BLANK WHEN ZERO se considera un campo editado.
- Si se especifica un entero de cero, se genera un mensaje informativo.

Sólo se puede aumentar o disminuir la parte del entero de la longitud de campo. No se puede cambiar el número de posiciones decimales de un ítem de datos.

Los atributos por omisión, SIGN IS TRAILING y USAGE IS DISPLAY, nunca se imprimen como comentarios después de una operación LIKE.

Cuando utiliza la cláusula LIKE, las reglas normales de calificación de nombres de datos se aplican en el nombre de datos principal; sin embargo, el nombre de datos referenciado debe estar calificado exclusivamente si se definió anteriormente más de una vez. Por ejemplo:

```
 01  COMBINATIONS.
     05  PHENOTYPE      PIC XX.
     05  GENOTYPE LIKE PHENOTYPE.
* PICTURE IS X(2)
 01  PHENOTYPE-TRAITS.
     05  PHENOTYPE      PIC X(30).
     05  PHENO-GROUP LIKE PHENOTYPE OF COMBINATIONS.
* PICTURE IS X(2)
```

Si el usuario no califica exclusivamente el nombre de datos principal, el compilador le asigna una cláusula PICTURE de X(2) y recibirá un mensaje de error.

La utilización de la cláusula LIKE a veces puede dar como resultado ítems de grupo que no sean válidos. Por ejemplo, si define un ítem de grupo COMP-4 y luego utiliza la cláusula LIKE para definir un ítem COMP-3 que está subordinado a él, se producirá un error.

El ejemplo siguiente es válido:

```
 77  SWITCHES-IN-STOCK  PIC S99.
 01  PARTS-ON-ORDER  SIGN IS LEADING SEPARATE.
     05  SWITCHES-ON-ORDER LIKE SWITCHES-IN-STOCK.
* PICTURE IS S9(2)
```

Nota: SWITCHES-ON-ORDER tiene el mismo atributo SIGN (SIGN IS TRAILING) que SWITCHES-IN-STOCK.

En el caso de B LIKE A, donde A es un ítem de grupo, B no puede estar subordinado a A. En el resto de casos, B se definirá como un ítem alfanumérico con una longitud en bytes igual a la longitud de grupo A.

```
 01  GARAGE-1.
     05  STD-PARKING-1  PIC 9(3).
 01  GARAGE-2.
     05  STD-PARKING-2  PIC 9(3)  COMP-3.
 77  VACANCIES-1 LIKE GARAGE-1.
* PICTURE IS X(3)
 77  VACANCIES-2 LIKE GARAGE-2.
* PICTURE IS X(2)
```

STD-PARKING-1 es un campo numérico con zona, de forma que VACANCIES-1 necesita 3 bytes de almacenamiento. STD-PARKING-2 es un campo numérico empaquetado, de manera que VACANCIES-2 sólo necesita 2 bytes de almacenamiento.

Podrá utilizar la cláusula LIKE con la cláusula USAGE IS POINTER:

```
 01  CUSTOMER-RECORD.
     05 CUST-NAME           PIC X(16).
     05 CUST-ADDR-POINTER   POINTER.
     05 CUST-STATS-POINTER  LIKE CUST-ADDR-POINTER.
* USAGE IS POINTER
     05 CUST-NUMBER         PIC S9(8).
```

Nota: No se puede utilizar la cláusula LIKE para cambiar la longitud de un puntero.

Para información adicional acerca de la cláusula LIKE, consulte la publicación *COBOL/400 Reference*.

Fin de Ampliación de IBM

Modificación de Referencias

La modificación de referencias permite referenciar subseries de un ítem de datos. Simplemente debe especificar la posición en el ítem de datos en el que desee iniciar la subserie, así como la longitud de la subserie. La longitud es opcional: si la omite, se amplía automáticamente hasta el final del ítem de datos.

El usuario puede escribir la posición inicial y el valor de la longitud como literales enteros, ítems de datos o expresiones aritméticas.

La posición inicial debe ser, como mínimo, 1, y no puede ser mayor que la longitud del ítem de datos referenciado. La longitud debe ser, como mínimo, 1.

El resultado de añadir la posición inicial a la especificación de longitud, y después sustraerle 1, debe estar entre 1 y la longitud total del ítem de datos referenciado, ambos inclusive. Cuando el valor de longitud es mayor que la longitud total del ítem de datos, se produce un error.

Para información adicional acerca de la modificación de referencias, vea la publicación *COBOL/400 Reference*.

La opción de generación *RANGE produce códigos para detectar situaciones de modificación de referencias fuera del rango, así como para señalar las violaciones con un mensaje en tiempo de ejecución.

Imagine que quiere recuperar la hora actual del sistema y visualizar su valor en un formato ampliado. Puede recuperarla con la instrucción ACCEPT, que devuelve las horas, minutos, segundos y centésimas de segundo en el formato:

```
HHMMSSss
```

Sin embargo, es posible que quiera visualizar la hora actual en el formato:

```
HH:MM:SS
```

Sin la modificación de referencias, deberá definir los ítems de datos siguientes:

```
01  TIME-GROUP.
  05  INTERESTING-FIELDS.
    10  HOURS                          PIC XX.
    10  MINUTES                        PIC XX.
    10  SECONDS                        PIC XX.
  05  UNINTERESTING-FIELDS.
    10  HUNDREDTHS-OF-SECONDS          PIC XX.
01  EXPANDED-TIME-GROUP.
  05  INTERESTING-FIELDS.
    10  HOURS                          PIC XX.
    10  FILLER                         PIC X   VALUE ":".
    10  MINUTES                        PIC XX.
    10  FILLER                         PIC X   VALUE ":".
    10  SECONDS                        PIC XX.
```

El código siguiente recuperaría el valor de hora, lo convertiría al formato ampliado y visualizaría el valor nuevo:

```
ACCEPT TIME-GROUP FROM TIME
MOVE CORRESPONDING
   INTERESTING-FIELDS OF TIME-GROUP TO
   INTERESTING-FIELDS OF EXPANDED-TIME-GROUP
DISPLAY "CURRENT TIME IS: " EXPANDED-TIME-GROUP
```

Con la modificación de referencias, el usuario no necesita proporcionar nombres para los subcampos que describen los ítems de la hora. La única definición de datos que debe tener es:

```
01  REFMOD-TIME-ITEM                      PIC X(8).
```

El código para recuperar y ampliar el valor de la hora aparece de la siguiente manera:

```
ACCEPT REFMOD-TIME-ITEM FROM TIME
DISPLAY "CURRENT TIME IS: "
   REFMOD-TIME-ITEM (1:2)
   ":"
   REFMOD-TIME-ITEM (3:2)
   ":"
   REFMOD-TIME-ITEM (5:2)
```

El ejemplo siguiente muestra una referencia que empieza en la posición de caracteres 1, para una longitud 2, de forma que se recupera la parte del valor de la hora que corresponde al número de horas:

```
REFMOD-TIME-ITEM (1:2)
```

El ejemplo siguiente muestra una referencia que empieza en la posición de caracteres 3, para una longitud 2, de forma que se recupera la parte del valor de la hora que corresponde al número de minutos:

```
REFMOD-TIME-ITEM (3:2)
```

El ejemplo siguiente muestra un referencia que comienza en la posición de caracteres 5, para una longitud 2, de forma que se recupera la parte del valor de la hora que corresponde al número de segundos:

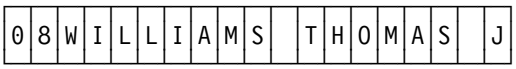
```
REFMOD-TIME-ITEM (5:2)
```

Modificación de Referencias con Tablas de Longitud Variable

Suponga que utiliza las tablas de longitud variable para contener nombres:

```
01  NAME-GROUP.
  05  NAME-LENGTH                        PIC 99.
  05  NAME-PORTION.
    10  FILLER                           PIC X
              OCCURS 1 TO 17 TIMES
              DEPENDING ON NAME-LENGTH.
01  NEW-NAME-GROUP.
  05  NEW-NAME-LENGTH                    PIC 99.
  05  NEW-NAME-PORTION.
    10  FILLER                           PIC X
              OCCURS 1 TO 17 TIMES
              DEPENDING ON NEW-NAME-LENGTH.
```

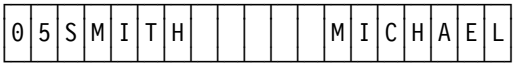
El objeto OCCURS DEPENDING ON de la tabla NAME-PORTION se coloca en 8 de forma que sólo se referencien las primeras ocho apariciones de la tabla, aunque se rellenen los 17 bytes de NAME-PORTION.



Suponga que quiere cambiar el valor en el ítem NAME-PORTION sin cambiar la parte del ítem que se define más allá de la longitud definida actualmente. Podría intentar la codificación:

```
MOVE NEW-NAME-GROUP TO NAME-GROUP
```

en la que el contenido de NEW-NAME-GROUP es:

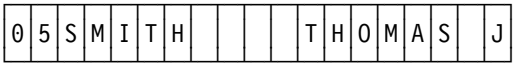


Según las reglas para la instrucción MOVE, se sustituirá todo el contenido del campo de recepción NAME-GROUP. Este problema puede evitarse utilizando la modificación de referencias en la instrucción MOVE:

```
MOVE NEW-NAME-GROUP TO NAME-GROUP ( 1 :LENGTH OF NAME-GROUP )
```

Al especificar la modificación de referencias con la longitud (LENGTH OF) de registro especial, la longitud de NAME-GROUP queda determinada por el valor de la variable NAME-LENGTH.

El nuevo valor de NAME-GROUP será:



Modificación de Referencias Utilizando los Nombres de Datos

Hasta aquí, todos los ejemplos de modificación de referencias han mostrado literales numéricos simples como posición inicial y valores de longitud de la modificación de referencias. Estos valores también pueden ser ítems de datos o expresiones aritméticas.

Suponga que un campo contiene algunos caracteres alineados a la derecha, y quiere moverlos a otro campo, pero alineados a la izquierda en lugar de estar a la derecha. Podrá hacerlo utilizando la modificación de referencias y una instrucción INSPECT.

El programa tendría los datos siguientes:

```
01  LEFTY                                  PIC X(30).
01  RIGHTY                                 PIC X(30)
                                           JUSTIFIED RIGHT.
01  I                                       PIC 9(9)
                                           USAGE BINARY.
```

A continuación, el programa cuenta el número de espacios iniciales y, utilizando expresiones aritméticas en una expresión de modificación de referencias, mueve los caracteres alineados a la derecha a otro campo (alineado a la izquierda):

```
MOVE SPACES TO LEFTY
MOVE ZERO TO I
INSPECT RIGHTY
   TALLYING I FOR LEADING SPACE
IF I IS LESS THAN 30 THEN
   MOVE RIGHTY ( I + 1 : 30 - I ) TO LEFTY
END-IF
```

La instrucción MOVE transfiere caracteres de RIGHTY, comenzando en la posición calculada en I + 1, para una longitud que se calcula en 30 – I, en el campo LEFTY.

Modificación de Referencias con Subíndices

Defina una tabla como ésta:

```
01  ANY-TABLE.
  05  TABLE-ELEMENT                          PIC X(10)
         OCCURS 3 TIMES
         VALUE "ABCDEFGHIJ".
```

Puede cambiar el tercer y cuarto byte del primer ítem de TABLE-ELEMENT al valor “??” con la instrucción MOVE siguiente:

```
MOVE "??" TO TABLE-ELEMENT ( 1 ) ( 3 : 2 )
```

Esta instrucción moverá el valor “??” al ítem de la tabla número 1, comenzando en la posición del carácter 3, para una longitud 2.

ANY-TABLE tendría este aspecto antes del cambio:

ABCDEFGHIJ
ABCDEFGHIJ
ABCDEFGHIJ

Y éste es el aspecto que tendría después del cambio:

AB??EFGHIJ
ABCDEFGHIJ
ABCDEFGHIJ

Deseditar

La acción **Deseditar** sirve para mover un ítem de datos editados de forma numérica a un ítem de datos de recepción numéricos o editados de forma numérica. El compilador realiza esta acción estableciendo en primer lugar el valor no editado del ítem editado de forma numérica. A continuación mueve el valor no editado al receptor.

La acción Deseditar puede producirse en operaciones del tipo MOVE e INITIALIZE. Una cláusula VALUE no deshace la edición.

Observe que los valores numéricos no editados pueden significar signos.

Imagine que utiliza un campo de caracteres para que contenga un valor numérico que se visualice en el terminal y para que contenga un valor suministrado por el operador del sistema. Suponga que este campo tiene la siguiente definición:

- Una posición de caracteres para un signo (para que contenga un espacio si el campo numérico es positivo o cero, o un signo menos si el campo numérico es negativo).
- Seis posiciones de dígitos, en las que los ceros iniciales se representen mediante espacios;
- Una coma decimal;
- Dos posiciones de dígitos decimales.

Los ítems de datos que utilice para definir este campo tendrían el siguiente aspecto:

```
01 NUM-EDIT  PIC –Z(6).9(2)  USAGE IS DISPLAY.
```

Se podría inicializar este campo utilizando la instrucción:

```
MOVE ZEROS TO NUM-EDIT
```

y cuando se visualice en el terminal, contendría el valor ␣␣␣␣␣␣␣.00.

Posteriormente, el operador del sistema podría utilizar este campo para la entrada de datos. Si el operador inserta ␣␣␣␣123,45 en el campo, el usuario podrá obtener el valor numérico del campo moviéndolo a un ítem de datos definido como:

```
01 NUMERIC-ITEM  PIC S9(6)V9(2)  USAGE IS PACKED-DECIMAL.
```

La instrucción:

```
MOVE NUM-EDIT TO NUMERIC-ITEM
```

provoca la acción Deshacer Edición, con lo que el ítem numérico recibe el valor numérico del campo editado de forma numérica NUM-EDIT. Como resultado, el ítem numérico contiene el valor +123,45.

Ejemplos de Deseditar

La Tabla 5 y la Tabla 6 en la página 279 muestran ejemplos de la acción deseditar de COBOL/400.

Tabla 5. Mover ítems Editados de Forma Numérica a Receptores Numéricos

Imagen Origen	Valor Origen	Imagen Receptora	Valor Receptor
$+++,+++.++	$␣␣␣+123.45	S9(5)V9(5) USAGE IS DISPLAY	+123.45
$+++,+++.++	$␣–1,234.56	S9(5)V9(5) USAGE IS BINARY	–1234.56
*****.999+	**123.450–	S9(5)V9(5) USAGE IS PACKED-DECIMAL	–123.45

Tabla 6. Mover ítems Editados de Forma Numérica a Receptores Editados de Forma Numérica			
Imagen Origen	**Valor Origen**	**Imagen Receptora**	**Valor Receptor**
$+++,+++.++	$bbb+123.45	$$$$,$$$.$$CR	bbbb$123.45bb
$+++,+++.++	$b–1,234.56	––––,–––.99	bb–1,234.56
*****.999+	**123.450–	ZZZBZZZBVZZZ	bbbb123b450
ZZZ999CR	b12345bb	$++++9999	$bb+12345
ZZZ999CR	b12345CR	999999.99–	012345.00–

Manejo de Errores de Datos

El compilador proporciona comprobaciones de errores en tiempo de compilación para operaciones de mover que impliquen deshacer edición.

El compilador no realiza esta comprobación para los valores origen de cero e ignora los caracteres de inserción simples (tales como / B 0 , .).

Prueba de Signo

El compilador valida los signos en ítems fuente editados de forma numérica, según las reglas que aparecen a continuación.

Definición PICTURE	**Contenidos admisibles**
+ fijo	+ o –
– fijo	b o –
CR	bb o CR
DB	bb o DB

Si no se obedecen estas reglas, se produce un error de signo y el programa se detiene.

Prueba de Flotantes

Si el fuente tiene una serie de caracteres flotantes, esta prueba verifica la corrección de caracteres flotantes iniciales en el campo de datos.

Las reglas para la prueba de flotantes son:

- Si la cláusula fuente PICTURE contiene símbolos $ flotantes, el primer carácter que no sea espacio en blanco de la parte pertinente del campo fuente (posiciones 2 a 7 en el ejemplo) debe ser un signo $, y su ubicación debe ser correcta según las reglas de edición de la cláusula PICTURE. (Vea la publicación *COBOL/400 Reference* para más información acerca de estas reglas).

  Por ejemplo:

Ubicación de un Carácter Flotante Inicial

```
01 A PIC +$$B,$$$.
  .
  .
/* Observe que "b" representa un espacio */
/* Serie PIC:             +$$B,$$$       */
/* Índices Posición:       12345678      */
MOVE 1 TO A.       /* A = "+bbbbb$1"     */
MOVE 12 TO A.      /* A = "+bbbb$12"     */
MOVE 123 TO A.     /* A = "+bbb$123"     */
MOVE 1234 TO A.    /* A = "+$1b,234"     */
```

En este ejemplo, el signo $ debe colocarse en la posición 2, 5, 6 o 7.

- Si la cláusula fuente PICTURE contiene símbolos + flotantes, el primer carácter que no sea en blanco de la parte pertinente del campo fuente debe ser un signo + o –, y su ubicación debe ser correcta según las reglas de edición de la cláusula PICTURE.
- Si la cláusula PICTURE contiene símbolos – flotantes, la parte pertinente del campo fuente deberá empezar con:
  - Uno o más espacios contiguos, el último de los cuales debe estar correctamente situado según las reglas de edición de la cláusula PICTURE.
  - Uno o más espacios contiguos, con un signo – justo a continuación. La ubicación del signo – debe ser correcta según las reglas de edición de la cláusula PICTURE.
  - A –.

Si estas reglas no se obedecen, se produce un error de flotante y el programa se detiene.

Consideraciones acerca del Rendimiento

Cláusulas PICTURE para Ítems Numéricos

Dado que las instrucciones de hardware utilizan signos, el usuario puede mejorar el rendimiento incluyendo una S en una cláusula de imagen siempre que sea posible.

También se puede mejorar el rendimiento especificando números impares o posiciones de caracteres numéricos en las cláusulas de imagen para ítems COMP-3 (decimales empaquetados). Internamente, el byte más a la derecha de un ítem decimal empaquetado contiene un dígito y un signo, y cualquier otro byte contiene dos dígitos. Si utiliza la configuración más eficiente, el compilador no necesita suministrar el dígito que falta.

Ítems Binarios de Ocho Bytes

Evite la utilización de ítems binarios de 8 bytes. Puede especificar estos ítems según su conveniencia, pero el compilador debe realizar conversiones para utilizarlos.

Segmentación

La utilización de la segmentación aumenta los tiempos de compilación y ejecución del programa COBOL. La característica de segmentación sólo se proporciona para lograr la compatibilidad con otros sistemas. Al utilizar programas COBOL/400 no tiene que preocuparse por la gestión de almacenamiento.

Llamadas a un Programa COBOL desde un Programa que no es COBOL

La repetición de llamadas desde un programa COBOL a otro que no sea COBOL puede dar como resultado una disminución sustancial en el rendimiento del compilador, puesto que al salir del programa principal COBOL (es decir, el programa que ha iniciado la unidad de ejecución COBOL) se desactiva el programa.

Se ha añadido una nueva función (MGTFUNC) a la rutina de tiempo de ejecución de COBOL (QRLMAIN) para evitar dicha desactivación, provocando que el programa principal COBOL se trate como un subprograma. Dado que esta corrección depende del tamaño de MGT, se recomienda llamar a la rutina de tiempo de ejecución, QLRMAIN, desde el programa principal COBOL con MGTFUNC = 9, tal y como aparece en el ejemplo siguiente:

```
01  mgtstruc.
    03  FILLER PIC X(277).
    03  mgtfunc pic 9(2) comp-4 value 9.
77  TEST-VAR  PIC X(10) value spaces.

      if test-var = spaces then
        display 'spaces'
        move 'faked' to test-var
        call 'QLRMAIN' using mgtstruc
      else
        display 'not spaces ' test-var.
```

Notas:

1. `01 Mgtstruc` debe estar en un límite de 16 bytes. Si se produce un error de límite, añada `77 aa PIC X.` delante de `01`.
2. Dado que la llamada a QLRMAIN cambia el programa principal COBOL por un subprograma, deberá utilizar el mandato EXIT PROGRAM y no STOP RUN, ya que podrían producirse errores.
3. RCLRSC desactivará el programa principal (ahora un subprograma).

Depuración

La depuración del lenguaje fuente COBOL se proporciona para ayudar al programador en COBOL a depurar un programa que no funciona según lo esperado. La utilización de este servicio aumenta los tiempos de compilación y ejecución de un programa COBOL.

Opción *NORANGE

Esta opción del parámetro GENOPT del mandato CRTCBLPGM suprime las comprobaciones en tiempo de ejecución para rangos de modificación de subíndices y referencias.

Esta opción puede aumentar el rendimiento cuando:

- Realice referencias frecuentes a las tablas, y los subíndices siempre hagan referencia a ítems que estén en las tablas
- Utilice frecuentemente la modificación de referencias.

Nota: La opción *RANGE genera códigos para comprobar rangos de subíndice. Por ejemplo, asegura que el usuario no está intentando acceder al ítem número 21 de una matriz de 20 ítems.

La opción *NORANGE no genera códigos para comprobar rangos de modificación de subíndices o referencias.

Estas opciones no eliminan la comprobación de subíndice cero realizada por el sistema operativo. Si se produce el subíndice cero, el sistema operativo no permitirá su utilización y emitirá el mensaje MCH0603.

Opción *DUPKEYCHK

Esta opción de parámetro GENOPT del mandato CRTCBLPGM indica que se efectuará la comprobación de clave duplicada para archivos INDEXED. La utilización de DUPKEYCHK durante la lectura de archivos INDEXED puede afectar negativamente al rendimiento.

Archivos Relativos

Pueden producirse retrasos duraderos si se abren o se cierran archivos relativos en los que se estén inicializando grandes volúmenes de registros para suprimir registros.

Consulte la Tabla 4 en la página 263 para más información.

Indicadores

Si utiliza los indicadores en un área de indicadores separada (palabra clave INDARA especificada en las DDS) en vez de hacerlo en un área de registros, la utilización de la cláusula OCCURS para especificar una tabla de hasta 99 indicadores puede aumentar el rendimiento. Consulte la Figura 60 en la página 163 para más información.

Control de Compromiso

Generalmente, la utilización del control de compromiso aumenta el tiempo de ejecución de un programa COBOL. Además, el bloqueo de registros resultante de la utilización del control de compromiso por un trabajo puede provocar retrasos para otros usuarios que intenten acceder al mismo archivo.

Lectura sin Bloqueos de Registros

Para evitar registros de bloqueos innecesarios, puede incluir la frase NO LOCK en la instrucción READ. Para más información acerca de esta frase, consulte la sección de la instrucción READ en la publicación *COBOL/400 Reference*.

Inicialización de Variables

Se puede reducir el tiempo de ejecución de un programa eligiendo la posibilidad de **no** inicializar variables de programa que no tengan cláusulas de valor asociadas. Se puede especificar esta no inicialización mediante `*NOSTDINZ` para el parámetro GENOPT del mandato CRTCBLPGM, o especificando `NOSTDINZ` en la instrucción PROCESS. A continuación, el compilador inicializa únicamente aquellas variables que tienen cláusulas de valor declaradas. Una ventaja adicional de esta opción consiste en que el usuario también puede compilar programas más grandes con un número mayor de variables.

Si se especifica `*NOSTDINZ`, deberá asegurarse de que todos los ítems de datos contengan datos válidos antes de intentar manipular los ítems. Si un ítem no contiene datos válidos, pueden producirse errores de datos decimales.

Bloqueo de Registros

Se puede utilizar el bloqueo de registros para mejorar el rendimiento del tiempo de ejecución. Las ventajas del bloqueo quedan patentes cuando se leen varios registros de manera secuencial, como una lectura aleatoria seguida de lecturas secuenciales.

Para más información acerca de los bloqueos, consulte el apartado “Desbloqueo de Registros de Entrada y Bloqueo de Registros de Salida” en la página 107.

Bucles en un Programa

Cuando un programa procesa repetidamente la misma serie de instrucciones, y parece evidente que así continuará indefinidamente, el programa se encuentra en un bucle. Para identificar los bucles, puede utilizar la información conocida acerca del propio programa, de la manera siguiente:

- Tiempo: Si el tiempo de ejecución real sobrepasa sustancialmente el tiempo de ejecución esperado, el programa puede estar en un bucle.
- Operaciones de E/S: Si no se dan operaciones de entrada/salida y se espera que la E/S se produzca repetidamente, es probable que el programa esté en un bucle.

Rastreo de un Bucle en un Programa

Frecuentemente, un bucle abarca muchas instrucciones en un programa. En este caso, puede utilizar las características de depuración COBOL que aparecen en el Capítulo 5, “Depuración del Programa” en la página 57.

Errores que Pueden Causar un Bucle

Una instrucción PERFORM con una cláusula UNTIL puede provocar un bucle si no se puede cumplir la condición especificada en la cláusula UNTIL. Por ejemplo:

```
PERFORM ... UNTIL COUNTR LESS THAN ZERO
```

donde COUNTR es un ítem numérico sin signo.

Una instrucción GO TO que hace referencia a un nombre de procedimiento anterior puede provocar un bucle si no hay ninguna instrucción condicional para impedir que la instrucción GO TO se procese de nuevo. Por ejemplo:

```
PARA-1.
  MOVE ...
  MOVE ...
  MOVE ...
PARA-2.
  MOVE ...
  GO TO PARA-1.
```

se produce una variación posible de este caso si existe una instrucción condicional, pero esta condición no puede cumplirse o la instrucción no se bifurca (mediante una sentencia GO TO) a un párrafo fuera del rango del bucle.

Capítulo 12. Comunicaciones entre Programas

En ocasiones, una aplicación es suficiente para que se codifique como un programa único y autosuficiente. Sin embargo, en la mayoría de los casos, la solución que ofrece una aplicación consta de varios programas compilados por separado y utilizados conjuntamente.

El sistema AS/400 proporciona la comunicación entre programas COBOL y entre programas COBOL y otros que no sean COBOL.

Una **unidad de ejecución** COBOL es un conjunto de uno o más programas que funcionan como una unidad en el tiempo de ejecución con el fin de proporcionar una solución a un problema. Una unidad de ejecución COBOL se inicia con el primer programa COBOL en la pila de programas, e incluye todos los programas (de cualquier tipo) que estén por debajo de él. Una **pila de programas** es una lista de programas enlazados como resultado de programas que llaman a otros programas, o de manera implícita desde cualquier otra acción dentro del mismo trabajo.

Cuando una unidad de ejecución consta de varios programas compilados por separado que se llaman mutuamente, dichos programas deben ser capaces de comunicarse entre sí. Necesitan transferir el control y normalmente necesitan tener acceso a datos comunes. Este capítulo describe los métodos seguidos por esta comunicación entre programas compilados por separado.

Transferencia de Control a Otro Programa

En la División de Procedimientos, un programa puede llamar a otro programa (lo que, en términos de COBOL, se suele denominar subprograma), y este programa llamado puede, a su vez, llamar a otro programa. El programa que llama a otro programa se denomina programa **de llamada**, y el programa al que ésta llama se denomina programa **llamado**.

El programa COBOL llamado empieza a ejecutarse al principio de la División de Procedimiento.

Cuando el proceso del programa llamado ha finalizado, el programa puede transferir de nuevo el control al programa de llamada o finalizar la unidad de ejecución.

Un programa llamado no debe ejecutar, directa o indirectamente, su llamador (por ejemplo, el programa X llamando al programa Y; el programa Y llamando al programa Z, y el programa Z llamando entonces al programa X). Recibe el nombre de llamada **recurrente**. COBOL/400 permite la recurrencia tanto en programas principales como en subprogramas. Sin embargo, si desea que sus programas se adecúen a estándares SAA, no utilice llamadas recurrentes.

Programas Principales y Subprogramas

El primer programa COBOL que se va a a ejecutar inicia la unidad de ejecución COBOL, y no es otro que el **programa principal**. No existen instrucciones fuente u opciones específicas que identifiquen un programa COBOL como un programa principal o un subprograma. Un **subprograma** es un programa de la unidad de ejecución que está por debajo del programa principal en la pila de programas. Para más información acerca de las pilas de programas y de otros términos relativos a la comunicación entre programas, consulte la publicación *CL Guía del Programador*.

Devolución de Control desde un Programa Llamado

Es importante conocer si un programa COBOL es un programa principal o un subprograma, con el fin de determinar el modo en que el control se devuelve desde un programa llamado cuando se produce un error o finaliza un programa.

Puede emitir una instrucción STOP RUN, EXIT PROGRAM o GOBACK para devolver el control de un programa llamado.

Si la ejecución finaliza en el programa principal, se utiliza STOP RUN o GOBACK. Estas instrucciones finalizan la unidad de ejecución, y se devuelve el control al llamador del programa principal.

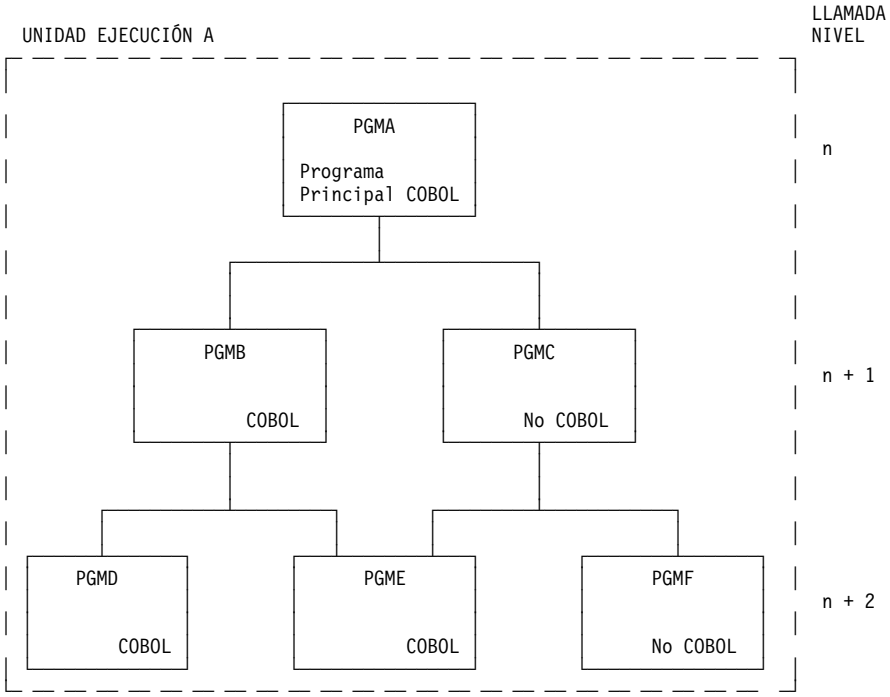
Si la ejecución finaliza en un subprograma, éste podrá finalizar con una instrucción EXIT PROGRAM, GOBACK o STOP RUN. Si el subprograma finaliza con una instrucción EXIT PROGRAM o GOBACK, el control vuelve a su inmediato llamador sin finalizar la unidad de ejecución. Se genera una instrucción EXIT PROGRAM implícita si no existe ninguna instrucción ejecutable posterior en un programa llamado. Si finaliza con una instrucción STOP RUN, el efecto es el mismo que se produce en el programa principal: todos los programas COBOL de la unidad de ejecución finalizan, y el control vuelve al llamador del programa principal.

Un subprograma se queda en un **estado utilizado por última vez** cuando finaliza con EXIT PROGRAM o GOBACK. La próxima vez que se llame en la unidad de ejecución, sus valores internos serán los que se habían dejado, con la salvedad de que los valores de retorno para las instrucciones PERFORM se reinicializarán a sus valores iniciales. Por contra, un programa principal se inicializará cada vez que se llame.

Los ejemplos siguientes ilustran la utilización de las instrucciones EXIT PROGRAM y STOP RUN en partes distintas de la unidad de ejecución.

- El ejemplo de la Figura 85 en la página 287 muestra una única unidad de ejecución.
- El ejemplo de la Figura 86 en la página 288 muestra múltiples unidades de ejecución que se ejecutan de manera consecutiva.
- El ejemplo de la Figura 87 en la página 289 muestra una unidad de ejecución con un programa compartido que es a la vez un subprograma y un programa principal.
- El ejemplo de la Figura 88 en la página 290 muestra varias unidades de ejecución que se ejecutan simultáneamente.

Nota: Puede sustituir una instrucción GOBACK por una instrucción EXIT PROGRAM que aparezca en un subprograma, o una instrucción STOP RUN que aparezca en un programa principal.



	PROGRAMA QUE EJECUTA LA INSTRUCCIÓN			
INSTRUCCIÓN	PGMA	PGMB	PGMD	PGME
EXIT PROGRAM	1	2	2	2
STOP RUN	3	3	3	3

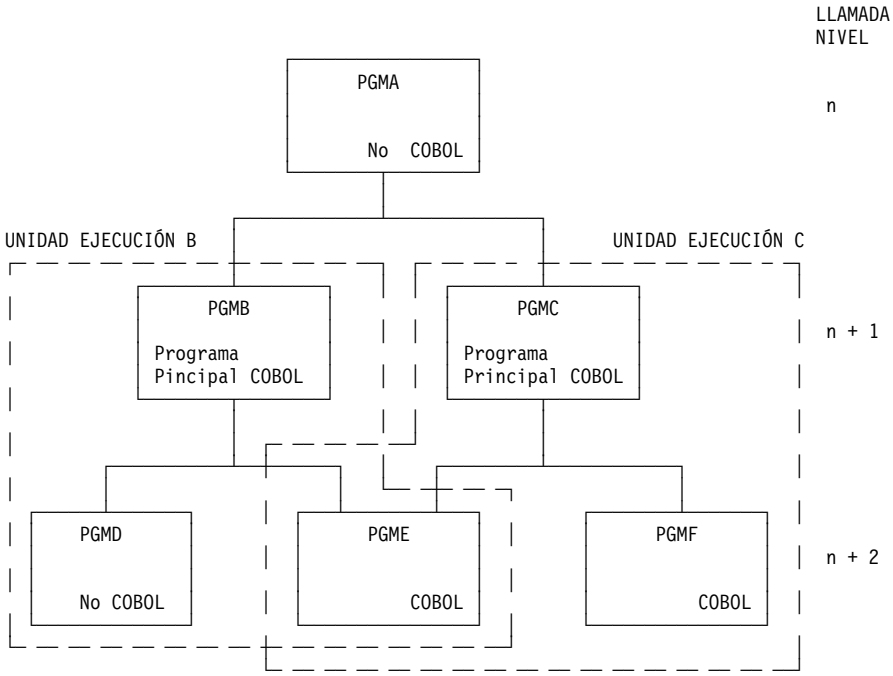
Figura 85. Ejemplo de una Única Unidad de Ejecución

1 No se procesa ninguna operación porque la instrucción se procesa en un programa principal. El proceso continúa con la instrucción siguiente en PGMA.

2 El control se devuelve al llamador del programa que procesa la instrucción EXIT PROGRAM.

3 La unidad de ejecución A finaliza. Para todos los programas en la unidad de ejecución, se cierran los archivos abiertos.Se libera el almacenamiento para todos los programas en la unidad de ejecución. El control se devuelve al programa que está en el nivel de llamada n-1. Si n=1, se aplican las consideraciones siguientes:

- La unidad de ejecución A opera como un paso de trabajo. Consulte la publicación *CL Guía del Programador* para más información.
- Para trabajos por lotes, la instrucción STOP RUN finaliza el trabajo. Para trabajos interactivos, el control se devuelve al sistema y éste el paso del trabajo.



PROGRAMA QUE EJECUTA LA INSTRUCCIÓN

INSTRUCCIÓN	PGMB	PGMC	PGME (UNIDAD DE EJEC. B)	PGME (UNIDAD EJEC.C)	PGMF
EXIT PROGRAM	1	1	2	2	2
STOP RUN	3	4	3	4	4

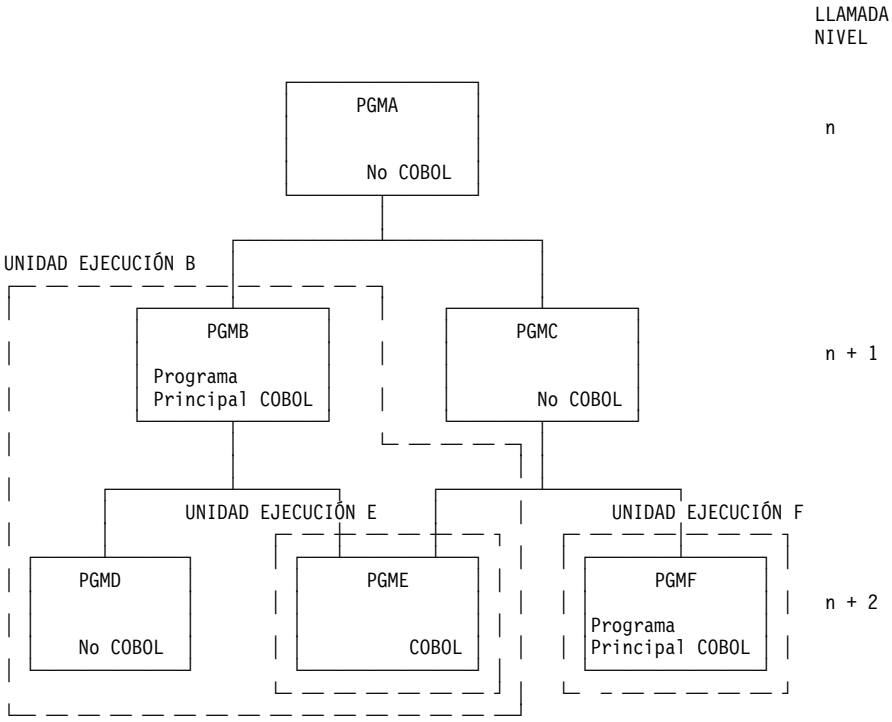
Figura 86. Ejemplo de Varias Unidades de Ejecución que se Ejecutan de Manera Consecutiva

1 No se procesa ninguna operación porque la instrucción se procesa en un programa principal. El proceso continúa con la instrucción siguiente en el programa principal.

2 El control se devuelve al llamador del programa que procesa la instrucción EXIT PROGRAM.

3 Se finaliza la unidad de ejecución B. Todos los archivos abiertos en la unidad de ejecución B se cierran. Se libera el almacenamiento para todos los programas en la unidad de ejecución B. El control se devuelve al llamador del programa principal para la unidad de ejecución (PGMA).

4 Se finaliza la unidad de ejecución C. Todos los archivos abiertos en la unidad de ejecución C se cierran. Se libera el almacenamiento para todos los programas en la unidad de ejecución C. El control se devuelve al llamador del programa principal para la unidad de ejecución (PGMA).



	PROGRAMA QUE EJECUTA LA INSTRUCCIÓN			
INSTRUCCIÓN	PGMB	PGME (UNIDAD DE EJEC. B)	PGME (UNIDAD DE EJEC. E)	PGMF
EXIT PROGRAM	1	2	1	1
STOP RUN	3	3	4	5

Figura 87. Ejemplo de una Unidad de Ejecución con un Programa Compartido que es tanto un subprograma como un programa principal

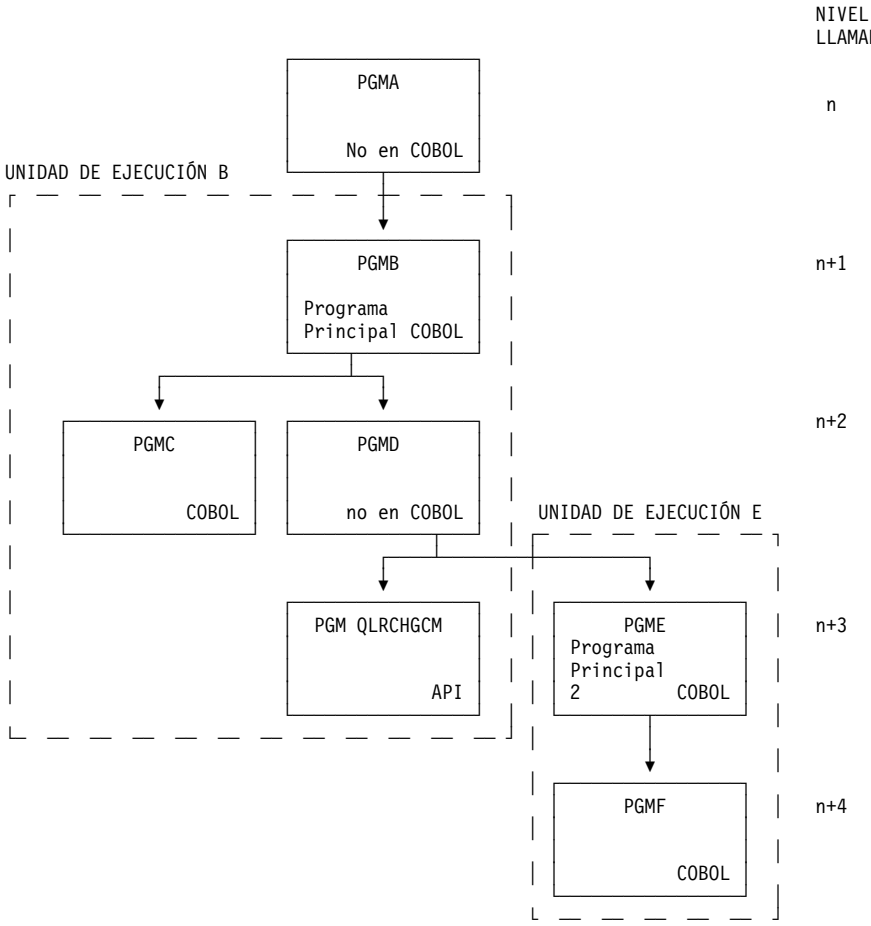
1 No se procesa ninguna operación porque la instrucción se procesa en un programa principal. El proceso continúa con la instrucción siguiente en el programa principal.

2 El control se devuelve al llamador del programa que procesa la instrucción EXIT PROGRAM.

3 Se finaliza la unidad de ejecución B. Todos los archivos abiertos en la unidad de ejecución B se cierran. Se libera el almacenamiento para todos los programas en la unidad de ejecución B. El control se devuelve al llamador del programa principal para la unidad de ejecución (PGMA).

4 Se finaliza la unidad de ejecución E. Todos los archivos abiertos en la unidad de ejecución E se cierran. Se libera el almacenamiento para PGME. El control se devuelve al llamador del programa principal para la unidad de ejecución (PGMC).

5 Se finaliza la unidad de ejecución F. Todos los archivos abiertos en la unidad de ejecución F se cierran. Se libera el almacenamiento para PGMF. El control se devuelve al llamador del programa principal para la unidad de ejecución (PGMC).



	PROGRAMA QUE EJECUTA LA INSTRUCCIÓN			
INSTRUCCIÓN	PGMB	PGMC (UNIDAD DE EJEC. B)	PGME (UNIDAD DE EJEC. E)	PGMF
EXIT PROGRAM	1	2	1	2
STOP RUN	3	3	4	4

Figura 88. Ejemplo de Varias Unidades de Ejecución que se Ejecutan Simultáneamente

1 No se procesa ninguna operación porque la instrucción se procesa en un programa principal. El proceso continúa con la instrucción siguiente en programa principal.

2 El control se devuelve al llamador del programa que procesa la instrucción EXIT PROGRAM.

3 La unidad de ejecución B sólo se puede finalizar una vez que la unidad de ejecución E realiza una instrucción STOP RUN. Cuando la unidad de ejecución B finaliza, todos los archivos abiertos en la unidad de ejecución B se cierran. Se libera el almacenamiento para todos los programas de la unidad de ejecución B, y el control se devuelve al llamador del programa principal (PGMA).

4 Se finaliza la unidad de ejecución E. Todos los archivos abiertos en la unidad de ejecución E se cierran. Se libera el almacenamiento para todos los programas en la unidad de ejecución E. El control se devuelve a PGMD en la unidad de ejecución B.

Las unidades de ejecución concurrentes se consiguen utilizando la API QLRCHGCM. Consulte la publicación *System Programmer's Interface Reference* para más información acerca de esta API.

Inicialización de Almacenamiento

La primera vez que se llama a un programa COBOL en una unidad de ejecución, se inicializa su almacenamiento. El almacenamiento se inicializa de nuevo con las condiciones siguientes:

- Se termina la unidad de ejecución, luego se vuelve a iniciar.
- El programa se cancela (utilizando la instrucción CANCEL para COBOL, la operación FREE para el lenguaje de programación RPG/400* o el mandato Reclamar Recurso (RCLRSC), y a continuación se llama de nuevo.

Si se nombra un programa que no sea COBOL en una instrucción CANCEL, su nombre debe respetar las reglas para la formación de un nombre de programa COBOL.

Llamada a Otro Programa

Con cierta frecuencia, deseará que sus programas COBOL se comuniquen con otros programas COBOL o que no sean COBOL.

Paso de Datos Utilizando BY REFERENCE o BY CONTENT

BY REFERENCE significa que el subprograma está consultando y procesando los ítems de datos en el almacenamiento del programa de llamada, y no está trabajando sobre una copia de los datos.

BY CONTENT significa que el programa de llamada está pasando únicamente el **contenido** del *literal* o *identificador*. Con una instrucción CALL . . . BY CONTENT, el programa llamado no puede cambiar el valor del *literal* o *identificador* en el programa de llamada, incluso si éste modifica la variable en la que ha recibido el *literal* o *identificador*.

El hecho de pasar ítems de datos con BY REFERENCE o con BY CONTENT depende de lo que el usuario desee que el programa haga con los datos:

- Si desea que la definición del argumento de la instrucción CALL del programa de llamada y la definición del parámetro en el programa llamado compartan la misma memoria, especifique:

  CALL . . . BY REFERENCE *identificador*.

  Todo cambio realizado por el subprograma en el parámetro afecta al argumento del programa de llamada.

  Un identificador en la frase USING de la instrucción CALL . . . BY REFERENCE puede ser un nombre de archivo, además de un nombre de datos.

  El compilador acepta los nombres de archivos CALL como operandos CALL como si fueran una extensión.

- Si desea pasar la dirección de un área de registro a un programa llamado, especifique:

  CALL . . . BY REFERENCE ADDRESS OF *nombre-registro*.

  El subprograma recibe la instrucción ADDRESS OF del registro especial para el nombre de registro especificado.

  El usuario deberá definir el nombre de registro como un ítem de nivel-01 o nivel-77 en la Sección de Enlace del programa de llamada y del programa llamado. Se proporciona una instrucción ADDRESS OF del registro especial para cada registro de la Sección de Enlace.

- Si no desea que la definición del argumento de la instrucción CALL del programa de llamada y la definición del parámetro en el programa llamado compartan la misma memoria, especifique:

  CALL . . . BY CONTENT *identificador*.

- Si desea pasar un valor de literal a un programa llamado, especifique:

  CALL . . . BY CONTENT *literal*.

  El programa llamado no puede cambiar el valor del literal. El literal no puede ser numérico.

- Si desea pasar la longitud de un ítem de datos, especifique:

  CALL . . . BY CONTENT LENGTH OF *identificador*.

  El programa de llamada pasa la longitud de *identificador* desde su longitud de (LENGTH OF) registro especial. Cuando se pasan los literales por contenido (BY CONTENT), el programa llamado no puede cambiar sus valores.

- Si desea pasar tanto un ítem de datos como su longitud a un subprograma, especifique una combinación de BY REFERENCE y BY CONTENT. Por ejemplo:

  ```
  CALL 'ERRPROC' USING BY REFERENCE A
       BY CONTENT LENGTH OF A.
  ```

Los ítems de datos de un programa de llamada pueden describirse en la Sección de Enlace de todos los programas a los que llama directa o indirectamente. En este caso, el almacenamiento para estos ítems se asigna al programa de llamada superior. Es decir, el programa A llama al programa B, el cual llama al programa C. Los ítems de datos del programa A pueden describirse en la Sección de Enlace de los programas B y C, de manera que un conjunto de datos pueda estar disponible para los tres programas.

Descripción de Argumentos en el Programa de Llamada

En el programa de llamada, los argumentos se describen en la División de Datos del mismo modo que el resto de los ítems de datos de la División de Datos. A menos que se encuentren en la Sección de Enlace, el almacenamiento de dichos ítems se asigna al programa de llamada. Si se consultan datos en un archivo, éste debe estar abierto. Codifique la cláusula USING de la instrucción CALL para pasar los argumentos.

Descripción de Parámetros en el Programa Llamado

En el programa llamado, los parámetros se describen en la Sección de Enlace. Codifique la cláusula USING después de la cabecera PROCEDURE-DIVISION para recibir los parámetros.

En la Sección de Enlace

El usuario deberá conocer lo que esté pasando desde el programa de llamada y configurar la Sección de Enlace en el programa llamado para aceptarlo. Al programa llamado no le importa la cláusula de la instrucción CALL que el usuario utilice para pasar los datos (BY REFERENCE o BY CONTENT). En cualquiera de los dos casos, el programa llamado debe describir los datos que está recibiendo, cosa que realiza en la Sección de Enlace.

El número de *nombres-datos* de la lista de *identificador* de un programa llamado no debe ser mayor que el número de *nombres-datos* de la lista de *identificador* del programa de llamada. Existe una correspondencia de posición de uno a uno; es decir, el primer *identificador* del programa de llamada se pasa al primer *identificador* del programa llamado, y así sucesivamente. El compilador no hace ningún intento de hacer coincidir argumentos y parámetros.

Agrupación de Datos a Pasar

Piense en agrupar todos los ítems de datos que desee pasar entre programas y en ponerlos bajo un ítem de nivel-01. Si lo hace, podrá pasar un registro único de nivel-01 entre programas. Para ver un ejemplo de este método, observe la Figura 89 en la página 294.

Para que la posibilidad de registros no coincidentes se vea aún reducida, transfiera el registro de nivel-01 a un miembro de copia y cópielo en ambos programas, es decir, cópielo en la Sección de Almacenamiento de Trabajo del programa de llamada y en la Sección de Enlace del programa llamado.

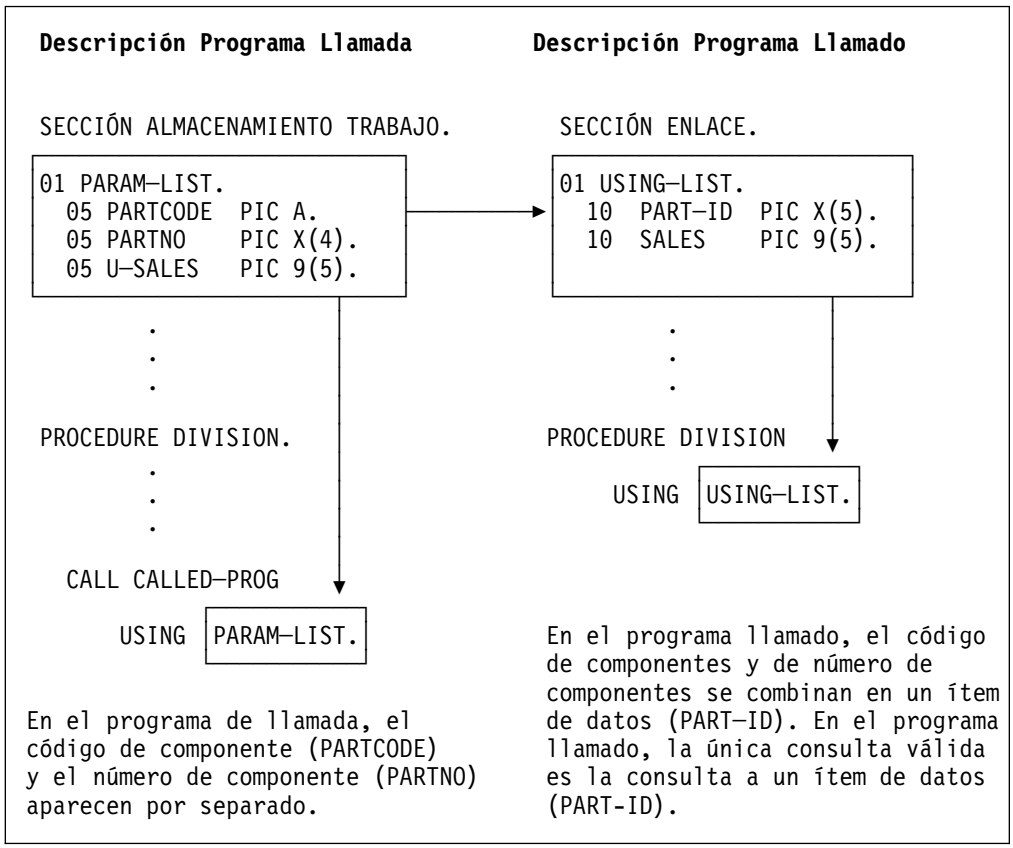


Figura 89. Ítem de Datos Comunes en el Enlace de Subprograma

Llamada por Identificador

Se establece un puntero del sistema que asocia un identificador con un objeto la primera vez que utilice el identificador en una instrucción CALL.

¡Importante para la compatibilidad!

Si realiza una llamada por identificador a un programa que posteriormente suprimirá o redenominará, debe utilizar la instrucción CANCEL para anular el puntero del sistema asociado con el identificador. Ello asegura que cuando utilice otra vez el identificador para llamar al programa, el puntero del sistema asociado se activará de nuevo.

El ejemplo siguiente muestra cómo aplicar la instrucción CANCEL para un identificador:

```
MOVE "ABCD" TO IDENT-1.
CALL IDENT-1.
CANCEL IDENT-1.
```

Si aplica la instrucción CANCEL directamente al literal "ABCD", *no* anula el puntero del sistema asociado con IDENT-1. En su lugar, puede continuar llamando al programa ABCD con sólo utilizar IDENT-1 en la instrucción CALL.

El valor del puntero del sistema también cambia si el usuario modifica el valor del identificador y realiza una llamada utilizando este nuevo valor.

Utilización de Punteros en un Programa COBOL/400

Podrá utilizar un **puntero** (un ítem de datos en el que pueden almacenarse los valores de la dirección) en un programa COBOL siempre que desee pasar y recibir direcciones de un ítem de datos asignado de manera variable, así como realizar direccionamientos de base limitada.

En el sistema AS/400, los punteros tienen una longitud de 16 bytes. Los punteros COBOL son **punteros de espacio** de AS/400, dado que se direccionan a objetos espacio de sistema. Una parte del puntero describe sus atributos (por ejemplo, a qué objeto espacio del AS/400 se está direccionando). Otra parte del puntero contiene el desplazamiento en el objeto espacio del sistema AS/400.

Para definir un puntero COBOL (llamado **ítem de datos puntero**), codifique una cláusula USAGE IS POINTER en el ítem de datos. Un ítem de datos puntero es un ítem que consta de 16 bytes que se puede comparar por igualdad o utilizar para establecer el valor de otros ítems de punteros.

Un ítem de datos de puntero sólo puede utilizarse:

En una instrucción SET (sólo Formato 5)
En una relación de condición
En la frase USING de la instrucción CALL o en la cabecera de la División de Procedimiento.
En el operando de la longitud (LENGTH OF) y dirección (ADDRESS OF) de registros especiales.

Si los punteros se utilizan en una condición relacional, los únicos operadores válidos son igual a o no igual a.

Dado que los ítems de datos punteros no son simples números binarios del sistema AS/400, el manejo de punteros como enteros no funciona.

Los ítems de datos punteros quedan definidos explícitamente con la cláusula USAGE IS POINTER, y son implícitos al utilizar una instrucción de dirección (ADDRESS OF) de registro especial o la dirección (ADDRESS OF) de un ítem.

Si un ítem de grupos se describe mediante la cláusula USAGE IS POINTER, los ítems básicos dentro del ítem de grupos son ítems de punteros. El grupo en sí no es un ítem de datos puntero, y no puede utilizarse en la sintaxis allí donde se permita un ítem de datos puntero. La cláusula USAGE de un ítem elemental no puede contradecir la cláusula USAGE de un grupo al que pertenezca el ítem.

Los ítems de datos puntero pueden formar parte de un grupo referenciado en una instrucción MOVE o en una instrucción de entrada/salida; sin embargo, si un ítem de datos puntero forma parte de un grupo, no se produce la conversión de los valores del puntero a otra forma de representación interna cuando la instrucción se ejecuta.

Definición y Alineación de Punteros

Los ítems de datos puntero pueden definirse a cualquier nivel (excepto el 88) en el archivo, en el almacenamiento de trabajo o en las secciones de enlace de un programa.

Cuando un puntero aparece referenciado en el sistema AS/400, debe estar en un límite de almacenamiento de 16 bits. La **alineación de puntero** se refiere al proceso del compilador de COBOL/400 de colocar los ítems de puntero de un ítem de grupos a desplazamientos que sean múltiplos de 16 bytes desde el principio del registro. Si un ítem de puntero no se encuentra en un límite de 16 bytes, se envía una excepción de alineación de puntero (MCH0602) al programa COBOL/400. En general, las excepciones de alineación de puntero se producen en la Sección de Enlace, donde corresponde al usuario decidir la alineación de dichos ítems.

En las secciones de Archivo y Almacenamiento de Trabajo, el compilador se asegura de que esta excepción no se produzca, añadiendo ítems FILLER implícitos. Cada vez que el compilador añade un ítem FILLER implícito, emite un mensaje de aviso. En la Sección de Enlace, el compilador no añade ningún ítem FILLER implícito; sin embargo, emite mensajes de aviso que indican la cantidad de bytes FILLER que se habrían añadido si el ítem de grupos hubiera aparecido en las secciones de Archivo o de Almacenamiento de Trabajo.

Podrá definir un ítem de datos como un puntero especificando la cláusula USAGE IS POINTER, tal como se muestra en el ejemplo siguiente:

```
       WORKING-STORAGE SECTION.
  77   APTR USAGE POINTER.
  01   AB.
       05 BPTR  USAGE POINTER.
       05 BVAR  PIC S9(3) PACKED-DECIMAL.
 LINKAGE SECTION.
  01   AVAR.
       05 CVAR  PIC X(30).
 PROCEDURE DIVISION.
       SET APTR TO ADDRESS OF AVAR.
```

Figura 90. Definición de un Ítem de Datos Puntero

En el ejemplo anterior, AVAR es un ítem de datos de nivel 01, de modo que la dirección de AVAR (ADDRESS OF AVAR) es la dirección (ADDRESS OF) del registro especial. Dado que un registro especial es un área de almacenamiento real, la instrucción SET traslada el contenido de ADDRESS OF AVAR a un ítem de datos de puntero APTR.

En el ejemplo que hemos visto, si la instrucción SET utilizara ADDRESS OF CVAR, no existiría ningún registro especial. En su lugar, al ítem de datos de puntero APTR se le asigna la dirección calculada de CVAR.

En las Secciones de Archivo y Almacenamiento de Trabajo

En las secciones de Archivo y Almacenamiento de Trabajo, todos los ítems de nivel 01 (y algunos ítems de nivel 66 y 77) se colocan en límites de 16 bytes.

Dentro de una estructura de grupos, los ítems de datos de puntero deben estar también en un límite de 16 bytes. Para asegurarse de ello, el compilador de COBOL/400 añade ítems FILLER inmediatamente antes de los ítems de datos de puntero. Para evitar dichos ítems FILLER, deberá colocar los ítems de datos de puntero al principio del ítem de grupo.

Si el ítem de datos de puntero forma parte de una tabla, el primer ítem de dicha tabla se sitúa en un límite de 16 bytes. Con el fin de asegurarse de que todas las apariciones posteriores se sitúen en un límite de 16 bytes, se añade un ítem FILLER al final de la tabla, siempre que sea necesario.

Un ejemplo de alineación de ítems de datos de puntero sería el siguiente:

```
 WORKING-STORAGE SECTION.
  77  APTR USAGE POINTER.
  01  AB.
      05 ALPHA-NUM PIC X(10).
      05 BPTR  USAGE POINTER.
  01  EF.
      05 ARRAY-1 OCCURS 3 TIMES.
         10 ALPHA-NUM-TWO PIC X(14).
         10 CPTR  USAGE POINTER.
         10 ALPHA-NUM-THREE PIC X(5).
```

Figura 91. Alineación de Ítems de Datos de Puntero

En el ejemplo anterior, APTR es un ítem de datos de puntero. Por lo tanto, el ítem de nivel 77 se sitúa en un límite de 16 bytes. El ítem de grupo AB es un ítem de nivel 01 y se sitúa automáticamente en un límite de 16 bytes. Dentro del ítem de grupo AB, BPTR no está en un límite de 16 bytes. Con el fin de alinearlo convenientemente, el compilador inserta un ítem FILLER de 6 bytes después de ALPHA-NUM. Finalmente, CPTR requiere un FILLER de 2 bytes para alinear su primera aparición. Dado que ALPHA-NUM-THREE tiene sólo una longitud de 5 bytes, debe añadirse otro ítem FILLER de 11 bytes al final ARRAY-1 para alinear todas las apariciones posteriores de CPTR.

Cuando se define un puntero en la Sección de Archivos, y existe un archivo que no tiene agrupaciones en activo, cada ítem de nivel 01 estará en un límite de 16 bytes. Si un archivo tiene bloques en activo, únicamente el primer registro de un bloque tiene la seguridad de estar en un límite de 16 bytes. De esta manera, no deberían definirse ítems de datos de puntero para archivos con bloques en activo. Para más información acerca de los bloques, consulte el apartado “Desbloqueo de Registros de Entrada y Bloqueo de Registros de Salida” en la página 107.

Los Punteros y la Cláusula REDEFINES

Un ítem de datos puntero puede ser el sujeto o el objeto de una cláusula REDEFINES.

Cuando un puntero es el sujeto de una cláusula REDEFINES, el ítem de datos objeto debe estar en un límite de 16 bytes.

Por ejemplo:

```
WORKING-STORAGE SECTION.
 01  AB.
     05 ALPHA-NUM PIC X(16).
     05 APTR REDEFINES ALPHA-NUM USAGE POINTER.
     05 BPTR  USAGE POINTER.
     05 CPTR REDEFINES BPTR USAGE POINTER.
```

Figura 92. REDEFINES e Ítems Alineados de Datos Puntero

En el ejemplo anterior, tanto APTR como CPTR son ítems de datos puntero que vuelven a definir ítems alineados de 16 bytes. En el ejemplo siguiente, el ítem redefinido daría como resultado un error grave de compilador:

```
WORKING-STORAGE SECTION.
 01  EF.
     05 ALPHA-NUM PIC X(5).
     05 HI.
        10 ALPHA-NUM-TWO PIC X(11).
        10 APTR  USAGE POINTER.
     05 BPTR REDEFINES HI USAGE POINTER.
```

Figura 93. REDEFINES e Ítems Alineados de Datos Puntero - Método Incorrecto

En el ejemplo anterior, APTR se alinea en un límite de 16 bytes. Es decir, el compilador de COBOL/400 no necesitaba añadir ítems FILLER para alinear APTR. El ítem de grupos HI no está en el límite de 16 bytes y, por lo tanto, tampoco lo está el ítem de datos puntero BPTR. Dado que el compilador de COBOL/400 no puede añadir ítems FILLER para colocar el BPTR en un límite de 16 bytes, se producirá un error grave. En el ejemplo que se muestra a continuación, similar al anterior, el compilador de COBOL/400 es capaz de colocar el ítem de datos puntero en un límite de 16 bytes:

```
WORKING-STORAGE SECTION.
 01  EF.
     05 ALPHA-NUM PIC X(5).
     05 HI.
        10 ALPHA-NUM-TWO PIC X(11).
        10 APTR  USAGE POINTER.
        10 ALPHA-NUM-THREE PIC X(5).
     05 KL REDEFINES HI.
        10 BPTR USAGE POINTER.
```

Figura 94. REDEFINES e Ítems no Alineados de Datos Puntero - Método Correcto

En el ejemplo anterior, el ítem de grupos KL no está en un límite de 16 bytes; no obstante, el compilador añade un ítem FILLER de 11 bytes antes del ítem de datos puntero BPTR para asegurarse de que se sitúa en un límite de 16 bytes.

Lectura y Grabación de Punteros

Los ítems de datos puntero pueden definirse en la Sección de Archivos, y su utilización y configuración puede ser la misma que la de cualquier ítem de datos puntero del Almacenamiento de Trabajo. Existen, no obstante, algunas restricciones:

- Si un archivo tiene bloques en activo, únicamente el primer registro de un bloque tiene la seguridad de estar en un límite de 16 bytes. Por consiguiente, no deberían definirse ítems de datos puntero para archivos con bloques en activo.
- Un registro que contenga punteros puede grabarse en un archivo; sin embargo, en una lectura ulterior de dicho registro, los ítems de datos puntero dan igual a NULL.

Inicialización de Punteros Utilizando la Constante Figurativa NULL

La constante figurativa NULL representa un valor que se utiliza para indicar que los ítems de datos definidos con USAGE IS POINTER, ADDRESS OF o la dirección (ADDRESS OF) del registro especial no contienen una dirección válida. Por ejemplo:

```
 WORKING-STORAGE SECTION.
  77 APTR  USAGE POINTER VALUE NULL.
 PROCEDURE DIVISION.
     IF APTR = NULL THEN
       DISPLAY 'APTR IS NULL'
     END-IF.
```

Figura 95. Utilización de NULL para Inicializar un Puntero

En el ejemplo anterior, el puntero APTR se establece en NULL en la sección de Almacenamiento de Trabajo. La comparación en la división de procedimiento será verdadera y la instrucción en pantalla se ejecutará.

En el sistema AS/400, el valor inicial de un ítem de datos de puntero, con o sin una cláusula de valor (VALUE) de NULL, da igual a NULL.

Longitud (LENGTH OF) de Registro Especial

La longitud (LENGTH OF) de registro especial contiene el número de bytes utilizado por un identificador. Éste devuelve un valor de 16 para un ítem de datos puntero.

Podrá utilizar la instrucción LENGTH OF en la División de Procedimiento allí donde se utilice un ítem de datos numéricos cuya definición sea la misma que la definición de la longitud (LENGHT OF) del registro especial; sin embargo, LENGHT OF no puede utilizarse como subíndice ni recibir ítems de datos. LENGTH OF tiene la siguiente definición implícita:

```
USAGE IS BINARY, PICTURE 9(9)
```

El siguiente ejemplo muestra de qué manera podrá utilizar LENGHT OF con punteros:

```
WORKING-STORAGE SECTION.
 77  APTR  USAGE POINTER.
 01  AB.
     05  BPTR  USAGE POINTER.
     05  BVAR  PIC S9(3) PACKED-DECIMAL.
     05  CVAR  PIC S9(3) PACKED-DECIMAL.
PROCEDURE DIVISION.
     MOVE LENGTH OF AB TO BVAR.
     MOVE LENGTH OF BPTR TO CVAR.
```

Figura 96. Utilización de LENGTH OF con Punteros

En el ejemplo anterior, la longitud del ítem de grupo AB se traslada a la variable BVAR. BVAR tiene un valor de 20 porque BPTR tiene 16 bytes de longitud, y ambas variables, BVAR y CVAR, tienen dos bytes de longitud. CVAR recibe un valor de 16.

También podrá utilizar el registro especial LENGHT OF para configurar estructuras de datos dentro de espacios de usuario, así como para incrementar direcciones recibidas desde otro programa. Para ver un ejemplo de un programa que utiliza el registro especial LENGTH OF para definir estructuras de datos dentro de espacios de usuario, consulte la Figura 99 en la página 305.

Configuración de la Dirección de Ítems de Enlace

Por norma general, cuando un programa COBOL llama a otro programa, los datos pasan entre los dos programas del siguiente modo: el programa de llamada utiliza la instrucción CALL USING para pasar operandos al programa llamado, y el programa llamado especifica la frase USING en la cabecera de la División de Procedimiento. Tiene que haber una correlación uno a uno entre los operandos de las frases USING de cada programa.

Al utilizar la dirección (ADDRESS OF) del registro especial, ya no será necesario que asegure una correlación una a una entre las frases USING de los dos programas. Para aquellos ítems de datos de la Sección de Enlace que no estén especificados en la frase USING de la cabecera de la División de Procedimiento, el usuario podrá utilizar una instrucción SET para especificar la dirección de inicio de la estructura de datos. Una vez ejecutada la instrucción SET, el ítem de datos se trata como si se hubiera pasado desde otro programa. Para ver un ejemplo de una instrucción SET utilizada de este modo, consulte la Figura 100 en la página 306. El apartado **16** de la página 309 muestra de qué manera la instrucción SET se utiliza para establecer la dirección de inicio de las estructuras de datos *ls-header-record* y *ls-user-space* al comienzo del espacio de usuario.

Utilización de ADDRESS OF y de ADDRESS OF del Registro Especial

Cuando especifique ADDRESS OF en un programa COBOL, el compilador determinará si utilizar la dirección calculada de un ítem de datos, conocida como ADDRESS OF, o la dirección (ADDRESS OF) del registro especial. La dirección (ADDRESS OF) del registro especial es la dirección de inicio de la estructura de datos desde la que se determinan todas las direcciones calculadas. Dado que la dirección (ADDRESS OF) del registro especial es una dirección de inicio de una estructura, debe consistir en un ítem de datos de nivel 01 o de nivel 77. Si modifica la referencia de este ítem de datos, dejará de ser la dirección de inicio de la estructura de datos. Se trata de una dirección calculada o ADDRESS OF. Si toma

la dirección (ADDRESS OF) de un ítem básico, y la dirección (ADDRESS OF) del ítem de nivel 01 se ha establecido en NULL, el resultado será una excepción de puntero (MCH3601).

No podrá utilizar la dirección calculada ADDRESS OF donde un ítem pueda ser susceptible de cambio. Sólo se puede modificar la dirección (ADDRESS OF) del registro especial. Por ejemplo, en la Figura 100, la instrucción SET del apartado **18** de la página 309 utiliza la dirección (ADDRESS OF) de registro especial porque se trata de un ítem de nivel 01. En el apartado **19** de la página 309 se utiliza ADDRESS OF porque, aunque se trata de un ítem de nivel 01, viene modificado por referencia.

Utilización de Punteros en una Instrucción MOVE

Los ítems básicos de datos puntero no se pueden mover mediante la instrucción MOVE, sino que debe utilizarse una instrucción SET. Sin embargo, los ítems de datos puntero se mueven de manera implícita cuando forman parte de un ítem de grupos.

Al compilar una instrucción MOVE, el compilador de COBOL/400 genera un código para mantener (puntero MOVE) o no mantener (no puntero MOVE) punteros dentro de un ítem de grupos.

Se realiza un puntero MOVE cuando se dan todas las condiciones siguientes:

1. El fuente del receptor de una instrucción MOVE contiene un puntero
2. Ambos ítems tienen, como mínimo, 16 bytes de longitud
3. Los ítems de datos están alineados convenientemente
4. Los ítems de datos son alfanuméricos o bien son ítems de grupos.

De todas las condiciones enumeradas anteriormente, la más compleja puede ser la de determinar si los dos ítems de datos están convenientemente alineados.

Si los ítems que se trasladan son ítems de nivel 01, o forman parte de un ítem de nivel 01, deben encontrarse en el mismo desplazamiento en relación a un límite de 16 bytes para que se produzca un puntero MOVE. (Se envía un mensaje de aviso si no es verdadero). El ejemplo siguiente muestra tres estructuras de datos, así como los resultados cuando se envía una instrucción MOVE:

```
WORKING-STORAGE SECTION.
   01  A.
       05  B      PIC X(10).
       05  C.
           10  D      PIC X(6).
           10  E      POINTER.
   01  A2.
       05  B2     PIC X(6).
       05  C2.
           10  D2     PIC X(10).
           10  E2     POINTER.
   01  A3.
       05  B3     PIC X(22).
       05  C3.
           10  D3     PIC X(10).
           10  E3     POINTER.

PROCEDURE DIVISION.
MOVE  A  to A2. 1
MOVE  A  to A3. 1
MOVE  C  to C2. 2
MOVE  C2 to C3. 3
```

1 Da como resultado un movimiento del puntero porque el desplazamiento de cada ítem de grupos que se han de mover es cero. La integridad del puntero se mantiene.

2 Da como resultado un movimiento sin puntero, puesto que los desplazamientos no coinciden. El desplazamiento del ítem de grupos C es 10 y el del ítem del grupos C2 es 6. La integridad del puntero no se mantiene.

3 Da como resultado un movimiento del puntero, puesto que el desplazamiento del grupo de artículos C2 es 6 y el desplazamiento de C3 relativo a un límite de 16 bytes es también de 6. (Cuando el desplazamiento es mayor que 16, el desplazamiento relativo al límite de 16 bytes se calcula dividiendo el desplazamiento entre 16. El recordatorio es el desplazamiento relativo. En este caso, el desplazamiento era 22, lo que, dividido entre 16, da un recordatorio, o desplazamiento relativo, de 6). La integridad del puntero se mantiene.

Si un ítem de grupos contiene un puntero, y el compilador no puede determinar el desplazamiento relativo a un límite de 16 bytes, el compilador envía un mensaje de aviso y se intenta el traslado del puntero. Sin embargo, es posible que la integridad no se mantenga. El compilador no puede determinar el desplazamiento si el ítem se define en la Sección de Enlace, o si el ítem viene modificado por referencia con una posición de inicio desconocida. Deberá asegurarse de que la alineación del puntero se mantiene; de lo contrario, puede darse un MCH0602.

El compilador de COBOL/400 coloca todos los ítems de nivel 01 en un límite de 16 bytes tanto si contienen ítems de datos puntero como si no.

Si uno de los ítems de la instrucción MOVE es un ítem de nivel 01 con un puntero, y el otro es un ítem de Almacenamiento de Trabajo de nivel 77, este último ítem va forzosamente a un límite de 16 bytes.

Utilización de Punteros en una Instrucción CALL

Cuando un ítem de datos puntero se pasa a una instrucción CALL, dicho ítem se trata como todos los ítems USING. Dicho de otro modo, se pasa al programa llamado un puntero al ítem de datos puntero (o una copia del ítem de datos puntero).

Deberá prestarse una especial atención cuando se utilice una instrucción CALL con la frase BY CONTENT para pasar punteros e ítems de grupos que contengan punteros. Es un caso parecido al de la instrucción MOVE. Para una instrucción CALL BY CONTENT, se efectúa una instrucción MOVE implícita de un ítem para crearlo en un área temporal. Si el compilador puede determinar el desplazamiento de un ítem relativo a un límite de 16 bytes, este mismo desplazamiento se utiliza cuando la instrucción implícita MOVE del ítem BY CONTENT se realiza en un área temporal. Cuando el compilador no puede determinar el desplazamiento de un ítem relativo a un límite de 16 bytes, la instrucción implícita MOVE del ítem BY CONTENT se realiza en un área temporal que se alinea en un límite de 16 bytes.

El compilador no puede determinar el desplazamiento de un ítem relativo a un límite de 16 bytes cuando el ítem BY CONTENT:

- Viene modificado por referencia con una posición de inicio desconocida, o
- Está definido en la Sección de Enlace.

Cuando un operando está modificado por referencia, el desplazamiento es la posición de inicio de la modificación de referencia menos uno, más el desplazamiento del operando de la estructura de datos. Cuando un operando se encuentre en la Sección de Enlace, podrá determinarse su desplazamiento desde el programa de llamada.

Para evitar problemas con la alineación del puntero, pase los ítems por referencia.

A continuación aparece un ejemplo de paso de ítems que contienen punteros, en el que su integridad se mantiene en algunos casos y en otros no.

```
WORKING-STORAGE SECTION.

01  A. 1
    05  B    PIC X(3).
    05  C. 2
        10  FILLER   PIC X(13).
        10  D        POINTER.

PROCEDURE DIVISION.

CALL "B" USING A C.
```

Figura 97. Programa A -- Programa Principal

```
WORKING-STORAGE SECTION.

01  E.
    05  F    PIC X(16).
    05   G   POINTER.
77  K   PIC S9(3)    VALUE 8.

LINKAGE SECTION.

01  A. 3
    05  B    PIC X(3).
    05  C.
        10   FILLER   PIC X(13).
        10   D   POINTER.
01  C2. 4
    05  FILLER   PIC X(13).
    05  D2   POINTER.

PROCEDURE DIVISION USING A C2.

CALL "C" USING BY CONTENT
      A, C2, 5  E(5: ), 6  E(K: ), 7  F.  8
```

Figura 98. Programa B -- Subprograma

En el ejemplo anterior, el Programa A pasa dos ítems de grupo al Programa B. 1 es un ítem de grupo de nivel 01, con un desplazamiento cero. 2 es un ítem de grupo de nivel 05 y tiene un desplazamiento de 3. Dado que los ítems se pasan por referencia, la integridad del puntero se mantiene para los dos ítems de grupos, A y C.

El programa B pasa cinco ítems a otro programa, C. Los ítems se pasan, por contenido, al Programa C. Puesto que se pasan por contenido, el Programa C recibe una copia de los ítems, y la integridad del puntero no se mantiene en todos los casos.

- 3 Dado que este ítem está definido en la Sección de Enlace, tiene un desplazamiento desconocido. El compilador asume que se alinea a 16 bytes; en este caso, cuando A se pasa, la integridad del puntero de D se mantiene, pero se envía un mensaje de aviso del compilador en la instrucción CALL.
- 4 Este ítem contiene un puntero, y un movimiento del puntero se realiza mediante 5 . Sin embargo, puesto que el ítem está definido en la Sección de Enlace y el desplazamiento es desconocido, la integridad del puntero no se mantiene. El compilador intenta mover C2 a un área de alineación de 16 bytes, y se envía un mensaje de aviso del compilador.
- 6 Puesto que E contiene un puntero, se produce un movimiento de puntero. El desplazamiento se puede calcular porque la posición de inicio de la modificación de referencia es un literal numérico. En este caso, la integridad del puntero se mantiene y el ítem se sitúa en un desplazamiento de 4 desde el límite de 16 bytes.
- 7 Dado que E contiene un puntero, se intenta un movimiento de puntero. Ya que E está modificado por referencia con una posición de inicio desconocida (K), el compilador no puede calcular el desplazamiento y asume que está alineado en un límite de 16 bytes. Se envía un mensaje de aviso del compilador.

Si el valor de K provoca que E se alinee en un límite de 16 bytes, la integridad del puntero se mantiene. Para que esto ocurra, K debe ser 1 ó 17.

- **8** F es un ítem definido en la Sección de Almacenamiento de Trabajo y no contiene punteros, de manera que no son de esperar movimientos de puntero.

Utilización de Punteros y API para Acceder a Espacios de Usuario

El ejemplo siguiente muestra de qué manera puede utilizar punteros para acceder a espacios de usuario y encadenar registros.

POINTA es un programa que lee los nombres y direcciones de los clientes en un espacio de usuario y a continuación visualiza la información en una lista. El programa asume que la información del cliente existe en un archivo llamado POINTACU.

El campo de la dirección del cliente es un campo de longitud variable, lo que permite la inserción de direcciones largas.

```
A* ESTE ES UN ARCHIVO DE INFORMACIÓN DEL CLIENTE - POINTACUST
A
A
A          R FSCUST                    TEXT('CUSTOMER MASTER RECORD')
A            FS_CUST_NO     8S00       TEXT('CUSTOMER NUMBER')
A                                       ALIAS(FS_CUST_NUMBER)
A            FS_CUST_NM     20         TEXT('CUSTOMER NAME')
A                                       ALIAS(FS_CUST_NAME)
A            FS_CUST_AD    100         TEXT('CUSTOMER ADDRESS')
A                                       ALIAS(FS_CUST_ADDRESS)
A                                       VARLEN
```

Figura 99. Ejemplo de Utilización de Punteros para Acceder a Espacios de Usuario -- DDS

```
5763CB1 V3R0M5  001000            IBM SAA COBOL/400             TESTER/POINTA         AS400SYS  05/01/94 18:01:14     Página  1
Programa . . . . . . . . . . . . . . :   POINTA
  Biblioteca . . . . . . . . . . . . :     TESTER
Archivo fuente . . . . . . . . . . . :   QLBLSRC
  Biblioteca . . . . . . . . . . . . :     TESTER
Miembro fuente . . . . . . . . . . . :   POINTA     05/01/94 17:55:27
Nivel de gravedad de generación  . . :   29
"Descripción" del texto. . . . . . . :   *BLANK
Opciones de listado fuente . . . . . :   *NONE
Opciones de generación . . . . . . . :   *NONE
Opciones de conversión . . . . . . . :   *NONE
Límite de mensaje:
  Cantidad de mensajes.. . . . . . . :   *NOMAX
  Gravedad límite de mensaje   . . . :   29
Imprimir archivo. . . . . . . . . . :   QSYSPRT
  Biblioteca . . . . . . . . . . . . :     *LIBL
Señalización FIPS. . . . . . . . . . :   *NOFIPS *NOSEG *NODEB *NOOBSOLETE
Señalización SAA . . . . . . . . . . :   *NOFLAG
Opciones de visualización ampliada . :
Gravedad de señalización . . . . . . :   0
Sustituir programa . . . . . . . . . :   *YES
Release de destino . . . . . . . . . :   *CURRENT
Perfil de usuario. . . . . . . . . . :   *USER
Autorización . . . . . . . . . . . . :   *LIBCRTAUT
Compilador . . . . . . . . . . . . . :   IBM SAA COBOL/400
Pantalla de Información del Cliente 1
5763CB1 V3R0M5  001000          Fuente COBOL AS/400             TESTER/POINTA         AS400SYS  05/01/94 18:01:14     Página  2
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
    1  000010 PROCESS extaccdsp varchar 2
    2  000020 ID DIVISION.                                              CBT00010

       000040* Este programa lee un archivo de registros de longitud variable
       000050*  en un espacio de usuario. A continuación muestra los
       000060*  registros en la pantalla.
    3  000070 PROGRAM-ID. pointa.
    4  000080 ENVIRONMENT DIVISION.
    5  000090 CONFIGURATION SECTION.
    6  000100 SPECIAL-NAMES. CONSOLE IS CRT,
    7  000110                CRT STATUS IS ws-crt-status. 3
    8  000120 INPUT-OUTPUT SECTION.
    9  000130 FILE-CONTROL.
   10  000140     SELECT cust-file ASSIGN TO DATABASE-pointacu
   11  000150            ORGANIZATION IS SEQUENTIAL
   12  000160            FILE STATUS IS ws-file-status.
   13  000170 DATA DIVISION.
   14  000180 FILE SECTION.
   15  000190 FD  cust-file.
   16  000200 01  fs-cust-record.
       000210* copiar nombres de campos transformando los subrayados
       000220*  en guiones y utilizando nombres de alias
   17  000230 COPY DDR-ALL-FORMATS-I OF pointacu.
   18 +000001       05  POINTACU-RECORD PIC X(130).                                  <-ALL-FMTS
      +000002*    FORMATO E-S:FSCUST    DESDE ARCHIVO POINTACU   DE BIBLO TESTER    <-ALL-FMTS
      +000003*                          CUSTOMER MASTER RECORD                      <-ALL-FMTS
   19 +000004       05  FSCUST        REDEFINES POINTACU-RECORD.                    <-ALL-FMTS
   20 +000005           06 FS-CUST-NUMBER        PIC S9(8).                        <-ALL-FMTS
      +000006*                          CUSTOMER NUMBER                             <-ALL-FMTS
   21 +000007           06 FS-CUST-NAME          PIC X(20).                        <-ALL-FMTS
      +000008*                          CUSTOMER NAME                               <-ALL-FMTS
   22 +000009           06 FS-CUST-ADDRESS. 4                                       <-ALL-FMTS
      +000010*               (Campo de longitud variable)                           <-ALL-FMTS
   23 +000011              49 FS-CUST-ADDRESS-LENGTH                               <-ALL-FMTS
   24 +000012                                 PIC S9(4) COMP-4.                    <-ALL-FMTS
   25 +000013              49 FS-CUST-ADDRESS-DATA                                 <-ALL-FMTS
   26 +000014                                 PIC X(100).                          <-ALL-FMTS
      +000015*                          DIRECCIÓN DEL CLIENTE                       <-ALL-FMTS
   27  000240 WORKING-STORAGE SECTION.
   28  000250 01  ws-file-status.
   29  000260     05 ws-file-status-1 PIC X.
   30  000270        88  ws-file-stat-good   VALUE "0".
   31  000280        88  ws-file-stat-at-end VALUE "1".
   32  000290     05 ws-file-status-2 PIC X.
   33  000300 01  ws-crt-status. 5
   34  000310     05 ws-status-1           PIC 9(2).
   35  000320        88 ws-status-1-ok       VALUE 0.
   36  000330        88 ws-status-1-func-key VALUE 1.
```

Figura 100 (Parte 1 de 7). Ejemplo de Utilización de Punteros para Acceder a Espacios de Usuario

```
Pantalla de Información del Cliente
 5763CB1 V3R0M5  001000            Fuente COBOL AS/400              TESTER/POINTA         AS400SYS  05/01/94 18:01:14     Página  3
  INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
    37  000340        88 ws-status-1-error    VALUE 9.
    38  000350     05 ws-status-2           PIC 9(2).
    39  000360        88 ws-func-03           VALUE 3.
    40  000370        88 ws-func-07           VALUE 7.
    41  000380        88 ws-func-08           VALUE 8.
    42  000390     05 ws-status-3           PIC 9(2).
    43  000400 01  ws-params.  6
    44  000410     05  ws-space.
    45  000420       10  ws-space-name      PIC X(10) VALUE "MYSPACE".
    46  000430       10  ws-space-lib       PIC X(10) VALUE "QTEMP".
    47  000440     05  ws-attr              PIC X(10) VALUE "PF".
    48  000450     05  ws-init-size         PIC S9(5) VALUE 32000 BINARY.
    49  000460     05  ws-init-char         PIC X     VALUE SPACE.
    50  000470     05  ws-auth              PIC X(10) VALUE "*ALL".
    51  000480     05  ws-text              PIC X(50) VALUE
    52  000490     "Registros de Información del Cliente".
    53  000500     05  ws-replace           PIC X(10) VALUE "*YES".
    54  000510     05  ws-err-data.  7
    55  000520         10 ws-input-l        PIC S9(6) BINARY VALUE ZERO.
    56  000530         10 ws-output-l       PIC S9(6) BINARY VALUE ZERO.
    57  000540         10 ws-exception-id   PIC X(7).
    58  000550         10 ws-reserved       PIC X(1).
    59  000560         10 ws-exception-data PIC X(87).
    60  000570     05  ws-space-ptr         POINTER.  8
    61  000580     05  ws-map-ptr           POINTER.
        000590
    62  000600 77  ws-accept-data           PIC X.
    63  000610   88 ws-acc-create-space      VALUE "Y", "y".
    64  000620   88 ws-acc-delete-space      VALUE "Y", "y".
    65  000630   88 ws-acc-no-space          VALUE "N", "n".
        000640
    66  000650 77  ws-prog-indicator        PIC X   VALUE "G".
    67  000660   88  ws-prog-continue       VALUE "G".
    68  000670   88  ws-prog-end            VALUE "C".
    69  000680   88  ws-prog-loop           VALUE "L".
        000690
    70  000700 77  ws-line                  PIC S99.
        000710* línea de mensajes de error
    71  000720 77  ws-error-msg             PIC X(50) VALUE SPACES.
        000730* más indicadores de información de dirección
    72  000740 77  ws-plus                  PIC X.
        000750* longitud la información de dirección a visualizar
    73  000760 77  ws-temp-size             PIC 9(2).
        000770
    74  000780 77  ws-current-rec           PIC S9(4) VALUE 1.
    75  000790 77  ws-old-rec               PIC S9(4) VALUE 1.
    76  000800 77  ws-old-space-ptr         POINTER.
        000810* cantidad máxima de líneas a visualizar
    77  000820 77  ws-displayed-lines       PIC S99 VALUE 20.
        000830* línea en la que iniciar la visualización de registros
    78  000840 77  ws-start-line            PIC S99 VALUE 5.
        000850* variables para crear un registro nuevo en el espacio
    79  000860 77  ws-addr-inc              PIC S9(4) PACKED-DECIMAL.
    80  000870 77  ws-temp                  PIC S9(4) PACKED-DECIMAL.
    81  000880 77  ws-temp-2                PIC S9(4) PACKED-DECIMAL.
        000890* puntero al registro anterior
    82  000900 77  ws-cust-prev-ptr         POINTER VALUE NULL.
    83  000910 LINKAGE SECTION.
    84  000920 01 ls-header-record.  9
    85  000930    05  ls-hdr-cust-ptr            USAGE POINTER.
        000940* cantidad de registros leídos desde el archivo
    86  000950    05  ls-record-counter          PIC S9(3) BINARY.
    87  000960    05  FILLER                     PIC X(14).  10
    88  000970 01 ls-user-space.  11
    89  000980     05  ls-customer-rec.
        000990* puntero al registro anterior del cliente
    90  001000         10 ls-cust-prev-ptr       USAGE POINTER.
    91  001010         10 ls-cust-rec-length     PIC S9(4) BINARY.
    92  001020         10 ls-cust-name           PIC X(20).
    93  001030         10 ls-cust-number         PIC S9(8).
```

Figura 100 (Parte 2 de 7). Ejemplo de Utilización de Punteros para Acceder a Espacios de Usuario

```
Pantalla de Información del Cliente
 5763CB1 V3R0M5  001000            Fuente COBOL AS/400              TESTER/POINTA         AS400SYS  05/01/94 18:01:14     Página  4
  INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
       001040* longitud total del registro incluyendo bytes de relleno para
       001050*  asegurarse de que el siguiente registro está en el límite de 16 bytes
    94 001060         10 ls-cust-address-length  PIC S9(4) BINARY.
    95 001070     05  ls-cust-address-data       PIC X(116).
       001080
       001090* Tamaño de ls-user-space es 16 más grande de lo necesario.
       001100*   Así pues, la dirección de inicio del siguiente registro
       001110*   permite establecerlo sin exceder el tamaño declarado
       001120* El tamaño es 16 más grande para permitir la alineación del puntero
       001130
    96 001140 PROCEDURE DIVISION.
       001150* nota no necesaria para la entrada "USING" en PROC... DIV.
       001160 DECLARATIVES.
       001170 cust-file-para SECTION.
       001180     USE AFTER ERROR PROCEDURE ON cust-file.
       001190 cust-file-para-2.
    97 001200     MOVE "Error XX on file pointacu" TO ws-error-msg.
    98 001210     MOVE ws-file-status TO ws-error-msg(7:2).
       001220 END DECLARATIVES.
       001230 main-section section.
       001240 main-proc.
       001250* siga leyendo la pantalla inicial hasta que los datos sean correctos
    99 001260     SET ws-prog-loop to TRUE.
   100 001270     PERFORM initial-display THRU read-initial-display
       001280         UNTIL NOT ws-prog-loop.
       001290* si desea continuar el programa y quiere crear un área de
       001300*   información de cliente, rellene el espacio con
       001310*   registros del archivo del cliente
   101 001320     IF ws-prog-continue and
       001330        ws-acc-create-space THEN
   102 001340       PERFORM read-customer-file
   103 001350       MOVE 1 TO ws-current-rec
       001360* establezca ptr en el registro de cabecera
   104 001370       SET ADDRESS OF ls-header-record TO ws-space-ptr
       001380* establezca en el primer registro del cliente en espacio
   105 001390       SET ADDRESS OF ls-user-space TO ls-hdr-cust-ptr
       001400     END-IF.
   106 001410     IF ws-prog-continue THEN
   107 001420       PERFORM main-loop UNTIL ws-prog-end
       001430     END-IF.
       001440 end-program.
   108 001450     PERFORM clean-up.
   109 001460     STOP RUN.
       001470 initial-display.  12
   110 001480     DISPLAY "Crear Área de Información del Cliente" AT 0118 WITH
       001490                BLANK SCREEN REVERSE-VIDEO
       001500              "Crear área de información del cliente (Y/N)=>   <="
       001510                 AT 1015
       001520              "F3=Salir" AT 2202.
   111 001530     IF ws-error-msg NOT = SPACES THEN
   112 001540       DISPLAY ws-error-msg at 2302 with beep highlight
   113 001550       MOVE SPACES TO ws-error-msg
       001560     END-IF.
       001570 read-initial-display.  13
   114 001580     ACCEPT ws-accept-data AT 1056 WITH REVERSE-VIDEO
       001590       ON EXCEPTION
   115 001600         IF ws-status-1-func-key THEN
   116 001610           IF ws-func-03 THEN
   117 001620             SET ws-prog-end TO TRUE
       001630           ELSE
   118 001640             MOVE "Tecla de Función Incorrecta" TO ws-error-msg
       001650           END-IF
       001660         ELSE
   119 001670           MOVE "Error Desconocido" TO we-error-msg
       001680         END-IF
       001690       NOT ON EXCEPTION
```

Figura 100 (Parte 3 de 7). Ejemplo de Utilización de Punteros para Acceder a Espacios de Usuario

```
Pantalla de Información del Cliente
 5763CB1 V3R0M5  001000            Fuente COBOL AS/400              TESTER/POINTA         AS400SYS  05/01/94 18:01:14     Página  5
   120  001700         IF ws-acc-create-space THEN
   121  001710           PERFORM create-space THRU get-space
   122  001720           SET ws-prog-continue TO TRUE
        001730         ELSE
   123  001740           IF NOT ws-acc-no-space THEN
   124  001750             MOVE "Entrado Carácter incorrecto" TO ws-error-msg
        001760           ELSE
   125  001770             SET ws-prog-continue TO TRUE
   126  001780             PERFORM get-space
        001790           END-IF
        001800         END-IF
        001810     END-ACCEPT.
        001820 create-space.
   127  001830     CALL "QUSCRTUS"  14
        001840             USING ws-space, ws-attr, ws-init-size,
        001850                   ws-init-char, ws-auth, ws-text,
        001860                   ws-replace, ws-err-data.
        001870* compruebe los errores al crear espacio
        001880 get-space.
   128  001890     CALL "QUSPTRUS" USING ws-space, ws-space-ptr.  15
        001900* establezca el registro de cabecera al comienzo del espacio
   129  001910     SET ADDRESS OF ls-header-record  16
        001920         ADDRESS OF ls-user-space  17
        001930         TO ws-space-ptr.
        001940* establezca el primer registro del cliente después del registro de cabecera
   130  001950     SET ADDRESS OF ls-user-space TO  18
        001960         ADDRESS OF ls-user-space(LENGTH OF ls-header-record  19
        001970                                     + 1:1).
        001980* guarde ptr para el primer registro del registro de cabecera
   131  001990     SET ls-hdr-cust-ptr TO ADDRESS OF ls-user-space.
        002000 delete-space.
   132  002010     CALL "QUSDLTUS" USING ws-space, ws-err-data.  20
        002020 read-customer-file.
        002030* lea todos los registros del archivo del cliente y colóquese en el espacio
   133  002040     OPEN INPUT cust-file.
   134  002050     IF ws-file-stat-good THEN
   135  002060       READ cust-file AT END CONTINUE
   136  002070       END-READ
   137  002080       PERFORM VARYING ls-record-counter FROM 1 BY 1
        002090             UNTIL not ws-file-stat-good
   138  002100         SET  ls-cust-prev-ptr  TO ws-cust-prev-ptr
        002110*  Coloque la información del archivo al espacio
   139  002120         MOVE fs-cust-name      TO ls-cust-name
   140  002130         MOVE fs-cust-number    TO ls-cust-number
   141  002140         MOVE fs-cust-address-length  TO ls-cust-address-length
   142  002150         MOVE fs-cust-address-data(1:fs-cust-address-length)
        002160              TO ls-cust-address-data(1:ls-cust-address-length)
        002170*  Guarde ptr para el registro actual
   143  002180         SET  ws-cust-prev-ptr  TO ADDRESS OF ls-user-space
        002190*  asegúrese de que el siguiente registro está en el límite de 16 bytes
   144  002200         ADD LENGTH OF ls-customer-rec  21
        002210             ls-cust-address-length TO 1 GIVING ws-addr-inc
   145  002220         DIVIDE ws-addr-inc BY 16 GIVING ws-temp
        002230             REMAINDER ws-temp-2
   146  002240         SUBTRACT ws-temp-2 FROM 16 GIVING ws-temp
        002250*  Guarde la longitud total del registro en el espacio del usuario
   147  002260         ADD ws-addr-inc TO ws-temp GIVING ls-cust-rec-length
   148  002270         SET ADDRESS OF ls-user-space
        002280          TO ADDRESS OF ls-user-space(ls-cust-rec-length + 1:1)
        002290*  Obtenga el siguiente registro del archivo
   149  002300         READ cust-file AT END CONTINUE
   150  002310         END-READ
        002320       END-PERFORM
        002330*  Al final del bucle coloque un registro más de los
        002340*   que debe haber
   151  002350       SUBTRACT 1 FROM ls-record-counter
        002360     END-IF.
   152  002370     CLOSE cust-file.
        002380
        002390 main-loop.  22
        002400* escriba los registros en la pantalla hasta que se pulse F3
```

Figura 100 (Parte 4 de 7). Ejemplo de Utilización de Punteros para Acceder a Espacios de Usuario

```
Pantalla de Información del Cliente
 5763CB1 V3R0M5  001000           Fuente COBOL AS/400             TESTER/POINTA         AS400SYS  05/01/94 18:01:14     Página  6
   153  002410     DISPLAY "Información del cliente" AT 0124 WITH
        002420                BLANK SCREEN REVERSE-VIDEO
        002430             "Cust     Customer Name        Customer"
        002440                AT 0305
        002450             " Address"
        002460             "Number"   AT 0405
        002470             "F3=Salir" AT 2202.
        002480* si aparece un error pendiente en la pantalla
   154  002490     IF ws-error-msg NOT = SPACES THEN
   155  002500       DISPLAY ws-error-msg at 2302 with beep highlight
   156  002510       MOVE SPACES TO ws-error-msg
        002520     END-IF.
        002530* si en la mitad de la lista se pulsa F7 en la pantalla
   157  002540     IF ws-current-rec > 1 THEN 23
   158  002550       DISPLAY "F7=Back" AT 2240
        002560     END-IF.
        002570* guarde el registro actual
   159  002580     MOVE ws-current-rec TO ws-old-rec.
   160  002590     SET  ws-old-space-ptr TO ADDRESS OF ls-user-space. 24
        002600* traslade cada registro a la pantalla
   161  002610     PERFORM VARYING ws-line FROM  ws-start-line BY 1
        002620         UNTIL ws-line > ws-displayed-lines or
        002630               ws-current-rec > ls-record-counter
        002640* si la dirección es mayor que el ancho de pantalla "+"
   162  002650         IF ls-cust-address-length > 40 THEN
   163  002660           MOVE "+" TO ws-plus
   164  002670           MOVE 40 TO ws-temp-size
        002680         ELSE
   165  002690           MOVE ls-cust-address-length TO ws-temp-size
   166  002700           MOVE SPACE TO ws-plus
        002710         END-IF
   167  002720         DISPLAY ls-cust-number at line ws-line column 5
        002730                 ls-cust-name ls-cust-address-data with
        002740                   size ws-temp-size ws-plus at line
        002750                   ws-line column 78
        002760* coloque el siguiente registro en la pantalla
   168  002770         ADD 1 TO ws-current-rec
   169  002780         SET ADDRESS OF ls-user-space
        002790          TO ADDRESS OF ls-user-space
        002800            (ls-cust-rec-length + 1:1)
        002810     END-PERFORM.
        002820* si se puede seguir pulse F8 en la pantalla
   170  002830     IF ws-current-rec < ls-record-counter THEN 23
   171  002840       DISPLAY "F8=Av Pág" AT 2250
        002850     END-IF.
        002860* compruebe si continúa, salga u obtenga nuevos registros
        002870*   o registros anteriores
   172  002880     ACCEPT ws-accept-data WITH SECURE 25
        002890       ON EXCEPTION
   173  002900         IF ws-status-1-func-key THEN
   174  002910           IF ws-func-03 THEN
   175  002920             SET ws-prog-end TO TRUE
        002930           ELSE
   176  002940            IF ws-func-07 THEN
   177  002950             PERFORM back-screen
        002960            ELSE
   178  002970            IF ws-func-08 THEN
   179  002980             PERFORM forward-screen
        002990            ELSE
   180  003000             MOVE "Invalid Function Key" TO ws-error-msg
   181  003010             MOVE ws-old-rec TO ws-current-rec
   182  003020             SET ADDRESS OF ls-user-space TO ws-old-space-ptr
        003030            END-IF
        003040           END-IF
        003050         ELSE
   183  003060           MOVE "Unknown Error" TO ws-error-msg
   184  003070           MOVE ws-old-rec TO ws-current-rec
   185  003080           SET ADDRESS OF ls-user-space TO ws-old-space-ptr
        003090         END-IF
        003100       NOT ON EXCEPTION
   186  003110         MOVE ws-old-rec TO ws-current-rec
   187  003120         SET ADDRESS OF ls-user-space TO ws-old-space-ptr
        003130     END-ACCEPT.
        003140 clean-up.
        003150* limpie el programa
```

Figura 100 (Parte 5 de 7). Ejemplo de Utilización de Punteros para Acceder a Espacios de Usuario

```
Pantalla de Información del Cliente
 5763CB1 V3R0M5  001000            Fuente COBOL AS/400              TESTER/POINTA         AS400SYS  05/01/94 18:01:14     Página  7
       003160* siga leyendo la pantalla final hasta que los datos sean correctos
   188  003170     SET ws-prog-loop to TRUE.
   189  003180     PERFORM end-display THRU read-end-display  26
       003190         UNTIL NOT ws-prog-loop.
       003200 end-display.
   190  003210     DISPLAY "Eliminar Área de Información del Cliente" AT 0118 WITH  27
       003220                BLANK SCREEN REVERSE-VIDEO
       003230              "Eliminar área de información del cliente (Y/N)=>   <="
       003240                 AT 1015
       003250              "F3=Salir" AT 2202.
   191  003260     IF ws-error-msg NOT = SPACES THEN
   192  003270       DISPLAY ws-error-msg at 2302 with beep highlight
   193  003280       MOVE SPACES TO ws-error-msg
       003290     END-IF.
       003300 read-end-display.
   194  003310     ACCEPT ws-accept-data AT 1056 WITH REVERSE-VIDEO
       003320       ON EXCEPTION
   195  003330         IF ws-status-1-func-key THEN
   196  003340           IF ws-func-03 THEN
   197  003350             SET ws-prog-end TO TRUE
       003360           ELSE
   198  003370             MOVE "Tecla de Función Incorrecta" TO ws-error-msg
       003380           END-IF
       003390         ELSE
   199  003400           MOVE "Error Desconocido" TO we-error-msg
       003410         END-IF
       003420       NOT ON EXCEPTION
   200  003430         IF ws-acc-delete-space THEN
   201  003440           PERFORM delete-space
   202  003450           SET ws-prog-continue TO TRUE
       003460         ELSE
   203  003470           IF NOT ws-acc-no-space THEN
   204  003480             MOVE "Entrado Carácter incorrecto" TO ws-error-msg
       003490           ELSE
   205  003500             SET ws-prog-continue TO TRUE
       003510           END-IF
       003520         END-IF
       003530     END-ACCEPT.
       003540 back-screen.  28
   206  003550     IF ws-old-rec <= 1 THEN
   207  003560       MOVE "Top of customer records" TO ws-error-msg
   208  003570       MOVE ws-old-rec TO ws-current-rec  29
   209  003580       SET ADDRESS OF ls-user-space TO ws-old-space-ptr
       003590     ELSE
   210  003600       MOVE ws-old-rec TO ws-current-rec  29
   211  003610       SET ADDRESS OF ls-user-space TO ws-old-space-ptr
   212  003620       PERFORM VARYING ws-line FROM ws-start-line BY 1
       003630         UNTIL ws-line > ws-displayed-lines or
       003640               ws-current-rec <= 1
       003650* Efectúa la copia de seguridad de un registro a la vez
   213  003660         SET ws-cust-prev-ptr TO ls-cust-prev-ptr
   214  003670         SET ADDRESS OF ls-user-space TO ws-cust-prev-ptr  30
   215  003680         SUBTRACT 1 FROM ws-current-rec
       003690       END-PERFORM
       003700     END-IF.
       003710 forward-screen.  31
       003720* si el registro actual es igual o mayor que los errores de impersión
       003730*   de registro máximo, alcance los registros máximos
   216  003740     IF ws-current-rec >= ls-record-counter
   217  003750       MOVE "No more customer records" TO ws-error-msg
   218  003760       MOVE ws-old-rec TO ws-current-rec
   219  003770       SET ADDRESS OF ls-user-space TO ws-old-space-ptr
       003780     ELSE
   220  003790       MOVE ws-current-rec TO ws-old-rec
   221  003800       SET ws-old-space-ptr TO ADDRESS OF ls-user-space
       003810     END-IF.
                          * * * * *  F I N  D E  F U E N T E  * * * * *
Pantalla de Información del Cliente
 5763CB1 V3R0M5  001000            Mensajes COBOL AS/400            TESTER/POINTA         AS400SYS  05/01/94 18:01:14     Página   8
  INST
*   15  MSGID: LBL0650  GRAVEDAD: 00  NUMSEC:  000190
       Mensaje . . . . :   Bloqueo/Desbloqueo para archivo 'CUST-FILE'
         se realizará por medio del código generado por el compilador.
                    * * * * *  F I N  D E  M E N S A J E S  * * * * *
```

Figura 100 (Parte 6 de 7). Ejemplo de Utilización de Punteros para Acceder a Espacios de Usuario

```
                                    Resumen Mensajes
 Total    Info(0-4)    Aviso(5-19)      Error(20-29)    Grave(30-39)    Terminal(40-99)
     1          1             0                0               0               0
Registros fuente leídos . . . . . :   381
Registros de copia leídos . . . . . :   15
Miembros de copia procesados  . . . :   1
Errores de secuencia  . . . . . :   0
Mensaje de gravedad emitido más alto:   0
 LBL0901 00  Programa POINTA creado en biblioteca TESTER.
               * * * * *  F I N  D E  C O M P I L A C I O N  * * * * *
```

Figura 100 (Parte 7 de 7). Ejemplo de Utilización de Punteros para Acceder a Espacios de Usuario

1 La directriz del compilador TITLE se utiliza para crear el título que aparece al principio de cada página.

3 CRT STATUS IS especifica un nombre de datos en el que se coloca un valor de estado después de la finalización de una instrucción ACCEPT ampliada. En este ejemplo, el valor de STATUS se utiliza para determinar la tecla de función que se ha pulsado.

4 *fs-cust-address* es un campo de longitud variable. Para ver nombres con significado real, en lugar de una instrucción FILLER, especifique *VARCHAR para el parámetro CVTOPT del mandato CRTCBLPGM, o VARCHAR en la instrucción PROCESS, tal como se muestra en el apartado **2** . Para más información acerca de los campos de longitud variable, consulte el apartado “Declaración de Ítems de Datos utilizando Tipos de Datos de CVTOPT” en la página 137.

5 En este apartado se define CRT STATUS tal como se menciona en el apartado **3** .

6 La estructura *ws-params* contiene los parámetros utilizados al llamar a las API para acceder a espacios de usuario.

7 *ws-err-data* es la estructura del parámetro de error para las API del espacio de usuario. Observe que *ws-input-l* es cero, lo que significa que toda excepción está señalada y no se pasa en el parámetro de códigos de error. Para más información acerca de los parámetros de códigos de error, consulte la publicación *System Programmer's Interface Reference*.

8 *ws-space-ptr* define un ítem de datos puntero establecido por la API QUSPTRUS. Esta estructura se direcciona al inicio del espacio de usuario y se utiliza para configurar las direcciones de los ítems en la Sección de Enlace.

9 La primera estructura de datos (*ls-header-record*) que se va a definir en el espacio de usuario.

10 FILLER se utiliza para mantener la alineación del puntero, porque convierte a *ls-header-record* en un múltiplo de 16 bytes de longitud.

11 La segunda estructura de datos (*ls-user-space*) que se va a definir en el espacio de usuario.

12 *initial-display* muestra la pantalla Create Customer Information Area (Crear área de información del cliente). Esta pantalla aparece en la Figura 101 en la página 314.

13 *read-initial-display* lee la primera pantalla y determina si el usuario elige continuar o concluir el programa. Si el usuario continúa el programa pulsando Intro, el programa comprobará la estructura *ws-accept-data* para ver si el área de información del cliente se va a crear.

14 QUSCRTUS es una API utilizada para crear espacios de usuario.

15 QUSPTRUS es una API utilizada para devolver un puntero al inicio de un espacio de usuario.

16 Correlaciona la primera estructura de datos (*ls-header-record*) sobre el inicio del espacio de usuario.

17 Correlaciona la segunda estructura de datos (*ls-user-space*) sobre el inicio del espacio de usuario.

18 Utiliza la dirección (ADDRESS OF) del registro especial.

19 Utiliza ADDRESS OF, no la dirección (ADDRESS OF) del registro especial, porque está modificado por referencia.

20 QUSDLTUS es una API utilizada para suprimir un espacio de usuario.

21 Las cuatro instrucciones aritméticas siguientes calculan la longitud total de cada registro y aseguran que cada registro sea un múltiplo de 16 bytes de longitud.

22 *main-loop* contiene la pantalla Información del Cliente. Consulte la Figura 102 en la página 314.

23 Estas instrucciones determinan si el programa debe visualizar las teclas de función F7 y F8.

24 Guarda un puntero al primer registro de cliente de la pantalla.

25 Esta instrucción ACCEPT espera la entrada de la pantalla de Información del Cliente. Partiendo de la tecla de función pulsada, llama al párrafo adecuado para visualizar el siguiente conjunto de registros (*forward-screen*), o el anterior conjunto de registros (*back-screen*), o establece un indicador para finalizar la rutina si se pulsa F3.

26 La rutina de borrado visualiza la pantalla Suprimir Área de Información del Cliente hasta que se pulsa la tecla adecuada.

27 Esta instrucción contiene la pantalla Suprimir Área de Información de Cliente.

28 Cada registro contiene un puntero al registro de cliente anterior. La dirección (ADDRESS OF) del registro especial se direcciona al registro de cliente actual. Al cambiar la dirección (ADDRESS OF) del registro especial, cambia el registro de cliente actual.

back-screen retrasa el puntero de registro actual un registro a la vez **30**, moviendo el puntero del registro de cliente anterior al puntero del registro del cliente actual (ADDRESS OF). Antes de retrasar un registro a la vez, el programa establece el registro de cliente actual en el primer registro visualizado actualmente **29**.

31 *Forward-screen* establece la estructura *ws-old-space-ptr* (que se direcciona al primer registro de la pantalla) para que se direccione al registro actual (que está después del último registro visualizado).

Un espacio de usuario siempre empieza en un límite de 16 bytes, de modo que el método descrito en este apartado asegura que **todos** los

registros están alineados. *ls-cust-rec-length* también se utiliza para encadenar los registros.

Cuando ejecute POINTA, observará las pantallas siguientes:

```
                Crear Área de Información del Cliente




            Crear área de información del cliente (Y/N)=> y <=









F3=Salir
```

Figura 101. Crear Pantalla del Área de Información del Cliente

Si se especifica `Y` para crear el espacio de usuario, el programa lee los registros del cliente del archivo y coloca la información en el espacio de usuario. Los registros se encadenan juntos.

Cuando se pulsa Intro en la pantalla anterior, aparece la pantalla de Información del Cliente:

```
                   Información del Cliente

   Núm      Nombre Cliente       Dirección Cliente
   Cliente
   00000001 Bakery Unlimited     30 Bake Way, North York
   00000002 Window World         150 Eglinton Ave E., North York, Ontario
   00000003 Jons Clothes         101 Park St, North Bay, Ontario, Canada
   00000004 Pizza World          254 Main Street, Toronto, Ontario         +
   00000005 Marv's Auto Body     9 George St, Peterborough, Ontario, Cana  +
   00000006 Jack's Snacks        23 North St, Timmins, Ontario, Canada
   00000007 Video World          14 Robson St, Vancouver, B.C, Canada
   00000008 Pat's Daycare        8 Kingston Rd, Pickering, Ontario, Canad  +
   00000009 Mary's Pies          3 Front St, Toronto, Ontario, Canada
   00000010 Carol's Fashions     19 Spark St, Ottawa, Ontario, Canada
   00000011 Grey Optical         5 Lundy's Lane, Niagara Falls, Ont. Cana  +
   00000012 Fred's Forage        33 Dufferin St, Toronto, Ontario, Canada  +
   00000013 Dave's Trucking      15 Water St, Guelph, Ontario, Canada
   00000014 Doug's Music         101 Queen St. Toronto, Ontario, Canada    +
   00000015 Anytime Copiers      300 Warden Ave, Scarborough, Ontario, Ca  +
   00000016 Rosa's Ribs          440 Avenue Rd, Toronto, Ontario, Canada

F3=Salir                                        F8=Av Pág
```

Figura 102. Pantalla del Área de Información del Cliente

Si hay más de 16 registros en el espacio de usuario (según la línea inicial en *ws-start-line*), el programa habilita la tecla F8=Av Pág, para permitir al usuario que avance las páginas de la lista. Cuando el usuario ha avanzado alguna página, se habilita la tecla F7=Re Pág para que el usuario pueda retroceder las páginas de la lista, tal como se muestra en la siguiente pantalla:

```
                    Información del Cliente

    Núm      Nombre Cliente       Dirección Cliente
    Cliente
    00000017 Picture It           33 Kingston Rd, Ajax, Ontario, Canada
    00000018 Paula's Flowers      144 Pape Ave, Toronto, Ontario, Canada
    00000019 Mom's Diapers        101 Ford St, Toronto, Ontario, Canada
    00000020 Chez Francois        1202 Rue Ste Anne, Montreal, PQ, Canada
    00000021 Vetements de Louise  892 Rue Sherbrooke, Montreal E, PQ, Cana   +
    00000022 Good Eats            355 Lake St, Port Hope, Ontario, Canada









F3=Salir                                F7=Re Pág
```

Figura 103. Pantalla de Información del Cliente (Segunda Pantalla)

Cuando el cliente sale de la pantalla anterior, se muestra la opción que permite suprimir el espacio de usuario, tal como se muestra en la pantalla siguiente:

```
                 Suprimir Área de Información del Cliente









            Surprimir área de información del cliente (Y/N)=> n <=









F3=Salir
```

Figura 104. Suprimir Pantalla de Información del Cliente

Proceso de una Lista Encadenada

Una aplicación típica para la utilización de ítems de datos puntero es el proceso de una lista encadenada (una serie de registros en la que cada uno direcciona al siguiente).

En este ejemplo, supongamos una lista encadenada compuesta de registros de salarios individuales. La Figura 105 muestra un modo de visualizar cómo se enlazan estos registros en el almacenamiento:

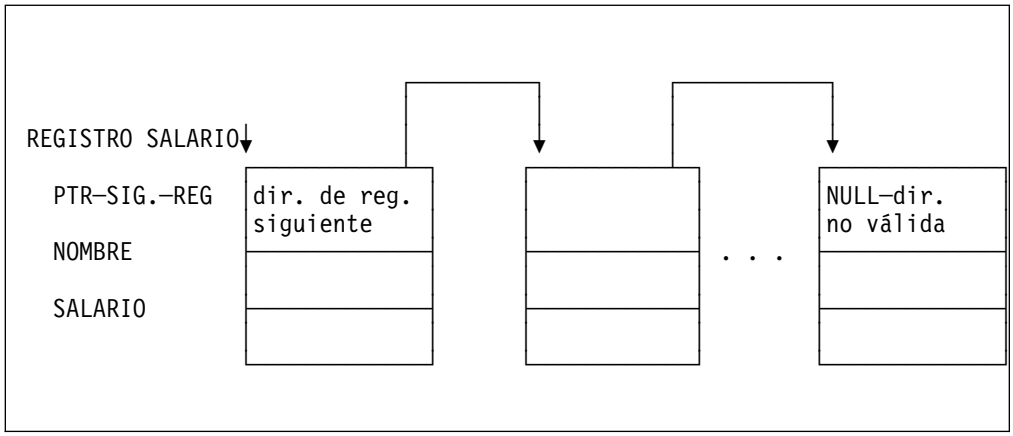


Figura 105. Representación de una Lista Encadenada que Acaba en NULL

El primer ítem de cada registro se direcciona al registro siguiente (excepto en el caso del último registro). El primer ítem del último registro contiene un valor nulo en lugar de una dirección, con el fin de indicar que se trata del último registro.

La lógica de nivel superior de una aplicación que procesa estos registros puede tener un aspecto parecido a éste:

```
OBTAIN ADDRESS OF FIRST RECORD IN CHAINED LIST FROM ROUTINE
CHECK FOR END OF THE CHAINED LIST
DO UNTIL END OF THE CHAINED LIST
   PROCESS RECORD
   GO ON TO THE NEXT RECORD
END
```

La Figura 106 en la página 317 contiene un diseño del programa de proceso, LISTS, utilizado en este ejemplo de proceso de una lista encadenada.

```
       IDENTIFICATION DIVISION.
       PROGRAM-ID. LISTS.
       ENVIRONMENT DIVISION.
       DATA DIVISION.
      ******
       WORKING-STORAGE SECTION.
       77  PTR-FIRST         POINTER  VALUE IS NULL.
       77  DEPT-TOTAL        PIC 9(4) VALUE IS 0.
      ******
       LINKAGE SECTION.
       01  SALARY-REC.
         02  PTR-NEXT-REC    POINTER.
         02  NAME            PIC X(20).
         02  DEPT            PIC 9(4).
         02  SALARY          PIC 9(6).
       01  DEPT-X            PIC 9(4).
      ******
       PROCEDURE DIVISION USING DEPT-X.
      ******
      * PARA TODOS EN EL DEPTO. RECIBIDOS COMO DEPT-X, REVISEN TODOS
      * LOS REGISTROS DE LA LISTA EN CADENA SEGÚN LA DIRECCIÓN DEL
      * PROGRAMA CHAIN-ANCH Y SUMEN LOS SALARIOS EN CADA REGISTRO.
      * EN CADA REGISTRO, PRT-NEXT-REC ES UN PUNTERO PARA EL
      * SIGUIENTE REGISTRO DE LA LISTA; EN EL ÚLTIMO REGISTRO,
      * PRT-NEXT-REC ES NULO.
      * VISUALIZAR EL TOTAL.
      ******
           CALL "CHAIN-ANCH" USING PTR-FIRST
           SET ADDRESS OF SALARY-REC TO PTR-FIRST
      ******
           PERFORM WITH TEST BEFORE UNTIL ADDRESS OF SALARY-REC = NULL
            IF DEPT = DEPT-X
              THEN ADD SALARY TO DEPT-TOTAL
              ELSE CONTINUE
            END-IF
            SET ADDRESS OF SALARY-REC TO PTR-NEXT-REC
           END-PERFORM
      ******
           DISPLAY DEPT-TOTAL
           GOBACK.
```

Figura 106. Programa para el Proceso de una Lista Encadenada

Paso de Direcciones entre Programas

Para obtener la dirección de la primera área de registro SALARY-REC, el programa LISTS llama al programa CHAIN-ANCH:

```
CALL "CHAIN-ANCH" USING PTR-FIRST
```

PTR-FIRST está definido en WORKING-STORAGE en el programa de llamada (LISTS) como un ítem de datos puntero:

```
WORKING-STORAGE SECTION.
77  PTR-FIRST         POINTER  VALUE IS NULL.
```

A la vuelta de la llamada a CHAIN-ANCH, PTR-FIRST contiene la dirección del primer registro en la lista encadenada.

PTR-FIRST se define inicialmente por tener un valor nulo como comprobación lógica. Si se produce un error con la llamada, y PTR-FIRST no llega a recibir el valor de la dirección del primer registro en la cadena, seguirá habiendo un valor nulo en PTR-FIRST y, en función de la lógica del programa, los registros no se procesarán.

NULL es una constante figurativa utilizada para asignar el valor de una dirección no válida a ítems puntero. Puede utilizarse en la cláusula VALUE IS NULL, en la instrucción SET y como operando en una condición de relación con un ítem de datos puntero.

La Sección de Enlace del programa de llamada contiene la descripción de los registros en la lista encadenada. Contiene asimismo la descripción del código de departamento que se pasa a través de la frase USING de la instrucción CALL.

```
LINKAGE SECTION.
01  SALARY-REC.
  02  PTR-NEXT-REC    POINTER.
  02  NAME            PIC X(20).
  02  DEPT            PIC 9(4).
  02  SALARY          PIC 9(6).
01  DEPT-X            PIC 9(4).
```

Para “basar” la descripción del registro SALARY-REC de la dirección contenida en PTR-FIRST, utilice la instrucción SET:

```
CALL "CHAIN-ANCH" USING PTR-FIRST
SET ADDRESS OF SALARY-REC TO PTR-FIRST
```

Comprobación de la Finalización de una Lista Encadenada

La lista encadenada de este ejemplo se configura de manera que el último registro contenga una dirección no válida. Para ello, el ítem de datos puntero del último registro tendría asignado el valor NULL.

Puede asignarse el valor NULL a un ítem de datos puntero de dos modos:

- Un ítem de datos puntero puede definirse con una cláusula VALUE IS NULL en su definición de datos.
- NULL puede ser el campo de envío en una instrucción SET.
- El valor inicial de un ítem de datos puntero con o sin una cláusula VALUE de NULL da igual a NULL.

En el caso de una lista encadenada en la que el puntero del último registro contenga un valor nulo, el código para comprobar la finalización de la lista sería:

```
IF PTR-NEXT-REC = NULL
⋮

(lógica de finalización de cadena)
```

Si no ha llegado al final de la lista, procese el registro y vaya al siguiente registro.

En el programa LISTS, esta prueba del final de la lista encadenada se realiza con una estructura “do while”:

```
PERFORM WITH TEST BEFORE UNTIL ADDRESS OF SALARY-REC = NULL
 IF DEPT = DEPT-X
   THEN ADD SALARY TO DEPT-TOTAL
   ELSE CONTINUE
 END-IF
 SET ADDRESS OF SALARY-REC TO PTR-NEXT-REC
END-PERFORM
```

Continuación del Proceso del Siguiente Registro

Para ir al siguiente registro, establezca la dirección de éste en la Sección de Enlace de manera que sea igual a la dirección del siguiente registro. Este acto se realiza a través del ítem de datos puntero enviado como primer campo en SALARY-REC:

```
SET ADDRESS OF SALARY-REC TO PTR-NEXT-REC
```

A continuación repita la rutina de proceso de registros, que procesará el siguiente registro en la lista encadenada.

Incremento de las Direcciones Recibidas de Otro Programa

Los datos desde un programa de llamada pueden contener información de cabecera de la cual decida prescindir (por ejemplo, en los datos recibidos desde una aplicación CICS que no se ha migrado al nivel de mandatos).

Dado que los ítems de datos puntero no son numéricos, el usuario no podrá efectuar aritmética directamente sobre ellos. No obstante, podrá utilizar el verbo SET para incrementar la dirección pasada para ignorar la información de cabecera.

Puede configurar la Sección de Enlace de la siguiente manera:

```
       LINKAGE SECTION.
       01  RECORD-A.
         02  HEADER          PIC X(16).
         02  REAL-SALARY-REC PIC X(30).
  ⋮
       01  SALARY-REC.
         02  PTR-NEXT-REC    POINTER.
         02  NAME            PIC X(20).
         02  DEPT            PIC 9(4).
         02  SALARY          PIC 9(6).
```

En la División de Procedimiento, sitúe la dirección de SALARY-REC en la dirección de REAL-SALARY-REC:

```
       SET ADDRESS OF SALARY-REC TO ADDRESS OF REAL-SALARY-REC
```

SALARY-REC se ha situado ahora en la dirección de RECORD-A + 16.

Áreas de Datos

Un área de datos es un objeto utilizado para comunicar datos tales como valores de variables entre programas dentro de un trabajo y entre trabajos. Un área de datos puede crearse y declararse en un programa antes de que se utilice en dicho programa o trabajo. Para más información acerca de la creación y declaración de un área de datos, consulte la publicación *CL Guía del Programador*.

Área de Datos Local

El área de datos local puede utilizarse para pasar cualquier información entre programas en un trabajo. Esta información puede consistir en datos en formato libre, como por ejemplo mensajes informales, o puede ser también un conjunto de campos totalmente estructurado o con formato.

El sistema crea automáticamente un área de datos local para cada trabajo. El área de datos local está definida fuera del programa COBOL como una área de 1024 bytes.

Cuando se somete un trabajo, el área de datos local del trabajo que se somete se copia al área de datos local del trabajo sometido. Si no hay ningún trabajo que se someta, el área de datos local se inicializa a espacios en blanco.

Un programa COBOL puede acceder al área de datos local para su trabajo con las instrucciones ACCEPT y DISPLAY, utilizando un nombre mnemotécnico asociado al nombre de función LOCAL-DATA.

Sólo hay un área de datos local asociada a cada trabajo. Incluso si se adquieren varias estaciones de trabajo mediante un único trabajo, sólo existe un área de datos local para dicho trabajo. *No* hay un área de datos local para cada estación de trabajo.

Área de Datos PIP (Parámetros de Inicialización de Programas)

El área de datos PIP es utilizada por un trabajo de prearranque. Generalmente, un trabajo de prearranque es un trabajo de un sistema remoto bajo ICF que el usuario inicia y mantiene listo para ejecutar hasta que lo llama.

Si utiliza un trabajo de prearranque, no tiene que esperar a que un programa que usted llama vaya a través del proceso de inicialización de un trabajo. La iniciación de trabajo se realiza antes de que un programa pueda iniciarse realmente. Dado que ya se ha realizado la iniciación del trabajo, un trabajo de prearranque permite que el programa se inicie más rápidamente una vez recibida la petición de arranque del programa.

Un programa COBOL puede acceder al área de datos PIP para su trabajo con la instrucción ACCEPT, utilizando un nombre mnemotécnico asociado con el nombre de función PIP-DATA.

El área de datos PIP es un ítem alfanumérico de 2000 bytes que contiene parámetros recibidos desde un programa de llamada. Proporciona los parámetros de inicialización que, en trabajos que no son de prearranque, se ofrece mediante los parámetros COBOL estándar.

Utilice una instrucción ACCEPT de Formato 5 para acceder al área de datos PIP, similar a la manera en que utiliza la instrucción ACCEPT de Formato 4 para leer desde el área de datos local. Observe que no puede actualizar el área de datos PIP utilizando COBOL. Consulte la publicación *COBOL/400 Reference* para obtener información de sintaxis detallada.

Para más información acerca de los trabajos de prearranque y del área de datos PIP, consulte la publicación *Guía para la Gestión de Trabajos* y la publicación *CL Guía del Programador*.

Consideraciones de Archivo

Puede pasar un nombre de archivo como parámetro en un programa COBOL, pero no podrá utilizar dicho archivo en el programa llamado. Si se define un archivo en un programa de llamada y un programa llamado, se trata como dos archivos distintos. El contenido del área de registro y el puntero del registro actual en cada programa son independientes, a menos que los archivos compartidos se especifiquen en mandatos CL. Consulte la publicación *Guía para la Gestión de Datos* para más información acerca de los archivos compartidos.

Las instrucciones siguientes afectan al estado de archivo de manera diferente:

- Una instrucción EXIT PROGRAM no cambia el estado de cualquier archivo en una unidad de ejecución.
- Una instrucción STOP RUN cierra todos los archivos de una unidad de ejecución.

Ampliación de IBM

- Una instrucción A GOBACK emitida desde un programa principal cierra todos los archivos de una unidad de ejecución. Una instrucción GOBACK emitida desde un subprograma no cambia el estado de ningún archivo en una unidad de ejecución.

Fin de Ampliación de IBM

- Una instrucción CANCEL no cambia el estado de ningún archivo en el programa que se cancela. Libera el almacenamiento que contiene información acerca del archivo. Si el programa tiene archivos que se abren cuando se procesa la instrucción CANCEL, esos archivos se cierran cuando se cancela el programa. El programa ya no puede seguir utilizando el archivo. Si el programa cancelado se llama de nuevo, el programa considera que el archivo está cerrado. Si el programa abre el archivo, se establece un nuevo enlace para dicho archivo. Esto puede hacer que se utilice el almacenamiento auxiliar del sistema.

Apéndice A. Características de segmentación

No se preocupe de la gestión del almacenamiento cuando utilice programas COBOL/400. Sin embargo, la segmentación de almacenamiento está disponible para ofrecer compatibilidad con otros sistemas.

La característica de segmentación proporciona optimización de almacenamiento de la División de Procedimientos controlada por el programador, permitiendo que esta división se subdivida tanto física como lógicamente.

Conceptos referentes a la segmentación

Aunque no es necesaria, la División de Procedimientos de un programa fuente suele grabarse como un grupo consecutivo de secciones, estando formada, cada una de ellas, por una serie de operaciones relacionadas que llevan a cabo una función determinada. Así pues, la totalidad de la División de Procedimientos se compone de un cierto número de subdivisiones lógicas. La segmentación permite que el programador divida físicamente la División de Procedimientos en segmentos, cada uno de los cuales tiene unos atributos físicos y lógicos específicos.

Cuando se utiliza la segmentación, la totalidad de la División de Procedimientos se divide en secciones. Cada sección debe clasificarse de acuerdo con sus atributos físicos y lógicos. La clasificación se especifica mediante los números de segmento. Todas las secciones que tengan el mismo número de segmento componen un segmento de programa.

Los números de segmento han de ser enteros del 0 al 99.

Segmentos de Programa

Hay tres tipos de segmentos de programa: permanente fijo, recubrible fijo e independiente.

Segmentos Fijos

Los segmentos permanentes fijos y los segmentos recubribles fijos componen la parte fija, la parte de la División de Procedimientos que se trata lógicamente como si estuviese siempre presente físicamente en almacenamiento principal. Los números de segmento de la parte fija han de ser enteros del 0 al 49.

Un segmento permanente fijo queda disponible siempre en su último estado de utilización.

Un segmento recubrible fijo está siempre lógicamente en el almacenamiento principal durante el proceso del programa; por lo tanto, siempre se encuentra disponible en su último estado de utilización. Cualquier recubrimiento de uno de esos segmentos es transparente para el usuario. De este modo, un segmento fijo recubrible es idéntico lógicamente a un segmento permanente fijo.

Segmentos Independientes

Obviamente, un segmento independiente puede recubrir y ser recubierto por otros segmentos durante la ejecución de un programa.

Un segmento independiente queda disponible en su estado inicial la primera vez que se le transfiere control (explícita o implícitamente) durante la ejecución de un programa.

Un segmento independiente queda disponible en su estado inicial durante posteriores transferencias de control cuando:

- La transferencia es el resultado de una transferencia implícita de control entre instrucciones consecutivas que se encuentren en segmentos distintos (es decir, cuando el control llega al segmento independiente desde el segmento físicamente anterior).
- La transferencia es el resultado de una transferencia implícita desde una instrucción SORT o MERGE en un segmento a un procedimiento de entrada SORT o un procedimiento de salida SORT/MERGE dentro de un segmento independiente.
- Una transferencia explícita de control se lleva cabo desde una sección con un número de segmento distinto (como sucede, por ejemplo, durante la transferencia de control en una instrucción PERFORM n TIMES).

Un segmento independiente queda disponible en su último estado de utilización durante posteriores transferencias de control, cuando:

- Exceptuando los dos tipos anteriores de transferencias implícitas, tiene lugar una transferencia implícita desde una sección con una prioridad diferente (como, por ejemplo, cuando se devuelve control al segmento independiente desde un procedimiento Declarativo).
- Se obtiene como resultado una transferencia explícita desde una instrucción EXIT PROGRAM o GOBACK.

Los segmentos independientes han de tener asignados números de segmento del 50 al 99.

Lógica de segmentación

En un programa segmentado, las secciones se clasifican por un sistema de números de segmento de acuerdo con los criterios siguientes:

- *Frecuencia de Referencia*–Las secciones más utilizadas, o aquéllas que han de estar disponibles en todo momento como referencia, deben estar normalmente dentro de segmentos fijos permanentes. Las secciones que se utilizan con menor frecuencia pueden estar en segmentos fijos recubribles o en segmentos independientes, según la lógica del programa.
- *Frecuencia de Utilización*–Cuanto mayor sea la frecuencia con la que se utilice una sección, menor será su número de segmento; cuanto menor sea la frecuencia con la que se la referencie, mayor será su número de segmento.
- *Relaciones Lógicas*–A las secciones que se comunican más frecuentemente entre ellas habría que darles números de segmento idénticos.

Control de Segmentación

Excepto para transferencias específicas de control, la secuencia lógica y la secuencia física de instrucciones de programa son iguales. El compilador inserta cualquier instrucción necesaria para inicializar un segmento. No es necesario transferir control al principio de un segmento, o al principio de una sección dentro de un segmento. En su lugar, el control puede transferirse a cualquier párrafo en la División de Procedimientos.

Consideraciones sobre el Programa Fuente COBOL

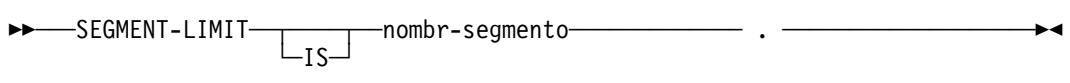
Los elementos siguientes de un programa fuente COBOL implantan la característica de segmentación:

- La cláusula SEGMENT-LIMIT en el párrafo OBJECT-COMPUTER de la División de Características de Entorno. Esta cláusula sirve para controlar la especificación de segmentos permanentes fijos y recubribles fijos.
- Los números de segmentos de la División de Procedimientos que agrupan secciones en segmentos. El esquema de numeración de segmentos permite además la especificación de segmentos independientes, segmentos permanentes fijos, y (junto con la cláusula SEGMENT-LIMIT) de segmentos recubribles fijos.

Segmentación–División del Entorno

En el párrafo OBJECT-COMPUTER, la cláusula SEGMENT-LIMIT permite que el usuario vuelva a clasificar los segmentos permanentes fijos mientras se retienen las propiedades de los segmentos de la parte fija para los segmentos vueltos a clasificar.

Formato



La cláusula SEGMENT-LIMIT permite que el programador especifique ciertos segmentos permanentes como susceptibles de ser recubiertos por segmentos independientes, sin perder las propiedades lógicas de los segmentos de la parte fija.

El `número-segmento` debe ser un entero con un valor dentro del rango de 1 a 49.

Cuando se especifica la cláusula SEGMENT-LIMIT:

- Los segmentos permanentes fijos son los que poseen números de segmento del 0 hasta el número de segmento especificado pero sin incluirlo.
- Los segmentos recubribles fijos son los que poseen números de segmento desde el número de segmento especificado hasta el 49.

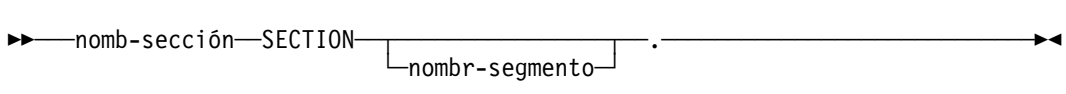
Por ejemplo, si se especifica SEGMENT-LIMIT IS 25, las secciones con números de segmento del 0 al 24 son segmentos permanentes fijos, y las secciones con números de segmento del 25 al 49 son segmentos recubribles fijos.

Cuando se omite la cláusula SEGMENT-LIMIT, todas las secciones con números de segmento del 0 al 49 son segmentos permanentes fijos.

Segmentación–División de Procedimientos

En la División de Procedimientos de un programa segmentado, la clasificación de secciones se especifica mediante los números de segmento en los encabezamientos de sección. El número de segmento ha de ser un entero del 0 al 99.

Formato



Todas las secciones con el mismo número de segmento componen un segmento de programa. No es necesario que dichas secciones sean contiguas en el programa fuente.

Los segmentos con números de segmento del 0 al 49 están en la parte fija del programa. Sólo es posible asignar estos números a las secciones declarativas. Los segmentos con números de segmento del 50 al 99 son segmentos independientes. Si se omite el número de segmento del encabezamiento de la sección, se asume que el número de segmento es 0.

Segmentación–Consideraciones Especiales

Cuando se utiliza la segmentación, existen una serie de restricciones en las instrucciones ALTER, PERFORM, SORT y MERGE. También hay consideraciones especiales para los programas de llamada y programas llamados.

Instrucción ALTER

No debe hacerse referencia a una instrucción GO TO en un segmento independiente por medio de una instrucción ALTER en un segmento distinto. Todas las otras utilizaciones de la instrucción ALTER son válidas y se realizan, incluso si la instrucción GO TO a la que se haga referencia está en un segmento recubrible fijo.

Instrucción PERFORM

Una instrucción PERFORM en la parte fija puede tener en su rango, además de cualquier procedimiento Declarativo, cuya ejecución tenga lugar dentro de ese rango, sólo uno de los elementos siguientes:

- Secciones y/o párrafos en la parte fija
- Secciones y/o párrafos dentro de un único segmento independiente.

Una instrucción PERFORM en un segmento independiente puede tener dentro de su rango, además de los procedimientos Declarativos, cuyo proceso tenga lugar dentro de ese rango, sólo uno de los elementos siguientes:

- Secciones y/o párrafos en la parte fija
- Secciones y/o párrafos contenidos totalmente en el mismo segmento independiente que la instrucción PERFORM.

Instrucciones SORT y MERGE

Si aparece una instrucción SORT o MERGE en la parte fija, cualquier procedimiento de entrada SORT o procedimiento de salida SORT/MERGE debe aparecer completamente en una de las siguientes ubicaciones:

- La parte fija
- Un único segmento independiente.

Si aparece una instrucción SORT o MERGE dentro de un segmento independiente, cualquier procedimiento de entrada SORT o procedimiento de salida SORT/MERGE debe aparecer por completo dentro de una de las siguientes ubicaciones:

- La parte fija
- El mismo segmento independiente que la instrucción SORT o MERGE.

Programas Llamados y Programas de Llamada

La instrucción CALL puede aparecer en cualquier parte dentro de un programa segmentado. Cuando una instrucción CALL aparece dentro de un segmento independiente, dicho segmento se encuentra en su último estado de utilización cuando se devuelve el control al programa de llamada.

Apéndice B. Características de Depuración

Las características de depuración especifican las condiciones bajo las que los procedimientos se supervisarán durante el tiempo de ejecución del programa.

Se proporcionan las instrucciones de depuración del lenguaje fuente COBOL. El usuario debe decidir qué supervisar, así como la información que necesita recuperar a propósito de la depuración. Las características de depuración COBOL permiten simplemente acceder a la información pertinente.

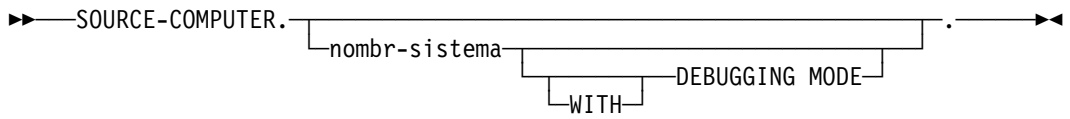
Depuración del Lenguaje Fuente COBOL

Los elementos del lenguaje COBOL que implantan las cracterísticas de depuración son un conmutador en tiempo de compilación (WITH DEBUGGING MODE), un conmutador en tiempo de ejecución, una sentencia declarativa USE FOR DEBUGGING, el registro especial DEBUG-ITEM y las líneas de depuración que pueden escribirse en las Divisiones de Características de Entorno, Datos y Procedimientos.

Conmutador de Tiempo de Compilación

En el párrafo SOURCE-COMPUTER de la Sección de Configuración, la cláusula WITH DEBUGGING MODE actúa como un conmutador en tiempo de compilación.

Formato



La cláusula WITH DEBUGGING MODE ejerce como conmutador en tiempo de compilación para las instrucciones de depuración escritas en el programa fuente.

Cuando se especifica WITH DEBUGGING MODE, todas las secciones y líneas de depuración se compilan tal como se especifica en este apéndice. Cuando se omite WITH DEBUGGING MODE, todas las secciones y líneas de depuración se tratan como documentación.

Conmutador en Tiempo de Ejecución

El conmutador en tiempo de ejecución activa dinámicamente el código de depuración que se genera cuando se especifica WITH DEBUGGING MODE.

Existen dos mandatos para controlar el conmutador en tiempo de ejecución. Para activar el conmutador en tiempo de ejecución, introduzca el mandato:

```
STRCBLDBG
```

y pulse F4.

Se visualiza la siguiente pantalla:

```
                        Iniciar Depuración COBOL (STRCBLDBG)

Escriba la elección, pulse Intro.

Programa . . . . . .                  Nombre
  Biblioteca.. . . . . .     *LIBL      Nombre, *LIBL, *CURLIB

                                                                    Final
F3=Salir  F4=Solicitud  F5=Renovar  F12=Cancelar  F13=Cómo util. esta pantalla
F24=Más teclas
```

El siguiente diagrama muestra la sintaxis del mandato STRCBLDBG:

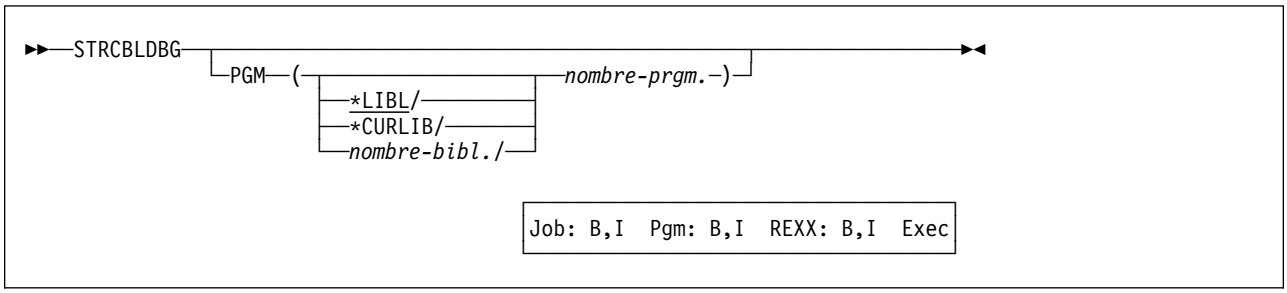


Figura 107. Sintaxis del mandato STRCBLDBG

Se permite utilizar este mandato tanto en procesos interactivos y por lotes como en programas CL.

Interfaz de Programación de Uso General

Se puede utilizar este mandato en QCMDEXC.

Fin de Interfaz de Programación de Uso General

Para desactivar el conmutador en tiempo de ejecución, introduzca el mandato:

```
ENDCBLDBG
```

y pulse F4.

Se visualiza el siguiente mandato:

```
                          Final Depuración COBOL (ENDCBLDBG)

Escriba la elección, pulse Intro.

Programa . . . . . .                    Nombre
  Biblioteca.. . . . . .     *LIBL       Nombre, *LIBL, *CURLIB









                                                                      Final
F3=Salir  F4=Solicitud  F5=Renovar  F12=Cancelar  F13=Cómo util. esta pantalla
F24=Más teclas
```

EL siguiente diagrama muestra la sintaxis del mandato ENDCBLDBG:

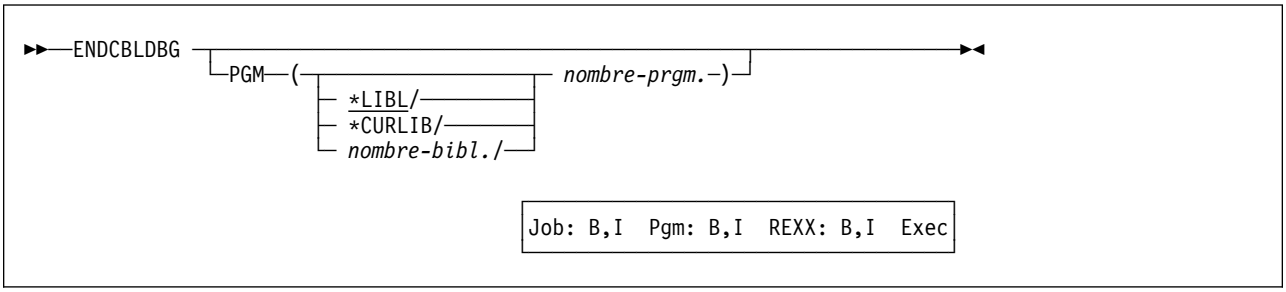


Figura 108. Sintaxis del mandato ENDCBLDBG

Se permite utilizar este mandato tanto en procesos interactivos y por lotes como en programas CL.

Interfaz de Programación de Uso General

Se puede utilizar este mandato en QCMDEXC.

Fin de Interfaz de Programación de Uso General

El valor por omisión para el conmutador en tiempo de ejecución se desactiva.

Cuando se especifica la modalidad de depuración mediante el conmutador en tiempo de ejecución, se activan todas las secciones y líneas de depuración (**D** en la columna 7) compiladas en el programa.

Es necesario introducir el mandato STRCBLDBG para cada programa COBOL (programa principal o programa llamado) que se deba depurar en la próxima unidad de ejecución COBOL. Al final de la unidad de ejecución, todos los conmutadores en tiempo de ejecución que estén activos, es desactivado. Si ha de desactivar un conmutador antes de arrancar una unidad de ejecución COBOL, utilice el mandato ENDCBLDBG. Los conmutadores en tiempo de ejecución pueden estar activos a la vez hasta para 15 programas.

Cuando se emiten los mandatos STRCBLDBG o ENDCBLDBG en un programa CL, las expresiones de concatenación pueden utilizarse para todos los valores de parámetros. Consulte la publicación *CL Guía del Programador* para obtener más información acerca de las expresiones de concatenación.

Cuando se suprime la modalidad de depuración, mediante el conmutador en tiempo de ejecución, se inhabilita cualquier procedimiento Declarativo USE FOR DEBUGGING. Todas las líneas de depuración (**D** en la columna 7) permanecen en vigor.

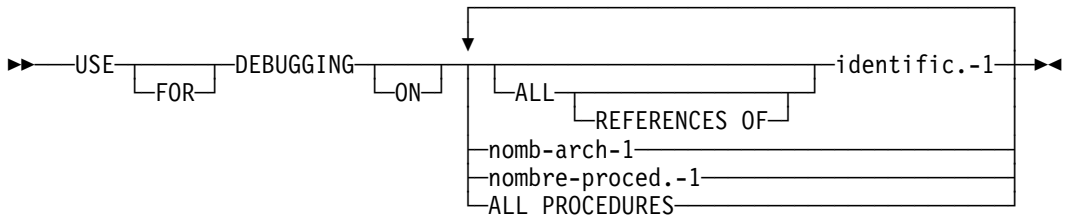
No es necesaria la recompilación del programa fuente para activar o desactivar el conmutador en tiempo de ejecución.

Cuando no se especifica WITH DEBUGGING MODE en el párrafo SOURCE-COMPUTER, el conmutador en tiempo de ejecución no tiene ningún efecto en la ejecución del programa.

Sentencia Declarativa USE FOR DEBUGGING

La sentencia USE FOR DEBUGGING en la División de Procedimientos identifica los ítems del programa fuente que se visualizan mediante el procedimiento declarativo de depuración asociado.

Formato



El `Identificador-1` no puede modificarse para su consulta.

Cuando se especifican, todas las secciones de depuración deben escribirse inmediatamente después del encabezamiento DECLARATIVES. Excepto para la sentencia USE FOR DEBUGGING, no debe haber ninguna referencia dentro del procedimiento de depuración a ningún procedimiento que no sea declarativo.

Observe que la declarativa USE FOR DEBUGGING provoca que se ignoren todas las instrucciones subsiguientes hasta llegar a una instrucción USE AFTER EXCEPTION/ERROR válida o a un delimitador END DECLARATIVES. Es posible que programas enteros se omitan por esta razón.

La aparición de una instrucción en una sección de depuración no provoca la ejecución automática de dicha sección.

Una sección de depuración para un operando específico se procesa una sola vez como resultado de la ejecución de una única instrucción, sin importar cuántas veces se especifica el operando en la instrucción. Una excepción a esta regla es que cada especificación de un identificador subindexado o indexado en la que los subíndices o índices sean distintos, provoca la invocación de la Declarativa de depuración. Para una instrucción PERFORM que cause la ejecución repetida de un procedimiento, cualquier nombre del procedimiento asociado a una sección Declarativa de depuración, se ejecuta una vez por cada ejecución del procedimiento.

A efectos de depuración, cada aparición por separado de un verbo imperativo dentro de una instrucción imperativa comienza una instrucción distinta.

Las instrucciones que aparecen fuera de las secciones de depuración no deben hacer referencia a los nombres de procedimientos definidos dentro de las secciones de depuración.

Excepto para la propia sentencia USE FOR DEBUGGING, las instrucciones dentro de una sección Declarativa de depuración sólo puede hacer referencia a nombres de procedimientos definidos en un procedimiento USE distinto mediante la instrucción PERFORM. Los nombres de Procedimientos dentro de las secciones Declarativas de depuración no deben aparecer en las sentencias USE FOR DEBUGGING.

En la Tabla 7 se definen los puntos en los que se ejecutan los procedimientos USE FOR DEBUGGING durante el tiempo de ejecución. El identificador-n, el nombre de archivo-n y el nombre de procedimiento-n hacen referencia a la primera y a todas las especificaciones posteriores de ese tipo de operando en una sentencia USE FOR DEBUGGING. Cualquier identificador, nombre de archivo, o nombre de procedimiento en particular puede aparecer en una sola sentencia USE FOR DEBUGGING, y sólo una vez en esa sentencia.

Un identificador en una sentencia USE FOR DEBUGGING:

- Debe especificarse sin la subindexación o la indexación necesaria normalmente si contiene una cláusula OCCURS o está subordinada a una entrada que contenga una cláusula OCCURS. (Una instrucción SEARCH o SEARCH ALL que haga referencia a uno de esos identificadores, no invoca a los procedimientos USE FOR DEBUGGING.)
- No debe ser un registro especial.

Cuando se especifica ALL PROCEDURES en una sentencia USE FOR DEBUGGING, no deben especificarse el nombre de procedimiento-1, el nombre de procedimiento-2, el nombre de procedimiento-3, etc., en ninguna sentencia USE FOR DEBUGGING. La frase ALL PROCEDURES sólo puede especificarse una vez en un programa.

Cuando se utiliza una operando USE FOR DEBUGGING como calificador, dicha referencia en el programa no activa los procedimientos de depuración.

Las referencias al registro especial DEBUG-ITEM sólo se pueden realizar desde dentro de un procedimiento Declarativo de depuración.

Tabla 7. Ejecución de las Declarativas de Depuración

Operando USE FOR DEBUGGING	**Los procedimientos USE FOR DEBUGGING se ejecutan después de lo siguiente:**
identificador-n	Antes de REWRITE/WRITE identificador-n y después del movimiento de la frase FROM, si es aplicable. Después de cada inicialización, modificación, o evaluación del identificador-n en PERFORM/VARYING/AFTER/UNTIL identificador-n. Después de cualquier otra instrucción COBOL que haga referencia explícitamente al identificador-n y pudiera cambiar su contenido. (Véase nota adjunta).
ALL REFERENCES OF identificador-n	Antes de GO TO DEPENDING ON identificador-n, se transfiere el control, y antes de que se ejecute ninguna sección de depuración asociada para el nombre de procedimiento. Antes REWRITE/WRITE identificador-n y del movimiento de la frase FROM, si es aplicable. Después de cada inicialización, modificación o evaluación del identificador-n en PERFORM/VARYING/AFTER/UNTIL identificador-n. Después de cualquier otra instrucción COBOL que se refiera explícitamente al identificador-n. (Véase nota adjunta).
nombre-archivo-n	Después de CLOSE/DELETE/OPEN/START nombre-archivo-n Después de READ nombre-archivo-n en el que no se ejecutó AT END/INVALID KEY.
nombre-procedimiento-n	Antes de cada ejecución del procedimiento nombrado. Después de la ejecución de la instrucción ALTER que haga referencia al procedimiento nombrado.
ALL PROCEDURES	Antes de cada ejecución de todos los procedimientos de no depuración. Después de la ejecución de toda instrucción ALTER (excepto las instrucciones ALTER en los procedimientos Declarativos).

Nota: Los operandos sobre los que actúe pero no se nombren explícitamente en las instrucciones tales como ADD, MOVE, o SUBTRACT CORRESPONDING nunca causan la activación de un procedimiento USE FOR DEBUGGING cuando se ejecutan dichas instrucciones. Si se especifica el identificador-n en una frase que no se procesa, no se ejecuta la sección de depuración asociada.

Registro Especial DEBUG-ITEM

El registro especial DEBUG-ITEM proporciona información para un procedimiento Declarativo de depuración. DEBUG-ITEM tiene la descripción implícita siguiente:

```
01 DEBUG-ITEM.
   02 DEBUG-LINE      PICTURE IS X(6).
   02 FILLER          PICTURE IS X VALUE SPACE.
   02 DEBUG-NAME      PICTURE IS X(30).
   02 FILLER          PICTURE IS X VALUE SPACE.
   02 DEBUG-SUB-1     PICTURE IS S9999 SIGN IS
                      LEADING SEPARATE CHARACTER.
   02 FILLER          PICTURE IS X VALUE SPACE.
   02 DEBUG-SUB-2     PICTURE IS S9999 SIGN IS
                      LEADING SEPARATE CHARACTER.
   02 FILLER          PICTURE IS X VALUE SPACE.
   02 DEBUG-SUB-3     PICTURE IS S9999 SIGN IS
                      LEADING SEPARATE CHARACTER.
   02 FILLER          PICTURE IS X VALUE SPACE.
   02 DEBUG-CONTENTS  PICTURE IS X(n).
```

El registro especial DEBUG-ITEM proporciona información acerca de las condiciones que provocan la ejecución de una sección de depuración.

Antes de que se procese cada sección de depuración, DEBUG-ITEM se llena de espacios. El contenido de los subcampos DEBUG-ITEM se actualiza según las reglas que rige la instrucción MOVE, con una excepción: DEBUG-CONTENTS se actualiza como si el movimiento fuera un movimiento alfanumérico a alfanumérico sin conversión de datos desde una forma de representación interna a otra. Después de la actualización, cada campo contiene:

- DEBUG-LINE: El número de instrucción generado por el compilador, ajustado a la derecha y rellenado con ceros por la izquierda. Por ejemplo, 000112.
- DEBUG-NAME: Los primeros 30 caracteres del nombre causante de la ejecución de la sección de depuración. Todos los calificadores se separan mediante la palabra OF (no se introducen subíndices o índices en DEBUG-NAME).
- DEBUG-SUB-1, DEBUG-SUB-2, DEBUG-SUB-3: Si DEBUG-NAME está subindexado o indexado, el número de aparición de cada nivel se introduce en el DEBUG-SUB-n correspondiente. Si el artículo no está subindexado o indexado, estos campos permanecen con espacios en blanco.
- DEBUG-CONTENTS: Los datos se mueven a DEBUG-CONTENTS tal como se muestra en la Tabla 8. DEBUG-CONTENTS tiene el mismo tamaño que el identificador más grande del programa.

Tabla 8. Contenido del Subcampo DEBUG-ITEM

Ítem que causa la ejecución de la sección de depuración	**DEBUG-LINE Contiene el número de la instrucción COBOL que hace referencia a**	**Contenido de DEBUG-NAME**	**Contenido de DEBUG-CONTENTS**
identificador-n	identificador-n	identificador-n	Contenido del identificador-n cuando el control se transfiere a la sección de depuración.
nombre-archivo-n	nombre-archivo-n	nombre-archivo-n	Para READ: contenido del registro recuperado. Otras referencias: espacios.
nombre-procedimiento-n referencia ALTER	instrucción ALTER	nombre-procedimiento-n	nombre-procedimiento-n en la frase TO PROCEED TO
GO TO nombre-procedimiento-n	instrucción GO TO	nombre-procedimiento-n	
nombre-procedimiento-n en SORT/MERGE INPUT/OUTPUT PROCEDURE	instrucción SORT/MERGE	nombre-procedimiento-n	“SORT INPUT” “SORT OUTPUT” “MERGE OUTPUT” según sea aplicable
Transferencia de control de la instrucción PERFORM	Esta instrucción PERFORM	nombre-procedimiento-n	“PERFORM LOOP”
nombre-procedimiento en un procedimiento USE	Instrucción causante de la ejecución del procedimiento USE	nombre-procedimiento-n	“USE PROCEDURE”
Transferencia implícita desde el procedimiento secuencial anterior	Instrucción anterior procesada en un procedimiento secuencial previo (véase nota adjunta)	nombre-procedimiento-n	“FALL THROUGH”
Primera entrada en el primer procedimiento no declarativo	Número de línea de la primera instrucción del procedimiento	Primer nombre de procedimiento no declarativo	“START PROGRAM”

Nota: Si este párrafo está precedido de un encabezamiento de sección y el control se pasa al encabezamiento de sección, el numero de instrucción hace referencia al encabezamiento de sección.

Líneas de depuración

Las líneas de depuración pueden ayudar a determinar la causa de un error. Una línea de depuración es cualquier línea en un programa fuente con una **D** codificada en la columna 7 (el área de continuación). Si una línea de depuración sólo contiene espacios en blanco en el Área A y B, se considera una línea en blanco.

Cada línea de depuración debe escribirse de tal forma que el resultado sea un programa sintácticamente correcto, ya sea mediante la compilación de las líneas de depuración o mediante una corrección sintáctica, aunque se traten como documentación.

Se permiten líneas de depuración sucesivas. Las líneas de depuración pueden continuarse. Sin embargo, cada línea de continuación debe contener una **D** en la columna 7, y la serie de caracteres no debe partirse en dos líneas.

Las líneas de depuración sólo pueden especificarse después del párrafo OBJECT-COMPUTER.

Cuando se especifica la cláusula WITH DEBUGGING MODE en el párrafo SOURCE-COMPUTER, se compilan todas las líneas de depuración como parte del programa objeto.

Cuando se omite la cláusula WITH DEBUGGING MODE, se comprueba la sintaxis de las líneas de depuración, pero se tratan como documentación.

Apéndice C. Nivel de Soporte del Lenguaje

Estándar ANSI X3.23-1985 COBOL

El estándar ANSI X3.23-1985 COBOL está compuesto por once módulos de proceso funcional, siete de los cuales son imprescindibles y cuatro opcionales.

Los siete módulos necesarios son: Núcleo, E-S Secuencial, E-S Relativa, E-S Indexada, Comunicación Entre Programas, Clasificación Fusión y Manipulación de Texto Fuente. Los cuatro módulos opcionales son: Transcriptor de Informes, Comunicaciones, Depuración y Segmentación.

Los elementos del lenguaje dentro de los módulos pueden clasificarse como elementos del nivel 1 y elementos de nivel 2. Los elementos de nueve de los módulos se dividen en elementos del nivel 1 y elementos del nivel 2. Dos de los módulos (SORT-MERGE y REPORT WRITER) sólo contienen elementos del nivel 1. Por ejemplo, los elementos del nivel 1 del Núcleo realizan operaciones internas básicas. Los elementos del nivel 2 del Núcleo proporcionan elementos para procesos internos más amplios y sofisticados.

Los tres subconjuntos del COBOL Estándar son el subconjunto superior, el subconjunto intermedio y el subconjunto inferior. Cada subconjunto está compuesto por un nivel de los siete módulos necesarios: Núcleo, E-S Secuencial, E-S Relativa, E-S Indexada, Comunicación Entre Programas, Clasificación Fusión, y Manipulación de Texto Fuente. Los cuatro módulos opcionales (Transcriptor de Informes, Comunicaciones, Depuración y Segmentación) no son necesarios en los tres subconjuntos del COBOL Estándar.

El subconjunto superior está compuesto por todos los elementos del lenguaje del nivel superior de los módulos necesarios. Es decir:

- Los elementos del nivel 2 del Núcleo, E-S Secuencial, E-S Relativa, E-S Secuencial, Comunicación Entre Programas, y Manipulación del Texto Fuente
- Los elementos del nivel 1 de la Clasificación-Fusión.

El subconjunto intermedio está compuesto por todos los elementos de lenguaje del nivel 1 de todos los módulos necesarios. Es decir:

- Los elementos del nivel 1 del Núcleo, E-S Secuencial, E-S Relativa, E-S Indexada, Comunicación entre Programas, y Manipulación del Texto Fuente.

El subconjunto inferior se compone de todos los elementos del lenguaje de nivel 1 de los módulos del Núcleo, E-S Secuencial, y Comunicación Entre Programas.

Los cuatro módulos opcionales no son parte integral de ninguno de los subconjuntos. Sin embargo, ninguna, todas o cualquier combinación de los módulos opcionales puede asociarse con cualquier subconjunto.

Nivel de Soporte del Lenguaje COBOL/400

El compilador COBOL/400 proporciona soporte para:

- El nivel 1 de los módulos del Núcleo, E-S Secuencial, E-S Relativa, E-S Indexada, Comunicación Entre Programas, Clasificación Fusión y Manipulación de Texto Fuente
- El nivel 2 de los módulos de Depuración y Segmentación.

El compilador COBOL/400 no proporciona soporte para los módulos del Transcriptor de Informes y Comunicaciones de ANSI X3.23-1985 COBOL.

El nivel de soporte proporcionado por el compilador COBOL/400 se representa en la tabla inferior. La siguiente tabla:

- Muestra el nivel de soporte del compilador COBOL/400 para cada módulo de proceso funcional del Estándar ANSI X3.23-1985 COBOL
- Describe cada módulo.

Seguidamente se da una explicación de la notación utilizada en la tabla:

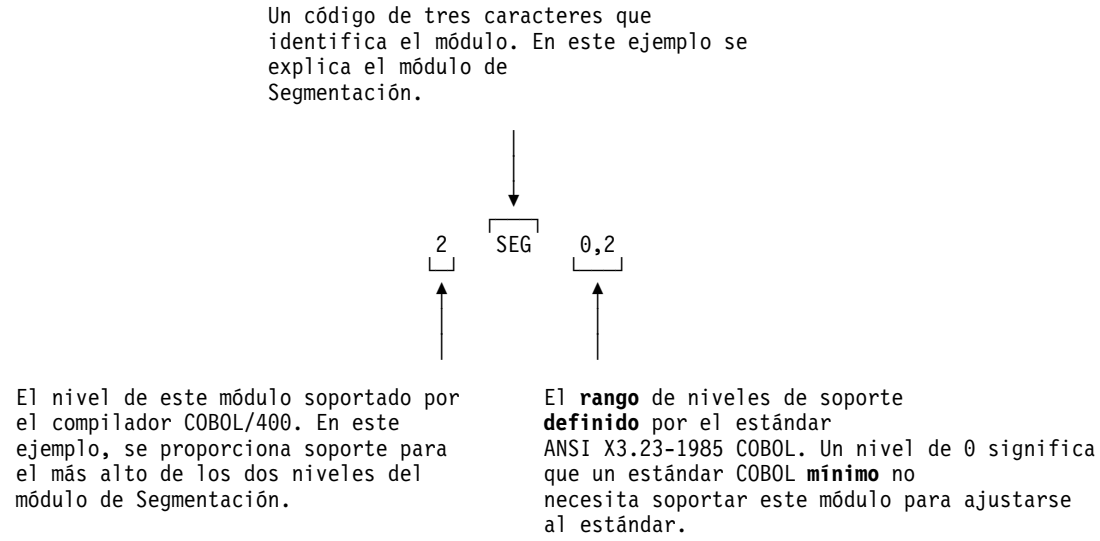


Tabla 9 (Página 1 de 2). Nivel de Soporte del Compilador COBOL/400

COBOL/400 Nivel de Lenguaje Soportado	**Descripción del Módulo**
Núcleo 1 NUC 1,2	Contiene los elementos del lenguaje necesarios para procesar los datos internamente dentro de las cuatro divisiones básicas de un programa y la posibilidad para la definición y acceso de tablas.
E-S Secuencial 1 SEQ 1,2	Proporciona acceso a los registros de los archivos mediante la secuencia establecida en que se grabaron los registros en el archivo.
E-S Relativa 1 REL 0,2	Proporciona acceso a registros de manera secuencial o al azar. Cada registro se identifica únicamente mediante un entero que representa la posición lógica del registro dentro del archivo.

Tabla 9 (Página 2 de 2). Nivel de Soporte del Compilador COBOL/400	
COBOL/400 Nivel de Lenguaje Soportado	**Descripción del Módulo**
E-S Indexada 1 INX 0,2	Proporciona acceso a registros de manera secuencial o al azar. Cada registro dentro de un archivo indexado se identifica mediante una clave de registro.
Comunicación Entre programas 1 IPC 1,2	Permite que un programa COBOL comunique con otros programas mediante transferencias de control y acceso a elementos de datos comunes.
Clasificación-Fusión 1 SRT 0,1	Ordena uno o más archivos de registros, o combina dos o más archivos ordenados de forma idéntica de acuerdo con las claves especificadas por el usuario.
Manipulación del Texto Fuente 1 STM 0,2	Permite insertar texto COBOL predefinido dentro de un programa en tiempo de ejecución.
Transcriptor de informes 0 RPW 0,1	Proporciona una producción semiautomática de informes impresos.
Comunicaciones 0 COM 0,2	Permite acceder, procesar, y crear mensajes o partes de mensajes; también permite la comunicación entre un Sistema de Control de Mensajes con dispositivos de comunicaciones locales y remotos.
Depuración 2 DEB 0,2	Permite especificar instrucciones y procedimientos para la depuración.
Segmentación 2 SEG 0,2	Proporciona el solapamiento en tiempo de objeto de secciones de la División de Procedimientos.

Soporte SAA de la Interfaz de Programación Común (CPI)

El archivo fuente QILBINC en las bibliotecas de productos QLBL y QLBLP contiene los miembros que mantienen las especificaciones para múltiples Interfaces de Programación Común SAA. Estas especificaciones describen las interfaces de parámetros. Este archivo es propiedad de IBM y no debe sufrir ninguna modificación.

Si desea personalizar cualquier especificación, debe copiar cualquier miembro que desee modificar en un archivo fuente de una de las bibliotecas. Puede utilizar el mandato Copiar Archivo (CPYF) para efectuar esta tarea. Para obtener más información acerca del mandato CPYF, consulte la publicación *CL Reference*.

Si copia estas especificaciones en la biblioteca, debe renovar las copias cuando cuando se instale una release nueva del producto, o cuando se efectúe cualquier modificación utilizando un Arreglo Temporal de Programa (PTF). IBM proporciona el mantenimiento para estas especificaciones sólo en las bibliotecas en las que esté distribuido.

Apéndice D. Mensajes COBOL/400, Señalizador FIPS y Señalización SAA

Mensajes COBOL/400

Este apéndice proporciona una descripción general de los mensajes que IBM suministra con el programa bajo licencia COBOL/400.

Mensajes Interactivos

En un entorno interactivo, los mensajes se visualizan en la pantalla de la estación de trabajo. Pueden aparecer en la pantalla actual como resultado de la ejecución del programa o como respuesta a las entradas introducidas desde la línea de mandatos, menús, pantalla de entrada de mandatos o Herramientas para el Desarrollo de Aplicaciones (Appl Dev Tools). Los mensajes también pueden aparecer si se solicitan mediante un mandato de visualización o de una opción del menú.

Los mensajes para el programa bajo licencia COBOL/400 empiezan con un prefijo LSC, LBE, o LBL.

El comprobador de sintaxis COBOL/400 emite los mensajes LSC cuando se utiliza el Programa de Utilidad de Entrada Fuente (SEU) para introducir el fuente COBOL/400. Por ejemplo, se visualiza la pantalla siguiente después de introducir incorrectamente el nombre del programa en el párrafo PROGRAM-ID.

```
 Columna . . . :    1  71           Editar                        XMPLIB/QLBLSRC
 SEU==> _____________________________________________________________     TESTPR
 FMT CB ......-A+++B++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
        *************** Inicio de datos ************************************
0000.10         IDENTIFICATION DIVISION.
0000.20         PROGRAM-ID. #TESTPR.
0000.70         ENVIRONMENT DIVISION.
0000.90         SOURCE-COMPUTER. IBM-AS400.
        ***************** Fin de datos ***************************************









 F3=Salir  F4=Solicitar F5=Renovar  F9=Recuperar  F10=Cursor
 F16=Repetir búsqueda  F17=Repetir cambio         F24=Más teclas
# no está en el juego de caracteres COBOL. Línea rechazada
```

Figura 109. Ejemplo de un Mensaje del Comprobador de Sintaxis COBOL/400.

Los mensajes LBE proporcionan información adicional acerca de la operación del sistema durante el tiempo de ejecución. Por ejemplo, puede visualizar la pantalla siguiente si comete un error en tiempo de ejecución:

```
                        Visualizar Mensajes del Programa

Trabajo 011111/PGMRS/E34 iniciado el 03/04/90 a las 14:35:02 en el
subsistema QINTER  en el Mensaje CPF4101 en XMPLDUMP en COBOLEX (C D F G).









Escriba respuesta, pulse Intro.
  Respuesta ... _____________________________________________________________
_____________________________________________________________________________

F3=Salir  F12=Cancelar
```

Figura 110. Mensaje de Error en Tiempo de Ejecución

Si desplaza el cursor hasta la línea en la que aparece el número de mensaje CPF4101 y pulsa la tecla AYUDA o F1, la información del mensale LBE se visualiza tal como se muestra a continuación:

```
                      Información de Mensaje Adicional

ID de mensaje . . . . . :   LBE7200          Gravedad . . . . . :   99
Tipo de mensaje . . . . :   INQUIRY
Fecha envío . . . . . :   03/04/90         Hora envío . . . . . :   14:37:15
Desde programa .. . . . :   QLREXHAN        Instrucción  . . . . :   0000
A programa  . . . . . :   *EXT            Instrucción  . . . . :   0000

Mensaje . . . . :   Mensaje CPF4101 en XMPLDUMP en COBOLEX (C D F).
Causa . . . . . :   Se detectó el mensaje CPF4101 en la instrucción COBOL. OPEN
  (instrucción MI 007E) en programa XMPLDUMP en COBOLEX.
Recuperación. . :   Entre una G para continuar el programa en el siguiente
  instrucción MI, o C si no desea vuelco, una  D si desea vuelco de los
  identificadores COBOL o una F si desea volcar los identificadores COBOL
  y las variable generadas por el compilador. El texto de mensaje para CPF4101
  es el siguiente: archivo SALES en biblioteca *LIBL no encontrado o archivo
  de datos en línea perdido.
Elecciones posibles de respuesta al mensaje. . . . . :
  C -- No se produce ningún vuelco con formato
  D -- Se produce un vuelco de los identificadores COBOL
  F -- Se produce un vuelco de todas las variables
  G -- Para continuar el programa en la siguiente instrucción MI.
                                                                        Final
Pulse Intro para continuar.

F3=Salir    F10=Visualizar mensajes en anotaciones de trabajo   F12=Cancelar
```

Figura 111. Mensaje de Error en Tiempo de Ejecución–Texto de Segundo Nivel

Los mensajes LBE 7900 a 7999 se utilizan como encabezamientos para la información impresa durante un vuelco con formato COBOL/400.

Los mensajes LBL se describen en el apartado “Mensajes de Compilación” que se muestra a continuación.

El apartado “Respuesta a Mensajes” en la página 345 explica cómo visualizar el texto de mensaje de segundo nivel y cómo contestar los mensajes.

Mensajes de Compilación

Los mensajes LBL se imprimen en el listado del programa cuando se detectan errores durante la compilación del programa. Los mensajes LBL incluyen el mensaje emitido cuando se solicita la señalización de Federal Information Processing Standard (FIPS); para obtener más información acerca de los mensajes FIPS, vea la página 347 de este apéndice.

Listados de Programas

En la salida del compilador, el listado de mensajes COBOL/400 sigue al listado fuente. El listado de mensajes COBOL/400 muestra el identificador del mensaje, la gravedad, el texto, normalmente la ubicación del error y el resumen de los mensajes.

Para obtener más información acerca de los Listados de Programas, consulte el apartado “Listado Fuente” en la página 42.

Respuestas a Mensajes

En un entorno interactivo, se indica un mensaje mediante una o varias de las condiciones siguientes:

- Un mensaje escueto (llamado texto de primer nivel) en la línea de mensajes
- Una imagen invertida resaltando el campo de entrada con error
- Un teclado bloqueado
- El sonido de una alarma (si se instaló la opción de alarma).

Los siguiente párrafos describen brevemente algunos métodos de respuestas a mensajes de error; hay más información disponible en la publicaciones *Guía para Nuevos Usuarios* y *Herramientas para el Desarrollo de Aplicaciones*.

Si la corrección necesaria es obvia desde la pantalla inicial, puede pulsar la tecla Restaurar Error (si el teclado está bloqueado), introduzca la información correcta y continúe el trabajo.

Si el mensaje indica que debe elegir una respuesta (como por ejemplo **C** para cancelar, **D** para volcar los identificadores COBOL, **F** para volcar todas las variables, o **G** para reanudar el proceso en la siguiente instrucción COBOL), las opciones de respuesta se muestran entre paréntesis en el texto de mensaje de primer nivel. Para ver un ejemplo, consulte la Figura 110 en la página 344.

Si la información en la pantalla de información inicial no ofrece los suficientes datos para manejar el error, puede pulsar la tecla AYUDA (después de colocar el cursor en la línea de mensajes, si es necesario) para obtener una pantalla de segundo nivel con información adicional que permite solucionar este error. Para volver a la pantalla inicial, pulse la tecla Intro; luego pulse la tecla Restaurar Error (si el teclado está bloqueado) y efectúe la corrección o respuesta.

Si el error se produce cuando compila o ejecuta un programa, puede que necesite modificar las instrucciones fuente COBOL/400 o los mandatos de lenguaje de control (CL). Consulte la publicación *SEU Guía del Usuario y Manual de Consulta* para obtener información sobre cómo modificar las instrucciones.

Descripciones de Mensajes COBOL

Los mensajes para el programa bajo licencia COBOL/400 empiezan con los prefijos LSC, LBE o LBL.

El comprobador de sintaxis emite los mensajes LSC cuando se utiliza el SEU para introducir la fuente COBOL.

Los mensajes LBE proporcionan información adicional acerca de la operación del sistema durante el tiempo de ejecución.

Los mensajes LBL son mensajes generados por el compilador.

Los números de mensajes se asignan de la siguiente forma:

Mensaje de Error	Descripción
LBE7000 al LBE7199	Mensajes de Escape
LBE7200 al LBE7999	Mensajes en Tiempo de Ejecución
LBE9001	Mensaje de Escape
LBL0000 al LBL0999	Mensajes con gravedad menor a 30
LBL1000 al LBL1999	Mensajes con gravedad superior o igual a 30
LBL8000 al LBL8799	Mensajes de Señalizador FIPS
LBL8800 al LBL8899	Mensajes de Señalización SAA
LSC0000 al LSC1999	Mensajes del controlador de sintaxis

Niveles de Gravedad

El programa bajo licencia COBOL/400 proporciona los siguientes niveles de gravedad de mensajes:

Gravedad Significado

00 Informativo: Este nivel se utiliza para comunicar información al usuario. No se ha producido ningún error. Los mensajes informativos sólo se listan cuando se especifica la opción FLAG (00).

10 Aviso: Este nivel indica que se ha detectado un error pero no es lo suficientemente grave como para interrumpir la ejecución del programa.

20 Error: Este nivel indica que se ha producido un error, pero el compilador está efectuando una operación que podría producir el código deseado.

30 Error Grave: Este nivel indica que se ha detectado un error grave. La compilación ha finalizado, pero el programa no se puede ejecutar.

40 Irrecuperable: Este nivel indica generalmente un error del usuario que provoca la finalización del proceso.

50 Irrecuperable: Este nivel indica generalmente un error del compilador que provoca la finalización del proceso.

99 Acción: Se requiere alguna acción manual, como por ejemplo introducir una respuesta, cambiar los formularios de impresora o sustituir los disquetes.

Nota: Los mensajes 00, 10 y 20 se suprimen cuando se utiliza la opción FLAG(30) de la instrucción PROCESS o cuando el mandato CRTCBLPGM especifica FLAG(30) y la instrucción PROCESS no lo altera temporalmente. Consulte el apartado “Uso de la Instrucción PROCESS para Especificar las Opciones del Compilador” en la página 33 para más información.

El compilador siempre intenta proporcionar diagnósticos completos de todos los textos fuente en el programa, incluso cuando se han detectado errores. Si el compilador no puede continuar en una instrucción determinada, el mensaje establece que el compilador no puede continuar y que el resto de la información se omitirá. Cuando se produce este error, el programador debe examinar la instrucción entera.

El servicio de mensajes OS/400 se utiliza para producir todos los mensajes. Los mensajes del compilador COBOL/400 residen en el archivo de mensajes QLBLMSG, mientras que los mensajes en tiempo de ejecución lo hacen en el archivo QLBLMSGE.

El programa que envía el mensaje, *no* la descripción del mensaje almacenada en el archivo de mensajes, es el que determina las variables de sustitución y los valores de respuesta válida. Sin embargo, pueden cambiarse ciertos elementos de una descripción de mensajes: por ejemplo, el texto, el nivel de gravedad, la respuesta por omisión o el listado del vuelco. Para ello, es preciso especificar otra definición de mensaje mediante el mandato Añadir Descripción de Mensaje (ADDMSGD), colocar la descripción modificada en un archivo de mensajes creado por el usuario, y especificar[1] dicho archivo en el mandato Alterar Temporalmente Archivo de Mensajes (OVRMSGF). El mandato OVRMSGF sirve para que el compilador recupere mensajes del archivo especificado. Consulte los mandatos ADDMSGD y OVRMSGF en la publicación *CL Reference* si desea obtener información adicional.

ATENCIÓN: La sustitución temporal de un mensaje suministrado por IBM con un mensaje creado por el usuario puede provocar resultados imprevisibles. Si no se retienen los valores de respuesta, el programa puede que no responda a ninguna respuesta. Cambiar las respuestas por omisión en los mensajes de tipo *NOTIFY puede afectar la ejecución del programa en modalidad desatendida. Cambiar la gravedad puede provocar la cancelación de un trabajo que no estuviera cancelado. Tenga cuidado al alterar los mensajes suministrados por IBM por mensajes creados por el usuario.

El señalizador Federal Information Processing Standard (FIPS)

El señalizador FIPS puede especificarse para supervisar un subconjunto FIPS COBOL, cualquier módulo opcional, todos los elementos de lenguaje obsoleto, o bien una combinación de un subconjunto FIPS COBOL, de módulos opcionales y de todos los elementos obsoletos.

[1] Si es preciso modificar y sustituir un mensaje suministrado por IBM en *su* archivo de mensajes, póngase en contacto con su representante de servicio.

La supervisión es un análisis que compara la sintaxis utilizada en el programa fuente con la sintaxis incluida en el subconjunto FIPS seleccionado por el usuario y los módulos opcionales. Se identifica cualquier sintaxis utilizada en el programa fuente que no se ajuste al subconjunto FIPS COBOL seleccionado y los módulos opcionales. Cualquier sintaxis para un elemento de lenguaje obsoleto utilizada en el programa fuente también se identificará (según el compilador escogido). Consulte la página 25 para obtener más información acerca de los parámetros para la señalización FIPS.

Las especificaciones 1986 FIPS COBOL son las especificaciones de lenguaje contenidas en ANSI X3.23-1985 COBOL. FIPS COBOL se subdivide en tres subconjuntos y en cuatro módulos opcionales. Los tres subconjuntos se identifican como Inferior, Intermedio y Superior. Los cuatro módulos opcionales son el Transcriptor de Informes, Comunicaciones, Depuración y Segmentación. Estos cuatro módulos opcionales no son parte integral de ninguno de los subconjuntos: sin embargo, ninguna, todas, o cualquier combinación de los módulos opcionales pueden asociarse a cualquier subconjunto. Cualquier programa grabado que se ajuste al estándar FIPS debe ajustarse a uno de los subcampos de 1986 FIPS COBOL. La Tabla 10 muestra los módulos de proceso estándar 1985 ANSI Standard COBOL que se incluyen en cada uno de los subcampos de 1986 FIPS COBOL.

A continuación se ofrece una explicación de la notación utilizada dentro de la tabla:

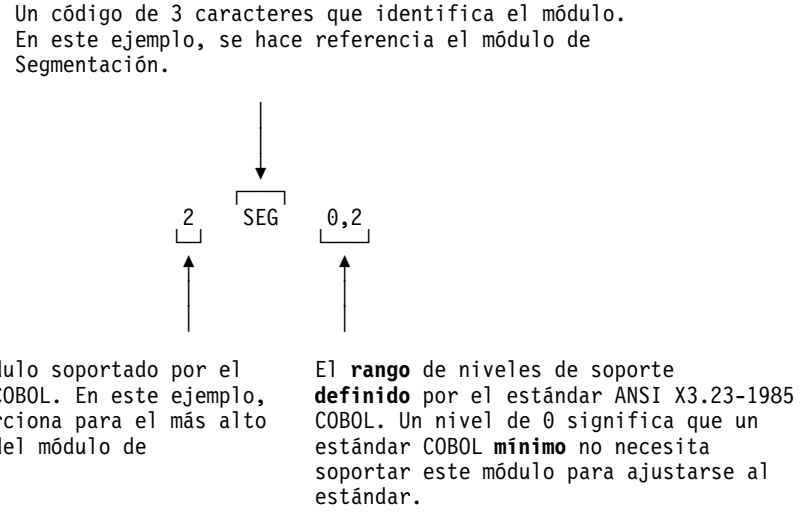


Tabla 10 (Página 1 de 2). Estándar 1985 American National COBOL y Niveles FIPS 1986

Nombre de Módulo 1985 ANSI	**FIPS Superior**	**FIPS Intermedio**	**FIPS Inferior**
Núcleo	2 NUC 1,2	1 NUC 1,2	1 NUC 1,2
E-S Secuencial	2 SEQ 1,2	1 SEQ 1,2	1 SEQ 1,2
E-S Relativa	2 REL 0,2	1 REL 0,2	0 REL 0,2
E-S Indexada	2 INX 0,2	1 INX 0,2	0 INX 0,2
Manipulación del Texto Fuente	2 STM 0,2	1 STM 0,2	0 STM 0,2

Tabla 10 (Página 2 de 2). Estándar 1985 American National COBOL y Niveles FIPS 1986			
Nombre de Módulo 1985 ANSI	**FIPS Superior**	**FIPS Intermedio**	**FIPS Inferior**
Clasificación-Fusión	1 SRT 0,1	1 SRT 0,1	0 SRT 0,1
Comunicación Entre Programas	2 IPC 1,2	1 IPC 1,2	1 IPC 1,2
Transcriptor de Informes	0, o 1 RPW 0,1	0, o 1 RPW 0,1	0, o 1 RPW 0,1
Segmentación	0,1 o 2 SEG 0,2	0,1 o 2 SEG 0,2	0,1 o 2 SEG 0,2
Depuración	0,1 o 2 DEB 0,2	0,1 o 2 DEB 0,2	0,1 o 2 DEB 0,2
Comunicaciones	0,1 o 2 COM 0,2	0,1 o 2 COM 0,2	0,1 o 2 COM 0,2

Nota: El compilador COBOL/400 soporta los módulos opcionales de Segmentación y Depuración.

Los elementos que se especifiquen en el programa fuente COBOL/400 y que no se incluyen en COBOL FIPS 1986 se señalan tal como se describe en el Apéndice C, “Nivel de Soporte del Lenguaje” en la página 339.

Señalización SAA

Puede escoger la realización de la señalización SAA para determinar si las funciones COBOL/400 que utiliza son portables o no a otros entornos SAA COBOL.

La señalización se realiza en aquellas funciones COBOL/400 que se encuentran *fuera* de SAA COBOL, como por ejemplo:

Ampliaciones COBOL/400
Límites del compilador COBOL/400
Palabras reservadas no de SAA
Opciones del compilador.

De esta manera, puede escribir programas que se ajusten a la definición COBOL SAA.

Como ejemplo de señalización SAA en un listado de compilador, consulte la Figura 12 en la página 48. Para realizar la señalización SAA mediante el mandato CRTCBLPGM CL, especifique SAAFLAG(*FLAG). Para realizar la señalización SAA mediante la instrucción PROCESS, especifique SAAFLAG.

Al compilar un programa para que se ajuste a la definición SAA mediante el mandato CRTCBLPGM, especifique lo siguiente:

```
OPTION(*QUOTE *NOSEQUENCE *NONUMBER)
GENOPT(*CRTF *DUPKEYCHK *SYNC)
SAAFLAG(*FLAG)
```

Si utiliza la instrucción PROCESS, especifique lo siguiente:

```
QUOTE, NOSEQUENCE, NONUMBER, CRTF,
DUPKEYCHK, SYNC, SAAFLAG.
```

Para obtener más información sobre cómo especificar la opción para la señalización SAA, consulte el Parámetro SAAFLAG en la página 26 y el apartado “Uso de la Instrucción PROCESS para Especificar las Opciones del Compilador” en la página 33.

Para obtener información acerca de los límites del compilador, consulte el apéndice Límites del Compilador en la publicación *COBOL/400 Reference*.

Apéndice E. Diferencias entre el COBOL ANSI 74 COBOL y el COBOL ANSI 85

Este apéndice identifica los elementos del lenguaje COBOL ANSI 85 que no son compatibles con el COBOL ANSI 74. Estos ítems identifican los cambios y condiciones que los usuarios del COBOL ANSI 74 necesitan para saber cuándo migrar al COBOL ANSI 85.

Consulte el apartado “Estándares Industriales Utilizados en el Diseño del Compilador” en la página xiii para obtener más información acerca del COBOL ANSI 85.

Migración de Programas COBOL ANSI 74 a COBOL ANSI 85

A continuación se muestran algunas de las características o modificaciones del COBOL ANSI 85 que pueden afectar a los programas COBOL ANSI 74:

- La palabra clave ALPHABET debe preceder al nombre del alfabeto en la cláusula nombre del alfabeto en el párrafo SPECIAL-NAMES. Un nombre del alfabeto es una palabra definida por el usuario en el párrafo SPECIAL-NAMES que da nombre a un juego de caracteres o a un orden de clasificación.
- El ítem de datos de clave relativa especificada en la frase RELATIVE KEY no debe contener el símbolo PICTURE “P”.
- La prueba de clase ALPHABETIC es cierta para letras mayúsculas, minúsculas y para el carácter en blanco.
- Cuando no hay más instrucciones para procesar en un programa llamado, se ejecuta una EXIT PROGRAM implícita.
- En una instrucción MERGE no se pueden especificar dos archivos en la misma cláusula SAME AREA o SAME SORT-MERGE AREA. Los únicos archivos en una instrucción MERGE que pueden especificarse en la cláusula SAME RECORD AREA son los asociados con la frase GIVING.
- Dentro de la instrucción READ, no puede especificarse la frase INTO a no ser que:

  Todos los registros asociados con el archivo y el ítem de datos especificado en la frase INTO sean elementos de grupo o ítems alfanuméricos elementales, o sólo esté subordinada una descripción de registro a la entrada de descripción del archivo.
- Dentro de la instrucción RETURN, la frase INTO no se puede especificar a no ser que:

  Todos los registros asociados con el archivo y el ítem de datos especificado en la frase INTO sean ítems de grupo o ítems alfanuméricos elementales, o sólo esté subordinada una descripción de registro a la entrada de descripción del archivo de clasificación-fusión.
- Indicador de posición del archivo - El concepto de un puntero de registro actual se ha cambiado a indicador de posición de archivo.
- Palabras reservadas - Se han añadido nuevas palabras reservadas.
- Estado de E/S - Se han añadido nuevos valores de estado de E/S.

- EL pseudotexto-1 de la instrucción COPY no debe estar únicamente compuesto por una coma o por dos puntos.
- Un ítem de datos que aparece en la frase USING de la cabecera de la División de Procedimiento no debe tener una cláusula REDEFINES en la entrada de descripción de datos.
- Si no se especifica la frase FOOTING, no existe ninguna condición de fin de página independiente de la condición de desbordamiento de página.
- La frase NO REWIND no puede especificarse en una instrucción CLOSE que tenga una frase REEL/UNIT.
- Las instrucciones CANCEL y STOP RUN cierran todos los archivos abiertos.
- Cuando un ítem receptor es un elemento de datos de longitud variable y contiene el objeto de la frase DEPENDING ON, se utilizará la longitud máxima del elemento.
- Dentro de la frase VARYING ... AFTER de la instrucción PERFORM, el identificador-2 se aumenta antes de establecer el identificador-5.
- Cualquier subindexación para el identificador-4 de la frase REMAINDER de la instrucción DIVIDE se evalúa después de que el resultado de la operación DIVIDE se haya almacenado en el identificador-3 de la frase GIVING.
- Las frases ADVANCING PAGE y END-OF-PAGE no deben estar en la misma instrucción WRITE.
- La serie de caracteres de PICTURE de un ítem alfanumérico sólo puede contener el símbolo “A”. No se permite ninguna edición para la categoría de datos alfabéticos.

  Nota: Un carácter alfabético es una letra o un carácter en blanco.
- Cuando se hace referencia a un ítem de datos descritos por una PICTURE y dicho elemento contiene una “P”, se considera que las posiciones del dígito especificadas por “P” contienen ceros en las operaciones siguientes:
  - Toda operación que precise un operando emisor de tipo numérico
  - Una instrucción MOVE donde el operando emisor sea numérico y cuya serie de caracteres PICTURE contenga el símbolo “P”
  - Una instrucción MOVE donde el operando emisor sea numérico editado y cuya serie de caracteres PICTURE contenga el símbolo “P”, y el operando sea numérico o numérico editado.
  - Una operación de comparación donde ambos operandos sean numéricos.
- El literal en la cláusula CURRENCY SIGN no puede ser una constante figurativa.
- Si la instrucción COPY aparece en una entrada de comentarios, se considera parte de la entrada de comentarios.
- Se definen los siguientes casos de exponenciación:
  - Si una expresión con un valor cero se eleva a una potencia negativa o igual a cero, se produce una condición de error de tamaño.
  - Si la evaluación de la exponenciación da un número real a la vez positivo y negativo, se devuelve el número positivo.

  - Si como resultado de la evaluación no existe un número real, existe una condición de error.

- Cuando el literal de constante figurativa ALL no se asocia a otro ítem de datos, la longitud de la serie de caracteres es la longitud del literal.

Apéndice F. Soporte del Juego de Caracteres de Idiomas Internacionales de Doble Byte

Ampliación de IBM

Este apéndice sólo describe las mejoras realizadas en el lenguaje de programación COBOL para escribir programas que procesan caracteres de doble byte.

En concreto, este apéndice describe dónde se pueden utilizar los caracteres del Juego de Doble Byte (DBCS) en cada parte de un programa COBOL y las consideraciones que hay que saber para trabajar con datos DBCS en el lenguaje COBOL/400.

Existen dos formas de especificar caracteres DBCS:

- Caracteres DBCS entre paréntesis
- Caracteres de datos DBCS gráficos

Por lo general, COBOL maneja caracteres DBCS entre paréntesis de la misma manera que maneja caracteres alfanuméricos. Los caracteres **DBCS entre paréntesis** pertenecen a una serie de caracteres en la que cada carácter está representado por dos bytes. El carácter empieza con un carácter de desplazamiento a teclado ideográfico (SO) y finaliza con un carácter de desplazamiento a teclado estándar (SI). Por lo tanto, depende exclusivamente del usuario conocer (o disponer de la comprobación del programa COBOL) los ítems de datos que contienen los caracteres DBCS, así como asegurarse de que el programa recibe y procesa correctamente esta información.

De este modo, el usuario ya puede utilizar las descripciones DDS que definen los campos de datos DBCS gráficos con los programas COBOL/400. Los caracteres **DBCS gráficos** pertenecen a una serie de caracteres en la que cada carácter está representado por dos bytes. Esta serie de caracteres no contiene ni caracteres de desplazamiento a teclado ideográfico ni caracteres de desplazamiento a teclado estándar. No es posible utilizar programas que contengan datos gráficos. Para obtener más información sobre cómo especificar ítems de datos gráficos mediante los programas COBOL/400, consulte el apartado “Campos Gráficos DBCS” en la página 139.

Utilización de Caracteres DBCS en Literales

Tipos de Literales

Hay dos tipos de literales en los que se pueden utilizar los caracteres DBCS: el literal DBCS y el literal mixto. Un literal mixto está compuesto por los caracteres del Juego de Caracteres de Doble Byte (DBCS) y el Juego de Caracteres de un Sólo Byte (SBCS).

Literales DBCS: El compilador COBOL reconoce los caracteres DBCS en los literales DBCS cuando se utiliza la opción GRAPHIC en la instrucción PROCESS.

Nota: La opción GRAPHIC en la instrucción PROCESS no debe confundirse con el valor *GRAPHIC del parámetro CVTOPT del mandato CRTCBLPGM ni con la opción CVTGRAPHIC en la instrucción PROCESS, que se utilizan

para especificar los datos gráficos de doble byte desde una descripción DDS. Para obtener más información sobre cómo especificar datos gráficos, consulte el apartado “Campos Gráficos DBCS” en la página 139.

Literales DBCS/SBCS: El compilador COBOL reconoce caracteres DBCS en literales DBCS/SBCS (mixtos), cuando se está en un sistema DBCS y no se especifica ni la opción GRAPHIC ni la instrucción PROCESS.

Especificación de Literales que contienen Caracteres DBCS

Cuando se especifica un literal que contiene caracteres DBCS, siga las mismas normas que se aplican al especificar literales alfanuméricos, además de las siguientes reglas específicas para los tipos de literal:

Especificación de un Literales DBCS: Cuando se especifica un literal DBCS, tenga en cuenta lo siguiente:

El formato de un literal DBCS es:

"0_EK1K2$0_F$"

- Unas comillas abren y cierran el literal.
- Un carácter de desplazamiento a teclado ideográfico (0_E) sigue inmediatamente a la comilla inicial y ocupa 1 byte. Un carácter de **desplazamiento a teclado ideográfico** es un carácter de control (hex 0E) que indica el inicio de una serie de caracteres de doble byte.
- Un carácter de desplazamiento a teclado estándar (0_F) precede inmediatamente la comilla final y ocupa 1 byte. Un carácter de **desplazamiento a teclado estándar** es un carácter de control (HEX 0F) que indica el final de una serie de caracteres de doble byte.
- Todos los caracteres DBCS aparecen entre los caracteres de desplazamiento a teclado ideográfico y de desplazamiento a teclado estándar.
- Sólo los caracteres DBCS pueden aparecer en el literal (las series nulas no son válidas).

La longitud máxima de un literal DBCS es de 80 caracteres DBCS, incluyendo los caracteres de control de desplazamiento. (Si se cuentan juntos son equivalentes en longitud a un carácter DBCS). Los caracteres de control de desplazamiento son parte del literal y toman parte en todas las operaciones.

Consulte el apartado “Continuación de los Literales DBCS en una Línea Nueva” en la página 358 para obtener más información sobre cómo ampliar los literales DBCS.

Especificación de un Literal DBCS/SBCS: cuando se especifica un literal DBCS/SBCS, recuerde que:

- Los literales DBCS/SBCS pueden tener diferentes formas. A continuación se muestra sólo un posible ejemplo:

  "SINGLE0_EK1K2K3$0_F$BYTES"

- USAGE DISPLAY debe ser explícita o implícita.
- Unas comillas abren y cierran el literal.

- Los caracteres EBCDIC pueden aparecer antes y después de cualquier serie de caracteres DBCS en el literal mixto.
- Todas las series de caracteres DBCS aparecen entre caracteres de desplazamiento a teclado ideográfico y a teclado estándar.
- Es preciso doblar todas las comillas DBCS que aparezcan dentro del literal. No es necesario doblar las comillas DBCS que se encuentren dentro del literal.
- Sólo puede utilizar series nulas DBCS (caracteres de desplazamiento a teclado ideográfico y a teclado estándar sin ningún carácter DBCS) *sólo* cuando el literal contenga al menos un carácter SBCS.

Los caracteres de desplazamiento a teclado ideográfico y a teclado estándar no pueden jerarquizarse.

Los caracteres de control de desplazamiento son parte del literal y toman parte en todas las operaciones.

Los literales DBCS/SBCS no pueden continuar en más líneas. Están restringidos al espacio del ÁREA B de una línea.

Otras Consideraciones

Comillas: Aunque el comentario anterior utiliza el término *comillas* para describir el carácter que identifica un literal, el carácter que realmente se ha utilizado puede variar según la opción especificada en el mandato CL CRTCBLPGM o en la instrucción PROCESS. Si se especifica la opción APOST, se utiliza un apóstrofe ('). De lo contrario, se utilizan las comillas ("). En este apéndice, las *comillas* hacen referencia tanto a los apóstrofes como a las comillas. El carácter que elija no afecta a las reglas que especifican el literal.

Caracteres de Desplazamiento: Los caracteres de desplazamiento a teclado ideográfico y a teclado estándar separan los caracteres EBCDIC de los caracteres DBCS. Forman parte de los literales DBCS y DBCS/SBCS. Por lo tanto, los caracteres de codificación de desplazamiento participan en todas las operaciones cuando aparecen en los literales DBCS o DBCS/SBCS.

Caracteres DBCS comprobados por el compilador COBOL

Cuando el compilador COBOL encuentra una serie DBCS, la comprueba explorando los caracteres DBCS uno a uno.

Las condiciones siguientes provocan que el compilador COBOL diagnostique un literal que contenga caracteres DBCS como *no válido:*

- La sintaxis del literal es incorrecta.
- El literal DBCS es más largo que una línea y *no* respeta las normas para la continuación de literales no numéricos. (Consulte el apartado “Continuación de los Literales DBCS en una Línea Nueva” en la página 358 para obtener más información.)
- El literal DBCS/SBCS es más largo que una línea.

Cuando el compilador COBOL encuentra un literal DBCS que no es válido, genera un mensaje de error y luego procesa el literal como un literal alfanumérico.

Para cada literal DBCS o SBCS que no es válido, el compilador genera un mensaje de error y acepta o ignora el literal.

Continuación de los Literales DBCS en una Línea Nueva

Para continuar un literal DBCS en otra línea del código fuente, realice *todos* los pasos siguientes:

- Coloque un carácter de desplazamiento a teclado estándar en la columna 71 ó 72 de la línea a continuar (si coloca el carácter en la columna 71, se ignora el espacio en blanco de la columna 72)
- Coloque un guión (-) en la columna 7 (área de continuación) de la línea nueva
- Coloque unas comillas, a continuación un carácter de desplazamiento a teclado ideográfico, y después el resto del literal en el Área B de la línea nueva.

Por ejemplo:

```
-A 1 B
  ⋮
 01  DBCS1             PIC X(12)                VALUE "0EK1K2K30F
-    "0EK4K50F".
  ⋮
```

El valor de DBCS1 es "0_EK1K2K3K4K50_F".

El carácter de desplazamiento a teclado estándar, las comillas y el carácter de desplazamiento a teclado ideográfico utilizados para continuar una línea no se cuentan en la longitud del literal DBCS. Se cuenta el primer carácter de desplazamiento a teclado ideográfico y el primer carácter de desplazamiento a teclado estándar.

En qué lugar puede utilizar los caracteres DBCS en un programa COBOL

En general, puede utilizar los literales DBCS o DBCS/SBCS allí dónde se permitan los literales no numéricos. Sin embargo, los literales que se utilicen para lo siguiente no pueden incluir caracteres de doble byte:

- Cláusula ALPHABET-NAME
- Cláusula CURRENCY SIGN
- Cláusula ASSIGN
- Cláusula CLASS
- Instrucción CALL
- Instrucción CANCEL

Nota: No puede utilizar los caracteres DBCS para palabras o nombres COBOL. Consulte la publicación *COBOL/400 Reference* para obtener más información sobre cómo dar formato a los nombres del sistema COBOL, palabras reservadas y palabras definidas por el usuario, como por ejemplo nombres de datos y nombres de archivos.

Forma de Escribir Comentarios

Se puede escribir un comentario que contenga caracteres DBCS en un programa COBOL colocando un asterisco (*) o barra (/) en la columna siete de la línea del programa. Los dos símbolos hacen que el compilador trate la información que sigue a la columna siete como documentación. La barra también provoca que se genere un salto de página. Puesto que el compilador COBOL no comprueba el contenido de las líneas de comentarios, los caracteres DBCS que se encuentren dentro de comentarios no se detectan. Los caracteres DBCS que no son válidos pueden hacer que el listado del compilador se imprima de forma incorrecta.

División de Identificaciones

Se pueden colocar entradas de comentarios que contengan caracteres DBCS en cualquier parte de la División de Identificaciones excepto en el párrafo PROGRAM-ID. El nombre del programa especificado en el párrafo PROGRAM-ID debe ser alfanumérico.

División de Entornos

Sección de Configuración

Es posible utilizar los caracteres DBCS en las entradas de comentarios sólo en el párrafo de la Sección de Configuración. Todos los nombres de función, nombres mnemotécnicos, nombres de condición y nombres del alfabeto deben especificarse con los caracteres alfanuméricos. Para las entradas SOURCE-COMPUTER y OBJECT-COMPUTER, utilice el nombre del sistema alfanumérico:

IBM-AS400

No es posible utilizar los literales DBCS o DBCS/SBCS en la Sección de Configuración. En su lugar, utilice los literales alfanuméricos para definir un nombre del alfabeto y el literal en la cláusula CURRENCY SIGN del párrafo SPECIAL-NAMES. No hay alfabeto DBCS. En su lugar, utilice el juego de caracteres EBCDIC.

Sección Entrada-Salida

Especifique todos los nombres de datos, nombres de archivos y nombres de asignación que utilicen caracteres alfanuméricos. Puede utilizar los caracteres DBCS en los comentarios.

Para archivos indexados, el nombre de datos en la cláusula RECORD KEY puede referirse a un ítem de datos DBCS o DBCS/SBCS dentro de un registro. El número de campos en el registro, más el número de posiciones que ocupa la clave del registro, no puede ser superior a 120.

Nota: Cada carácter DBCS ocupa dos posiciones y los caracteres de control de desplazamiento ocupan cada uno una posición. Asegúrese de que la descripción de datos de la clave y la posición de la clave en el archivo coincidan con las especificadas en el momento de creación del archivo.

No puede utilizar datos DBCS y DBCS/SBCS como RELATIVE KEY en los archivos relativos.

Párrafo de Control de Archivos

Cláusula ASSIGN: No puede utilizar los literales que contengan caracteres DBCS en la cláusula ASSIGN para especificar un medio externo como una impresora o una base de datos.

División de Datos

Sección de Archivos

Para la entrada FD (Descripción de Archivo), se puede utilizar ítems de datos DBCS o DBCS/SBCS o literales en la cláusula VALUE OF. La cláusula DATA RECORDS puede referirse sólo a ítems de datos. Debido a que el compilador COBOL/400 trata la cláusula VALUE OF y la cláusula DATA RECORDS en la Sección de Archivos como documentación, ninguna cláusula tiene efecto alguno cuando ejecuta el programa. No obstante, el compilador COBOL comprueba todos los literales de la cláusula VALUE OF para asegurarse de que son válidos.

Para cintas magnéticas, el sistema sólo puede leer caracteres DBCS o bien grabar a caracteres DBCS a la cinta en el formato EBCDIC. El sistema no puede realizar funciones de cinta que involucren una cinta con formato ASCII. Defina el nombre del alfabeto en la cláusula CODE-SET como NATIVE. Utilice los caracteres alfanuméricos para especificar el nombre del alfabeto.

Sección de Almacenamiento de Trabajo

Cláusula REDEFINES: las normas existentes para los datos redefinidos también se aplican a los datos que contienen caracteres DBCS. Cuando determina la longitud de un ítem de datos redefinido o para redefinir, recuerde que cada carácter DBCS es el doble de largo que un carácter alfanumérico.

Asegúrese también de que los ítems de datos redefinidos contengan los caracteres de control de desplazamiento cuando y donde sea necesario.

Cláusula OCCURS: Utilice esta cláusula para definir las tabla para el almacenamiento de datos DBCS o DBCS/SBCS. Si especifica la frase ASCENDING/DESCENDING KEY, el COBOL asume que los contenidos de la tabla están en la secuencia de clasificación EBCDIC del programa. Los caracteres de control de desplazamiento en los datos DBCS y DBCS/SBCS forman parte del orden de clasificación.

Para obtener más información sobre el manejo de tablas que contienen caracteres DBCS, consulte el apartado “Manejo de Tablas–Instrucción SEARCH” en la página 367.

Cláusula JUSTIFIED RIGHT: Utilice la cláusula JUSTIFIED RIGHT para alinear los datos DBCS o DBCS/SBCS en la posición más a la derecha de un campo receptor elemental. Si el campo receptor es inferior al campo emisor, COBOL trunca los caracteres que estén más a la derecha. Si el campo receptor es más largo que el campo emisor, COBOL llena el espacio inutilizado en la parte izquierda del campo receptor con espacios en blanco.

La cláusula JUSTIFIED no afecta al valor inicial en la cláusula VALUE.

Cláusula VALUE: Se pueden utilizar los literales DBCS o DBCS/SBCS para especificar un valor inicial para un ítem de datos que no sea numérico, o definir los valores para las entradas del nombre de condición del nivel 88.

Todos los caracteres de control de desplazamiento en el literal se consideran parte de la serie PICTURE del literal, excepto cuando se utiliza para continuar una línea nueva. Cuando continúa un literal DBCS, el compilador *no* incluye el carácter de desplazamiento a teclado estándar en la columna 71 ó 72, o la comilla inicial (") y el carácter de desplazamiento a teclado ideográfico en la siguiente línea como parte del literal DBCS. De todas formas, asegúrese de que el literal DBCS no exceda el tamaño del ítem de datos especificado en la cláusula PICTURE, de lo contrario se produce el truncamiento.

Nota: Los literales DBCS/SBCS mixtos no pueden continuar en una línea nueva.

Cuando utilizan literales que contienen caracteres DBCS en la cláusula VALUE para las entradas del nombre de condición de nivel 88, COBOL trata los caracteres DBCS como alfanuméricos. Por lo tanto, siga las normas que rigen la especificación de datos alfanuméricos, incluyendo la permisión de una opción THROUGH. Esta opción utiliza el orden de clasificación EBCDIC normal, pero recuerde que los caracteres de control de desplazamiento y los datos DBCS/SBCS forman parte de la secuencia de clasificación.

Cláusula PICTURE: Utilice el símbolo X de PICTURE para definir los ítems de datos DBCS y DBCS/SBCS. Debido a que los caracteres DBCS son el doble de largos que los alfanuméricos, y están dentro de los caracteres de control de desplazamiento, debe definir un ítem de datos DBCS que contenga *n* caracteres DBCS como

```
PICTURE X(2n+2)
```

Un elemento de datos DBCS/SBCS que contenga *m* caracteres SBCS, y una serie de *n* caracteres DBCS se definiría como

```
PICTURE X(m+2n+2)
```

Puede utilizar todos los símbolos PICTURE alfanuméricos editados para los ítems de datos DBCS y DBCS/SBCS. Los símbolos de edición causan el mismo efecto en los datos DBCS de estos ítems que en los ítems de datos alfanuméricos. Compruebe que haya obtenido los resultados deseados.

Cláusula RENAMES: Utilice esta cláusula para especificar los agrupamientos alternativos de ítems de datos elementales. Las normas existentes para la redenominación de ítems de datos alfanuméricos también se aplica para los ítems de datos DBCS y DBCS/SBCS.

División de Procedimientos

Declarativas

Un identificador en la sentencia USE FOR DEBUGGING de la sección DECLARATIVES puede referirse a un ítem de datos DBCS o DBCS/SBCS.

No puede utilizar los caracteres DBCS para los nombres de archivos o de procedimientos en la sentencia USE FOR DEBUGGING.

Expresiones Condicionales

Debido a que los nombres de condición (entradas de nivel 88) pueden referirse a los ítems de datos que contienen caracteres DBCS, puede utilizar la condición de nombre de condición para comprobar estos datos. (Consulte el apartado “Cláusula VALUE” en la página 361). Siga las normas que se listan en el manual *COBOL/400 Reference* para conocer mejor el uso de las variables condicionales y de los nombres de condición.

Puede utilizar los ítems de datos DBCS o DBCS/SBCS o los literales como operandos en una condición de relación. Debido a que COBOL trata los datos DBCS como alfanuméricos, todas las comparaciones se producen de acuerdo con las normas de los operandos alfanuméricos. Tenga siempre presente que:

- El sistema no reconoce el contenido mixto.
- El sistema utiliza los códigos de desplazamiento en las comparaciones de datos DBCS y DBCS/SBCS.
- El sistema compara los datos utilizando la secuencia de clasificación EBCDIC, o una secuencia definida por el usuario.
- En una comparación de ítem DBCS o DBCS/SBCS con ítems similares de tamaño desigual, se llena el ítem más pequeño de la derecha con espacios EBCDIC.

Consulte la sección “Párrafo SPECIAL-NAMES” en el manual *COBOL/400 Reference* para obtener más información.

Puede utilizar las condiciones de clase y de estado de conmutador tal y como se describe en la publicación *COBOL/400 Reference*.

Instrucciones de Entrada/Salida

Instrucción ACCEPT: Los datos de entrada recibidos desde un dispositivo utilizando la instrucción ACCEPT de Formato 1 pueden incluir datos DBCS o DBCS/SBCS. Todos los datos DBCS y DBCS/SBCS deben identificarse mediante la sintaxis adecuada. Los datos de entrada, incluyendo los caracteres de control de desplazamiento, sustituyen los contenidos existentes del identificador. COBOL no realiza la edición ni la comprobación de error en los datos.

Si utiliza la instrucción ACCEPT de Formato 3 para obtener información OPEN-FEEDBACK acerca del archivo, dicha información incluye un campo que muestra si el archivo contiene datos DBCS o DBCS/SBCS.

La información que se recibe desde el área de datos local por medio de una instrucción ACCEPT de Formato 4 puede incluir series de caracteres DBCS o DBCS/SBCS. La información recibida sustituye los contenidos existentes. COBOL no realiza ninguna edición o comprobación de errores. Esto también se aplica a la información recibida desde el área de datos PIP por medio de una instrucción ACCEPT de Formato 5.

Al utilizar la instrucción ACCEPT de Formato 6, se pueden obtener los atributos de una pantalla de estación de trabajo y de su teclado. Para estaciones de pantalla que pueden visualizar caracteres DBCS, el sistema establece el valor apropiado en el ítem de datos ATTRIBUTE-DATA. No se puede utilizar los caracteres DBCS para nombrar un dispositivo.

Si utiliza una instrucción ACCEPT (Formato 7) ampliada para la entrada de la estación de trabajo a nivel de campo, debe asegurarse de que los datos DBCS no se dividen entre líneas. COBOL no realiza ninguna edición o comprobación de errores.

Instrucción DISPLAY: Puede especificar los ítems de datos DBCS o DBCS/SBCS o los literales en la instrucción DISPLAY. Puede mezclar los tipos de datos. Los datos DBCS y DBCS/SBCS, de ítems de datos o literales, se envían tal y como aparecen en el dispositivo del programa o en el área de datos local que es el destino nombrado en la instrucción DISPLAY.

Debido a que COBOL ignora las características del dispositivo en que se visualizan los datos, debe asegurarse de que los datos DBCS y DBCS/SBCS sean correctos. Es posible que sea preciso especificar la opción de visualización ampliada *NOUNDSPCHAR (o la opción del parámetro de instrucción del proceso equivalente) cuando se compile el programa, para asegurarse de que la estación de trabajo puede manejar correctamente los datos DBCS.

Nota: ALL es una opción válida para literales mixtos.

Si utiliza una instrucción DISPLAY (Formato 3) ampliada para la salida de la estación de trabajo a nivel de campo, debe asegurarse de que los datos DBCS no se dividan entre líneas.

Instrucción READ: Puede utilizar los ítems de datos DBCS o DBCS/SBCS como la RECORD KEY para un archivo indexado. Consulte el apartado “Sección Entrada-Salida” en la página 359 para obtener más información.

Frase INTO: Es posible leer un registro de un ítem de datos DBCS o DBCS/SBCS mediante la frase INTO. Esta frase provoca que se realice una instrucción MOVE (pero sin la opción CORRESPONDING). El compilador mueve los datos DBCS y DBCS/SBCS de la misma manera que mueve los datos alfanuméricos. No asegura que estos datos sean válidos.

Instrucción REWRITE: Utilice la frase FROM de esta instrucción para transferir datos DBCS o DBCS/SBCS desde un ítem de datos DBCS o DBCS/SBCS a un registro existente. La frase FROM provoca que ambos tipos de datos se muevan de la misma manera que la frase INTO en la instrucción READ. (Consulte el apartado “Instrucción READ”.)

Instrucción START: Si utiliza caracteres DBCS en la clave de un archivo indexado, especifique un ítem de datos correspondientes en la frase KEY de la instrucción START.

Debe cumplirse una de las condiciones siguientes:

- El elemento de datos debe ser el mismo que el ítem de datos especificado en la cláusula RECORD KEY del párrafo FILE-CONTROL.
- El ítem de datos tiene el mismo primer carácter que la clave de registro y no es más largo que la clave de registro.

Puede especificar los operadores válidos (como por ejemplo EQUAL, GREATER THAN, NOT LESS THAN) en la frase KEY. El sistema puede seguir la secuencia de clasificación EBCDIC o la secuencia definida por el usuario.

Instrucción WRITE: Utilice la frase FROM de esta instrucción para grabar los datos DBCS o DBCS/SBCS en un registro. Esta frase mueve los datos de la misma manera que la instrucción REWRITE. (Consulte el apartado “Instrucción REWRITE”).

Debe incluir los caracteres de control de desplazamiento cuando grabe los datos en un archivo de dispositivo.

Instrucciones para la Manipulación de Datos

Instrucciones Aritméticas: Debido a que COBOL trata los caracteres DBCS de la misma manera que lo hace con los caracteres alfanuméricos, no utilice los caracteres DBCS en las operaciones numéricas, ni los manipule con instrucciones aritméticas.

Instrucción INSPECT: Puede utilizar cualquier ítem de datos DBCS o DBCS/SBCS como operando para la instrucción INSPECT. En este tipo de operaciones, el sistema cuenta y sustituye cada mitad de un carácter DBCS, incluyendo los caracteres de control de desplazamiento. Por lo tanto, puede que los datos no coincidan adecuadamente.

Puede utilizar cualquier combinación de operandos de doble byte y alfanuméricos con literales o ítems de datos de caracteres de doble byte. Si utiliza la frase REPLACING, puede sustituir las partes del ítem inspeccionado por datos alfanuméricos, o viceversa.

No puede sustituir una serie de caracteres por otra de una longitud distinta. Téngalo presente cuando sustituya caracteres alfanuméricos por caracteres DBCS, o viceversa.

Si quiere controlar la utilización de la instrucción INSPECT con ítems que contengan caracteres DBCS, defina los ítems de datos que contengan caracteres de control de desplazamiento. Utilice los caracteres de desplazamiento a teclado ideográfico o a teclado estándar como los operandos BEFORE/AFTER en la instrucción INSPECT.

El ejemplo siguiente muestra cómo puede utilizar la instrucción INSPECT para sustituir un carácter DBCS por otro.

```
01  SUBJECT-ITEM                PICTURE X(50).
01  DBCS-CHARACTERS             VALUE "0EK1K20F".
    05  SHIFT-OUT               PICTURE X.
    05  DBCS-CHARACTER-1        PICTURE XX.
    05  DBCS-CHARACTER-2        PICTURE XX.
    05  SHIFT-IN                PICTURE X.
```

La instrucción INSPECT debe codificarse de la manera siguiente:

```
INSPECT SUBJECT-ITEM
    REPLACING ALL DBCS-CHARACTER-1
              BY  DBCS-CHARACTER-2
    AFTER INITIAL SHIFT-OUT.
```

Nota: El uso de la frase AFTER INITIAL SHIFT-OUT sirve para evitar el riesgo de sustituir de forma accidental dos caracteres alfanuméricos consecutivos que tengan los mismos valores EBCDIC como DBCS-CHARACTER-1 (en casos dónde SUBJECT-ITEM datos DBCS/SBCS).

También puede utilizar la instrucción INSPECT para determinar si un ítem de datos contiene caracteres DBCS, de forma que se pueda producir el proceso apropiado. Por ejemplo:

```
01  SUBJECT-FIELD          PICTURE X(50).
01  TALLY-FIELD            PICTURE 9(3) COMP.
01  SHIFTS                 VALUE "0E0F".
    05  SHIFT-OUT          PICTURE X.
    05  SHIFT-IN           PICTURE X.
```

En la División de Procedimientos debe introducir lo siguiente:

```
MOVE ZERO TO TALLY-FIELD.
INSPECT SUBJECT-FIELD TALLYING TALLY-FIELD
                                FOR ALL  SHIFT-OUT.
IF TALLY-FIELD IS GREATER THAN ZERO THEN
    PERFORM DBCS-PROCESSING
ELSE
    PERFORM A-N-K-PROCESSING.
```

Instrucción MOVE: Todos los caracteres DBCS se mueven como series de caracteres alfanuméricos. El sistema no convierte ni examina los datos.

Puede mover los literales DBCS/SBCS a ítems de grupo e ítems alfanuméricos.

Si la longitud del campo receptor es distinta a la del campo emisor, COBOL realiza una de las acciones siguientes:

- Trunca los caracteres del ítem emisor si es más largo que el ítem receptor. Esta operación puede reducir la integridad de los datos.
- Llena el ítem emisor con espacios en blanco si es más corto que el ítem receptor.

Para comprender mejor el efecto de los símbolos de edición en la cláusula PICTURE del ítem de datos receptor, consulte la publicación *COBOL/400 Reference*.

Instrucción SET (Formato Nombre-Condición): Cuando se establece el nombre de condición TRUE en esta instrucción, COBOL desplaza el literal de la cláusula VALUE hasta el ítem de datos asociado. Se puede mover un literal con caracteres DBCS.

Instrucción STRING: Puede utilizar la instrucción STRING para construir un ítem de datos que contenga subcampos DBCS o DBCS/SBCS. Todos los datos de los ítems de datos fuente o literales, incluyendo los caracteres de control de desplazamiento, la mitad de un carácter DBCS a la vez, se desplazan hasta el ítem de datos receptor.

Instrucción UNSTRING: La instrucción UNSTRING trata los datos DBCS y DBCS/SBCS de la misma forma que los datos alfanuméricos. La operación UNSTRING se realiza en una mitad del carácter DBCS a la vez.

Los ítems de datos pueden contener caracteres alfanuméricos y DBCS dentro del mismo campo.

Utilice la frase DELIMITED BY para localizar subcampos de doble byte y alfanuméricos dentro de un campo de datos. Identifique los ítems de datos que

contengan caracteres de control de desplazamiento, y utilícelos como identificadores en la frase DELIMITED BY. Observe con detenimiento los ejemplos siguientes para comprender mejor cómo se lleva a cabo lo expuesto anteriormente. Utilice la variable POINTER para continuar examinando los subcampos del campo emisor.

Después de que el sistema realice la operación UNSTRING, puede comprobar los delimitadores almacenados mediante las frases DELIMITER IN con los caracteres de control de desplazamiento para ver qué subcampos contienen caracteres DBCS y cuáles contienen caracteres alfanuméricos.

El ejemplo siguiente muestra cómo puede establecer los campos para preparar la operación no de serie en la serie de caracteres que contienen datos DBCS/SBCS:

```
01  SUBJECT-FIELD      PICTURE X(40)
01  FILLER.
    05  UNSTRING-TABLE OCCURS 4 TIMES.
        10  RECEIVER   PICTURE X(40).
        10  DELIMTR    PICTURE X.
        10  COUNTS     PICTURE 99 COMP.
01  SHIFTS             VALUE "0E0F".
    05  SHIFT-OUT      PICTURE X.
    05  SHIFT-IN       PICTURE X.
```

Codifique la instrucción UNSTRING de la manera siguiente:

```
UNSTRING SUBJECT-FIELD  DELIMITED BY SHIFT-OUT
                                     OR SHIFT-IN
INTO RECEIVER (1) DELIMITER IN DELIMTR (1)
                  COUNT     IN COUNTS (1)
INTO RECEIVER (2) DELIMITER IN DELIMTR (2)
                  COUNT     IN COUNTS (2)
INTO RECEIVER (3) DELIMITER IN DELIMTR (3)
                  COUNT     IN COUNTS (3)
INTO RECEIVER (4) DELIMITER IN DELIMTR (4)
                  COUNT     IN COUNTS (4)
ON OVERFLOW PERFORM UNSTRING-OVERFLOW-MESSAGE.
```

Esta instrucción UNSTRING divide una serie de caracteres en partes alfanuméricas y DBCS. Teniendo en cuenta que los datos en la serie de caracteres son válidos, un valor del delimitador de desplazamiento a teclado ideográfico indica que el campo receptor correspondiente contiene datos alfanuméricos, mientras que un valor de desplazamiento a teclado estándar indica que el campo receptor correspondiente tiene datos DBCS. Puede comprobar los ítems de datos COUNT para determinar si cada campo receptor ha recibido todos los caracteres. La figura siguiente es un ejemplo que muestra los resultados de la operación UNSTRING descrita anteriormente:

```
SUBJECT-FIELD = ABC0EK1K2K30FD0EK4K5K60F
RECEIVER (1) = ABC       DELIMTR (1) = 0E   COUNTS (1) = 3
RECEIVER (2) = K1K2K3    DELIMTR (2) = 0F   COUNTS (2) = 6
RECEIVER (3) = D         DELIMTR (3) = 0E   COUNTS (3) = 1
RECEIVER (4) = K4K5K6    DELIMTR (4) = 0F   COUNTS (4) = 6


SUBJECT-FIELD = 0EK1K2K30FABC0EK40F
RECEIVER (1) = (blanks)  DELIMTR (1) = 0E   COUNTS (1) = 0
RECEIVER (2) = K1K2K3    DELIMTR (2) = 0F   COUNTS (2) = 6
RECEIVER (3) = ABC       DELIMTR (3) = 0E   COUNTS (3) = 3
RECEIVER (4) = K4        DELIMTR (4) = 0F   COUNTS (4) = 2
```

Instrucciones de Bifurcación de Procedimientos

Puede utilizar un literal DBCS o DBCS/SBCS como el operando para la instrucción STOP. Cuando lo haga, el sistema visualiza el literal y al mismo tiempo lo introduce en la estación de trabajo para tareas interactivas. Para trabajos por lotes, el sistema subrayada la ubicación habitual en la que aparecerían literales en la cola de mensajes del operador del sistema. El sistema no edita ni comprueba los contenidos del literal.

Manejo de Tablas–Instrucción SEARCH

Puede realizar una instrucción SEARCH de Formato 1 (búsqueda secuencial de una tabla) en una tabla que contenga datos DBCS o DBCS/SBCS en medio carácter DBCS a la vez.

También puede realizar la instrucción SEARCH de Formato 2 (SEARCH ALL) con una tabla DBCS o DBCS/SBCS. Ordene la tabla de acuerdo con la secuencia de clasificación escogida.

Nota: Los caracteres de control de desplazamiento en los datos DBCS y DBCS/SBCS están incluidos dentro de la comparación.

SORT/MERGE

No se puede realizar una clasificación alfabética DBCS utilizando COBOL. Sin embargo, puede utilizar ítems de datos DBCS o DBCS/SBCS como claves en una instrucción SORT o MERGE. La operación de clasificación ordena los datos de acuerdo con la secuencia de clasificación especificada en el párrafo SORT, MERGE o SPECIAL NAMES. El sistema ordena cualquier carácter de control de desplazamiento que se encuentre dentro de las claves DBCS y DBCS/SBCS.

Utilice la instrucción RELEASE para transferir registros que contengan caracteres DBCS desde un área de entrada/salida hasta la fase inicial de una operación de clasificación. El sistema realiza la frase FROM con la instrucción RELEASE de la misma manera que realiza la frase FROM con la instrucción WRITE. (Consulte el apartado “Instrucción WRITE” en la página 364).

También es posible utilizar la instrucción RETURN para transferir registros que contengan caracteres DBCS desde la fase final de una operación de fusión o clasificación hasta un área de entrada/salida. El sistema realiza la frase INTO con la instrucción RETURN de la misma manera que realiza la frase INTO con la instrucción READ. (Consulte el apartado “Instrucción READ” en la página 363).

Instrucciones Dirigidas al Compilador

Instrucciones COPY

Puede utilizar la instrucción COPY para copiar el texto fuente que contiene caracteres DBCS en un programa COBOL. Cuando lo haga, asegúrese de que especifica el nombre de miembro, nombre de archivo y el nombre de la biblioteca utilizando datos alfanuméricos y de que los especifica de acuerdo con las normas establecidas en la *COBOL/400 Reference*

Utilice la instrucción COPY de Formato 2 para copiar campos definidos en las especificaciones de descripción de datos (DDS). Los ítems de datos DBCS y DBCS/SBCS (el valor en la columna 35 del formato DDS es O) se copian en un programa COBOL en el formato PICTURE X(n). El listado del compilador no indica que estos campos contengan caracteres DBCS, a no ser que un campo sea un campo clave. Si es así, el sistema imprime una O en la tabla de comentarios para las claves.

Los ítems de datos DBCS gráficos se copian a un programa COBOL en formato PICTURE X(N). El listado del compilador indica que estos campos contienen datos gráficos. Consulte el apartado “Campos Gráficos DBCS” en la página 139 si desea consultar la descripción de los tipos de datos DBCS gráficos.

Puede colocar los caracteres DBCS en los comentarios del texto que se copian de las DDS si el campo de las DDS asociado tiene comentarios.

Si especifica la frase REPLACING de la instrucción COPY, tenga en cuenta que:

- El pseudotexto puede contener cualquier combinación de caracteres DBCS y alfanuméricos.
- Puede utilizar literales con contenido DBCS o DBCS/SBCS.
- Los identificadores pueden referirse a un ítem de datos que contiene caracteres DBCS.

Instrucción TITLE

Puede utilizar literales DBCS/SBCS como el literal de la instrucción TITLE.

Comunicaciones entre programas

Puede especificar entradas para los ítems de datos DBCS o DBCS/SBCS en la Sección de Enlace de la División de Datos.

Se pueden pasar caracteres DBCS desde un programa a otro especificando los ítems de datos de la frase USING. No puede utilizar caracteres DBCS en la instrucción CALL para el nombre de programa del programa llamado.

No puede utilizar caracteres DBCS en la instrucción CANCEL porque especifican nombres de programas.

Distintivo FIPS

Las mejoras para el lenguaje COBOL que permiten utilizar caracteres DBCS se distinguen (identifican) mediante el distintivo FIPS (Federal Information Processing Standard) proporcionado por el compilador como ampliaciones de IBM.

Listados de Programa COBOL

Los caracteres DBCS pueden aparecer en listados que se originan desde archivos fuente que pueden contener DBCS y que se producen mediante sistemas que también admiten este tipos de caracteres.

Los caracteres DBCS que aparecen en un listado de programa tienen como origen un archivo fuente, un texto fuente generado por una instrucción COPY o un mensaje del compilador COBOL.

Un listado que contenga caracteres DBCS debe disponer de salida a un archivo de impresora que pueda procesar datos DBCS. Los listados que contienen caracteres DBCS se manejan correctamente si una de las siguientes condiciones es verdadera:

- El archivo de impresora que especifica el parámetro PRTFILE del mandato CRTCBLPGM se define con los atributos necesarios utilizando el mandato CRTPRTF o CHGPRTF.
- El archivo fuente puede contener datos DBCS utilizando el parámetro IGCDTA del mandato CRTSRCPF. En este caso, el programa altera temporalmente el valor existente del atributo para el archivo de impresora de salida.
- El usuario ha especificado el atributo necesario para la impresora de salida utilizando el parámetro IGCDTA del mandato OVRPRT antes de compilar el programa.

Nota: El parámetro IGCDTA sólo está disponible en sistemas DBCS, y no puede definirse o visualizarse en sistemas que no sean DBCS. El usuario puede, no obstante, crear objetos con atributos DBCS en un sistema que no sea DBCS copiándolos desde el sistema DBCS. Compruebe si se producen posibles incompatibilidades si realiza esto.

El compilador puede utilizar caracteres del programa fuente como parámetros de sustitución en mensajes del comprobador de sintaxis y del compilador. El sistema no comprueba ni edita los parámetros de sustitución. Si no se especifican correctamente los caracteres DBCS, el sistema puede imprimir o visualizar defectuosamente partes de los mensajes.

Fin de Ampliación de IBM

Apéndice G. Ejemplos de Procesos de Archivos AS/400

Este apéndice contiene programas ejemplo que ilustran las técnicas básicas de programación asociadas con cada tipo de organización de archivos AS/400. Estos ejemplos se han concebido para que sirvan únicamente como pautas de planificación, así como para ilustrar las sentencias de entrada/salida que se necesitan para determinados métodos de acceso. Se utilizan de forma casual otras características COBOL (como por ejemplo la instrucción PERFORM). Los programas que se indican son:

- Creación de archivos secuenciales
- Actualización y ampliación de archivos secuenciales
- Creación de archivos indexados
- Actualización de archivos indexados
- Creación de archivos relativos
- Actualización de archivos relativos
- Recuperación de archivos relativos.

Creación de Archivos Secuenciales

Este programa crea un archivo secuencial con los registros de salarios de empleados. Los registros de entrada se ordenan en orden ascendente por número de empleado. El archivo de salida tiene el mismo orden. (Un **archivo de salida** es un archivo que se ha abierto en modalidad de salida o en modalidad ampliada).

```
5763CB1 V3R0M5                  Fuente COBOL AS/400
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
    1  000010 IDENTIFICATION DIVISION.
    2  000020 PROGRAM-ID.  CRTSEQ.
       000030
    3  000040 ENVIRONMENT DIVISION.
    4  000050 CONFIGURATION SECTION.
    5  000060 SOURCE-COMPUTER.   IBM-AS400.                                          05/24/94
    6  000070 OBJECT-COMPUTER.   IBM-AS400.                                          05/24/94
    7  000080 SPECIAL-NAMES.  CONSOLE IS TYPEWRITER.
    8  000090 INPUT-OUTPUT SECTION.
    9  000100 FILE-CONTROL.
   10  000110     SELECT INPUT-FILE ASSIGN TO DISK-FILEA
   11  000120         FILE STATUS IS INPUT-FILE-STATUS.
   12  000130     SELECT OUTPUT-FILE ASSIGN TO DISK-FILEB
   13  000140         FILE STATUS IS OUTPUT-FILE-STATUS.
   14  000150 DATA DIVISION.
   15  000160 FILE SECTION.
   16  000170 FD  INPUT-FILE LABEL RECORDS STANDARD.
   17  000180 01  INPUT-RECORD.
   18  000190     05  INPUT-EMPLOYEE-NUMBER      PICTURE 9(6).
   19  000200     05  INPUT-EMPLOYEE-NAME        PICTURE X(28).
   20  000210     05  INPUT-EMPLOYEE-CODE        PICTURE 9.
   21  000220     05  INPUT-EMPLOYEE-SALARY      PICTURE 9(6)V99.
   22  000230 FD  OUTPUT-FILE LABEL RECORDS STANDARD.
   23  000240 01  OUTPUT-RECORD.
   24  000250     05  OUTPUT-EMPLOYEE-NUMBER     PICTURE 9(6).
   25  000260     05  OUTPUT-EMPLOYEE-NAME       PICTURE X(28).
   26  000270     05  OUTPUT-EMPLOYEE-CODE       PICTURE 9.
   27  000280     05  OUTPUT-EMPLOYEE-SALARY     PICTURE 9(6)V99.
   28  000290 WORKING-STORAGE SECTION.
   29  000300 77  INPUT-FILE-STATUS              PICTURE XX.
   30  000310 77  OUTPUT-FILE-STATUS             PICTURE XX.
   31  000320 01  INPUTEND                       PICTURE X VALUE SPACE.
   32  000330     88  THE-END-OF-INPUT           VALUE "E".
   33  000340 01  DISP-RECORD.
   34  000350     05  OP-NAME                    PICTURE X(7).
   35  000360     05  FILLER                     PICTURE XX VALUE SPACE.
   36  000370     05  FILE-NAME                  PICTURE X(11).
   37  000380     05  FILLER                     PICTURE XX VALUE SPACE.
   38  000390     05  FILLER                     PICTURE X(14)
   39  000400                                    VALUE "FILE STATUS IS".
   40  000410     05  FILLER                     PICTURE XX VALUE SPACE.
   41  000420     05  SK                         PICTURE XX.
   42  000430 PROCEDURE DIVISION.
       000440 DECLARATIVES.
       000450 I-O-ERROR SECTION.
       000460           USE AFTER STANDARD ERROR PROCEDURE ON INPUT-FILE,
       000470                                              OUTPUT-FILE.
       000480 I-O-ERROR-PARA.
       000490*************************************************************
       000500* PARA ASEGURAR CONTROL SE DEVUELVEN DECLARATIVAS FICTICIAS*
       000510* A ESTE PROGRAMA CUANDO SE PRODUCE ERROR DURANTE PROCESO  *
       000520* ARCHIVO. MANEJO ERRORES DESPUÉS DE CADA INSTRUCCIÓN E/S. *
       000530*************************************************************
       000540 END DECLARATIVES.
       000550 MAIN-PROGRAM SECTION.
       000560 OPEN-FILES.
   43  000570     OPEN INPUT INPUT-FILE
       000580         OUTPUT OUTPUT-FILE.
   44  000590     IF INPUT-FILE-STATUS NOT = "00"
   45  000600         MOVE "OPEN" TO OP-NAME
   46  000610         MOVE "INPUT-FILE" TO FILE-NAME
   47  000620         MOVE INPUT-FILE-STATUS TO SK
   48  000630         PERFORM ERROR-OUT-1 THROUGH ERROR-OUT-2.
   49  000640     IF OUTPUT-FILE-STATUS NOT = "00"
   50  000650        MOVE "OPEN" TO OP-NAME
   51  000660        MOVE "OUTPUT-FILE" TO FILE-NAME
   52  000670        MOVE OUTPUT-FILE-STATUS TO SK
   53  000680        PERFORM ERROR-OUT-1 THROUGH ERROR-OUT-2.
   54  000690     PERFORM BUILD-FILE UNTIL THE-END-OF-INPUT.
       000700 CLOSE-FILES.
   55  000710        CLOSE INPUT-FILE
       000720              OUTPUT-FILE.
   56  000730        STOP RUN.
       000740 BUILD-FILE.
```

Figura 112 (Parte 1 de 2). Ejemplo de un Archivo Secuencial de Registros del Salario de un Empleado

```
5763CB1 V3R0M5                  Fuente COBOL AS/400
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
   57  000750     READ INPUT-FILE INTO OUTPUT-RECORD
   58  000760         AT END SET THE-END-OF-INPUT TO TRUE.
   59  000770     IF INPUT-FILE-STATUS NOT = "00"
   60  000780           MOVE "WRITE" TO OP-NAME
   61  000790           MOVE "OUTPUT-FILE" TO FILE-NAME
   62  000800           MOVE OUTPUT-FILE-STATUS TO SK
   63  000810           PERFORM ERROR-OUT-1 THROUGH ERROR-OUT-2
   64  000820           GO TO CLOSE-FILES.
   65  000830       WRITE OUTPUT-RECORD.
   66  000840       IF OUTPUT-FILE-STATUS NOT = "00"
   67  000850           MOVE "WRITE" TO OP-NAME
   68  000860           MOVE "OUTPUT-FILE" TO FILE-NAME
   69  000870           MOVE OUTPUT-FILE-STATUS TO SK
   70  000880           PERFORM ERROR-OUT-1 THROUGH ERROR-OUT-2
   71  000890           GO TO CLOSE-FILES.
       000900 ERROR-OUT-1.
   72  000910           DISPLAY "FILE PROCESSING ERROR" UPON TYPEWRITER.
   73  000920           DISPLAY DISP-RECORD UPON TYPEWRITER.
   74  000930           CLOSE INPUT-FILE
       000940                 OUTPUT-FILE.
   75  000950           STOP RUN.
       000960 ERROR-OUT-2.
       000970     EXIT.
                        * * * * *  F I N  D E  F U E N T E  * * * * *
 5763CB1 V3R0M5                  Mensajes COBOL AS/400
  INST
*   16  IDMEN: LBL0650  GRAVEDAD: 00  NUMSEC:  000170
       Mensaje . . . . :   Bloqueo/Desbloqueo para archivo 'INPUT-FILE'
         se realizará por medio del código generado por compilador.
*   22  IDMEN: LBL0650  GRAVEDAD: 00  NUMSEC:  000230
       Mensaje . . . . :   Bloqueo/Desbloqueo para archivo 'OUTPUT-FILE'
         se realizará por medio del código generado por compilador.
*   43  IDMEN: LBL0335  GRAVEDAD: 00  NUMSEC:  000540
       Mensaje . . . . :   Párrafo vacío o sección precede el párrafo
         o sección 'END DECLARATIVES'

                      * * * * *  F I N  D E  M E N S A J E S  * * * * *
                                      Resumen de mensajes
  Total    Info(0-4)    Aviso(5-19)    Error(20-29)    Grave(30-39)    Terminal(40-99)
      3          3             0               0               0                0
 Registros fuente leídos. . . . . :   97
 Registros copia leídos . . . . . :   0
 Miembros copia procesados  . . . . . :   0
 Errores secuencia  . . . . . :   0
 Mensaje gravedad más alta enviado  . :   0
  LBL0901 00  Programa CRTSEQ creado en biblioteca XMPLIB.

                     * * * * *  F I N  D E  C O M P I L A C I Ó N * * * * *
```

Figura 112 (Parte 2 de 2). Ejemplo de un Archivo Secuencial de Registros del Salario de un Empleado

Actualización y Ampliación de Archivos Secuenciales

Este programa actualiza y amplía el archivo que el programa `CRTSEQ` ha creado. Se leen el `INPUT-FILE` y el `MASTER-FILE`. Cuando se encuentra una coincidencia entre el `INPUT-EMPLOYEE-NUMBER` y el `MST-EMPLOYEE-NUMBER`, el registro de entrada sustituye al registro original. Cuando se procesa `MASTER-FILE`, se añaden nuevos registros de empleados al final del archivo.

```
5763CB1 V3R0M5                  Fuente COBOL AS/400
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
    1  000010 IDENTIFICATION DIVISION.
    2  000020 PROGRAM-ID.  UPDTSEQ.
    3  000030 ENVIRONMENT DIVISION.
    4  000040 CONFIGURATION SECTION.
    5  000050 SOURCE-COMPUTER.  IBM-AS400.                                          05/24/94
    6  000060 OBJECT-COMPUTER.  IBM-AS400.                                          05/24/94
    7  000070 INPUT-OUTPUT SECTION.
    8  000080 FILE-CONTROL.
    9  000090     SELECT INPUT-FILE ASSIGN TO DISK-FILES
   10  000100         FILE STATUS IS INPUT-FILE-STATUS.  A
   11  000110     SELECT MASTER-FILE ASSIGN TO DISK-MSTFILEB
   12  000120         FILE STATUS IS MASTER-FILE-STATUS.  B
       000130
   13  000140 DATA DIVISION.
   14  000150 FILE SECTION.
   15  000160 FD  INPUT-FILE LABEL RECORDS STANDARD.
   16  000170 01  INPUT-RECORD.
   17  000180     05  INPUT-EMPLOYEE-NUMBER      PICTURE 9(6).
   18  000190     05  INPUT-EMPLOYEE-NAME        PICTURE X(28).
   19  000200     05  INPUT-EMPLOYEE-CODE        PICTURE 9.
   20  000210     05  INPUT-EMPLOYEE-SALARY      PICTURE 9(6)V99.
   21  000220 FD  MASTER-FILE LABEL RECORDS STANDARD.
   22  000230 01  MASTER-RECORD.
   23  000240     05 MST-EMPLOYEE-NUMBER         PICTURE 9(6).
   24  000250     05 MST-EMPLOYEE-NAME           PICTURE X(28).
   25  000260     05 MST-EMPLOYEE-CODE           PICTURE 9.
   26  000270     05 MST-EMPLOYEE-SALARY         PICTURE 9(6)V99.
   27  000280 WORKING-STORAGE SECTION.
   28  000290 77  INPUT-FILE-STATUS             PICTURE XX.
   29  000300 77  MASTER-FILE-STATUS            PICTURE XX.
   30  000310 01  INPUTEND                      PICTURE X VALUE SPACE.
   31  000320     88 THE-END-OF-INPUT           VALUE "E".
   32  000330 01  MASTEREND                     PICTURE X VALUE SPACE.
   33  000340     88 THE-END-OF-MASTER          VALUE "E".
   34  000350 01  ERROR-INFO.
   35  000360     05 OP-NAME                     PICTURE X(12).
   36  000370     05 FILLER                      PICTURE XX VALUE SPACE.
   37  000380     05 FILE-NAME                   PICTURE X(11).
   38  000390     05 FILLER                      PICTURE XX VALUE SPACE.
   39  000400     05 FILLER                      PICTURE X(14)
   40  000410                                   VALUE "FILE STATUS IS".
   41  000420     05 FILLER                      PICTURE XX VALUE SPACE.
   42  000430     05 SK                          PICTURE XX.
   43  000440 PROCEDURE DIVISION.
       000450 DECLARATIVES.
       000460 INPUT-FILE-ERROR SECTION.
       000470      USE AFTER STANDARD ERROR PROCEDURE ON INPUT-FILE.  C
       000480 INPUT-FILE-ERROR-PARA.
   44  000490        MOVE INPUT-FILE-STATUS TO SK.
   45  000500        MOVE "INPUT-FILE" TO FILE-NAME.
   46  000510        DISPLAY "FILE PROCESSING ERROR".
   47  000520        DISPLAY ERROR-INFO.
   48  000530        DISPLAY "PROCESSING TERMINATED DUE TO I-O ERROR".
   49  000540        STOP RUN.
       000550 I-O-FILE-ERROR SECTION.
       000560     USE AFTER STANDARD ERROR PROCEDURE ON MASTER-FILE.  D
       000570 MASTER-FILE-ERROR-PARA.
   50  000580        MOVE MASTER-FILE-STATUS TO SK.
   51  000590        MOVE "MASTER-FILE" TO FILE-NAME.
   52  000600        DISPLAY "FILE PROCESSING ERROR".
   53  000610        DISPLAY ERROR-INFO.
   54  000620        DISPLAY "PROCESSING TERMINATED DUE TO I-O ERROR".
   55  000630        STOP RUN.
       000640 END DECLARATIVES.
       000650 MAIN-PROGRAM SECTION.
       000660 OPEN-FILES.
   56  000670        MOVE "OPEN" TO OP-NAME.
   57  000680        OPEN INPUT INPUT-FILE
       000690             I-O   MASTER-FILE.
       000700 PROCESSING-LOGIC.
   58  000710       PERFORM READ-INPUT-FILE.
   59  000720       PERFORM READ-MASTER-FILE.
   60  000730       PERFORM PROCESS-FILES UNTIL THE-END-OF-INPUT.
```

Figura 113 (Parte 1 de 2). Ejemplo de un Programa de Actualización de Archivo Secuencial

```
5763CB1 V3R0M5                   Fuente COBOL AS/400
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
      000740 CLOSE-FILES.
   61 000750       MOVE "CLOSE" TO OP-NAME.
   62 000760       CLOSE MASTER-FILE
      000770             INPUT-FILE.
   63 000780       STOP RUN.
      000790 READ-INPUT-FILE.
   64 000800       MOVE "READ" TO OP-NAME.
   65 000810       READ INPUT-FILE
   66 000820           AT END SET THE-END-OF-INPUT TO TRUE.
      000830 READ-MASTER-FILE.
   67 000840       MOVE "READ" TO OP-NAME.
   68 000850       READ MASTER-FILE
      000860           AT END
   69 000870              SET THE-END-OF-MASTER TO TRUE
   70 000880              MOVE "AT END CLOSE" TO OP-NAME
   71 000890              CLOSE MASTER-FILE
   72 000900              MOVE "OPEN EXTEND" TO OP-NAME
   73 000910              OPEN EXTEND MASTER-FILE.
      000920 PROCESS-FILES.
   74 000930     IF THE-END-OF-MASTER
   75 000940         WRITE MASTER-RECORD FROM INPUT-RECORD
   76 000950         PERFORM READ-INPUT-FILE
      000960     ELSE
   77 000970         IF MST-EMPLOYEE-NUMBER LESS THAN INPUT-EMPLOYEE-NUMBER
   78 000980            PERFORM READ-MASTER-FILE
      000990        ELSE
   79 001000           IF MST-EMPLOYEE-NUMBER = INPUT-EMPLOYEE-NUMBER
   80 001010              MOVE "REWRITE" TO OP-NAME
   81 001020              REWRITE MASTER-RECORD FROM INPUT-RECORD
   82 001030              PERFORM READ-INPUT-FILE
   83 001040              PERFORM READ-MASTER-FILE
      001050           ELSE
   84 001060               DISPLAY "ERROR RECORD -> ", INPUT-EMPLOYEE-NUMBER
   85 001070               PERFORM READ-INPUT-FILE.
                          * * * * *  F I N  D E  F U E N T E  * * * * *
5763CB1 V3R0M5                   Mensajes COBOL AS/400
 INST
*   15  IDMEN: LBL0650  GRAVEDAD: 00  NUMSEC:  000160
       Mensaje . . . . :   Bloqueo/Desbloqueo para archivo 'INPUT-FILE'
         se realizará por medio del código generado por compilador.

                     * * * * *  F I N  D E  M E N S A J E S  * * * * *
                                    Resumen de mensajes
  Total    Info(0-4)    Aviso(5-19)    Error(20-29)    Grave(30-39)    Terminal(40-99)
      1          1            0               0               0                0
 Registros fuente leídos. . . . . . . :   107
 Registros copia leídos . . . . . . . :   0
 Miembros copia procesados  . . . . . :   0
 Errores secuencia  . . . . . . . . . :   0
 Mensaje gravedad más alta enviado  . :   0
  LBL0901 00  Programa UPDTSEQ creado en biblioteca XMPLIB.

                    * * * * *  F I N  D E  C O M P I L A C I Ó N * * * * *
```

Figura 113 (Parte 2 de 2). Ejemplo de un Programa de Actualización de Archivo Secuencial

El ejemplo de la Figura 113 en la página 374 incluye:

A Una cláusula FILE STATUS para que el programa registre el estado de las solicitudes de E/S más recientes relacionadas con `INPUT-FILE`.

B Una cláusula FILE STATUS para que el programa registra el estado de las solicitudes de E/S más recientes relacionadas con `MASTER-FILE`.

C Un procedimiento USE que se ejecuta cuando se produce un error de E/S durante el proceso del `INPUT-FILE`.

D Un procedimiento USE que se ejecuta cuando se produce un error de E/S durante el proceso de `MASTER-FILE`.

Los valores de estado del archivo y los procedimientos USE juegan un papel muy importante en el *manejo de errores*. Para obtener más información, consulte el Capítulo 6, “Manejo de Errores y Excepciones COBOL/400”.

Creación de Archivos Indexados

Un **archivo indexado** es un archivo que registra la clave y la posición de cada registro en una parte separada del archivo llamada índice.

Este programa crea un archivo indexado de registros de resumen para depósitos de banco. La clave dentro de cada registro del archivo indexado es INDEX-KEY (número de cuenta de depósito); los registros de entrada se ordenan en secuencia ascendente según esta clave. Los registros se leen del archivo de entrada y se transfieren al área de registros del archivo indexado. Entonces se graba el registro del archivo indexado.

```
5763CB1 V3R0M5                  Fuente COBOL AS/400
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
    1  000010 IDENTIFICATION DIVISION.
    2  000020 PROGRAM-ID.  CRTIND.
       000030
    3  000040 ENVIRONMENT DIVISION.
    4  000050 CONFIGURATION SECTION.
    5  000060 SOURCE-COMPUTER. IBM-AS400.                                                    05/24/94
    6  000070 OBJECT-COMPUTER. IBM-AS400.                                                    05/24/94
    7  000080 INPUT-OUTPUT SECTION.
    8  000090 FILE-CONTROL.
    9  000100     SELECT INDEXED-FILE ASSIGN TO DISK-INDEXFILE
   10  000110         ORGANIZATION IS INDEXED
   11  000120         ACCESS IS SEQUENTIAL
   12  000130         RECORD KEY IS INDEX-KEY
   13  000140         FILE STATUS IS INDEXED-FILE-STATUS.
   14  000150     SELECT INPUT-FILE ASSIGN TO DISK-FILEG
   15  000160         FILE STATUS IS INPUT-FILE-STATUS.
   16  000170 DATA DIVISION.
   17  000180 FILE SECTION.
   18  000190 FD  INDEXED-FILE LABEL RECORDS STANDARD.
   19  000200 01  INDEX-RECORD.
   20  000210     05  INDEX-KEY                    PICTURE X(10).
   21  000220     05  INDEX-FLD1                   PICTURE X(10).
   22  000230     05  INDEX-NAME                   PICTURE X(20).
   23  000240     05  INDEX-BAL                    PICTURE S9(5)V99.
   24  000250 FD  INPUT-FILE LABEL RECORDS STANDARD.
   25  000260 01  INPUT-RECORD.
   26  000270     05  INPUT-KEY                    PICTURE X(10).
   27  000280     05  INPUT-NAME                   PICTURE X(20).
   28  000290     05  INPUT-BAL                    PICTURE S9(5)V99.
   29  000300 WORKING-STORAGE SECTION.
   30  000310 77  INDEXED-FILE-STATUS              PICTURE XX.
   31  000320 77  INPUT-FILE-STATUS                PICTURE XX.
   32  000330 77  OP-NAME                          PICTURE X(7).
   33  000340 01  INPUTEND                         PICTURE X VALUE SPACES.
   34  000350     88  THE-END-OF-INPUT             VALUE "E".
   35  000360 01  ERRORFLAG                        PICTURE X VALUE SPACES.
   36  000370     88  ERROR-OCCURRED               VALUE "1".
   37  000380 PROCEDURE DIVISION.
       000390 DECLARATIVES.
       000400 INPUT-ERROR SECTION.
       000410       USE AFTER STANDARD ERROR PROCEDURE ON INPUT.
       000420 INPUT-ERROR-PARA.
   38  000430     DISPLAY "UNEXPECTED ERROR ON ", OP-NAME, " FOR INPUT-FILE ".
   39  000440     DISPLAY "FILE STATUS IS ", INPUT-FILE-STATUS.
   40  000450     SET ERROR-OCCURRED TO TRUE.
       000460 OUTPUT-ERROR SECTION.
       000470       USE AFTER STANDARD ERROR PROCEDURE ON OUTPUT.
       000480 OUTPUT-ERROR-PARA.
   41  000490     DISPLAY "UNEXPECTED ERROR ON ", OP-NAME, " FOR INDEXED-FILE ".
   42  000500     DISPLAY "FILE STATUS IS ", INDEXED-FILE-STATUS.
   43  000510     SET ERROR-OCCURRED TO TRUE.
       000520 END DECLARATIVES.
       000530 MAIN-PROCESSING SECTION.
       000540 MAIN-PROCEDURE.
   44  000550     MOVE "OPEN" TO OP-NAME.
   45  000560     OPEN INPUT INPUT-FILE
       000570         OUTPUT INDEXED-FILE.
   46  000580     IF ERROR-OCCURRED GO TO ERROR-TERMINATION.
   48  000590     PERFORM READ-INPUT-FILE.
   49  000600     PERFORM LOAD-INDEXED-FILE THRU READ-INPUT-FILE
       000610                               UNTIL THE-END-OF-INPUT.
```

Figura 114 (Parte 1 de 2). Ejemplo de un Programa de Archivo Indexado

```
5763CB1 V3R0M5                  Fuente COBOL AS/400
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
   50  000620     MOVE "CLOSE" TO OP-NAME.
   51  000630     CLOSE INPUT-FILE
       000640           INDEXED-FILE.
   52  000650     IF ERROR-OCCURRED GO TO ERROR-TERMINATION.
   54  000660     STOP RUN.
       000670 LOAD-INDEXED-FILE.
   55  000680     MOVE INPUT-KEY TO INDEX-KEY.
   56  000690     MOVE INPUT-NAME TO INDEX-NAME.
   57  000700     MOVE INPUT-BAL TO INDEX-BAL.
   58  000710     MOVE SPACES TO INDEX-FLD1.
   59  000720     MOVE "WRITE" TO OP-NAME.
   60  000730     WRITE INDEX-RECORD
       000740           INVALID KEY
   61  000750               DISPLAY "WRITE FAILED FOR KEY ", INDEX-KEY.
   62  000760     IF ERROR-OCCURRED GO TO ERROR-TERMINATION.
       000770 READ-INPUT-FILE.
   64  000780     MOVE "READ" TO OP-NAME.
   65  000790     READ INPUT-FILE
   66  000800          AT END SET THE-END-OF-INPUT TO TRUE.
   67  000810     IF ERROR-OCCURRED GO TO ERROR-TERMINATION.
       000820 ERROR-TERMINATION.
   69  000830     DISPLAY "I-O ERROR OCCURRED - PROCESS TERMINATING".
   70  000840     STOP RUN.
                        * * * * *  F I N  D E  F U E N T E  * * * * *
5763CB1 V3R0M5                  Mensajes COBOL AS/400
 INST
*   18  IDMEN: LBL0650  GRAVEDAD: 00  NUMSEC:  000190
       Mensaje . . . . :   Bloqueo/Desbloqueo para archivo 'INDEXED-FILE'
         se realizará por medio del código generado por compilador.
*   24  IDMEN: LBL0650  GRAVEDAD: 00  NUMSEC:  000250
       Mensaje . . . . :   Bloqueo/Desbloqueo para archivo 'INPUT-FILE'
         se realizará por medio del código generado por compilador.

                     * * * * *  F I N  D E  M E N S A J E S  * * * * *
                                    Resumen de mensajes
  Total    Info(0-4)    Aviso(5-19)    Error(20-29)    Grave(30-39)    Terminal(40-99)
      2          2            0              0               0               0
 Registros fuente leídos. . . . . :   84
 Registros copia leídos . . . . . :   0
 Miembros copia procesados  . . . :   0
 Errores secuencia  . . . . . :   0
 Mensaje gravedad más alta enviado  . :   0
  LBL0901 00  Programa CRTIND creado en biblioteca XMPLIB.

                    * * * * *  F I N  D E  C O M P I L A C I Ó N * * * * *
```

Figura 114 (Parte 2 de 2). Ejemplo de un Programa de Archivo Indexado

Actualización de Archivos Indexados

Este programa actualiza, mediante el acceso dinámico, el archivo indexado creado en el programa `CRTIND`.

Los registros de entrada tienen la clave para el registro, el número de depósito y la cantidad de la transacción.

Cuando el programa lee el registro de entrada, comprueba:

- Si es un registro de transacciones (en cuyo caso se llenan todos los campos del registro)
- Si es un registro que solicita recuperación secuencial de una clase genérica específica (en cuyo caso sólo el campo `INPUT-GEN-FLD` del registro de entrada contendrá datos).

El acceso al azar se utiliza para actualizar e imprimir los registros de las transacciones. El acceso secuencial se utiliza para la recuperación e impresión de todos los registros dentro de una clase genérica.

```
5763CB1 V3R0M5                  Fuente COBOL AS/400
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
    1  000010 IDENTIFICATION DIVISION.
    2  000020 PROGRAM-ID.  UPDTIND.
       000030
    3  000040 ENVIRONMENT DIVISION.
    4  000050 CONFIGURATION SECTION.
    5  000060 SOURCE-COMPUTER. IBM-AS400.                                                    05/24/94
    6  000070 OBJECT-COMPUTER. IBM-AS400.                                                    05/24/94
    7  000080 INPUT-OUTPUT SECTION.
    8  000090 FILE-CONTROL.
    9  000100     SELECT MASTER-FILE ASSIGN TO DISK-INDXFILE
   10  000110         ORGANIZATION IS INDEXED
   11  000120         ACCESS IS DYNAMIC
   12  000130         RECORD KEY IS MASTER-KEY
   13  000140         FILE STATUS IS MASTER-FILE-STATUS.
   14  000150     SELECT INPUT-FILE ASSIGN TO DISK-FILEH
   15  000160         FILE STATUS IS INPUT-FILE-STATUS.
   16  000170     SELECT PRINT-FILE ASSIGN TO PRINTER-QSYSPRT
   17  000180         FILE STATUS IS PRINT-FILE-STATUS.
   18  000190 DATA DIVISION.
   19  000200 FILE SECTION.
   20  000210 FD  MASTER-FILE LABEL RECORDS STANDARD.
   21  000220 01  MASTER-RECORD.
   22  000230     05  MASTER-KEY.
   23  000240       10    MASTER-GEN-FLD   PICTURE X(5).
   24  000250       10    MASTER-DET-FLD   PICTURE X(5).
   25  000260     05  MASTER-FLD1          PICTURE X(10).
   26  000270     05  MASTER-NAME          PICTURE X(20).
   27  000280     05  MASTER-BAL           PICTURE S9(5)V99.
   28  000290 FD  INPUT-FILE LABEL RECORDS STANDARD.
   29  000300 01  INPUT-REC.
   30  000310     05  INPUT-KEY.
   31  000320       10  INPUT-GEN-FLD      PICTURE X(5).
   32  000330       10  INPUT-DET-FLD      PICTURE X(5).
   33  000340     05  INPUT-NAME           PICTURE X(20).
   34  000350     05  INPUT-AMT            PICTURE S9(5)V99.
   35  000360 FD  PRINT-FILE LABEL RECORDS OMITTED
   36  000370     LINAGE 12 LINES FOOTING AT 9.
   37  000380 01  PRINT-RECORD-1.
   38  000390     05  PRINT-KEY            PICTURE X(10).
   39  000400     05  FILLER               PICTURE X(5).
   40  000410     05  PRINT-NAME           PICTURE X(20).
   41  000420     05  FILLER               PICTURE X(5).
   42  000430     05  PRINT-BAL            PICTURE $$$,$$9.99-.
   43  000440     05  FILLER               PICTURE X(7).
   44  000450     05  PRINT-AMT            PICTURE $$$,$$9.99-.
   45  000460     05  FILLER               PICTURE X(5).
   46  000470     05  PRINT-NEW-BAL        PICTURE $$$,$$9.99-.
   47  000480 01  PRINT-RECORD-2           PICTURE X(89).
   48  000490 WORKING-STORAGE SECTION.
   49  000500 77  MASTER-FILE-STATUS       PICTURE XX.
   50  000510 77  INPUT-FILE-STATUS        PICTURE XX.
   51  000520 77  PRINT-FILE-STATUS        PICTURE XX.
   52  000530 77  LINES-TO-FOOT            PICTURE 99.
   53  000540 01  PAGE-HEAD.
   54  000550     05  FILLER               PICTURE X(38) VALUE SPACES.
   55  000560     05  FILLER               PICTURE X(13) VALUE "UPDATE REPORT".
   56  000570     05  FILLER               PICTURE X(38) VALUE SPACES.
   57  000580 01  COLUMN-HEAD.
   58  000590     05  FILLER               PICTURE X(6)  VALUE "KEY ID".
   59  000600     05  FILLER               PICTURE X(9)  VALUE SPACES.
   60  000610     05  FILLER               PICTURE X(4)  VALUE "NAME".
   61  000620     05  FILLER               PICTURE X(21) VALUE SPACES.
   62  000630     05  FILLER               PICTURE X(11) VALUE "CUR BALANCE".
   63  000640     05  FILLER               PICTURE X(6)  VALUE SPACES.
   64  000650     05  FILLER               PICTURE X(13) VALUE "UPDATE AMOUNT".
   65  000660     05  FILLER               PICTURE X(4)  VALUE SPACES.
   66  000670     05  FILLER               PICTURE X(11) VALUE "NEW BALANCE".
   67  000680     05  FILLER               PICTURE X(4)  VALUE SPACES.
   68  000690 01  PAGE-FOOT.
   69  000700     05  FILLER               PICTURE X(81)  VALUE SPACES.
   70  000710     05  FILLER               PICTURE A(6)   VALUE "PAGE  ".
   71  000720     05  PG-NUMBER            PICTURE 99     VALUE 00.
       000730
   72  000740 01  INPUTEND                 PICTURE X VALUE SPACE.
   73  000750     88  THE-END-OF-INPUT              VALUE "E".
```

Figura 115 (Parte 1 de 4). Ejemplo de un Programa de Actualización de Archivo Indexado

```
5763CB1 V3R0M5                  Fuente COBOL AS/400
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
   74  000760 01  ERRORFLAG               PICTURE X VALUE SPACE.
   75  000770     88  ERROR-OCCURRED                VALUE "1".
   76  000780 01  ERROR-DATA.
   77  000790     05  FILLER               PICTURE X(21)
   78  000800                                VALUE "STATEMENT FAILING IS ".
   79  000810     05  OP-NAME              PICTURE X(9).
   80  000820     05  FILLER               PICTURE X(16)
   81  000830                                VALUE "FILE STATUS IS".
   82  000840     05  STATUS-VALUE         PICTURE XX.
   83  000850 01  INPUT-MESSAGE.
   84  000860     05  FILLER               PICTURE X(30)
   85  000870         VALUE "UNEXPECTED ERROR ON INPUT-FILE" .
   86  000880 01  I-O-MESSAGE.
   87  000890     05  FILLER               PICTURE X(31)
   88  000900         VALUE "UNEXPECTED ERROR ON MASTER-FILE" .
   89  000910 01  OUTPUT-MESSAGE.
   90  000920     05  FILLER               PICTURE X(30)
   91  000930         VALUE "UNEXPECTED ERROR ON PRINT-FILE" .
   92  000940 PROCEDURE DIVISION.
       000950 DECLARATIVES.
       000960 INPUT-ERROR SECTION.
       000970       USE AFTER STANDARD ERROR PROCEDURE ON INPUT.
       000980 INPUT-ERROR-PARA.
   93  000990       DISPLAY INPUT-MESSAGE.
   94  001000       MOVE INPUT-FILE-STATUS TO STATUS-VALUE.
   95  001010       DISPLAY ERROR-DATA.
   96  001020       SET ERROR-OCCURRED TO TRUE.
       001030 I-O-ERROR SECTION.
       001040     USE AFTER STANDARD ERROR PROCEDURE ON I-O.
       001050 I-O-ERROR-PARA.
   97  001060       DISPLAY I-O-MESSAGE.
   98  001070       MOVE MASTER-FILE-STATUS TO STATUS-VALUE.
   99  001080       DISPLAY ERROR-DATA.
  100  001090       SET ERROR-OCCURRED TO TRUE.
       001100 OUTPUT-ERROR SECTION.
       001110       USE AFTER STANDARD ERROR PROCEDURE ON OUTPUT.
       001120 OUTPUT-ERROR-PARA.
  101  001130       DISPLAY OUTPUT-MESSAGE.
  102  001140       MOVE PRINT-FILE-STATUS TO STATUS-VALUE.
  103  001150       DISPLAY ERROR-DATA.
  104  001160     SET ERROR-OCCURRED TO TRUE.
       001170 END DECLARATIVES.
       001180 MAIN-PROCESSING SECTION.
       001190 MAIN-PROCEDURE.
  105  001200     MOVE "OPEN" TO OP-NAME.
  106  001210     OPEN INPUT INPUT-FILE
       001220         I-O MASTER-FILE
       001230         OUTPUT PRINT-FILE.
  107  001240     IF ERROR-OCCURRED GO TO ERROR-TERMINATION.
  109  001250     PERFORM PAGE-START.
  110  001260     PERFORM READ-INPUT-FILE.
  111  001270     PERFORM PROCESS-DATA THRU READ-INPUT-FILE
       001280                          UNTIL THE-END-OF-INPUT.
  112  001290     PERFORM PAGE-END.
  113  001300     MOVE "CLOSE" TO OP-NAME.
  114  001310     CLOSE INPUT-FILE
       001320           MASTER-FILE
       001330           PRINT-FILE.
  115  001340     IF ERROR-OCCURRED GO TO ERROR-TERMINATION.
  117  001350     STOP RUN.
       001360
       001370 PROCESS-DATA.
  118  001380     IF INPUT-DET-FLD EQUAL SPACES
  119  001390         PERFORM INIT-SEQUENTIAL-PROCESS
       001400     ELSE
  120  001410         PERFORM DYNAMIC-PROCESS.
       001420 READ-INPUT-FILE.
  121  001430     MOVE "READ" TO OP-NAME.
  122  001440     READ INPUT-FILE
  123  001450         AT END SET THE-END-OF-INPUT TO TRUE.
  124  001460     IF ERROR-OCCURRED GO TO ERROR-TERMINATION.
       001470
       001480 INIT-SEQUENTIAL-PROCESS.
```

Figura 115 (Parte 2 de 4). Ejemplo de un Programa de Actualización de Archivo Indexado

```
5763CB1 V3R0M5                   Fuente COBOL AS/400
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
  126  001490     MOVE INPUT-GEN-FLD TO MASTER-GEN-FLD.
  127  001500     MOVE "START" TO OP-NAME.
  128  001510     START MASTER-FILE
       001520          KEY IS NOT LESS THAN MASTER-GEN-FLD
       001530          INVALID KEY
  129  001540              DISPLAY "MASTER-FILE START FAILED: INVALID KEY ",
       001550                      MASTER-GEN-FLD
  130  001560              MOVE HIGH-VALUE TO MASTER-GEN-FLD.
  131  001570     IF ERROR-OCCURRED GO TO ERROR-TERMINATION.
  133  001580     PERFORM SEQUENTIAL-PROCESS
       001590         UNTIL INPUT-GEN-FLD NOT EQUAL MASTER-GEN-FLD.
       001600
       001610 SEQUENTIAL-PROCESS.
  134  001620     MOVE "READ NEXT" TO OP-NAME.
  135  001630     READ MASTER-FILE NEXT RECORD
  136  001640         AT END MOVE HIGH-VALUE TO MASTER-GEN-FLD.
  137  001650     IF ERROR-OCCURRED GO TO  ERROR-TERMINATION.
  139  001660     IF INPUT-GEN-FLD EQUAL MASTER-GEN-FLD
  140  001670         MOVE MASTER-KEY TO PRINT-KEY
  141  001680         MOVE MASTER-NAME TO PRINT-NAME
  142  001690         MOVE MASTER-BAL TO PRINT-NEW-BAL
  143  001700         PERFORM PRINT-DETAIL.
       001710
       001720 DYNAMIC-PROCESS.
  144  001730     MOVE INPUT-KEY TO MASTER-KEY.
  145  001740     MOVE "READ" TO OP-NAME.
  146  001750     READ MASTER-FILE
       001760         INVALID KEY
  147  001770           DISPLAY "MASTER-FILE READ FAILED: INVALID KEY ",
       001780                   MASTER-KEY
  148  001790           MOVE HIGH-VALUE TO MASTER-GEN-FLD.
  149  001800     IF ERROR-OCCURRED GO TO ERROR-TERMINATION.
  151  001810     IF INPUT-GEN-FLD EQUAL MASTER-GEN-FLD
  152  001820         MOVE MASTER-KEY TO PRINT-KEY
  153  001830         MOVE MASTER-NAME TO PRINT-NAME
  154  001840         MOVE MASTER-BAL TO PRINT-BAL
  155  001850         MOVE INPUT-AMT TO PRINT-AMT
  156  001860         ADD INPUT-AMT TO MASTER-BAL
  157  001870         MOVE MASTER-BAL TO PRINT-NEW-BAL
  158  001880         PERFORM PRINT-DETAIL
  159  001890         MOVE "REWRITE" TO OP-NAME
  160  001900         REWRITE MASTER-RECORD
       001910            INVALID KEY
  161  001920             DISPLAY "MASTER-FILE REWRITE FAILED: INVALID KEY ",
       001930                     MASTER-KEY
  162  001940             MOVE HIGH-VALUE TO MASTER-GEN-FLD.
  163  001950     IF ERROR-OCCURRED GO TO ERROR-TERMINATION.
       001960 PRINT-DETAIL.
  165  001970     MOVE "WRITE" TO OP-NAME.
  166  001980     WRITE PRINT-RECORD-1
       001990          AT END-OF-PAGE
  167  002000          PERFORM PAGE-END THROUGH PAGE-START.
  168  002010     IF ERROR-OCCURRED GO TO ERROR-TERMINATION.
  170  002020     MOVE SPACES TO PRINT-RECORD-1.
       002030
       002040 PAGE-END.
  171  002050     MOVE "WRITE" TO OP-NAME.
  172  002060     ADD 1 TO PG-NUMBER.
  173  002070     SUBTRACT LINAGE-COUNTER OF PRINT-FILE FROM 12
       002080          GIVING LINES-TO-FOOT.
  174  002090     MOVE SPACES TO PRINT-RECORD-1.
  175  002100     WRITE PRINT-RECORD-1
       002110          AFTER ADVANCING LINES-TO-FOOT.
  176  002120     WRITE PRINT-RECORD-2 FROM PAGE-FOOT
       002130          BEFORE ADVANCING PAGE.
  177  002140     IF ERROR-OCCURRED GO TO ERROR-TERMINATION.
```

Figura 115 (Parte 3 de 4). Ejemplo de un Programa de Actualización de Archivo Indexado

```
 5763CB1 V3R0M5                  Fuente COBOL AS/400
  INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
       002150 PAGE-START.
   179 002160     WRITE PRINT-RECORD-2 FROM PAGE-HEAD
       002170          AFTER ADVANCING 0 LINES.
   180 002180     IF ERROR-OCCURRED GO TO ERROR-TERMINATION.
   182 002190     MOVE SPACES TO PRINT-RECORD-2.
   183 002200     WRITE PRINT-RECORD-2 FROM COLUMN-HEAD
       002210          AFTER ADVANCING 1 LINE.
   184 002220     IF ERROR-OCCURRED GO TO ERROR-TERMINATION.
   186 002230     MOVE SPACES TO PRINT-RECORD-2.
       002240 ERROR-TERMINATION.
   187 002250     DISPLAY "PROCESS TERMINATING ABNORMALLY".
   188 002260     STOP RUN.
                        * * * * *  F I N  D E  F U E N T E  * * * * * 5763CB1 V3R0M5          Mensajes COBOL AS/400
  INST
*   28  IDMEN: LBL0650  GRAVEDAD: 00  NUMSEC:  000290
       Mensaje . . . . :   Bloqueo/Desbloqueo para archivo 'INPUT-FILE'
         se realizará por medio del código generado por compilador.

                     * * * * *  F I N  D E  M E N S A J E S  * * * * *
                                    Resumen de mensajes
  Total    Info(0-4)    Aviso(5-19)    Error(20-29)    Grave(30-39)    Terminal(40-99)
      1          1            0               0               0                0
 Registros fuente leídos. . . . . . . :   226
 Registros copia leídos . . . . . . . :   0
 Miembros copia procesados  . . . . . :   0
 Errores secuencia  . . . . . . . . . :   0
 Mensaje gravedad más alta enviado  . :   0
  LBL0901 00  Programa UPDTIND creado en biblioteca XMPLIB.

                    * * * * *  F I N  D E  C O M P I L A C I Ó N * * * * *
```

Figura 115 (Parte 4 de 4). Ejemplo de un Programa de Actualización de Archivo Indexado

Creación de Archivo Relativos

Este programa crea un archivo relativo de registros de resumen de ventas utilizando acceso secuencial. Cada registro contiene un resumen de cinco años de ventas en unidades y pesetas de una semana del año; en total hay 52 registros en el archivo, que representa cada uno una semana.

Cada registro de entrada contiene un resumen de las ventas de una semana. Los registros de la primera semana de los cinco últimos años (en orden ascendente) son los primeros cinco registros de entrada. Los siguientes cinco registros de entrada corresponden a los registros de la segunda semana de los últimos cinco años. Así pues, cinco registros de entrada llenan un registro de salida.

`RELATIVE KEY` para `RELATIVE-FILE` no se especifica porque no es necesario para el acceso secuencial, a no ser que se utilice la instrucción START. (Sin embargo, para actualizar, la clave es `INPUT-WEEK`.)

```
5763CB1 V3R0M5                  Fuente COBOL AS/400
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
    1  000010 IDENTIFICATION DIVISION.
    2  000020 PROGRAM-ID.   CRTREL.
       000030
    3  000040 ENVIRONMENT DIVISION.
    4  000050 CONFIGURATION SECTION.
    5  000060 SOURCE-COMPUTER.   IBM-AS400.                                                  05/24/94
    6  000070 OBJECT-COMPUTER.   IBM-AS400.                                                  05/24/94
    7  000080 SPECIAL-NAMES.  REQUESTOR IS REQUESTOR.
    8  000090 FILE-CONTROL.
    9  000100     SELECT RELATIVE-FILE ASSIGN TO DISK-FILED
   10  000110         ORGANIZATION IS RELATIVE
   11  000120         ACCESS IS SEQUENTIAL
   12  000130         FILE STATUS RELATIVE-FILE-STATUS.
   13  000140     SELECT INPUT-FILE ASSIGN TO DISK-FILEC
   14  000150         FILE STATUS INPUT-FILE-STATUS.
       000160
   15  000170 DATA DIVISION.
   16  000180 FILE SECTION.
   17  000190 FD  RELATIVE-FILE LABEL RECORDS ARE STANDARD.
   18  000200 01  RELATIVE-RECORD-01.
   19  000210     05  RELATIVE-RECORD OCCURS 5 TIMES INDEXED BY REL-INDEX.
   20  000220         10  RELATIVE-YEAR          PICTURE 99.
   21  000230         10  RELATIVE-WEEK          PICTURE 99.
   22  000240         10  RELATIVE-UNIT-SALES    PICTURE S9(6).
   23  000250         10  RELATIVE-DOLLAR-SALES  PICTURE S9(9)V99.
   24  000260 FD  INPUT-FILE LABEL RECORDS STANDARD.
   25  000270 01  INPUT-RECORD.
   26  000280     05  INPUT-YEAR                PICTURE 99.
   27  000290     05  INPUT-WEEK                PICTURE 99.
   28  000300     05  INPUT-UNIT-SALES          PICTURE S9(6).
   29  000310     05  INPUT-DOLLAR-SALES        PICTURE S9(9)V99.
   30  000320 WORKING-STORAGE SECTION.
   31  000330 77  INPUT-FILE-STATUS             PICTURE XX.
   32  000340 77  RELATIVE-FILE-STATUS          PICTURE XX.
   33  000350 01  WORK-RECORD.
   34  000360     05  WORK-YEAR                 PICTURE 99 VALUE 00.
   35  000370     05  WORK-WEEK                 PICTURE 99.
   36  000380     05  WORK-UNIT-SALES           PICTURE S9(6).
   37  000390     05  WORK-DOLLAR-SALES         PICTURE S9(9)V99.
   38  000400 01  ERROR-INFO.
   39  000410     05  OP-NAME                   PICTURE X(5).
   40  000420     05  FILLER                    PICTURE X(10)
   41  000430                                     VALUE " ERROR ON ".
   42  000440     05  FILE-NAME                 PICTURE X(13).
   43  000450     05  FILLER                    PICTURE X(16)
   44  000460                                     VALUE " FILE STATUS IS ".
   45  000470     05  STATUS-VALUE              PICTURE XX.
   46  000480 01  ERROR-FLAG                    PICTURE X VALUE SPACE.
   47  000490     88  ERROR-OCCURRED            VALUE "1".
   48  000500 01  INPUTEND                      PICTURE X VALUE SPACE.
   49  000510     88  THE-END-OF-INPUT          VALUE "E".
       000520
   50  000530 PROCEDURE DIVISION.
       000540 DECLARATIVES.
       000550
       000560 INP-FILE-ERROR SECTION.
       000570           USE AFTER STANDARD ERROR PROCEDURE ON INPUT-FILE.
       000580 INPUT-FILE-ERROR.
   51  000590     MOVE "INPUT-FILE" TO FILE-NAME.
   52  000600     MOVE INPUT-FILE-STATUS TO STATUS-VALUE.
   53  000610     SET ERROR-OCCURRED TO TRUE.
       000620 REL-FILE-ERROR SECTION.
       000630           USE AFTER STANDARD ERROR PROCEDURE ON RELATIVE-FILE.
       000640 RELATIVE-FILE-ERROR.
   54  000650     MOVE "RELATIVE-FILE" TO FILE-NAME.
   55  000660     MOVE RELATIVE-FILE-STATUS TO STATUS-VALUE.
   56  000670     SET ERROR-OCCURRED TO TRUE.
       000680 END DECLARATIVES.
       000690 BEGIN-PROCESSING SECTION.
       000700 PROCESSING-CONTROL.
   57  000710     MOVE "OPEN" TO OP-NAME.
   58  000720     OPEN INPUT INPUT-FILE
       000730         OUTPUT RELATIVE-FILE.
   59  000740     IF ERROR-OCCURRED GO TO ERROR-TERMINATION.
   61  000750     SET REL-INDEX TO 1.
   62  000760     PERFORM READ-INPUT-FILE.
```

Figura 116 (Parte 1 de 2). Ejemplo de un Programa de Archivo Relativo

```
 5763CB1 V3R0M5                  Fuente COBOL AS/400
  INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
    63  000770     PERFORM PROCESS-DATA THRU READ-INPUT-FILE
        000780                         UNTIL THE-END-OF-INPUT.
    64  000790     CLOSE RELATIVE-FILE INPUT-FILE.
    65  000800     IF ERROR-OCCURRED GO TO ERROR-TERMINATION.
    67  000810     STOP RUN.
        000820 ERROR-TERMINATION.
    68  000830     DISPLAY ERROR-INFO UPON REQUESTOR.
    69  000840     DISPLAY "PROCESSING TERMINATED DUE TO I-O ERROR"
        000850             UPON REQUESTOR.
    70  000860     STOP RUN.
        000870 PROCESS-DATA.
    71  000880     MOVE INPUT-RECORD TO RELATIVE-RECORD (REL-INDEX).
    72  000890     IF REL-INDEX NOT = 5
    73  000900         SET REL-INDEX UP BY 1
        000910     ELSE
    74  000920         SET REL-INDEX TO 1
    75  000930         PERFORM RELATIVE-FILE-WRITE.
        000940 READ-INPUT-FILE.
    76  000950     MOVE "READ" TO OP-NAME.
    77  000960     READ INPUT-FILE
    78  000970         AT END SET THE-END-OF-INPUT TO TRUE.
    79  000980     IF ERROR-OCCURRED GO TO ERROR-TERMINATION.
        000990 RELATIVE-FILE-WRITE.
    81  001000     MOVE "WRITE" TO OP-NAME.
    82  001010     WRITE RELATIVE-RECORD-01.
    83  001020     IF ERROR-OCCURRED GO TO ERROR-TERMINATION.
                          * * * * *   F I N   D E   F U E N T E   * * * * * 5763CB1 V3R0M5          Mensajes COBOL AS/400
  INST
*       IDMEN: LBL0027  GRAVEDAD: 10  NUMSEC:
        Mensaje . . . . :   No se encuentra I-O SECTION. Se asume la presente
*   17  IDMEN: LBL0650  GRAVEDAD: 00  NUMSEC:  000190
        Mensaje . . . . :   Bloqueo/Desbloqueo para archivo 'RELATIVE-FILE'
          se realizará por medio del código generado por compilador.
*   24  IDMEN: LBL0650  GRAVEDAD: 00  NUMSEC:  000260
        Mensaje . . . . :   Bloqueo/Desbloqueo para archivo 'INPUT-FILE'
          se realizará por medio del código generado por compilador.

                      * * * * *   F I N   D E   M E N S A J E S   * * * * *
                                    Resumen de mensajes
   Total    Info(0-4)    Aviso(5-19)    Error(20-29)    Grave(30-39)    Terminal(40-99)
       3          2            1              0               0               0
 Registros fuente leídos. . . . . :   102
 Registros copia leídos . . . . . :   0
 Miembros copia procesados  . . . . . :   0
 Errores secuencia  . . . . . :   0
 Mensaje gravedad más alta enviado  . :   10
  LBL0901 00  Programa CRTREL creado en biblioteca XMPLIB.

                     * * * * *   F I N   D E   C O M P I L A C I Ó N * * * * *
```

Figura 116 (Parte 2 de 2). Ejemplo de un Programa de Archivo Relativo

Actualización de Archivos Relativos

Este programa utiliza el acceso secuencial para actualizar el archivo de registros de resumen de ventas que se ha creado en el programa `CRTREL`. El programa de actualización añade un registro para cada año nuevo y elimina los registros del año más antiguo desde `RELATIVE-FILE`.

El registro de entrada representa el registro de resumen de ventas para una semana del año anterior. RELATIVE KEY para `RELATIVE-FILE` es el registro de entrada como `INPUT-WEEK`. RELATIVE KEY se utiliza para comprobar que el registro se ha grabado correctamente.

```
5763CB1 V3R0M5                  Fuente COBOL AS/400
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
    1  000010 IDENTIFICATION DIVISION.
    2  000020 PROGRAM-ID.  UPDTREL.
       000030
    3  000040 ENVIRONMENT DIVISION.
    4  000050 CONFIGURATION SECTION.
    5  000060 SOURCE-COMPUTER.   IBM-AS400.                                                  05/24/94
    6  000070 OBJECT-COMPUTER.   IBM-AS400.                                                  05/24/94
    7  000080 INPUT-OUTPUT SECTION.
    8  000090 FILE-CONTROL.
    9  000100     SELECT RELATIVE-FILE ASSIGN TO DISK-FILED
   10  000110         ORGANIZATION IS RELATIVE
   11  000120         ACCESS IS SEQUENTIAL
   12  000130         RELATIVE KEY INPUT-WEEK
   13  000140         FILE STATUS STATUS-VALUE.
   14  000150     SELECT INPUT-FILE ASSIGN TO DISK-FILES2
   15  000160         FILE STATUS STATUS-VALUE.
       000170
   16  000180 DATA DIVISION.
   17  000190 FILE SECTION.
   18  000200 FD  RELATIVE-FILE LABEL RECORDS STANDARD.
   19  000210 01  RELATIVE-RECORD               PICTURE X(105).
   20  000220 FD  INPUT-FILE LABEL RECORDS STANDARD.
   21  000230 01  INPUT-RECORD.
   22  000240     05  INPUT-YEAR                PICTURE 99.
   23  000250     05  INPUT-WEEK                PICTURE 99.
   24  000260     05  INPUT-UNIT-SALES          PICTURE S9(6).
   25  000270     05  INPUT-DOLLAR-SALES        PICTURE S9(9)V99.
   26  000280 WORKING-STORAGE SECTION.
       000290
   27  000300 01  INPUTEND                      PICTURE X VALUE SPACE.
   28  000310     88  THE-END-OF-INPUT          VALUE "E".
   29  000320 01  WORK-RECORD.
   30  000330     05  FILLER                    PICTURE X(21).
   31  000340     05  CURRENT-WORK-YEARS        PICTURE X(84).
   32  000350     05  NEW-WORK-YEAR.
   33  000360     10  WORK-YEAR                 PICTURE 99.
   34  000370     10  WORK-WEEK                 PICTURE 99.
   35  000380     10  WORK-UNIT-SALES           PICTURE S9(6).
   36  000390     10  WORK-DOLLAR-SALES         PICTURE S9(9)V99.
   37  000400 66  WORK-OUT-RECORD RENAMES
   38  000410     CURRENT-WORK-YEARS THROUGH NEW-WORK-YEAR.
   39  000420 01  ERROR-MESSAGE.
   40  000430     05  OP-NAME                  PICTURE X(7).
   41  000440     05  FILLER                   PICTURE X(10)
   42  000450                                     VALUE " ERROR ON ".
   43  000460     05  FILE-NAME                PICTURE X(13).
   44  000470     05  FILLER                   PICTURE X(16)
   45  000480                                     VALUE " FILE STATUS IS ".
   46  000490     05  STATUS-VALUE             PICTURE X(2).
       000500
   47  000510 PROCEDURE DIVISION.
       000520 DECLARATIVES.
       000530 I-O-ERROR SECTION.
       000540     USE AFTER STANDARD ERROR PROCEDURE ON RELATIVE-FILE,
       000550                                          INPUT-FILE.
       000560 ERROR-PROCEDURE.
   48  000570     DISPLAY ERROR-MESSAGE.
   49  000580     DISPLAY "PROCESSING TERMINATING".
   50  000590     STOP RUN.
       000600 END DECLARATIVES.
       000610 MAIN-PROCEDURE SECTION.
       000620 BEGIN-PROCESSING.
   51  000630     MOVE "OPEN" TO OP-NAME.
   52  000640     MOVE "INPUT-FILE" TO FILE-NAME.
   53  000650     OPEN INPUT INPUT-FILE.
   54  000660     MOVE "RELATIVE-FILE" TO FILE-NAME.
   55  000670     OPEN I-O RELATIVE-FILE.
   56  000680     PERFORM READ-FILES.
   57  000690     PERFORM UPDATE-RELATIVE-FILE THRU READ-FILES
       000700                         UNTIL THE-END-OF-INPUT.
   58  000710     MOVE "CLOSE" TO OP-NAME.
   59  000720     MOVE "INPUT-FILE" TO FILE-NAME.
   60  000730     CLOSE INPUT-FILE.
   61  000740     MOVE "RELATIVE-FILE" TO FILE-NAME.
   62  000750     CLOSE RELATIVE-FILE.
   63  000760     STOP RUN.
```

Figura 117 (Parte 1 de 2). Ejemplo de un Programa de Actualización de Archivo Relativo

```
5763CB1 V3R0M5                  Fuente COBOL AS/400
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
      000770 UPDATE-RELATIVE-FILE.
   64 000780     MOVE "REWRITE" TO OP-NAME.
   65 000790     MOVE "RELATIVE-FILE" TO FILE-NAME.
   66 000800     REWRITE RELATIVE-RECORD FROM WORK-OUT-RECORD.
      000810 READ-FILES.
   67 000820     MOVE "READ" TO OP-NAME.
   68 000830     MOVE "RELATIVE-FILE" TO FILE-NAME.
   69 000840     READ RELATIVE-FILE INTO WORK-RECORD
   70 000850         AT END SET THE-END-OF-INPUT TO TRUE.
   71 000860     MOVE "INPUT-FILE" TO FILE-NAME.
   72 000870     READ INPUT-FILE INTO NEW-WORK-YEAR
   73 000880         AT END SET THE-END-OF-INPUT TO TRUE.
                       * * * * *  F I N  D E  F U E N T E  * * * * *
5763CB1 V3R0M5                  Mensajes COBOL AS/400
 INST
*  20  IDMEN: LBL0650  GRAVEDAD: 00  NUMSEC:  000220
       Mensaje . . . . :   Bloqueo/Desbloqueo para archivo 'INPUT-FILE'
         se realizará por medio del código generado por compilador.

                     * * * * *  F I N  D E  M E N S A J E S  * * * * *
                                    Resumen de mensajes
 Total    Info(0-4)    Aviso(5-19)    Error(20-29)    Grave(30-39)    Terminal(40-99)
     1          1            0              0               0               0
Registros fuente leídos. . . . . :   88
Registros copia leídos . . . . . :   0
Miembros copia procesados  . . . . . :   0
Errores secuencia  . . . . . :   0
Mensaje gravedad más alta enviado  . :   0
 LBL0901 00  Programa UPDTREL creado en biblioteca XMPLIB.

                    * * * * *  F I N  D E  C O M P I L A C I Ó N * * * * *
```

Figura 117 (Parte 2 de 2). Ejemplo de un Programa de Actualización de Archivo Relativo

Recuperación de Archivos Relativos

Este programa recupera el archivo resumen creado por el programa `CRTREL`, utilizando el acceso dinámico.

Los registros del `INPUT-FILE` contienen un campo obligatorio (INPUT-WEEK), que es el campo RELATIVE KEY para el `RELATIVE-FILE` y un campo opcional (END-WEEK). Un registro de entrada que contienen datos `INPUT-WEEK` y espacios `END-WEEK` solicita una salida impresa para el `RELATIVE-RECORD`; el registro de recupera por medio del acceso al azar. (El **proceso al azar** es un método de procesar en el que los registros se pueden leer, escribir o eliminar de un archivo en un orden solicitado por el programa que las utiliza.) Un registro de entrada que contenga datos tanto en `INPUT-WEEK` y `END-WEEK` solicita una salida impresa de todos los registros `RELATIVE-FILE` dentro del rango RELATIVE KEY que se incluye entre `INPUT-WEEK` y `END-WEEK`. Estos registros se recuperan mediante acceso secuencial.

```
5763CB1 V3R0M5                  Fuente COBOL AS/400
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
    1  000010 IDENTIFICATION DIVISION.
    2  000020 PROGRAM-ID.  RTRVREL.
       000030
    3  000040 ENVIRONMENT DIVISION.
    4  000050 CONFIGURATION SECTION.
    5  000060 SOURCE-COMPUTER.   IBM-AS400.                                                  05/24/94
    6  000070 OBJECT-COMPUTER.   IBM-AS400.                                                  05/24/94
    7  000080 SPECIAL-NAMES.  REQUESTOR IS REQUESTOR.
    8  000090 INPUT-OUTPUT SECTION.
    9  000100 FILE-CONTROL.
   10  000110     SELECT RELATIVE-FILE ASSIGN TO DISK-FILED
   11  000120         ORGANIZATION IS RELATIVE
   12  000130         ACCESS IS DYNAMIC
   13  000140         RELATIVE KEY INPUT-WEEK
   14  000150         FILE STATUS IS RELATIVE-FILE-STATUS.
   15  000160     SELECT INPUT-FILE ASSIGN TO DISK-FILEF
   16  000170         FILE STATUS IS INPUT-FILE-STATUS.
   17  000180     SELECT PRINT-FILE ASSIGN TO PRINTER-QSYSPRT
   18  000190         FILE STATUS IS PRINT-FILE-STATUS.
       000200
   19  000210 DATA DIVISION.
   20  000220 FILE SECTION.
   21  000230 FD  RELATIVE-FILE LABEL RECORDS STANDARD.
   22  000240 01  RELATIVE-RECORD-01.
   23  000250     05  RELATIVE-RECORD OCCURS 5 TIMES INDEXED BY REL-INDEX.
   24  000260         10  RELATIVE-YEAR         PICTURE 99.
   25  000270         10  RELATIVE-WEEK         PICTURE 99.
   26  000280         10  RELATIVE-UNIT-SALES   PICTURE S9(6).
   27  000290         10  RELATIVE-DOLLAR-SALES PICTURE S9(9)V99.
   28  000300 FD  INPUT-FILE LABEL RECORDS STANDARD.
   29  000310 01  INPUT-RECORD.
   30  000320     05 INPUT-WEEK                 PICTURE 99.
   31  000330     05  END-WEEK                  PICTURE 99.
   32  000340 FD  PRINT-FILE LABEL RECORDS OMITTED.
   33  000350 01  PRINT-RECORD.
   34  000360     05 PRINT-WEEK                 PICTURE 99.
   35  000370     05 FILLER                     PICTURE X(5).
   36  000380     05 PRINT-YEAR                 PICTURE 99.
   37  000390     05 FILLER                     PICTURE X(5).
   38  000400     05 PRINT-UNIT-SALES           PICTURE ZZZ,ZZ9.
   39  000410     05 FILLER                     PICTURE X(5).
   40  000420     05 PRINT-DOLLAR-SALES         PICTURE $$$$,$$$,$$$.99.
   41  000430 WORKING-STORAGE SECTION.
   42  000440 77  RELATIVE-FILE-STATUS          PICTURE XX.
   43  000450 77  INPUT-FILE-STATUS             PICTURE XX.
   44  000460 77  PRINT-FILE-STATUS             PICTURE XX.
   45  000470 77  HIGH-WEEK                     PICTURE 99 VALUE 53.
   46  000480 77  OP-NAME                       PICTURE X(9).
   47  000490 01  INPUTEND                      PICTURE X(9).
   48  000500     88  THE-END-OF-INPUT          VALUE "E".
   49  000510 PROCEDURE DIVISION.
       000520 DECLARATIVES.
       000530 RELATIVE-FILE-ERROR SECTION.
       000540          USE AFTER STANDARD ERROR PROCEDURE ON RELATIVE-FILE.
       000550 RELATIVE-ERROR-MSG.
   50  000560     DISPLAY OP-NAME, " ERROR ON RELATIVE-FILE ".
   51  000570     DISPLAY "FILE STATUS VALUE IS ", RELATIVE-FILE-STATUS.
   52  000580     DISPLAY "PROCESSING TERMINATED ".
   53  000590     STOP RUN.
       000600 INPUT-FILE-ERROR SECTION.
       000610          USE AFTER STANDARD ERROR PROCEDURE ON INPUT-FILE.
       000620 INPUT-ERROR-MSG.
   54  000630     DISPLAY OP-NAME, " ERROR ON INPUT-FILE ".
   55  000640     DISPLAY "FILE STATUS VALUE IS ", INPUT-FILE-STATUS.
   56  000650     DISPLAY "PROCESSING TERMINATED ".
   57  000660     STOP RUN.
       000670 PRINT-FILE-ERROR SECTION.
       000680          USE AFTER STANDARD ERROR PROCEDURE ON PRINT-FILE.
       000690 PRINT-ERROR-MSG.
   58  000700     DISPLAY OP-NAME, " ERROR ON PRINT-FILE ".
   59  000710     DISPLAY "FILE STATUS VALUE IS ", PRINT-FILE-STATUS.
   60  000720     DISPLAY "PROCESSING TERMINATED ".
   61  000730     STOP RUN.
       000740 END DECLARATIVES.
```

Figura 118 (Parte 1 de 2). Ejemplo de un Programa de Recuperación de Archivo Relativo

```
5763CB1 V3R0M5                  Fuente COBOL AS/400
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
      000750 MAIN-PROCEDURE SECTION.
      000760 MAIN-PROCESSING.
   62 000770     MOVE "OPEN" TO OP-NAME.
   63 000780     OPEN INPUT INPUT-FILE RELATIVE-FILE
      000790          OUTPUT PRINT-FILE.
   64 000800     MOVE SPACES TO PRINT-RECORD.
   65 000810     PERFORM READ-INPUT-FILE.
   66 000820     PERFORM CONTROL-PROCESS THRU READ-INPUT-FILE
      000830                        UNTIL THE-END-OF-INPUT.
   67 000840     MOVE "CLOSE" TO OP-NAME.
   68 000850     CLOSE RELATIVE-FILE
      000860          INPUT-FILE
      000870          PRINT-FILE.
   69 000880     STOP RUN.
      000890 CONTROL-PROCESS.
   70 000900     IF (END-WEEK = SPACES OR END-WEEK = 00)
   71 000910         PERFORM RANDOM-PROCESS
      000920     ELSE
   72 000930         PERFORM SEQUENTIAL-PROCESS.
      000940 READ-INPUT-FILE.
   73 000950     MOVE "READ" TO OP-NAME.
   74 000960     READ INPUT-FILE
   75 000970         AT END SET THE-END-OF-INPUT TO TRUE.
      000980 RANDOM-PROCESS.
   76 000990     MOVE "READ" TO OP-NAME.
   77 001000     READ RELATIVE-FILE
   78 001010          INVALID KEY MOVE HIGH-WEEK TO RELATIVE-WEEK(1).
   79 001020     IF RELATIVE-WEEK(1) NOT EQUAL HIGH-WEEK
   80 001030          PERFORM PRINT-SUMMARY VARYING REL-INDEX FROM 1 BY 1
      001040                                UNTIL REL-INDEX > 5.
      001050 SEQUENTIAL-PROCESS.
   81 001060     MOVE "READ" TO OP-NAME.
   82 001070     READ RELATIVE-FILE
   83 001080          INVALID KEY MOVE HIGH-WEEK TO RELATIVE-WEEK(1).
   84 001090     PERFORM READ-REL-SEQ
      001100          UNTIL RELATIVE-WEEK(1) GREATER THAN END-WEEK.
      001110
      001120 READ-REL-SEQ.
   85 001130     PERFORM PRINT-SUMMARY VARYING REL-INDEX FROM 1 BY 1
      001140                           UNTIL REL-INDEX > 5.
   86 001150     MOVE "READ NEXT" TO OP-NAME.
   87 001160     READ RELATIVE-FILE NEXT RECORD
   88 001170         AT END MOVE HIGH-WEEK TO RELATIVE-WEEK(1).
      001180 PRINT-SUMMARY.
   89 001190     MOVE RELATIVE-YEAR (REL-INDEX) TO PRINT-YEAR.
   90 001200     MOVE RELATIVE-WEEK (REL-INDEX) TO PRINT-WEEK.
   91 001210     MOVE RELATIVE-UNIT-SALES (REL-INDEX) TO PRINT-UNIT-SALES.
   92 001220     MOVE RELATIVE-DOLLAR-SALES(REL-INDEX) TO PRINT-DOLLAR-SALES.
   93 001230     MOVE "WRITE" TO OP-NAME.
   94 001240     WRITE PRINT-RECORD AFTER ADVANCING 2 LINES.
                        * * * * *   F I N   D E   F U E N T E   * * * * *
5763CB1 V3R0M5                  Mensajes COBOL AS/400
 INST
*   28  IDMEN: LBL0650  GRAVEDAD: 00  NUMSEC:  000300
       Mensaje . . . . :   Bloqueo/Desbloqueo para archivo 'INPUT-FILE'
         se realizará por medio del código generado por compilador.

                     * * * * *   F I N   D E   M E N S A J E S   * * * * *
                                    Resumen de mensajes
 Total    Info(0-4)    Aviso(5-19)    Error(20-29)    Grave(30-39)    Terminal(40-99)
     1            1              0               0               0                 0
Registros fuente leídos. . . . . :   124
Registros copia leídos . . . . . :   0
Miembros copia procesados  . . . . :   0
Errores secuencia  . . . . . :   0
Mensaje gravedad más alta enviado  . :   0
 LBL0901 00  Programa RTRVREL creado en biblioteca XMPLIB.

                   * * * * *   F I N   D E   C O M P I L A C I Ó N   * * * *
```

Figura 118 (Parte 2 de 2). Ejemplo de un Programa de Recuperación de Archivo Relativo

Archivos de Fusión y Clasificación

La Figura 119 ilustra la creación de archivos clasificados de ventas actuales y ventas anuales hasta la fecha.

Ante todo se ejecuta la instrucción SORT para las ventas actuales. El procedimiento de entrada para esta operación de clasificación es SCREEN-DEPT. Los registros se clasifican en orden ascendente por departamento, y dentro de cada departamento, en orden descendente por ventas netas. Después se imprime la salida para esta clasificación.

Una vez se completa la operación de clasificación, el registro de ventas actuales se fusiona con los registros de ventas anuales hasta la fecha. Los registros de este archivo se fusionan en orden ascendente por número de departamento y, dentro de cada departamento, en orden ascendente por número de empleado y, para cada empleado, en orden ascendente por meses para crear un archivo maestro actualizado hasta la fecha.

Cuando el proceso de fusión finaliza, se imprime el archivo maestro actualizado hasta la fecha.

```
5763CB1 V3R0M5  910524            Fuente COBOL AS/400
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
    1  000010 IDENTIFICATION DIVISION.
    2  000020 PROGRAM-ID.  SORTMERGE.
      000030*********************************************************
       000040* EJEMPLO SORT/MERGE QUE UTILIZA PROCEDIMIENTO ENTRADA  *
       000050*********************************************************
    3  000060 ENVIRONMENT DIVISION.
    4  000070 CONFIGURATION SECTION.
    5  000080 SOURCE-COMPUTER. IBM-AS400.
    6  000090 OBJECT-COMPUTER. IBM-AS400.
    7  000100 SPECIAL-NAMES.
    8  000110    REQUESTOR IS CONSOLE.
    9  000120 INPUT-OUTPUT SECTION.
   10  000130 FILE-CONTROL.
   11  000140     SELECT WORK-FILE ASSIGN TO DISK-WRK.
   12  000150     SELECT CURRENT-SALES-FILE-IN ASSIGN TO DISK-CURRIN.
   13  000160     SELECT CURRENT-SALES-FILE-OUT ASSIGN TO DISK-CURROUT.
   14  000170     SELECT YTD-SALES-FILE-IN ASSIGN TO DISK-YTDIN.
   15  000180     SELECT YTD-SALES-FILE-OUT ASSIGN TO DISK-YTDOUT.
   16  000190     SELECT PRINTER-OUT ASSIGN TO PRINTER-QPRINT.
   17  000200 DATA DIVISION.
   18  000210 FILE SECTION.
   19  000220 SD  WORK-FILE
   20  000230     DATA RECORD IS SALES-RECORD.
   21  000240 01  SALES-RECORD.
   22  000250   05 EMPL-NO                 PIC 9(6).
   23  000260   05 DEPT                    PIC 9(2).
   24  000270   05 SALES                   PIC 9(7)V99.
   25  000280   05 NAME-ADDR               PIC X(61).
   26  000290   05 MONTH                   PIC X(2).
   27  000300 FD  CURRENT-SALES-FILE-IN
   28  000310     LABEL RECORDS STANDARD
   29  000320     DATA RECORD CURRENT-SALES-IN.
   30  000330 01 CURRENT-SALES-IN.
   31  000340   05 EMPL-NO                 PIC 9(6).
   32  000350   05 DEPT                    PIC 9(2).
   33  000360     88 ON-SITE-EMPLOYEE     VALUES 0
   34  000370                             THRU 6, 8.
   35  000380   05 SALES                    PIC 9(7)V99.
   36  000390   05 NAME-ADDR                PIC X(61).
   37  000400   05 MONTH                    PIC X(2).
   38  000410 FD  CURRENT-SALES-FILE-OUT
   39  000420     LABEL RECORDS STANDARD
   40  000430     DATA RECORD CURRENT-SALES-OUT.
   41  000440 01 CURRENT-SALES-OUT.
```

Figura 119 (Parte 1 de 3). Ejemplo de Utilización de SORT/MERGE

```
5763CB1 V3R0M5  910524           Fuente COBOL AS/400
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
   42  000450   05 EMPL-NO                 PIC 9(6).
   43  000460   05 DEPT                    PIC 9(2).
   44  000470   05 SALES                   PIC 9(7)V99.
   45  000480   05 NAME-ADDR               PIC X(61).
   46  000490   05 MONTH                   PIC X(2).
   47  000500 FD  YTD-SALES-FILE-IN
   48  000510     LABEL RECORDS STANDARD
   49  000520     DATA RECORD YTD-SALES-IN.
   50  000530 01 YTD-SALES-IN.
   51  000540   05 EMPL-NO                 PIC 9(6).
   52  000550   05 DEPT                    PIC 9(2).
   53  000560   05 SALES                   PIC 9(7)V99.
   54  000570   05 NAME-ADDR               PIC X(61).
   55  000580   05 MONTH                   PIC X(2).
   56  000590 FD  YTD-SALES-FILE-OUT
   57  000600     LABEL RECORDS STANDARD
   58  000610     DATA RECORD YTD-SALES-OUT.
   59  000620 01 YTD-SALES-OUT.
   60  000630   05 EMPL-NO                 PIC 9(6).
   61  000640   05 DEPT                    PIC 9(2).
   62  000650   05 SALES                   PIC 9(7)V99.
   63  000660   05 NAME-ADDR               PIC X(61).
   64  000670   05 MONTH                   PIC X(2).
   65  000680 FD  PRINTER-OUT
   66  000690     LABEL RECORDS OMITTED
   67  000700     DATA RECORD PRINT-LINE.
   68  000710 01 PRINT-LINE.
   69  000720   05 RECORD-LABEL            PIC X(25).
   70  000730   05 DISK-RECORD-DISPLAY     PIC X(80).
   71  000740 WORKING-STORAGE SECTION.
   72  000750 01 SALES-FILE-IN-EOF-STATUS  PIC X        VALUE "F".
   73  000760    88 SALES-FILE-IN-END-OF-FILE          VALUE "T".
   74  000770 01 SALES-FILE-OUT-EOF-STATUS PIC X        VALUE "F".
   75  000780    88 SALES-FILE-OUT-END-OF-FILE         VALUE "T".
   76  000790 01 YTD-SALES-OUT-EOF-STATUS  PIC X        VALUE "F".
   77  000800    88 YTD-SALES-OUT-END-OF-FILE          VALUE "T".
   78  000810 PROCEDURE DIVISION.
       000820 OPEN-PRINTER-FILE SECTION.
       000830 005-PRINTER-FILE.
   79  000840     OPEN OUTPUT PRINTER-OUT.
       000850 LIST-SORT-LIST-CURRENT-SALES SECTION.
       000860 010-LIST-SORT-CURRENT-SALES.
   80  000870     SORT WORK-FILE
       000880          ON ASCENDING KEY DEPT OF SALES-RECORD
       000890          ON DESCENDING KEY SALES OF SALES-RECORD
       000900          INPUT PROCEDURE SCREEN-DEPT
       000910          GIVING CURRENT-SALES-FILE-OUT.
       000920 020-LIST-SORTED-SALES.
   81  000930     OPEN INPUT CURRENT-SALES-FILE-OUT.
   82  000940     PERFORM 100-PRINT-SALES-FILE-OUT
       000950       THRU 110-END-PRINT-SALES-FILE-OUT
       000960         UNTIL SALES-FILE-OUT-END-OF-FILE.
   83  000970     CLOSE CURRENT-SALES-FILE-OUT.
       000980 UPDATE-YEARLY-REPORT SECTION.
       000990 040-MERGE-CURRENT-PREVIOUS.
   84  001000     MERGE WORK-FILE
       001010          ON ASCENDING KEY DEPT OF SALES-RECORD
       001020          ON ASCENDING KEY EMPL-NO OF SALES-RECORD
       001030          ON ASCENDING KEY MONTH OF SALES-RECORD
       001040          USING YTD-SALES-FILE-IN
       001050                CURRENT-SALES-FILE-IN
       001060          GIVING YTD-SALES-FILE-OUT.
       001070 040-PRINT-YTD-SALES-OUT.
   85  001080     OPEN INPUT YTD-SALES-FILE-OUT.
   86  001090     PERFORM 120-READ-PRINT-YTD-SALES-OUT
       001100         UNTIL YTD-SALES-OUT-END-OF-FILE.
   87  001110     CLOSE YTD-SALES-FILE-OUT
       001120           PRINTER-OUT.
   88  001130     STOP RUN.
       001140 SCREEN-DEPT SECTION.
       001150 060-S-D-1.
   89  001160     OPEN INPUT CURRENT-SALES-FILE-IN
   90  001170     PERFORM 070-READ-SELECT-DEPT THRU 080-END-READ-SELECT-DEPT
       001180         UNTIL SALES-FILE-IN-END-OF-FILE.
```

Figura 119 (Parte 2 de 3). Ejemplo de Utilización de SORT/MERGE

```
5763CB1 V3R0M5  910524            Fuente COBOL AS/400
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
   91  001190     CLOSE CURRENT-SALES-FILE-IN.
   92  001200     GO TO 090-END-S-D-1.
       001210 070-READ-SELECT-DEPT.
   93  001220     READ CURRENT-SALES-FILE-IN
   94  001230         AT END MOVE "T" TO SALES-FILE-IN-EOF-STATUS
   95  001240                GO TO 080-END-READ-SELECT-DEPT.
   96  001250     MOVE "UNSORTED CURRENT SALES ",
       001260       TO RECORD-LABEL OF PRINT-LINE.
   97  001270     MOVE CURRENT-SALES-IN TO DISK-RECORD-DISPLAY.
   98  001280     WRITE PRINT-LINE.
   99  001290     IF ON-SITE-EMPLOYEE
  100  001300         MOVE CURRENT-SALES-IN TO SALES-RECORD
  101  001310         RELEASE SALES-RECORD.
       001320 080-END-READ-SELECT-DEPT.
       001330     EXIT.
  102  001340 090-END-S-D-1.
       001350 END-SCREEN-DEPT SECTION.
       001360 100-PRINT-SALES-FILE-OUT.
  103  001370     READ CURRENT-SALES-FILE-OUT
  104  001380         AT END MOVE "T" TO SALES-FILE-OUT-EOF-STATUS
  105  001390                GO TO 110-END-PRINT-SALES-FILE-OUT.
  106  001400     MOVE "SORTED CURRENT SALES "
       001410       TO RECORD-LABEL OF PRINT-LINE.
  107  001420     MOVE CURRENT-SALES-OUT TO DISK-RECORD-DISPLAY.
  108  001430     WRITE PRINT-LINE.
       001440 110-END-PRINT-SALES-FILE-OUT.
       001450     EXIT.
  109  001460 120-READ-PRINT-YTD-SALES-OUT.
  110  001470     READ YTD-SALES-FILE-OUT
  111  001480         AT END MOVE "T" TO YTD-SALES-OUT-EOF-STATUS
  112  001490                GO TO 130-END-READ-PRT-YTD-SALES-OUT.
  113  001500     MOVE "MERGED YTD SALES ",
       001510       TO RECORD-LABEL OF PRINT-LINE.
  114  001520     MOVE YTD-SALES-OUT TO DISK-RECORD-DISPLAY.
  115  001530     WRITE PRINT-LINE.
       001540 130-END-READ-PRT-YTD-SALES-OUT.
       001550     EXIT.
                        * * * * *  F I N  D E  F U E N T E  * * * * *
```

Figura 119 (Parte 3 de 3). Ejemplo de Utilización de SORT/MERGE

Apéndice H. Ejemplo de Vuelco con Formato COBOL

La Figura 120 en la página 394 muestra un ejemplo de un vuelco con formato COBOL. Para estar seguro de que un vuelco está disponible cuando se produce alguna anomalía al ejecutar el programa, cambie el parámetro INQMSGRPY del trabajo (por ejemplo, mediante el mandato CHGJOB) por *RQD. Cuando se solicita, es posible especificar la generación de un vuelco.

La siguiente lista describe las áreas etiquetadas de la figura:

A La excepción por la que se ha pedido el vuelco y la ubicación en el programa en la que se ha producido la excepción.

B El número de instrucción COBOL de la última operación de E/S que se ha ejecutado antes de que se haya producido la excepción. Esta información sólo aparece si se ha procesado como mínimo una operación de E/S.

C La información actual para cada archivo. Esta información sólo aparece si el programa tiene archivos.

D Comienzo de los campos generados por el compilador (incluidos en el vuelco si el usuario responde con una opción F).

E Distintivos de E/S del archivo actual:

Bit	Significado
1	El archivo está abierto
2	El archivo está bloqueado
3	Fin de archivo
4	(Reservado)
5	Archivo optativo
6	Comprobación de duplicados en archivo indexado durante la apertura
7	Fin de página
8	(Reservado).

F Código de estado anterior.

G Comienzo de la Tabla Global de Módulos (MGT).[3]

H Último código de excepción.

I Número de invocación del programa actual.

J Nombre calificado de programa y biblioteca.

K Comienzo de la Tabla Global del Programa (PGT).[4]

L Número de invocación del programa principal COBOL.

M Fecha del trabajo (AAMMDD).

N Comienzo de los campos del usuario.

O Campo con zona no válido impreso en hexadecimal.

---

[3] La Tabla Global de Módulos (MGT) define un área común para el módulo. La tabla se utiliza para pasar información a las subrutinas en tiempo de ejecución.

[4] La Tabla Global del Programa (PGT) es un área de comunicaciones para la unidad de ejecución COBOL entera. Sólo hay una PGT para la unidad de ejecución.

```
5763CB1 V3R0M5                   Fuente COBOL AS/400
 INST NUMSEC -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  NOMCOPIA   FECH/CAM
    1  000100 IDENTIFICATION DIVISION.                                                          03/07/94
    2  000200 PROGRAM-ID.     XMPLDUMP.                                                         03/22/94
    3  000300   AUTHOR.       PROGRAMMER NAME.                                                  03/07/94
    4  000400   INSTALLATION. COBOL DEVELOPMENT CENTRE.                                        03/07/94
    5  000500   DATE-WRITTEN. 11/27/88.                                                        03/07/94
    6  000600   DATE-COMPILED.  05/24/94 12:21:54.                                             03/07/94
    7  000700 ENVIRONMENT DIVISION.                                                           03/07/94
    8  000800 CONFIGURATION SECTION.                                                          03/07/94
    9  000900 SOURCE-COMPUTER. IBM-AS400.                                                     03/07/94
   10  001000 OBJECT-COMPUTER. IBM-AS400.                                                     03/07/94
   11  001100 INPUT-OUTPUT SECTION.                                                           03/07/94
   12  001200 FILE-CONTROL.                                                                   03/07/94
   13  001300     SELECT FILE-1 ASSIGN TO DISK-SALES.                                        03/22/94
   14  001400 DATA DIVISION.                                                                  03/07/94
   15  001500 FILE SECTION.                                                                   03/07/94
   16  001600 FD  FILE-1                                                                      03/07/94
   17  001700     LABEL RECORDS ARE STANDARD.                                                02/17/94
   18  001800 01  RECORD-1.                                                                   03/07/94
   19  001900   05 R-TYPE             PIC X(1).                                              02/17/94
   20  002000   05 R-AREA-CODE        PIC 9(2).                                              02/17/94
   21  002100     88 R-NORTH-EAST VALUES 15 THROUGH 30.                                      02/17/94
   22  002200   05 R-SALES-CAT-1      PIC S9(5)V9(2) COMP-3.                                 02/17/94
   23  002300   05 R-SALES-CAT-2      PIC S9(5)V9(2) COMP-3.                                 02/17/94
   24  002400   05 FILLER             PIC X(1).                                              02/17/94
       002500                                                                                 02/17/94
   25  002600 WORKING-STORAGE SECTION.                                                        03/07/94
   26  002700 01  W-SALES-VALUES.                                                            02/17/94
   27  002800     05 W-CAT-1          PIC S9(8)V9(2).                                        02/17/94
   28  002900     05 W-CAT-2          PIC S9(8)V9(2).                                        02/17/94
   29  003000     05 W-TOTAL          PIC S9(8)V9(2).                                        02/17/94
       003100                                                                                 02/17/94
   30  003200 01  W-EDIT-VALUES.                                                            02/17/94
   31  003300     05 FILLER           PIC X(8) VALUE "TOTALS: ".                             02/17/94
   32  003400     05 W-EDIT-1         PIC Z(7)9.9(2)-.                                       02/17/94
   33  003500     05 FILLER           PIC X(3) VALUE SPACES.                                 02/17/94
   34  003600     05 W-EDIT-2         PIC Z(7)9.9(2)-.                                       02/17/94
   35  003700     05 FILLER           PIC X(3) VALUE SPACES.                                 02/17/94
   36  003800     05 W-EDIT-TOTAL     PIC Z(7)9.9(2)-.                                       02/17/94
       003900                                                                                 02/17/94
   37  004000 01  END-FLAG           PIC X(1) VALUE SPACE.                                  02/17/94
   38  004100     88 END-OF-INPUT VALUE "Y".                                                 02/17/94
       004200                                                                                 02/17/94
   39  004300 PROCEDURE DIVISION.                                                             02/17/94
       004400*****************************************************************               02/17/94
       004500* ABRE LA ENTRADA, BORRA LOS TOTALES, LLAMA AL PROCESO         *                02/17/94
       004600* PRINCIPAL, VISUALIZA LOS RESULTADOS Y TERMINA LA EJECUCIÓN   *                02/17/94
       004700*****************************************************************               02/17/94
       004800 P-START.                                                                        02/17/94
   40  004900     OPEN INPUT FILE-1.                                                         02/17/94
   41  005000     MOVE ZEROS TO W-SALES-VALUES.                                              02/17/94
   42  005100     PERFORM P-MAIN UNTIL END-OF-INPUT.                                         02/17/94
       005200                                                                                 02/17/94
   43  005300     MOVE W-CAT-1 TO W-EDIT-1.                                                  02/17/94
   44  005400     MOVE W-CAT-2 TO W-EDIT-2.                                                  02/17/94
   45  005500     MOVE W-TOTAL TO W-EDIT-TOTAL.                                              02/17/94
   46  005600     DISPLAY W-EDIT-VALUES.
   47  005700     STOP RUN.
       005800
       005900*****************************************************************
       006000* LEE ARCH. ENTRADA Y SÓLO PROCESA LOS REG. CORRESPONDIENTES   *
       006100* AL ÁREA NOROESTE. AL LLEGAR FINAL DE ENTRADA, PONGA EL DIST. *
       006200*****************************************************************
       006300 P-MAIN.
   48  006400     READ FILE-1 AT END SET END-OF-INPUT TO TRUE.
   50  006500     IF R-NORTH-EAST AND NOT END-OF-INPUT
   51  006600         ADD R-SALES-CAT-1 TO W-CAT-1, W-TOTAL
   52  006700         ADD R-SALES-CAT-2 TO W-CAT-2, W-TOTAL.
                        * * * * *   F I N   D E   F U E N T E   * * * * *

Excepción MCH1202 en programa XMPLDUMP en QTEMP en número de instrucción
MI 005C sentencia COBOL número 51.
La última operación de E/S ha ocurrido en la instrucción 48. B
LBE7903-Información perteneciente al archivo FILE-1. C
LBE7905-El archivo está abierto.
LBE7906-Última operación de E/S completada para el archivo era READ.
LBE7907-Último estado de archivo era 04.
LBE7910-Último estado de archivo ampliado era.
```

Figura 120 (Parte 1 de 10). Ejemplo de un Vuelco con Formato COBOL

```
 VUELCO DE DATOS PARA EL PROGRAMA XMPLDUMP.QTEMP           13:39:08
05/24/94
 NOMBRE         DESPL.  ATRIBUTOS    VALOR  D
 .ADBUF         000480  POINTER(SPP)  NULL
 .ADBUFVL       000B90  CHAR(68)      '                                                                    '
                000B90  VALUE IN HEX  '00000000000000000000000000000000000000000000000000000000000000000000000000000000'X
                000BB8    +41         '00000000000000000000000000000000000000000000000000000000'X
 .ADDEV         0004B4  CHAR(10)      '          '                   '0000000000000000000'X
 .ADENV         000493  CHAR(1)       'I'
 .ADFILE        0004C0  POINTER(SPP)  NULL
 .ADFUNC        000490  CHAR(1)       ' '                           '00'X
 .ADLN          000494  BINARY(2)     0
 .ADMID         000496  BINARY(2)     0
 .ADPGM         00049B  CHAR(10)      'XMPLDUMP  '
 .ADRLN         000498  BINARY(2)     0
 .ADRTN         000470  POINTER(IP)   NULL
 .ADRTYP        00049A  CHAR(1)       ' '                           '00'X
 .ADTOD         0004A5  CHAR(15)      '               '             '000000000000000000000000000000'X
 .ADTYP         000491  BINARY(2)     0
 .BINSUB        000558  BINARY(4)     0
 .BIN2          00055C  BINARY(2)     0
 .BPCA          0004B0  CHAR(32767)   '          A        T            D01   <HHH               D12   <HHH               D15   <H'
                00050A    +91         'HH              D15   <HHH              D22   <HHH               D99   <HHH             '
                000564    +181        '  D01   <HHH              D23   <HHH               D25   <HHH              '
                0004B0  VALUE IN HEX  '800000000000000003DC19EB7000A40000100A30019000E40404040404040C4F0F14444444C88'X
                0004D8    +41         '88888C4040400001000000010000000040C4F1F24444444C8888888C4040400001000000002000000'X
                000500    +81         '0040C4F1F54444444C8888888C4040400001000000030000000040C4F1F54444444C8888888C4040'X
                000528    +121        '400001000000040000000040C4F2F24444444C8888888C4040400001000000050000000040C4F9F9'X
                000550    +161        '4444444C8888888C4040400001000000060000000040C4F0F14444444C8888888C40404000010000'X
                000578    +201        '00070000000040C4F2F34444444C8888888C4040400001000000080000000040C4F2F54444444C88'X
                0005A0    +241        '88888C4040400001000000090000000'X
 .BPCACTR       0004C0  BINARY(2)     1
 .BPCAFB        0004C6  BINARY(2)     14
 .BPCAMXR       0004C2  BINARY(2)     163
 .BPCARCD       0004B0  POINTER(SPP)  SPACE OFFSET    1632     '00000660'X
                                      OBJECT    SALES     COBOLEX   SALESFILE
 .BPCARIO       0004C4  BINARY(2)     25
 .BP01CA        0004B0  CHAR(32767)   '          A        T            D01   <HHH               D12   <HHH               D15   <H'
                00050A    +91         'HH              D15   <HHH              D22   <HHH               D99   <HHH             '
                000564    +181        '  D01   <HHH              D23   <HHH               D25   <HHH              '
                0004B0  VALUE IN HEX  '800000000000000003DC19EB7000A40000100A30019000E40404040404040C4F0F14444444C88'X
                0004D8    +41         '88888C4040400001000000010000000040C4F1F24444444C8888888C4040400001000000002000000'X
                000500    +81         '0040C4F1F54444444C8888888C4040400001000000030000000040C4F1F54444444C8888888C4040'X
                000528    +121        '400001000000040000000040C4F2F24444444C8888888C4040400001000000050000000040C4F9F9'X
                000550    +161        '4444444C8888888C4040400001000000060000000040C4F0F14444444C8888888C40404000010000'X
                000578    +201        '00070000000040C4F2F34444444C8888888C4040400001000000080000000040C4F2F54444444C88'X
                0005A0    +241        '88888C4040400001000000090000000'X
 .BP01CTR       0004C0  BINARY(2)     1
 .BP01FB        0004C6  BINARY(2)     14
 .BP01MXR       0004C2  BINARY(2)     163
 .BP01RCD       0004B0  POINTER(SPP)  SPACE OFFSET    1632     '00000660'X
                                      OBJECT    SALES     COBOLEX   SALESFILE
 .BP01RIO       0004C4  BINARY(2)     25
 .BSTRING               NOT ADDRESSABLE
 .BUFFER                NOT ADDRESSABLE
 .BUFPTR        000770  POINTER(SPP)  NULL
 .CALERP        000580  POINTER(SPP)  SPACE OFFSET    1376     '00000560'X
                                      OBJECT    PSSA
 .CALLOWR       000C70  CHAR(27)      ' ETAOINSHRDLUCMFWYPVBGKQJXZ'   '4085A381968995A288998493A4839486A6A897A58287929891A7A9'X
 .CALPHAB       000C20  CHAR(53)      ' ETAOINSHRDLUCMFWYPVBGKQJXZETAOINSHRDLUCMFWYPVBGKQJXZ'
                000C20  VALUE IN HEX  '40C5E3C1D6C9D5E2C8D9C4D3E4C3D4C6E6E8D7E5C2C7D2D8D1E7E985A381968995A288998493A483'X
                000C48    +41         '9486A6A897A58287929891A7A9'X
 .CALUPPR       000C55  CHAR(27)      ' ETAOINSHRDLUCMFWYPVBGKQJXZ'
 .CIMBSGN       000BDA  CHAR(60)      '0123456789 JKLMNOPQR  STUVWXYZ           ABCDEFGHI  STUVWXYZ'
                000BDA  VALUE IN HEX  'F0F1F2F3F4F5F6F7F8F9D0D1D2D3D4D5D6D7D8D9A0A1A2A3A4A5A6A7A8A9B0B1B2B3B4B5B6B7B8B9'X
                000C02    +41         'C0C1C2C3C4C5C6C7C8C9E0E1E2E3E4E5E6E7E8E9'X
 .CNUMERC       000C16  CHAR(10)      '0123456789'
 .CPADCHR       000C8B  CHAR(1)       ' '
 .CRCLEAR       000D00  POINTER(SYP)  OBJECT    QLRCLEAR
                                      CONTEXT   QSYS
 .CSEPSGN       000BD8  CHAR(2)       '+-'
 .DBUGRTN       000450  POINTER(IP)   NULL
 .DEVPTR        000730  POINTER(SPP)  SPACE OFFSET    324      '00000144'X
                                      OBJECT    SALES     COBOLEX   SALESFILE
 .DISPPOS       000CB0  BINARY(2)     0
 .DISPPTR       000CA0  POINTER(SPP)  NULL
 .DLINENO       000552  CHAR(6)       '      '                      '000000000000'X
 .DMCACIN       000870  BINARY(2)     121
 .DMCACQR       000872  BINARY(2)     66
```

Figura 120 (Parte 2 de 10). Ejemplo de un Vuelco con Formato COBOL

```
.DMCBLKR       0003BD  CHAR(1)       '*'
.DMCCPCL       000178  BINARY(2)     13
.DMCCPOP       00017A  BINARY(2)     17
.DMCDBOF       000020  BINARY(4)     704
.DMCDDS        0002C0  CHAR(298)     '                                        R             A    0          A    0          '
               00031A    +91         ' TY                                                                                    '
               000374    +181        '                                                  3               CPF     *  '
               0002C0  VALUE IN HEX  '00000000000000000000000000000000000000000000000000000000000000000000000000000000000000008000000000008000'X
               0002E8    +41         '001000990F0004B0800000000000000000003DC19EB70006F0800000000000000000003DC19EB70006F0'X
               000310    +81         '480000000000000000000000A3E8000011000000000000000000000000000000000000000000000000000019'X
               000338    +121        '0000000E000000000000000000000000000000000000000000000000000000000000000000000000000000000000'X
               000360    +161        '00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000'X
               000388    +201        '000000000000000000000000000000000000000000000000000000000000F3000001000000000000000000'X
               0003B0    +241        '00FF000000C3D7C600000000005CE000'X
.DMCDELT       000166  BINARY(2)     69
.DMCDROP       000874  BINARY(2)     71
.DMCFDEL       0003E9  CHAR(1)       ' '                            '00'X
.DMCFEOD       000168  BINARY(2)     111
.DMCFRCE       00016A  BINARY(2)     69
.DMCGET        000158  BINARY(2)     770
.DMCGETD       00015A  BINARY(2)     14
.DMCGETK       00015C  BINARY(2)     69
.DMCLINK       0007A5  BINARY(2)     0
.DMCODP        000000  CHAR(32767)   'E      M   M           F                                                    A              '
               00005A    +91         '                                  A                                                    DBSA'
               0000B4    +181        'LES     COBOLEX                                SALESFILE                      A'
               000000  VALUE IN HEX  '85000002000014D4000014D4000000B0000001400000001C60000028000000000000002C000000000'X
               000028    +41         '000000000000014000000000000000000000000000000000000000000000000003DC19EB7000DFF'X
               000050    +81         '00000000000000000000000000000008000000000000000003DC00036000AE08000000000000000'X
               000078    +121        '003DC19EB700189B000000000000000000000000000000001900000000000000000000C00000000'X
               0000A0    +161        '0C000000000000000004B00000000000C4C2E2C1D3C5E24040404040C3D6C2F3F8C5E74040400000'X
               0000C8    +201        '00000000000000000000000000000000000000000000E0000E2C1D3C5E2C6C9D3C5400000010040000'X
               0000F0    +241        '000000000000000000000000000011C1'X
.DMCOFFS       000010  BINARY(4)     320
.DMCPTGT       000162  BINARY(2)     69
.DMCPUT        000160  BINARY(2)     69
.DMCPUTD       00015E  BINARY(2)     69
.DMCRLSE       000170  BINARY(2)     69
.DMCRSTD       00016E  BINARY(2)     69
.DMCSPDD       00016C  BINARY(2)     69
.DMCSPTB       00017C  BINARY(2)     0
.DMCTBLE       00017E  BINARY(2)     1
.DMCUPD        000164  BINARY(2)     69
.DMPBDMJ               NOT ADDRESSABLE
.DMPBDSE               NOT ADDRESSABLE
.DMPCDFO       0001C6  BINARY(2)     144
.DMPCDFP       000790  POINTER(SPP)  NULL
.DMPDBFB               NOT ADDRESSABLE
.DMPDBFL       000190  CHAR(1)       ' '                            '00'X
.DMPDENT       000144  CHAR(130)     DIMENSION(250)
               000144    (1)         'DATABASE                           ?                                                  '
               00019E    +91         '                                        '
               000144  VALUE IN HEX  'C4C1E3C1C2C1E2C5404000000000000000000000000302000E00450045004500450045006F0045'X
               00016C    +41         '0045004500450BFD00450045000D00110000000100000000000000000000000000000000000000000000'X
               000194    +81         2 LINES OF ZEROES SUPPRESSED
               0001C6    (2)         '                   SALESFILE                                                          '
               000220    +91         '                                        '
               0001C6  VALUE IN HEX  '009000000000000000010000000000000000001E2C1D3C5E2C6C9D3C5400000000000000000000000'X
               0001EE    +41         '00000000000E000000000000000000000000000000000000000000000000000000000000000000000000000000'X
               000216    +81         '00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000'X
               00023E    +121        '00000000000000110000'X
               000248    (3)         '                                                                                    R      '
               0002A2    +91         '        R    AA                         '
               000248  VALUE IN HEX  '00000000000000000000000000000000220000000000000000000100000480000400000000000000'X
               000270    +41         '00000001000000110000000000000000000000002003000000000000000030000000000000000000'X
               000298    +81         '001000990E000DFF000000000000001800100099B000B10818100000000000000000000000000000'X
               0002C0    +121        '0000000000000000000'X
               0002CA    (4)         '                                  R             A    0          A    0          TY       '
               000324    +91         '                                        '
               0002CA  VALUE IN HEX  '00000000000000000000000000000000000000000000000800000000000800000100099F0004B08000'X
               0002F2    +41         '000000000000003DC19EB70006F08000000000000000003DC19EB70006F04800000000000000000000'X
               00031A    +81         '00A3E8000011000000000000000000000000000000000000000000000190000000E000000000000'X
               000342    +121        '00000000000000000000'X
               00034C    (5)         '                                                                                    3 '
               0003A6    +91         '              CPF     *       T        '
               00034C  VALUE IN HEX  '00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000'X
               000374    +41         '00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000'X
               00039C    +81         '0000000000000000F3000001000000000000000000FF000000C3D7C600000000005CE00000000003C'X
```

Figura 120 (Parte 3 de 10). Ejemplo de un Vuelco con Formato COBOL

```
              0003C4    +121        '00A30000000000000000'X
              0003CE    (6)         '                                                                                '
              000428    +91         '                        SALESFILE  2A248'
              0003CE  VALUE IN HEX  '00000000000000000000000000000000000000000000000000000000000000000010000'X
              0003F6    +41         '00000000000000000000000000000000000000000000000000000000000000000000000000000000'X
              00041E    +81         '00000000000000000000001100000000000000000000000000000000E000000000001E2C1D3C5E2C6'X
              000446    +121        'C9D3C54000F2C1F2F4F8'X
              000450    (7)         '33FF03CF                                                    U      &                '
              0004AA    +91         '              A        T            D0'
              000450  VALUE IN HEX  'F3F3C6C6F0F3C3C600000000000E000100000001000000000000000000000000000000000000000'X
              000478    +41         '00000000000000000000000000000000000000000000000A40000000000050000000000000000'X
              0004A0    +81         '00000000000000000000000000000800000000000000003DC19EB7000A40000100A30019000E'X
              0004C8    +121        '4040404040404040C4F0'X
              0004D2    (8)         '1   <HHH               D12   <HHH               D15   <HHH               D15   <HHH       '
              00052C    +91         '        D22   <HHH               D99   <'
              0004D2  VALUE IN HEX  'F14444444C8888888C40404000010000000100000000040C4F1F24444444C8888888C404040000100'X
              0004FA    +41         '00000200000000040C4F1F54444444C8888888C40404000010000000300000000040C4F1F54444444C'X
              000522    +81         '8888888C40404000010000000400000000040C4F2F24444444C8888888C40404000010000000050000'X
              00054A    +121        '000040C4F9F94444444C'X
              000554    (9)         'HHH               D01   <HHH               D23   <HHH               D25   <HHH           '
              0005AE    +91         '   D88   <HHH               D99   <HHH '
              000554  VALUE IN HEX  '8888888C40404000010000000600000000040C4F0F14444444C8888888C40404000010000000070000'X
              00057C    +41         '000040C4F2F34444444C8888888C40404000010000000800000000040C4F2F54444444C8888888C40'X
              0005A4    +81         '40400001000000090000000040C4F8F84444444C8888888C40404000010000000A0000000040C4F9'X
              0005CC    +121        'F94444444C8888888C40'X
              0005D6    (10)        '            D22   <HHH               D01   <HHH               D66   <HHH               D2'
              000630    +91         '2   <HHH               D77   <HHH       '
              0005D6  VALUE IN HEX  '4040000100000000B0000000040C4F2F24444444C8888888C40404000010000000C0000000040C4F0'X
              0005FE    +41         'F14444444C8888888C40404000010000000D0000000040C4F6F64444444C8888888C404040000100'X
              000626    +81         '00000E0000000040C4F2F24444444C8888888C40404000010000000F0000000040C4F7F74444444C'X
              00064E    +121        '8888888C404040000100'X
              000658    (11)        '        H2500000000  <          '
              000658  VALUE IN HEX  '00001000000000040C8F2F5F0F0F0F0F0F0F0F040404C000100000011000000004040404040404040'X
              000680    +41         '40404040404040404040404040404040404040404040404040404040404040404040404040404040'X
              0006A8    +81         '40404040404040404040404040404040404040404040404040404040404040404040404040404040'X
              0006D0    +121        '4040404040404040404040'X
              00140E    (12-38)     ' '
              001490    (39)        '                                                                                '
              0014EA    +91         '                                          '
              001490  VALUE IN HEX  '40404040404040404040404040404040404040404040404040404040404040404040404040404040'X
              0014B8    +41         '40404040404040404040404040404040404040404040404040404040404040000000000000000000000000'X
              0014E0    +81         2 LINES OF ZEROES SUPPRESSED
                                    CANNOT DUMP - SPACE ADDRESSING OR BOUNDARY ALIGNMENT EXCEPTION
.DMPDEVN      0001E6  CHAR(10)      '          '                   '00000000000000000000'X
.DMPDIOF              NOT ADDRESSABLE
.DMPDRN               NOT ADDRESSABLE
.DMPDSEK              NOT ADDRESSABLE
.DMPDVNM      000144  CHAR(10)      'DATABASE  '
.DMPENT       000144  CHAR(130)     'DATABASE                              ?                                         '
              00019E    +91         '                                        '
              000144  VALUE IN HEX  'C4C1E3C1C2C1E2C5404000000000000000000000000302000E00450045004500450045006F0045'X
              00016C    +41         '0045004500450BFD00450045000D00110000000100000000000000000000000000000000000000'X
              000194    +81         2 LINES OF ZEROES SUPPRESSED
.DMPFBAC      0000B0  CHAR(32767)   'DBSALES     COBOLEX                              SALESFILE                      AR NU      '
              00010A    +91         '&              T                                        DATABASE                          '
              000164    +181        '     ?                                                                                 '
              0000B0  VALUE IN HEX  'C4C2E2C1D3C5E24040404040C3D6C2F3F8C5E7404040000000000000000000000000000000000000000'X
              0000D8    +41         '00000000000E0000E2C1D3C5E2C6C9D3C5400000100400000000000000000000000000000011C1'X
              000100    +81         'D900D5A400000000000005000000000000000000000000000000A300000019000003E00000000000'X
              000128    +121        '00000000000000001000000010200000000000000000000000000010001C4C1E3C1C2C1E2C540400000'X
              000150    +161        '0000000000000000302000E00450045004500450045006F00450045004500450BFD00450045'X
              000178    +201        '000D00110000000100000000000000000000000000000000000000000000000000000000000000'X
              0001A0    +241        '00000000000000000000000000000000'X
.DMPFBAT      0000FF  CHAR(2)       'AR'
.DMPFBCL      0000F9  BINARY(2)     0
.DMPFBCT      008038  BINARY(2)     CANNOT DUMP - SPACE ADDRESSING OR BOUNDARY ALIGNMENT EXCEPTION
.DMPFBDC      0000F2  BINARY(2)     0
.DMPFBDE      00803C  CHAR(50)      DIMENSION(32)
                                    CANNOT DUMP - SPACE ADDRESSING OR BOUNDARY ALIGNMENT EXCEPTION
.DMPFBDU      000101  CHAR(1)       ' '                                 '00'X
.DMPFBFN      0000B2  CHAR(10)      'SALES     '
.DMPFBH1              NOT ADDRESSABLE
.DMPFBH2              NOT ADDRESSABLE
.DMPFBIB      0000EA  BINARY(4)     4100
.DMPFBLN      0000BC  CHAR(10)      'COBOLEX   '
.DMPFBLO      000117  BINARY(2)     0
.DMPFBLP      000740  POINTER(SPP)  NULL
.DMPFBLS              NOT ADDRESSABLE
```

Figura 120 (Parte 4 de 10). Ejemplo de un Vuelco con Formato COBOL

```
.DMPFBL1       0000DC  BINARY(2)     14
.DMPFBL2       0000DE  BINARY(2)     0
.DMPFBMF       000123  CHAR(1)       ' '                              '00'X
.DMPFBMN       0000E0  CHAR(10)      'SALESFILE '
.DMPFBND       00803A  BINARY(2)     CANNOT DUMP - SPACE ADDRESSING OR BOUNDARY ALIGNMENT EXCEPTION
.DMPFBOB       0000EE  BINARY(4)     0
.DMPFBOF       00010D  CHAR(10)      '          '                     '00000000000000000000'X
.DMPFBOL       0000F4  CHAR(3)       '   '                           '000000'X
.DMPFBPO       00011F  BINARY(4)     992
.DMPFBQN       000124  CHAR(10)      '          '                     '00000000000000000000'X
.DMPFBRC       0000FB  BINARY(4)     17
.DMPFBRW       0000F7  BINARY(2)     0
.DMPFBSC       000102  CHAR(1)       'N'
.DMPFBSF       0000C6  CHAR(10)      '          '                     '00000000000000000000'X
.DMPFBSL       0000D0  CHAR(10)      '          '                     '00000000000000000000'X
.DMPFBSN       0000DA  BINARY(2)     0
.DMPFBTY       0000B0  CHAR(2)       'DB'
.DMPFBUF       000103  CHAR(10)      'U      &  '                     'A4000000000000500000'X
.DMPFBVL               NOT ADDRESSABLE
.DMPIOFB       0001C6  CHAR(32767)   '                    SALESFILE                                                                '
               000220    +91         '                                                                                            '
               00027A    +181        '                                  R              R    AA                    '
               0001C6  VALUE IN HEX  '0090000000000000000100000000000000000001E2C1D3C5E2C6C9D3C5400000000000000000000000'X
               0001EE    +41         '00000000000E0000000000000000000000000000000000000000000000000000000000000000000000000000'X
               000216    +81         '00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000'X
               00023E    +121        '00000000000000110000000000000000000000000000000000000000220000000000000000000010000'X
               000266    +161        '04800004000000000000000000010000001100000000000000000000000200300000000000000000'X
               00028E    +201        '3000000000000000000000001000990E000DFF000000000000001800100099OB000B10818100000000'X
               0002B6    +241        '00000000000000000000000000000000'X
.DMPIOFS       0001C6  CHAR(144)     '                    SALESFILE                                                                '
               000220    +91         '                                                    '
               0001C6  VALUE IN HEX  '0090000000000000000100000000000000000001E2C1D3C5E2C6C9D3C5400000000000000000000000'X
               0001EE    +41         '00000000000E0000000000000000000000000000000000000000000000000000000000000000000000000000'X
               000216    +81         '00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000'X
               00023E    +121        '00000000000000110000000000000000000000000000000000000000'X
.DMPKYLN               NOT ADDRESSABLE
.DMPNDEV       000142  BINARY(2)     1
.DMPOFBS       0000B0  CHAR(17126)   DIMENSION(2)
                                     CANNOT DUMP - SPACE ADDRESSING OR BOUNDARY ALIGNMENT EXCEPTION
.DMPRCD                NOT ADDRESSABLE
.DMPRCDN               NOT ADDRESSABLE
.DMPRDUP               NOT ADDRESSABLE
.DMPRFMT       0001DA  CHAR(10)      'SALESFILE '
.DMPRRN                NOT ADDRESSABLE
.DMPSRC                NOT ADDRESSABLE
.EXCODE        000D30  CHAR(1)       ' '                              '00'X
.EXMSGID       000D35  CHAR(4)       '    '                           '00000000'X
.EXPARMS       000D30  CHAR(12)      '           '                     '000000000000000000000000'X
.EXPTR         000D40  POINTER(SPP)  SPACE OFFSET    3376     '00000D30'X
                                     OBJECT    PSSA
.FCLPP         0006DF  CHAR(3)       '   '                            '000000'X
.FCLSTC        0006DC  CHAR(12)      '           '                     '000000000000000000000000'X
.FCLSTC#       0006D0  CHAR(12)      '   PU      '                     '0D0003D7E4400600020001FF'X
.FCLSTP        0006FF  CHAR(21)      '                  '              '000000000000000000000000000000000000000000'X
.FCLSTP#       0006EA  CHAR(21)      '                   '             '09000200000A000200000B000200000C00020000FF'X
.FCPARM        0005B0  CHAR(22)      '                    '
.FCPARMP       0005D0  POINTER(SPP)  SPACE OFFSET    1456     '000005B0'X
                                     OBJECT    PSSA
.FCPTR         0005A0  POINTER(SYP)  OBJECT    QLREXHAN
                                     CONTEXT   QSYS
.FIB           0008A0  CHAR(32767)   'FILE-1                                  0400                                                 '
               0008FA    +91         '                                                                        A                   '
               000954    +181        '                                                                                  '
               0008A0  VALUE IN HEX  'C6C9D3C560F1404040404040404040404040404040404040404040404040404000800300000 1F0F4F0F0'X
               0008C8    +41         '000400000000000000000000000000000000040404040404040404040000000001000100010000C00'X
               0008F0    +81         '8000000000000000003DC00036000AE000000000000000000000000000000000000000000000000000'X
               000918    +121        '00000000000000000000000000000000000000000000000000000000800000000000000003DC19EB7000890'X
               000940    +161        '4040404040404040404040001000000000000000000000000000000000000000000000000000000000000000'X
               000968    +201        2 LINES OF ZEROES SUPPRESSED
.FIB#OPT       00062C  CHAR(8)       '        '                       '0000000000000000'X
.FIB#OP1               NOT ADDRESSABLE
.FIBACC        0008E9  BINARY(2)     1
.FIBACQ        0006E8  BINARY(2)     0
.FIBACTL       0006C0  CHAR(8)       '        '                       '160004000000000FF'X
.FIBALT        0008BE  CHAR(1)       ' '                              '00'X
.FIBCA         000955  CHAR(22)      '                      '         '00000000000000000000000000000000000000000000'X
.FIBCFMT       000961  CHAR(10)      '          '                     '00000000000000000000'X
.FIBCFS        0008C4  CHAR(2)       '04'
```

Figura 120 (Parte 5 de 10). Ejemplo de un Vuelco con Formato COBOL

```
.FIBCFS1       0008C4  CHAR(1)       '0'
.FIBCFS2       00094C  CHAR(4)       '    '                          '00000000'X
.FIBCHAN       000920  POINTER(SPP)  NULL
.FIBCKID       000955  ZONED(2,0)    **                              '0000'X
.FIBCOP        0008C0  CHAR(4)       '    '                          '03000001'X
.FIBCRP        0008E4  CHAR(1)       ' '                             '00'X
.FIBCTID       000957  CHAR(10)      '          '                    '00000000000000000000'X
.FIBCTL        000750  POINTER(SPP)  NULL
.FIBCUR        0008C0  CHAR(6)       '    04'                        '03000001F0F4'X
.FIBCURK       000634  CHAR(123)     '                                                                                                  '
               00068E    +91         '                                  '
               000634  VALUE IN HEX  '000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000'X
               00065C    +41         3 LINES OF ZEROES SUPPRESSED
.FIBDEVC       0008E5  BINARY(2)     0
.FIBDEVI       00094A  BINARY(2)     1
.FIBDEVN       000940  CHAR(10)      '          '
.FIBFLGS       0008BF  CHAR(1)       ' '                             '80'X E
.FIBFMT        0008DA  CHAR(10)      '          '
.FIBFN         0008A0  CHAR(30)      'FILE-1                        '
.FIBK#LN       00062D  BINARY(2)     0
.FIBK#R#       00062F  BINARY(4)     0
.FIBK#RK       000631  BINARY(2)     0
.FIBK#TP       00062C  CHAR(1)       ' '                             '00'X
.FIBKCGK       000600  CHAR(8)       '        '                      '0800040000000009'X
.FIBKCGR       000608  CHAR(8)       '        '                      '020004000000000FF'X
.FIBKCPD       000610  CHAR(8)       '        '                      '040004000000000FF'X
.FIBKCTL       000618  BINARY(2)     0
.FIBKDLN       000628  BINARY(2)     2
.FIBKDM#       00062A  BINARY(2)     0
.FIBKDTP       000627  CHAR(1)       ' '                             '0F'X
.FIBKEY        0008CA  BINARY(4)     0
.FIBKFLN       00061B  BINARY(2)     10
.FIBKFMT       00061D  CHAR(10)      '          '
.FIBKFTP       00061A  CHAR(1)       ' '                             '01'X
.FIBKKEY       000636  CHAR(121)     '                                                                                                  '
               000690    +91         '                                  '
               000636  VALUE IN HEX  '000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000'X
               00065E    +41         3 LINES OF ZEROES SUPPRESSED
.FIBKKLN       000634  BINARY(2)     0
.FIBKKTP       000633  CHAR(1)       ' '                             '00'X
.FIBKLEN       0008CE  BINARY(2)     0
.FIBKSTC       00062C  CHAR(1)       ' '                             '00'X
.FIBKSTE       000631  CHAR(1)       ' '                             '00'X
.FIBKSTL       00062D  BINARY(2)     0
.FIBKSTT       00062F  BINARY(2)     0
.FIBLBO        0008D0  BINARY(2)     0
.FIBLFT        0008CC  BINARY(2)     0
.FIBLIN        0008CA  BINARY(2)     0
.FIBLINE       0008D2  BINARY(2)     0
.FIBLTO        0008CE  BINARY(2)     0
.FIBMBRN       0009D6  CHAR(10)      'SALESFILE '
.FIBOFMT       0009CC  CHAR(10)      '          '                    '00000000000000000000'X
.FIBOFS        0008C6  CHAR(2)       '00' F
.FIBOFS1       0008C6  CHAR(1)       '0'
.FIBOKEY       000953  CHAR(121)     '                                                                                                  '
               0009AD    +91         '                                  '
               000953  VALUE IN HEX  '000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000'X
               00097B    +41         3 LINES OF ZEROES SUPPRESSED
.FIBOKLN       000951  BINARY(2)     0
.FIBOLDK       000951  CHAR(123)     '                                                                                                  '
               0009AB    +91         '                                    '
               000951  VALUE IN HEX  '000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000'X
               000979    +41         3 LINES OF ZEROES SUPPRESSED
.FIBOP         0007A1  CHAR(4)       '    '                          '03000001'X
.FIBOP1        0007A1  CHAR(1)       ' '                             '03'X
.FIBOP2        0007A2  CHAR(1)       ' '                             '00'X
.FIBOP3        0007A3  CHAR(1)       ' '                             '00'X
.FIBOP4        0007A4  CHAR(1)       ' '                             '01'X
.FIBORG        0008E7  BINARY(2)     1
.FIBORRN       000951  BINARY(4)     0
.FIBOTP        0008EB  BINARY(2)     1
.FIBPTR        0003B0  POINTER(SPP)  SPACE OFFSET    2208     '000008A0'X
                                     OBJECT    PSSA
.FIBP1         000930  POINTER(SPP)  SPACE OFFSET    1200     '000004B0'X
                                     OBJECT    SALES     COBOLEX   SALESFILE
.FIBRECS       0008ED  BINARY(2)     12
.FIBREL                NOT ADDRESSABLE
.FIBRLPT       0006B0  POINTER(SPP)  NULL
```

Figura 120 (Parte 6 de 10). Ejemplo de un Vuelco con Formato COBOL

```
.FIBROLC              NOT ADDRESSABLE
.FIBROLE              NOT ADDRESSABLE
.FIBROLL              NOT ADDRESSABLE
.FIBRSL               NOT ADDRESSABLE
.FIBRVAL              NOT ADDRESSABLE
.FIBSPC       0008CA  CHAR(14)      '              '               '00000000000000000000000000'X
.FIBTAPE      0006C8  CHAR(8)       '        '                     '11000400000000FF'X
.FIBTLEN      0006CB  BINARY(4)     0
.FIBUBTO      000900  POINTER(IP)   NULL
.FIBUFCB      0008F0  POINTER(SPP)  SPACE OFFSET    2528     '000009E0'X
                                    OBJECT    PSSA
.FIBURTN      000910  POINTER(SPP)  NULL
.FIBUSAV      0005F0  POINTER(IP)   NULL
.FIBUSE#      0008D8  BINARY(2)     0
.FIBVERB      0008C8  BINARY(2)     4
.FSKA         00070C  BINARY(2)     0
.FSKB         000702  BINARY(2)     0
.FSPA         000711  BINARY(2)     0
.FSPB         000707  BINARY(2)     0
.FSTKS        0006E5  BINARY(2)     0
.FWTRCD       0006C3  BINARY(4)     0
.F01ACC       0008E9  BINARY(2)     1
.F01ALTS      0008BE  CHAR(1)       ' '                           '00'X
.F01CFS2      00094C  CHAR(4)       '    '                        '00000000'X
.F01CHAN      000920  POINTER(SPP)  NULL
.F01COP       0008C0  CHAR(4)       '    '                        '03000001'X
.F01CRP       0008E4  CHAR(1)       ' '                           '00'X
.F01CUR       0008C0  CHAR(6)       '    04'                      '03000001F0F4'X
.F01DEVC      0008E5  BINARY(2)     0
.F01DEVI      00094A  BINARY(2)     1
.F01DEVN      000940  CHAR(10)      '          '
.F01FLGS      0008BF  CHAR(1)       ' '                           '80'X
.F01FMT       0008DA  CHAR(10)      '          '
.F01FN        0008A0  CHAR(30)      'FILE-1                      '
.F01MBRN      0009D6  CHAR(10)      'SALESFILE '
.F01OFMT      0009CC  CHAR(10)      '          '                  '00000000000000000000'X
.F01OFS       0008C6  CHAR(2)       '00'
.F01OKLN      000951  BINARY(2)     0
.F01OLDK      000953  CHAR(121)     '                                                                                                  '
              0009AD    +91         '                                '
              000953  VALUE IN HEX  '00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000'X
              00097B    +41         3 LINES OF ZEROES SUPPRESSED
.F01ORG       0008E7  BINARY(2)     1
.F01OTP       0008EB  BINARY(2)     1
.F01P1        000930  POINTER(SPP)  SPACE OFFSET    1200     '000004B0'X
                                    OBJECT    SALES     COBOLEX   SALESFILE
.F01RECS      0008ED  BINARY(2)     12
.F01SPC       0008CA  CHAR(14)      '              '               '00000000000000000000000000'X
.F01UBTO      000900  POINTER(IP)   NULL
.F01UFCB      0008F0  POINTER(SPP)  SPACE OFFSET    2528     '000009E0'X
                                    OBJECT    PSSA
.F01URTN      000910  POINTER(SPP)  NULL
.F01USE#      0008D8  BINARY(2)     0
.F01VERB      0008C8  CHAR(2)       '  '                          '0004'X
.IOCPTR       000860  POINTER(SPP)  SPACE OFFSET    1952     '000007A0'X
                                    OBJECT    PSSA
.IOEPTR       000840  POINTER(SYP)  OBJECT    QDBGETM
                                    CONTEXT   QSYS
.IOFDBEX      000780  POINTER(SPP)  NULL
.IOOPTR       000850  POINTER(SPP)  SPACE OFFSET    2240     '000008C0'X
                                    OBJECT    PSSA
.IORTN        0003A0  POINTER(IP)   STMT 48                       INSTR # 0000004F
                                    OBJECT    XMPLDUMP
                                    CONTEXT   QTEMP
.IP00001      000C90  POINTER(IP)   STMT 52                       INSTR # 00000065
                                    OBJECT    XMPLDUMP
                                    CONTEXT   QTEMP
.MAINRTN      000440  POINTER(SYP)  OBJECT    QLRMAIN
                                    CONTEXT   QSYS
.MGT          000230  CHAR(16)      'COBOL MGT 00.0LR' G
.MGTBIN8      000347  CHAR(8)       '        '                    '0000000000000000'X
.MGTB81       000347  BINARY(4)     0
.MGTCNTR      0002C0  BINARY(4)     DIMENSION(20)
              00030C    (40-20)     0
.MGTCPGM      000390  POINTER(SYP)  NULL
.MGTDBUG      000328  CHAR(1)       '0'
.MGTEXCP      00031C  CHAR(7)       '       ' H
.MGTFIB       000250  POINTER(SPP)  SPACE OFFSET    2208     '000008A0'X
```

Figura 120 (Parte 7 de 10). Ejemplo de un Vuelco con Formato COBOL

```
                                    OBJECT    PSSA
.MGTFUNC        000345  BINARY(2)   2
.MGTIND         000323  CHAR(1)     DIMENSION(32)
                000329    (1-7)     '0'
                00032A    (8)       '1'
                000342    (9-32)    '0'
.MGTINVC        00031A  BINARY(2)   3 I
.MGTLIB         00041A  CHAR(10)    'QTEMP     '
.MGTMSGI        0003F2  CHAR(7)     '       '                       '00000000000000'X
.MGTMSGN        0003F0  BINARY(2)   0
.MGTMSGR        0003E0  POINTER(SPP) NULL
.MGTMSGS        0003C0  POINTER(IP) NULL
.MGTMSGT        0003D0  POINTER(SPP) NULL
.MGTNAME        000310  CHAR(10)    'XMPLDUMP  '
.MGTNEXT        000240  POINTER(SPP) NULL
.MGTOSZ         000323  CHAR(1)     '0'
.MGTOVFL        000325  CHAR(1)     '0'
.MGTPACK        00034F  PACKED(31,0) ****************************** '00000000000000000000000000000000'X
.MGTPARM        000400  POINTER(SPP) NULL
.MGTPASA        000270  POINTER(SPP) SPACE OFFSET    5760     '00001680'X
                                    OBJECT    PASA
.MGTPASC        000270  CHAR(16)    '                '              '80000000000000000003DC00037001780'X
.MGTPCS         000370  POINTER(SPP) NULL
.MGTPFM         000327  CHAR(1)     '0'
.MGTPGM         0002A0  POINTER(SYP) OBJECT    XMPLDUMP J
                                    CONTEXT   QTEMP
.MGTPGT         000260  POINTER(SPP) SPACE OFFSET    5952     '00001740'X
                                    OBJECT    PSSA
.MGTPLVL        000361  BINARY(2)   0
.MGTPROG        000410  CHAR(10)    'XMPLDUMP  '
.MGTPTP         000380  POINTER(SPP) SPACE OFFSET    2864     '00000B30'X
                                    OBJECT    PSSA
.MGTPTR         000460  POINTER(SPP) SPACE OFFSET    560     '00000230'X
                                    OBJECT    PSSA
.MGTRST         0002B0  POINTER(IP) NULL
.MGTSEG         00035F  BINARY(2)   0
.MGTSEPT        000280  POINTER(SPP) SPACE OFFSET    0     '00000000'X
                                    OBJECT    QINSEPT
                                    CONTEXT   QSYS
.MGTSOSZ        000324  CHAR(1)     '0'
.MGTSPCD        000329  CHAR(1)     '0'
.MGTSW          000343  CHAR(1)     ' '                              '80'X
.MGTTYPE        000344  CHAR(1)     'I'
.MGTUPTR        000290  POINTER(SPP) SPACE OFFSET    1984     '000007C0'X
                                    OBJECT    E34       PGMRS     011111
.MGT9001        00032A  CHAR(1)     '1'
.NULLCL         0007A0  CHAR(1)     ' '                              'FF'X
.ODPBPTR        000760  POINTER(SPP) SPACE OFFSET    0     '00000000'X
                                    OBJECT    SALES     COBOLEX   SALESFILE
.ODPDBAS        000890  POINTER(SPP) SPACE OFFSET    704     '000002C0'X
                                    OBJECT    SALES     COBOLEX   SALESFILE
.ONSAVE         0004D0  CHAR(32)    '                                '
                0004D0  VALUE IN HEX '00000000000000000000000000000000000000000000000000000000000000000000000000000000'X
.PBPDUM         0008A0  POINTER(IP) NULL
.PBP0003        000B40  POINTER(IP) STMT 42                          INSTR # 00000030
                                    OBJECT    XMPLDUMP
                                    CONTEXT   QTEMP
.PERFCTR        000550  BINARY(2)   1
                                     K
.PGT            001740  CHAR(32767) 'PGT 00.0                                        0100000000000000000000000000000     (   .'
                00179A    +91       '   QTEMP                            XMPLDUMP                                .              '
                0017F4    +181      '                                                                                    '
                001740  VALUE IN HEX 'D7C7E340F0F04BF040404040404040408000000000000000003DC00036000330000000000000000'X
                001768    +41       '00000000000000000F0F1F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0F0'X
                001790    +81       '00030000004D0000004B010401D8E3C5D4D7404040404040404040404040404040404040404040'X
                0017B8    +121      '4040400201E2C1D4D7C4E4D4D7404040404040404040404040404040404040404040404040'X
                0017E0    +161      '0000004B000000000100000000000000000000000000000000000000000000000000000000000000'X
                001808    +201      '00000000000000000000000000000000000000000000000000000000000000000000000000002900'X
                001830    +241      '000000010000000000000000000000'X
.PGTIND         001770  CHAR(1)     DIMENSION(32)
                001770    (1)       '0'
                001771    (2)       '1'
                00178F    (3-32)    '0'
.PGTINVC        001790  BINARY(2)   3 L
.PGTLVL         001740  CHAR(16)    'PGT 00.0        '
.PGTMGTL        001760  POINTER(SPP) NULL
.PGTMGT1        001750  POINTER(SPP) SPACE OFFSET    560     '00000230'X
                                    OBJECT    PSSA
```

Figura 120 (Parte 8 de 10). Ejemplo de un Vuelco con Formato COBOL

```
.PNP0003       000B40  CHAR(48)      '           T              T                   '
               000B40  VALUE IN HEX  '40000000000000000003D03E33C00061A40000000000000000003D03E33C00085600030000000000000'X
               000B68    +41         '0000000000000000'X
.PTABLE        000B30  CHAR(16)      'PT 01.0    0    '              'D7E340F0F14BF000100001F000000000'X
.PTHSIZE       000B37  BINARY(2)     16
.PTNUM         000B39  BINARY(2)     1
.PTSEG         000B3B  CHAR(1)       '0'
.P020001       000CD0  PACKED(2,0)   25
.QLRDISP       000CC0  POINTER(SYP)  OBJECT    QLRADRTN
                                     CONTEXT   QSYS
.QLRXHAN       000D50  POINTER(SYP)  OBJECT    QLREXHAN
                                     CONTEXT   QSYS
.RCDFDBK       000880  POINTER(SPP)  NULL
.RETURNP       000820  POINTER(IP)   NULL
.RTNPTR        000560  POINTER(SPP)  SPACE OFFSET    5760     '00001680'X
                                     OBJECT    PASA
.RUNRTN        000830  POINTER(IP)   NULL
.SAVKKEY       0007A7  CHAR(121)     '                                                                                    '
               000801    +91         '                                '
               0007A7  VALUE IN HEX  '00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000'X
               0007CF    +41         3 LINES OF ZEROES SUPPRESSED
.SEPTP         000430  POINTER(SPP)  NULL
.SIZERP        000590  POINTER(SPP)  SPACE OFFSET    1376     '00000560'X
                                     OBJECT    PSSA
.SUBLEN        000D10  BINARY(2)     0
.SUBNAME       000D12  CHAR(10)      '          '                   '00000000000000000000'X
.SUBTXT        000D10  CHAR(12)      '            '                 '000000000000000000000000'X
.SUBTXTA       000D20  POINTER(SPP)  SPACE OFFSET    3344     '00000D10'X
                                     OBJECT    PSSA
.SUB2FST       000D62  CHAR(10)      '          '                   '00000000000000000000'X
.SUB2LEN       000D60  BINARY(2)     0
.SUB2SCD       000D6C  CHAR(10)      '          '                   '00000000000000000000'X
.SUB2TP        000D80  POINTER(SPP)  SPACE OFFSET    3424     '00000D60'X
                                     OBJECT    PSSA
.SUB2TXT       000D60  CHAR(22)      '                      '        '0000000000000000000000000000000000000000000000'X
.TC00001       000BD4  CHAR(2)       '  '                            '0000'X
.TC00002       000BD6  CHAR(2)       '  '                            '0000'X
.TMPN001       000CD0  CHAR(32)      ' ¬                              '
               000CD0  VALUE IN HEX  '025F00000000000000000000000000000000000000000000000000000000000000000000000000000000'X
.T000001               NOT ADDRESSABLE
.T000002       000B70  PACKED(7,2)   88888.88
.T000003       000B70  PACKED(7,2)   88888.88
.T1            0004F0  CHAR(32)      '                                '
               0004F0  VALUE IN HEX  '00000000000000000000000000000000000000000000000000000000000000000000000000000000'X
.T2            000510  CHAR(32)      '                                '
               000510  VALUE IN HEX  '00000000000000000000000000000000000000000000000000000000000000000000000000000000'X
.T3            000530  CHAR(32)      '                                '
               000530  VALUE IN HEX  '00000000000000000000000000000000000000000000000000000000000000000000000000000000'X
.UCB           0009E0  CHAR(32767)   '          A              A              A              A              A    W          '
               000A3A    +91         '                                    SALES       *LIBL                                0100'
               000A94    +181        '   ¢                                          "  ¬       0031111108006222'
               0009E0  VALUE IN HEX  '8000000000000000003DC19EB70003E08000000000000000003DC19EB70008908000000000000000'X
               000A08    +41         '003DC19EB70008908000000000000000003DC19EB70004908000000000000000003DC19EB70005A6'X
               000A30    +81         '00000000000000000000000000000000000000000000000000000000000000000000000000000000'X
               000A58    +121        '0000000000000000E2C1D3C5E24040404040FFB55CD3C9C2D34040404040FFB940404040404040040'X
               000A80    +161        '4040404040404040404040400001012 0F0F1F0F00001654A2000000020000000000000000000000'X
               000AA8    +201        '000000000000000000001000C00148000038000 3C80003A8000000006007FFF015F030F0000000000'X
               000AD0    +241        'F0F0F3F1F1F1F1F1F0F8F0F0F6F2F2F2'X
.UCBCLMG       000A98  CHAR(1)       ' '                              '20'X
.UCBEDOP       000A99  CHAR(1)       ' '                              '00'X
.UCBFILE       000A60  CHAR(10)      'SALES     '
.UCBFLGS       000A8E  CHAR(2)       '  '                             '0120'X
.UCBFLG1       000A8E  CHAR(1)       ' '                              '01'X
.UCBFLG2       000A8F  CHAR(1)       ' '                              '20'X
.UCBIBR@       0009F0  POINTER(SPP)  SPACE OFFSET    1200     '000004B0'X
                                     OBJECT    SALES     COBOLEX   SALESFILE
.UCBINDX       000A8C  BINARY(2)     1
.UCBIOF@       000A20  POINTER(SPP)  SPACE OFFSET    454      '000001C6'X
                                     OBJECT    SALES     COBOLEX   SALESFILE
.UCBLAST       000A82  CHAR(10)      '          '
.UCBLBID       000A6A  BINARY(2)     -75
.UCBLIB        000A6C  CHAR(10)      '*LIBL     '
.UCBLIBS       000A6A  CHAR(12)      '  *LIBL     '                 'FFB55CD3C9C2D34040404040'X
.UCBMBID       000A76  BINARY(2)     -71
.UCBMBRS       000A76  CHAR(12)      '            '                 'FFB9404040404040404040'X
.UCBMLIB       000A78  CHAR(10)      '          '
.UCBNXT@       000A30  POINTER(SPP)  NULL
.UCBOBR@       000A00  POINTER(SPP)  SPACE OFFSET    1200     '000004B0'X
```

Figura 120 (Parte 9 de 10). Ejemplo de un Vuelco con Formato COBOL

```
                                    OBJECT    SALES     COBOLEX   SALESFILE
.UCBODP@       0009E0  POINTER(SPP)  SPACE OFFSET    0     '00000000'X
                                    OBJECT    SALES     COBOLEX   SALESFILE
.UCBOPF@       000A10  POINTER(SPP)  SPACE OFFSET    176     '000000B0'X
                                    OBJECT    SALES     COBOLEX   SALESFILE
.UCBPARM       000AB0  BINARY(2)     1
.UCBRLEN       000AB2  BINARY(2)     12
.UCBRLVR       000A90  CHAR(4)       '0100'
.UCBSEP@       000A40  POINTER(SPP)  NULL
.UFCBPTR       000720  POINTER(SPP)  SPACE OFFSET    2528     '000009E0'X
                                    OBJECT    PSSA
.UFLGSAV       000CF0  CHAR(2)       '  '                              '0000'X
.USERTN        0005E0  POINTER(IP)   NULL
.USEWRK@       000D90  POINTER(SPP)  NULL
.U01CLMG       000A98  CHAR(1)       ' '                               '20'X
.U01FLGS       000A8E  CHAR(2)       '  '                              '0120'X
.U01IBF@       0009F0  POINTER(SPP)  SPACE OFFSET    1200     '000004B0'X
                                    OBJECT    SALES     COBOLEX   SALESFILE
.U01OBF@       000A00  POINTER(SPP)  SPACE OFFSET    1200     '000004B0'X
                                    OBJECT    SALES     COBOLEX   SALESFILE
.U01SEQO       000ABF  CHAR(1)       ' '                               '80'X
.U01UFCB       0009E0  POINTER(SPP)  SPACE OFFSET    0     '00000000'X
                                    OBJECT    SALES     COBOLEX   SALESFILE
.VALT001       000B70  CHAR(32)      'HHH                               '
               000B70  VALUE IN HEX  '8888888F000000000000000000000000000000000000000000000000000000000000'X
.V005622       000662  CHAR(1)       '5'
.WCBCNLS       0007D0  CHAR(1)       '0'
.WCBJDAT       0007D1  CHAR(7)       '0890623'  M
.WCBLURC       0007C0  BINARY(2)     2
.WCBPINF       0007C2  BINARY(2)     0
.WCBSWTC       0007D8  CHAR(8)       '00000000'
.WCBUDTA       0007C0  CHAR(32767)   '                0089062300000000                                                                  '
               00081A    +91         2 LINES OF BLANKS SUPPRESSED
               0007C0  VALUE IN HEX  '00020000000000000000000000000000F0F0F8F9F0F6F2F3F0F0F0F0F0F0F0F00000000000000000'X
               0007E8    +41         6 LINES OF ZEROES SUPPRESSED
.WCBURC        0007CE  CHAR(2)       '  '                              '0000'X
.WCBU0         0007D8  CHAR(1)       '0'
.WCBU1         0007D9  CHAR(1)       '0'
.WCBU2         0007DA  CHAR(1)       '0'
.WCBU3         0007DB  CHAR(1)       '0'
.WCBU4         0007DC  CHAR(1)       '0'
.WCBU5         0007DD  CHAR(1)       '0'
.WCBU6         0007DE  CHAR(1)       '0'
.WCBU7         0007DF  CHAR(1)       '0'
END-FLAG       000B28  CHAR(1)       ' '  N
END-OF-INPUT   000B29  CHAR(1)       'Y'
FILE-1         000660  CHAR(12)      'H2500000000 '
FILLER         00066B  CHAR(1)       ' '
FILLER         000B13  CHAR(3)       '   '
FILLER         000B04  CHAR(3)       '   '
FILLER         000AF0  CHAR(8)       'TOTALS: '
R-AREA-CODE    000661  ZONED(2,0)    25
R-NORTH-EAST   000AC9  PACKED(2,0)   30
R-NORTH-EAST   000AC7  PACKED(2,0)   15
R-SALES-CAT-1  000663  PACKED(7,2)
                     **INVALID DATA  'F0F0F0F0'X  O
R-SALES-CAT-2  000667  PACKED(7,2)
                     **INVALID DATA  'F0F0F0F0'X
R-TYPE         000660  CHAR(1)       'H'
RECORD-1       000660  CHAR(12)      'H2500000000 '
W-CAT-1        000AD0  ZONED(10,2)   311111.08
W-CAT-2        000ADA  ZONED(10,2)   622222.16
W-EDIT-TOTAL   000B16  CHAR(12)      '          '
W-EDIT-VALUES  000AF0  CHAR(50)      'TOTALS:                                          '
W-EDIT-1       000AF8  CHAR(12)      '           '
W-EDIT-2       000B07  CHAR(12)      '           '
W-SALES-VALUES 000AD0  CHAR(30)      '003111110800622222160093333324'
W-TOTAL        000AE4  ZONED(10,2)   933333.24
STATIC STORAGE FOR PROGRAM XMPLDUMP.QTEMP        BEGINS AT OFFSET 000230 IN THE PROGRAM STATIC STORAGE AREA (PSSA)
AUTOMATIC STORAGE FOR PROGRAM XMPLDUMP.QTEMP        BEGINS AT OFFSET 0016C0 IN THE PROGRAM AUTOMATIC STORAGE AREA (PASA)
```

Figura 120 (Parte 10 de 10). Ejemplo de un Vuelco con Formato COBOL

Bibliografía

Para obtener información acerca de los temas relacionados con la programación COBOL/400 en el sistema AS/400, consulte las siguientes publicaciones AS/400 IBM:

- *Comunicaciones: Guía para la Gestión, SC10-8974 (SC41-0024)*
  Título abreviado: *Guía para la Gestión de Comunicaciones*
- *Device Configuration Guide, SC41-8106*
  Título abreviado: *Device Configuration Guide*
- *Instalación de Software*, SC10-9279 (SC41-3120)
  Título abreviado: *Instalación de Software*
- *System Programmer's Interface Reference, SC41-8223*
  Título abreviado: *System Programmer's Interface Reference*
- *Guía para la Base de Datos, SC10-9009 (SC41-9659)*
  Título abreviado: *DDS Reference*
- *Data Description Specifications Coding Form, SX41-9891*
  Título abreviado: *DDS Coding Form*
- *Communications: Intersystem Communications Function Programmer's Guide, SC41-9590*
  Título abreviado: *ICF Programmer's Guide*
- *Operación del Sistema*, SC10-9280 (SC41-3203)
  Título abreviado: *Operación del Sistema*
- *Guía Básica para la Seguridad, SC10-9238 (SC41-0047)* y *Seguridad Manual de Consulta, SC10-8981 (SC41-8083)*
  Títulos abreviados: *Guía Básica para la Seguridad* y *Seguridad Manual de Consulta*
- *Distributed Data Management Guide, SC41-9600*
  Título abreviado: *DDM Guide*
- *Guía para la Base de Datos, SC10-9009 (SC41-9659)*
  Título abreviado: *Guía para la Base de Datos*
- *Utilidades: Programa de Utilidad para la Definición Interactiva de Datos (IDDU) Guía del Usuario, SC10-9007 (SC41-9657)*
  Título abreviado: *IDDU Guía del Usuario*
- *System Programmer's Interface Reference, SC41-8223*
  Título abreviado: *System Programmer's Interface Reference*
- *CICS/400 Application Programming Guide*, SC33-0822
  Título abreviado: CICS/400 Application Programming Guide
- *Communications: Remote Work Station Guide, SC41-0002*
  Título abreviado: *Remote Work Station Guide*
- *Advanced Backup and Recovery Guide, SC41-8079*
  Título abreviado: *Advanced Backup and Recovery Guide*
- *Programación: Lenguaje de Control Guía del Programador, SC10-8977 (SC41-8077)*
  Título abreviado: *CL Guía del Programador*
- *Guía para Nuevos Usuarios, SC10-8881 (SC41-8211)*
  Título abreviado: *Guía para Nuevos Usuarios*
- *Programming: Control Language Reference, SC41-0030*
  Título abreviado: *CL Reference*
- *Guía de Publicaciones, GC10-9237 (GC41-9678)*
  Título abreviado: *Guía de Publicaciones*
- *Programación: Guía para la Gestión de Trabajos, SC10-8978 (SC41-8078)*
  Título abreviado: *Guía para la Gestión de Trabajos*
- *SAA* Lenguaje de Consulta Estructurada SQL/400 Manual de Consulta, SC10-8997 (SC41-9608)*
  Título abreviado: *SQL/400* Manual de Consulta*
- *Guía para la Gestión de Datos, SC10-9008 (SC41-9658)*
  Título abreviado: *Guía para la Gestión de Datos*
- *COBOL/400 Reference*, SC09-1813
  Título abreviado: *COBOL/400 Reference*
- *American National Standard Programming Language COBOL, ANSI X3.23-1985, ISO 1989-1985*
  Título abreviado: *American National Standard Programming Language COBOL, ANSI X3.23-1985, ISO 1989-1985*

Para obtener más información sobre la Interfaz Común de Programación (CPI) de COBOL, consulte la siguiente publicación:

- *Systems Application Architecture Common Programming Interface COBOL Reference*, SC26-4354.

Glosario de Abreviaturas

Abreviatura	Significado	Explicación
Appl Dev Tools	Herramientas de desarrollo de aplicaciones (Application Development Tools)	Consta de programas para el sistema AS/400, como la Ayuda para el Diseño de Pantallas (SDA) y el Programa de Utilidad para la Entrada del Fuente (SEU).
ANSI	American National Standards Institute	Organización formada por fabricantes, consumidores y grupos de intereses generales, que establece los procedimientos por los que las organizaciones acreditadas crean y mantienen unos estándares industriales voluntarios en los Estados Unidos.
ASCII	American National Standard Code for Information Interchange (Código estándar americano para el intercambio de información).	Código desarrollado por el American National Standards Institute para el intercambio de información entre sistemas de proceso de datos, sistemas de comunicaciones de datos y equipos asociados. El juego de caracteres ASCII consta de caracteres de 8 bits, que a su vez constan de caracteres de control de 7 bits y de caracteres simbólicos, además de un bit de comprobación de paridad.

Abreviatura	Significado	Explicación
CICS	Customer Information Control Service (Servicio de control de información al cliente).	Programa bajo licencia IBM que habilita las transacciones entradas en las estaciones de trabajo remotas para que los programas de aplicación escritos por el usuario las procesen simultáneamente. Este programa con licencia incluye funciones para la creación, utilización y mantenimiento de bases de datos, así como para la comunicación con CICS o cualquier otro sistema operativo.
CL	Lenguaje de control (Control Language)	Conjunto de todos los mandatos con los que un usuario solicita funciones del sistema.

Abreviatura	Significado	Explicación
DBCS	Juego de Caracteres de Doble Byte (Double-Byte Character Set)	Juego de caracteres en el que cada carácter está representado por 2 bytes. Los idiomas como el japonés, chino y coreano, que contienen más símbolos de los que pueden representarse mediante 256 puntos de código, necesitan juegos de caracteres de doble byte. Dado que cada carácter necesita 2 bytes, el teclado, la visualización en pantalla y la impresión de caracteres DBCS precisan un hardware y unos programas que den soporte a DBCS. El sistema da soporte a cuatro juegos de caracteres de doble byte: japonés, coreano, chino simplificado y chino tradicional. Compárese con Juego de Caracteres de un Solo Byte.

Abreviatura	Significado	Explicación
DDM	Distributed Data Management (Gestión de Datos Distribuidos)	Función del sistema operativo que permite que un programa de aplicación o un usuario de un sistema utilicen archivos de datos almacenados en sistemas remotos. Los sistemas deben estar conectados por una red de comunicaciones; asimismo, los sistemas remotos deben utilizar DDM.
DDS	Especificaciones de descripción de datos (Data Description Specifications)	Descripción de la base de datos del usuario o de archivos de dispositivo que se entra en el sistema en un formato fijo. La descripción se utiliza para crear archivos.
EBCDIC	Extended Binary-Coded Decimal Interchange Code.	Juego de caracteres codificados que consta de 256 caracteres de ocho bits.
FIPS	Federal Information Processing Standard	Estándar oficial que sirve para mejorar la utilización y gestión de los ordenadores y del proceso de datos en las empresas.
ICF	Intersystem Communications Function	Función del sistema operativo que permite que un programa se comunique de modo interactivo con otro programa o sistema.
I/O	Entrada/Salida (Input/Output)	Datos proporcionados al ordenador o datos resultantes del proceso informático.
LVLCHK	Comprobación de Nivel	Función que compara los identificadores de registro de nivel de formato de un archivo que se va a abrir con la descripción de archivo que forma parte de un programa compilado, con el fin de determinar si el formato de registro para el archivo cambiado desde el programa se ha compilado.

Abreviatura	Significado	Explicación
ODT	Tabla de definición de objetos (Object Definition Table)	Tabla creada por el sistema en el tiempo de compilación para seguir la pista de los objetos declarados en el programa. Los objetos de programa que están en la tabla incluyen variables, constantes, etiquetas, listas de operandos y descripciones de excepciones. La tabla reside en el objeto programa compilado.
OS/400	Operating System/400	Sistema operativo del AS/400.
SDA	Ayuda para el Diseño de Pantallas	Función del programa bajo licencia Herramientas de Desarrollo de Aplicaciones AS/400 (AS/400 Application Development Tools) que proporciona ayuda al usuario para diseñar, crear y mantener pantallas y menús.
SEU	Programa de Utilidad para la Entrada del Fuente (Source Entry Utility)	Función del programa bajo licencia IBM Herramientas de Desarrollo de Aplicaciones AS/400 que se utiliza para crear y cambiar miembros fuente.
SQL/400	Lenguaje de Consulta Estructurado/400 (Structured Query Language/400)	Programa bajo licencia IBM que da soporte a la base de datos relacional que se utiliza para insertar información en una base de datos y para obtener y organizar información seleccionada desde una base de datos.
UPSI	Conmutador del indicador del estado de programa del usuario	Conmutador externo de programa que efectúa las funciones de un conmutador de hardware. Se proporcionan ocho conmutadores: UPSI 0 - 7.

Nota: Las abreviaturas de los mandatos OS/400 no aparecen en este apartado. Vea el manual *CL Reference* para más información acerca de los mandatos de OS/400 y de su utilización.

Índice

Caracteres Especiales

A

B

C

D

E

F

G

H

I

N

O

P

Q

R

S

T

U

V

X

Hoja de Comentarios

AS/400
COBOL/400 Guía del usuario
Versión 3 Release 1.0

Número de Publicación SC10-9424-00

En general, ¿está Ud. satisfecho con la información de este libro?

	Muy satisfecho	Satisfecho	Normal	Insatisfecho	Muy insatisfecho
Satisfacción general	□	□	□	□	□

¿Cómo valora los siguientes aspectos de este libro?

	Muy bien	Bien	Acep-table	Insatisfecho	Muy insatisfecho
Organización	□	□	□	□	□
Información completa y precisa	□	□	□	□	□
Información fácil de encontrar	□	□	□	□	□
Utilidad de las ilustraciones	□	□	□	□	□
Claridad de la redacción	□	□	□	□	□
Calidad de la edición	□	□	□	□	□
Adaptación a los formatos, unidades, etc. del país	□	□	□	□	□

Comentarios y sugerencias:

Nombre

Dirección

Compañía u Organización

Teléfono

Hoja de Comentarios
SC10-9424-00

Corte o Doble
Por la Línea

Dóblese por la línea de puntos **Por favor no lo grape** Dóblese por la línea de puntos

PONER
EL
SELLO
AQUI

International Business Machines, S.A.
Centro de Traducciones y Publicaciones
Avda. Diagonal, 571
08029 Barcelona, España

Dóblese por la línea de puntos **Por favor no lo grape** Dóblese por la línea de puntos

SC10-9424-00

Corte o Doble
Por la Línea

Número de Programa: 5763-CB1

Printed in Denmark by IBM Danmark A/S

SC10-9424-00