

Resumen de Nuevas Características en RM/COBOL en las últimas versiones, versión por versión

Software RM/COBOL

Ambiente de Aplicaciones y Lógica de Negocios de Misión Crítica. Le da a los programadores un fenomenal valor agregado, permitiéndoles con un mínimo de programación y mantenimiento, distribuir aplicaciones en un amplio abanico de plataformas y sistemas operativos.

RM/COBOL contiene un gran número de archivos y programas individuales. La cantidad de archivos y programas depende de la versión específica del producto y si se trata de un sistema de desarrollo o de ejecución solamente.

Compilador RM/COBOL

El compilador RM/COBOL lee código fuente COBOL y produce archivos objeto que pueden ser ejecutados utilizando un runtime. Estos archivos objetos son portátiles, y pueden ser ejecutados por un runtime de RM/COBOL en muchas configuraciones de computadoras -aún configuraciones de computadoras que son diferentes de la que se usó para compilar los programas objeto.

RM/COBOL Runtime System

El runtime de RM/COBOL se usa para ejecutar programas COBOL compilados. Liant Software Corporation provee un runtime diferente para cada computadora soportada, y ayuda a separar al programador COBOL de las diferencias entre computadoras y ambientes. El runtime también incluye un depurador para asistir en el desarrollo de programas COBOL.

CodeWatch

CodeWatch es un sistema de desarrollo integrado para Windows que se incluye en el development system de RM/COBOL. CodeWatch soporta el ciclo entero de desarrollo, incluyendo la edición, compilación, y depuración de aplicaciones RM/COBOL.

CodeWatch puede utilizarse para depurar y cambiar programas que son compilador independientemente, sin requerir que se construyan proyectos.

CodeBridge

CodeBridge es un sistema que permite hacer llamadas entre lenguajes de programación que se incluye con el development system de RM/COBOL. Esta herramienta simplifica la comunicación entre programas COBOL y subprogramas no-COBOL (como aquellos escritos en C o C++). CodeBridge permite a los programadores COBOL llamar APIs externas o subprogramas desarrollados a medida sin introducir otro lenguaje y tipos de datos en sus programas.

InstantSQL. Un paquete para embeber sentencias SQL en los programas fuente COBOL para que los programas puedan acceder bases de datos relacionales ODBC usando sentencias SQL.

Bibliotecas Internas y Programas Utilitarios

El runtime de RM/COBOL también incluye gran cantidad de bibliotecas de rutinas para realizar funciones no especificadas en el estándar COBOL. Entre otras cosas, estas rutinas pueden ser usadas para obtener información sobre los parámetros en los programas, controlar la pantalla, y ejecutar otros programas (no-COBOL).

Además, una biblioteca de subprogramas P\$ es provista con el runtime de RM/COBOL para Windows que permite acceder a las características de impresión de Windows.

Hay varios programas utilitarios que vienen con RM/COBOL. Estos programas utilitarios se usan para administrar y manipular archivos de datos y archivos objeto RM/COBOL.

Paquetes Adicionales que no se incluyen en las herramientas de desarrollo ni de producción

Hay varios paquetes adicionales disponibles para programas RM/COBOL. Estos incluyen los siguientes:

Xcentrity Business Information Server (BIS)

Es un Servidor de Aplicaciones Web basado en estándares específico para COBOL. Construido sobre el poder de XML como la base de conectividad trabaja en conjunto con los servidores web standard de la industria como Microsoft IIS y Apache. BIS ofrece a los desarrolladores de aplicaciones una oportunidad única para construir Aplicaciones Web avanzadas basadas en el browser o Web Services basados en SOAP abarcando programas RM/COBOL y archivos de datos COBOL y bases de datos.

Xcentrity Framework

Una solución completa diseñada específicamente para acelerar la migración de sus aplicaciones de gestión a la Web, y para incrementar su funcionalidad actual con los últimos avances disponibles en tecnologías Web. Basado en una arquitectura Java (J2EE), proporciona una solución completa y una rápida metodología para construir y obtener una verdadera funcionalidad Web dentro de una Arquitectura Orientada a Servicios (SOA).

BIS-Express

BIS-Express es un generador de aplicaciones para la web diseñado específicamente para el Business Information Server (BIS) de Liant Software Corporation. BIS-Express le permite diseñar su aplicación web de una forma intuitiva, sencilla y ágil, completando parámetros en un formulario web, luego genera todas las páginas, programas, y archivos de soporte que son requeridos por una aplicación completa. La aplicación creada por BIS-Express está lista para correr. Esto significa que usted puede agregar su lógica de negocios a una aplicación web completa y libre de errores, en vez de comenzar desde cero.

XML Extensions

Permite a las aplicaciones RM/COBOL interoperar libre y fácilmente con otras aplicaciones que usan el standard XML. XML Extensions proporciona la habilidad de importar y exportar documentos XML hacia y desde variables de la working storage de COBOL en una forma natural e intuitiva para el programador COBOL. XML Extensions también es la base de BIS, nuestro servidor de aplicaciones y servicios COBOL para la web.

WOW Extensions (COBOL-WOW)

Una herramienta de desarrollo visual para crear aplicaciones con todas las características de Windows en forma rápida, fácil y completamente en Cobol. Aplicaciones COBOL 100% GUI.

Relativity

Utilice sus datos COBOL como una Base de Datos Relacional. La información debe estar al alcance de los usuarios de forma fácil y rápida para su actualización y manipulación. Los usuarios quieren acceder a estos datos con sus herramientas favoritas, como Microsoft Excel, Microsoft Word, Crystal Reports, Microsoft Access, PHP y otras. Relativity brinda a cceso SQL a Archivos COBOL (RM y MF) (ODBC/JDBC)

RM/InfoExpress

Un administrador de archivos diseñado para optimizar el acceso a archivos de datos RM/COBOL en varias redes de area local y de area extendida. Disponible para TCP/IP.

Cobol RPC

Una herramienta para construir aplicaciones RM/COBOL distribuidas para LANs, WANs e Internet. Permite llamar e iniciar programas en otro servidor.

Todas las versiones

[- RM/COBOL 12](#)

- [- RM/COBOL 11](#)
 - [- RM/COBOL 10](#)
 - [- RM/COBOL 9](#)
 - [- RM/COBOL 8.0](#)
 - [- RM/COBOL 7.5](#)
 - [- RM/COBOL 7.1 para UNIX](#)
 - [- RM/COBOL 7.0 para Windows](#)
-

Nuevas Características en RM/COBOL 12

- # Compatibilidad mejorada con Microsoft Windows Server 2008 y Windows Vista.
- # Depuración de aplicaciones Xcentrisity BIS dentro de Codewatch.
- # El largo máximo de las opciones de línea de comando del runtime ha sido incrementado a 4095 caracteres.
- # El número máximo de identificadores que pueden ser definidos en un programa fuente ha sido aumentado desde 65.534 a 840.000.
- # El largo de las variables en la Working-Storage y la Linkage Section ahora puede ser mayor de 65.280 caracteres. Los registros de archivo aún están limitados a 65.280 caracteres.
- # Ahora puede usarse reference modification sobre ítems mayores a 65.280 caracteres en largo.
- # La cláusula OCCURS ahora permite más de 64k ocurrencias.
- # Se ha agregado la cláusula SAME AS como una nueva cláusula de descripción de un dato. Esta cláusula permite declarar una variable que tenga la misma descripción que una variable definida previamente. 01 registro-historia-usuarios. 10 clave-historia-usuarios. 20 id-usuario same as id-usuario in archivo-usuarios.
- # La declaración INSPECT ha sido mejorada con el adjetivo TRAILING para contar o reemplazar los caracteres finales en el valor de una variable. Esta nueva capacidad es particularmente útil para contar el número de espacios al final de una variable no numérica.
- # Reference modification ha sido mejorada con la frase JUSTIFIED que permite la fácil remoción de los espacios finales en un dato que se envía o justificación a la derecha en una variable que se recibe.
- # Los nombres de índice ya no tienen que ser únicos.
- # Se ha agregado la opción de compilación condicional a través del uso de secuencias en el área de Identification en el programa fuente. Los patrones de las secuencias para incluir o excluir se especifican en el archivo de configuración.
- # Las reglas de la cláusula OCCURS con la frase DEPENDING ON se han flexibilizado para permitir múltiples especificaciones de ocurrencia variable de datos dentro de un registro.
- # El compilador ahora soporta los siguientes nuevos registros especiales: HIGHEST-VALUE, LOWEST-VALUE, INITIAL-VALUE, MAX-VALUE, MIN-VALUE, PROCEDURE-NAME.
- # Se ha agregado un nuevo valor para la palabra clave ACTION del registro de configuración TERM-INPUT. El valor ENTER-DEBUGGER causa que el Debugger Interactivo de RM/COBOL se inicie en la siguiente sentencia ejecutada luego de una sentencia ACCEP

Características agregadas para soportar XML Extensions Incluye un compilador de RM/COBOL capaz de generar y embeber una tabla de símbolos en formato XML dentro del archivo objeto COBOL. Esta característica elimina la necesidad de correr un utilitario separado luego de la compilación para crear los archivos de modelo de XML Extensions. **RM/InfoExpress Versión 12** # Compatible con el Manejador de Archivos RM actual.

Logging del lado del cliente es una nueva capacidad para diagnosticar y resolver problemas de conexión inicial del cliente de RM/InfoExpress al Servidor de RM/InfoExpress.

El nuevo archivo de log contiene la siguiente información sobre cada conexión:

- el nombre de host del cliente
- la dirección IP del cliente (mediante gethostbyname)
- el nombre de host del servidor
- la dirección IP del servidor
- el resultado del pedido de conexión

Además del status de conexión, el archivo de log también contiene el resultado del pedido de establecimiento de sesión, y el resultado del primer pedido de Open y la respuesta.

Xcentrisity v12

- # Debugging de programas de servicio BIS usando CodeWatch.
- # Total soporte de Microsoft Windows Server 2008.
- # Internet Protocol Version 6 (IPv6) ahora es soportado en Windows Vista y Windows Server 2008.

Relativity v12

- # El utilitario del Designer para marcar columnas Nulas ahora muestra los nombres de las tablas seleccionadas para procesar.
- # UNIX data client calcula y muestra el número de filas en las tablas; muestra los índices de las tablas.
- # El Designer permite invocar el applet de Configuración de Relativity del panel de control y su archivo de ayuda.

Nuevas Características en RM/COBOL 11

Soporta **Windows Vista**, así como Windows XP, 2000 y 2003 Server.
Ya no se soportan versiones anteriores de Windows como 98, 98SE, ME y NT 4.0.

La sentencia CALL soporta hasta 2047 elementos en la lista de USING.

El formato de línea del código fuente permite una longitud total de hasta 65000 caracteres. Se han hecho varios cambios en las opciones de configuración para permitir la definición y uso de líneas de más de 80 caracteres en los programas fuente. Se ha mejorado la sentencia START permitiéndose utilizar

sentencias de tipo START WHILE KEY LIKE para seleccionar registros que concuerden con una determinada expresión.

La comprobación en tiempo de ejecución de índices ahora puede activarse mediante una entrada en el archivo de configuración.

Las propiedades configuración mediante RMConfig e INI2REG pueden ser establecidas para el usuario actual o para todos los usuarios de esa PC.

Soporte para PDFlib versiones 6.0.3 y 7.0.0 (www.pdflib.com).

CodeWatch incluye preferencias para establecer la longitud máxima de línea y la posición del margen derecho.

CodeWatch ahora soporta el uso de varios monitores.

Nuevas Características en RM/COBOL 10

Los servidores COM del compilador y el runtime se registran de

forma automática y silenciosa cuando sea necesario si el usuario tiene los permisos suficientes para realizar el registro. Ésto elimina errores tipo "clase no registrada" que pueden ocurrir cuando el registro de Windows está dañado. El registro automático es también útil cuando las máquinas cliente acceden al compilador o al runtime desde la red y no han registrado los servidores COM en la máquina cliente como parte de la instalación del cliente. Se incluye el adaptador de RM/COBOL a Pervasive.SQL para Linux (Ya se incluía la versión para Windows).

Se ha incrementado el número máximo de caracteres permitidos para el argumento principal de programa de 100 caracteres a 2048.

La función SYSTEM ahora permite una cadena de caracteres de línea de comando de hasta 4096 caracteres en sistemas operativos Windows tipo NT.

Las funciones C\$CARG y C\$DARG ahora pueden devolver un puntero al elemento referenciado, y un puntero a la cadena codificada para un elemento argumento que es un elemento editado. Adicionalmente, se ha añadido una función C\$PARG para obtener el puntero al elemento enésimo de los argumentos cuando otros atributos no son requeridos. Éstos cambios permiten tratar una lista de argumentos como un array, además de acceder a argumentos no especificados en la cabecera de Procedure Division.

Los mensajes de error del runtime ahora pueden manejar mejor rutas de fichero largas, que ahora son más comunes que en el pasado.

El runtime ahora soporta ficheros objeto de versión 13, lo que permite elementos mayores que 64K, como en las sentencias OCCURS.

Se han rebajado algunos límites del lenguaje RM/COBOL con respecto a las anteriores versiones:

Un elemento incluido en una sentencia OCCURS ya no está limitado a una longitud de 65280 caracteres.

La cabecera fija de un grupo variable en tamaño, es decir, un grupo que contenga un elemento descrito con la sintaxis DEPENDING ON de la sentencia OCCURS, ya no está limitado a una longitud de 65280 caracteres.

El tamaño total de los literales especificados en un segmento de programa no está limitado a 65535 caracteres.

Nueva extensión de posicionamiento de columna AT para los ACCEPT y DISPLAYS para compatibilidad con la sintaxis de otros dialectos COBOL.

El compilador avisa cuando se compila un programa con identificadores tipo EXTERNAL con más de 30 caracteres de longitud y los trunca. Generalmente se considera un límite de 240 caracteres para los nombre de identificadores.

Se ha corregido un problema del compilador que ignora la primera sentencia COPY de un fichero copiado utilizando la sentencia COPY que especifica un REPLACING.

Varias mejoras en CodeBridge

Mejoras en Codewatch

- El editor de código fuente de Codewatch versión 10 permite mostrar y ocultar los ficheros COPY.
- La ventana de Codewatch permite realizar doble-clic en la columna derecha para activar o desactivar puntos de ruptura.
- El menu File incluye opciones para imprimir todo el código fuente actual o la parte seleccionada.
- Las ventanas mostradas como pestañas muestran el nombre de cada fichero o ventana abierta en la parte baja del entorno.
- Ahora es posible cargar grandes proyectos y guardar ficheros fuentes grandes mucho mas rápido que en versiones anteriores.
- Se han solucionado varios fallos menores.

Mejoras en XML Extensions

* La versión 10 incluye numerosos arreglos, principalmente relacionados con Xcentrisity Business Information Server (BIS)

Nuevas Características en RM/COBOL 9

RM/COBOL 9 contiene nuevas mejoras y características significativas tanto para programadores Windows como UNIX.

Lo más destacable de RM/COBOL 9 y WOW Extensions

Modo ANSI/OEM en Windows

El programador puede indicar el código de página ANSI u OEM que se

utilizará durante la ejecución del programa. En versiones anteriores, siempre se utilizaba el código de página OEM. Si bien esto proporcionaba una necesaria compatibilidad para los programas existentes con respecto a MS-DOS, resultó que aumentaba la incompatibilidad con otras aplicaciones Windows y con las versiones UNIX/Linux de RM/COBOL cuando se utilizaban caracteres no-ASCII (por ejemplo á, é, í, ñ) en programas RM/COBOL bajo Windows. La versión 9 permite al programador escoger un código de página ANSI compatible con los editores de texto y otras aplicaciones Windows, y que es mucho más compatible con los grupos de caracteres que se suelen utilizar en UNIX.

C\$GetNativeCharset{ se ha añadido para permitir a un programa obtener detalles con respecto al grupo de caracteres y código de página que se están utilizando.

{style:type=span}Se pueden generar programas WOW anidados

El WOW Designer ahora permite generar de manera opcional programas utilizando un subprograma anidado por cada evento. Esto proporciona una mejor forma de aislar la lógica de manejo de cada evento, además de un programa más manejable. Cada evento puede tener ahora su propia Working-Storage Section.

Los programas WOW pueden procesar parámetros de eventos de controles ActiveX

Los programas WOW tienen acceso a los parámetros de eventos, permitiendo la utilización de un mayor número de controles ActiveX disponibles comercialmente.

Se agregaron características de compatibilidad con Micro Focus

Se han añadido varias características compatibles con Micro Focus COBOL para solucionar problemas que usualmente se encuentran en una migración. Se han incluido sentencias de ACCEPT y DISPLAY específicas de MF, y las cláusulas CRT, CRT STATUS y CURSOR. Además, se han aligerado las reglas de modificación de referencias para ajustarse a la forma menos estricta de comprobación que tienen el runtime de Micro Focus y otros sistemas COBOL.

El registro especial WHEN-COMPILED ahora también es soportado.

Nombres de identificadores largos (>30 caracteres)

La versión 9 ahora permite utilizar completamente nombres de variables y de condiciones largos, además de otros nombres de identificadores definidos por el usuario hasta un máximo de 240 caracteres de longitud. Esto es especialmente útil cuando se utiliza XML Extensions para RM/COBOL para guardar y procesar datos XML externos.

Muchas mejoras en WOW Designer

El WOW Designer tiene nuevas características que permiten alinear controles dentro de contenedores, una propiedad Tag que permite contener datos especificados por el usuario para la mayoría de los controles intrínsecos, ToolTips definidas por el usuario en muchos controles, obtener la propiedad Name de los controles, y otras muchas funciones. La barra de herramientas ahora puede ser configurada por el programador. Se han agregado al conjunto de controles básicos un control para tomar la fecha y la hora (Date Time Picker) y un control de Mes del Calendario. Se ha agregado un editor del control Tab, además de otras muchas características muy convenientes.

Mejoras en WOW Runtime plus

El WOW Thin Client tiene ahora su propio archivo de configuración y

soporta un nuevo grupo de funciones de capacidades RPC dentro de un programa WOW Thin Client. Se han agregado nuevas funciones del runtime para controlar iconos y manipular controles, además de iterar a través de todos los controles en un formulario WOW.

Mejoras en la importación de RM/Panels en WOW

La capacidad de importación de RM/Panels del WOW Designer ha sido ampliada para permitir especificar ciertas características de representación gráfica. Esto puede resultar en una mejor apariencia GUI.

Se agregaron nuevas rutinas P\$

Dos nuevas funciones P\$ de impresión en Windows han sido agregadas: P\$ResetPrinter y P\$SetLineSpacing.

Control de listado de compilación mejorado

La frase SUPPRESS puede usarse con una sentencia COPY para evitar la impresión de las líneas de código de los copys. Los mensajes de diagnóstico del compilador ahora también pueden ser eliminados de

manera selectiva.

Manejo más conveniente de los patrones de condición de "LIKE"

Si se usa un patrón variable con la nueva condición "LIKE", ahora se omiten los espacios en blanco sobrantes.

Carácter de relleno de memoria configurable

El programador puede ahora designar un carácter de relleno para utilizar para la memoria no inicializada.

Cientos de otras mejoras y correcciones.

Nuevas Características en RM/COBOL 8.0

Nuevas Características del Runtime (Windows y Unix)

. **Datos Binarios Nativos.** El runtime ahora soporta items de datos numéricos de formato binario en formato nativo. Estas variables se identifican en el programa fuente como COMPUTATIONAL-5 o COMP-5.

. **Nuevo Nivel de Versión de Objeto.** Se introduce el nivel de version de objetos 11 para soportar el uso de COMPUTATIONAL-5 y COMP-5.

Nuevas Características del Compilador (Windows y Unix)

. COMPUTATIONAL-5 (COMP-5) para items de datos numéricos de formato binario en formato nativo de máquina.

. El compilador ahora provee variables en la Working-Storage Section para parámetros de archivos y datos que son especificados con un nombre de variable que no está definida en la Data División. Esto elimina errores de compilación causados por el olvido en la definición de una o más de éstos parámetros y por lo tanto acelera el tiempo de desarrollo. Los siguientes parámetros son afectados por esta característica:

- Nombre de acceso al archivo especificado en la cláusula ASSIGN.
- Variable Padding character especificada en la cláusula PADDING CHARACTER.
- Variable para una clave relativa especificada en la frase RELATIVE KEY de la cláusula ACCESS MODE.
- Variable de status de I/O especificada en la cláusula FILE STATUS.
- Variable de largo de registro especificada en la frase DEPENDING de la cláusula RECORD IS VARYING.
- Variables de la Linage especificadas en la cláusula LINAGE (las variables de líneas de lineage, de footing, de top y bottom).
- Variable de etiqueta especificada en la cláusula VALUE OF.
- Variable de contador de ocurrencias especificada en la frase DEPENDING de la cláusula OCCURS.

. El compilador ahora soporta expresiones de concatenación usando el operador & para concatenar literales no numéricos.

. La cláusula OCCURS ahora se permite en los niveles 01 y 77 en la Working-Storage Section.

. Un grupo vacío, o sea, un grupo que no contiene ningún item elemental, ahora es permitido. Los elementos vacíos corresponden a elementos vacíos XML y esto se hace para alinear aún más el modelo de datos COBOL con el modelo de datos XML.

. El compilador ahora soporta la cláusula NUMERIC SIGN en el párrafo Special-Names. Esto permite al programa fuente especificar la convención de signo por defecto para elementos numéricos con signo de display que son descritos sin la cláusula SIGN en su descripción.

. La cláusula ASSIGN ya no requiere un nombre de dispositivo, aún cuando el nombre de acceso a archivo se especifique con una variable. El nombre de dispositivo solo se requiere ahora si no se especifica ningún nombre de acceso al archivo en la cláusula ASSIGN (variable1 o

literal1).

Nuevas Características en RM/COBOL 7.5

Introducción del Ambiente del Desarrollo de Aplicaciones CodeWatch

RM/COBOL 7.5 incluye la última versión de CodeWatch, un ambiente de desarrollo para Windows completamente integrado. Iniciado para esta versión 7.5, CodeWatch ahora soporta el ciclo completo de desarrollo, incluyendo edición, compilación y depuración de aplicaciones RM/COBOL. CodeWatch puede ser utilizado para depurar y cambiar programas que son compilados en forma independiente, sin el requerimiento de la construcción de proyectos. En lugar de ello, el conocimiento requerido acerca de la estructura de su aplicación es creada durante las sesiones de depuración.

Mejoras a CodeBridge

CodeBridge , el sistema de Liant para llamadas de sistema de lenguaje-cruzado, ha sido mejorado para manipular enteros de 64-bits sobre plataformas UNIX que posean un compilador C que soporte enteros de 64-bits. Un nuevo callback en tiempo de ejecución, GetCallerInfo, ha sido agregado para permitir a los subprogramas no-COBOL mejorar los mensajes de error con información adicional acerca del llamador. Este nuevo callback provee el nombre del programa llamador y el número de línea, el nombre del archivo objeto, y la fecha y hora en que el programa llamador fue compilado. La definición y comentario de esta nueva llamada en tiempo de ejecución y su estructura de datos asociado, CALLER_INFO, están disponibles en **rtcallbk.h** , un cabezal provisto con el sistema RM/COBOL. Los ejemplos de su uso están incluidos en el subprograma de ejemplo **msgbox.c** para Windows y en el subprograma de ejemplo **usrsub.c** para UNIX. Dos nuevos parámetros de atributos, llamados atributos de error de base, han sido agregados a CodeBridge para la recuperación del set de información de error de la biblioteca C y de las funciones API de Windows. Los nuevos atributos de error de base, **errno** y **get_last_error** , permiten el retorno de la información del error mediante la edición de la plantilla CodeBridge en lugar del código generado.

Compilador en Modo-Consola en Windows

El compilador RM/COBOL puede ahora ser ejecutado como una aplicación modo-consola en Windows con el comando `rmcobolc` . El comando `rmcobolg` puede ser aún utilizado para disparar la versión de interfase gráfica de usuario Windows (modo-GUI) del compilador. La aplicación en modo-consola es más pequeña y veloz que la versión modo-GUI y satisface las compilaciones en lote de un gran número de programas. Más allá de la interfase, los dos compiladores son idénticos ya que ambos utilizan la misma DLL para la implementación del compilador. Una opción de instalación permite que usted seleccione una u otra versión del compilador a ser invocada con el comando `rmcobol` .

Compilaciones Múltiples y en Lotes Son Más Fáciles y Más Rápidas

La caja de diálogo al compilar en RM/COBOL para Windows permite que más de un archivo sea seleccionado. Puede seleccionar archivos adicionales mediante la pulsación de la tecla Ctrl mientras se clickea sobre ellos, o por el uso de Ctrl+A para seleccionarlos a todos. Los subdirectorios son automáticamente ignorados. Las compilaciones se detienen cuando todos los archivos han sido compilados o una compilación retorna un código de salida no-cero. Esta forma de compilación es más rápida que mediante una secuencia de órdenes porque el compilador no necesita ser recargado entre los fuentes. Tanto para el compilador de modo-consola como para el de modo-GUI, los caracteres comodines sobre la línea de comando también pueden ser aplicados para la selección de múltiples archivos a compilar. Los caracteres comodines soportados son "?" (coincide con cualquier carácter único) y "*" (coincide con ninguno o varios caracteres).

Archivos Indexados Más Confiables

El soporte a los archivos indexados ha sido hecho más confiable por el agregado de nuevas características integrales como parte de la versión de archivo de nivel 4. Adicionalmente, la versión 4 de archivos indexados opcionalmente soporta la nueva capacidad "atomic I/O" la cual provee un recurso al usuario para evitar casi todos los errores 98 causados por fallas que ocurren cuando un archivo es abierto. Los archivos creados con atomic I/O casi nunca necesitan recuperación. De ocurrir una caída durante una operación I/O COBOL, el archivo será automática y rápidamente recuperado la próxima vez que sea abierto o cuando una operación de escritura sea realizada. La palabra clave ENABLE-ATOMIC-IO ha sido agregada al registro de configuración RUN-INDEX-FILES para determinar si los archivos indexados creados por el runtime system utilizarán atomic I/O.

La versión por defecto de archivo indexado para los nuevos archivos ha sido cambiada de 2 a 4 automáticamente para proveer el más alto nivel de confiabilidad a los nuevos archivos. La palabra clave DEFAULT-FILE-VERSION-NUMBER del registro de configuración RUN-INDEX-FILES puede ser utilizada para especificar un valor diferente.

La versión 4 de archivos indexados puede, tal como la versión 3, crecer a un tamaño mayor que los de versiones 0 o 2. Sin embargo, a diferencia de los de versión 3, los archivos de versión 4 pueden ser tan largos o de tamaño regular, dependiendo de la nueva palabra clave USE-LARGE-FILE-LOCK-LIMIT en el registro de configuración RUN-INDEX-FILE. Esta nueva palabra clave determina si las palabras claves LARGE-FILE-LOCK-LIMIT o FILE-LOCK-LIMIT del registro de configuración RUN-FILES-ATTR son utilizadas para determinar la dirección más larga que puede ser bloqueada en el archivo. Esto, de cumplirse, determina cuán grande puede ser el archivo.

Mejor Performance de los Archivos Indexados

Varios cambios han sido introducidos para incrementar la performance de los archivos indexados en la creación de nuevos archivos indexados mediante tamaños de bloques más razonables y mediante el incremento del tamaño máximo del buffer pool de archivo en al menos de un millón a diez millones de bytes.

Ahora, el runtime system de la versión 7.5 de RM/COBOL crea nuevos archivos indexados con un tamaño mínimo de bloque de 1024 bytes y asegura que el tamaño del bloque para los nuevos archivos indexados es un múltiplo del tamaño de sector del disco. El procesamiento del archivo indexado es generalmente más eficiente con tamaños de bloques largos y con tamaños de bloques que son también múltiplos del tamaño del sector del disco (512 bytes para Windows y, normalmente, 1024 para UNIX).

Dos nuevas palabras claves, MINIMUM-BLOCK-SIZE y ROUND-TO-NICE-BLOCK-SIZE, se han agregado al registro de configuración RUN-INDEX-FILES para permitir que el tamaño del bloque sea computado de la misma manera que las versiones previas del runtime system de RM/COBOL.

La palabra clave BUFFER-POOL-SIZE del registro de configuración RUN-FILES-ATTR permite ahora que el tamaño del buffer sea de 10.000.000 bytes. Generalmente, un tamaño tal de buffer produce mejor performance que un buffer de tamaño más pequeño. Algunos chequeos pueden ser requeridos para encontrar el tamaño óptimo para su aplicación.

Archivo de Configuración Automática Disponible para Windows

Los archivos de configuración pueden ahora ser automáticamente cargados en Windows por el runtime, el compilador, y por el utilitario de recuperación en una forma similar a la capacidad de UNIX. Aún así, la facultad de vincular un archivo de configuración al ejecutable en Windows mediante el utilitario **rmattach** es aún posible.

Comentarios de Cola para los Registros de Configuración

Los registros de configuración pueden ahora contener un comentario de cola, es decir, un comentario que no comienza en la columna uno del registro de configuración.

Mejoras para Subprogramas No-COBOL en Windows

El runtime system de RM/COBOL 7.5 para Windows ha sido mejorado para cargar las DLLs automáticamente desde el subdirectorio especial **RmAutoLd**, del directorio de ejecución del **runcobol** sin necesidad de especificar el nombre del archivo con la opción L (Library) del comando **runcobol**. Todas las DLLs en este subdirectorio especial serán cargadas automáticamente. Aunque esto ya no hace necesario el especificar bibliotecas no-COBOL en el comando **runcobol**, la opción L (Library) del Runtime Command está igualmente soportada para hacerlo.

El runtime system de Windows fue también mejorado para soportar los símbolos especiales predefinidos (puntos de entrada y nombres de variables), tales como RM_EntryPoints, RM_AddOnInit, y RM_AddOnTerminate, los cuales, previamente, estaban disponibles solo bajo UNIX. Dado que ninguno de estos puntos de entrada especiales son requeridos, si están presentes, la DLL pueden proveer una lista de puntos de entrada de llamada-COBOL

sin necesidad de especificar la sección .EDATA en tiempo de enlace, y puede proveer un código especial de inicialización y terminación que puede ser llamado automáticamente cuando el runtime system inicializa y finaliza.

Agregados a la Biblioteca de Subprogramas de RM/COBOL

La biblioteca de subprogramas de RM/COBOL ha sido ampliada con los siguientes subprogramas C\$:

- **C\$CompilePattern** compila una expresión de variable patrón para usar en la nueva condición LIKE la cual ha sido agregada al lenguaje RM/COBOL.
- **C\$ConvertAnsiToOem** puede ser usado para convertir un buffer conteniendo caracteres ANSI a un buffer conteniendo los correspondientes caracteres OEM. La configuración del carácter euro ANSI/OEM del runtime es preservado en la conversión.
- **C\$ConvertOemToAnsi** puede ser usado para convertir un buffer conteniendo caracteres OEM a un buffer conteniendo los correspondientes caracteres ANSI. La configuración del carácter euro ANSI/OEM del runtime es preservado en la conversión.
- **C\$DARG** puede ser usado para obtener la descripción de un argumento actual mediante el uso de un número de argumento para referirse al argumento deseado.
- Se han agregado varios de los subprogramas de la biblioteca para hacer operaciones lógicas. Estos incluyen **C\$LogicalAnd** , **C\$LogicalComplement** , **C\$LogicalOr** , **C\$LogicalShiftLeft** , **C\$LogicalShiftRight** , y **C\$LogicalXor** . Cada uno de estos subprogramas pueden operar tanto cadenas no numéricas como valores numéricos.
- **C\$SecureHash** produce un compendio de mensaje de 20 bytes desde una cadena de texto de mensaje de entrada de 229 bytes (2 * * 29) o menos de largo usando el seguro algoritmo de hash (SHA-1).

Eliminación de los Archivos de Mensajes

Los archivos de mensajes para los programas ejecutables RM/COBOL (**runcobol** , **rmcobol** y **recover1**) han sido eliminados. **RUN.MSG** , **RMC.MSG**, y **REC.MSG** , presentes en las versiones previas de RM/COBOL, no existen en la versión 7.5. Los mensajes contenidos en estos archivos residen ahora dentro de cada ejecutable. De esta manera, no hay ninguna búsqueda del archivo de mensajes y entonces ninguna posibilidad de que ocurra un desencuentro en la ejecución con las versiones de los archivos de

mensajes.

Eliminación del Archivo Overlay del Compilador

El archivo de overlay para el compilador RM/COBOL de los programas ejecutables al compilar (**rmcobol** y **rmcobolc**) ha sido eliminado. El archivo **RMCOBOL.OVY** , presente en las versiones de los sistemas de desarrollo de RM/COBOL ya no existe en la versión 7.5. De esta manera, no hay ninguna búsqueda del archivo de overlay y entonces ninguna posibilidad de que ocurra un desencuentro en la ejecución con las versiones de los archivos de overlay.

Nuevas Características del Runtime System

Además de los nuevos subprogramas C\$ a la biblioteca de subprogramas para tiempo de ejecución, el runtime system de la versión 7.5 de RM/COBOL para Windows y UNIX ha sido mejorado con las siguientes características:

- **Pipe de sendas** . En UNIX, donde direccionar una salida de impresión a un spooler de impresión es común, un nombre de acceso de archivo que comienza con un carácter pipe ("|") puede ahora ser usado para crear el pipe sin tener que usar un registro de configuración DEFINE-DEVICE. Esto permite que las opciones de spooler sean construidas dinámicamente en una variable por el programa COBOL. El chequeo del carácter pipe es realizado después de que el nombre de acceso del archivo es mapeado mediante cualquier variable de ambiente aplicable, así un programa también puede ser conducido a un pipe mediante la carga de una variable de ambiente que mapee el nombre de acceso del archivo especificado en el programa a un valor que tenga al carácter pipe como el primer carácter.

- **Pipes de Entrada.** Ahora los pipes de entrada son soportados bajo UNIX. Un pipe es usado para entrada cuando un archivo es abierto en modo de entrada y uno de dos, o el path que comience con el carácter pipe ("|") o el nombre de acceso al archivo se refiera al registro de configuración DEFINE-DEVICE que especifique un pipe con la palabra clave PIPE=YES. Por ejemplo, un archivo abierto en modo de entrada con un nombre de acceso de archivo teniendo el valor "| ls -1 *.txt" leerá una lista de archivos de texto (asumiendo archivos de texto a aquellos identificados por la extensión ".txt") en el directorio corriente.

- **Configuración por Defecto del Procedimiento Use.** La acción a tomar cuando no hay procedimiento USE para un error de I/O sobre un archivo puede ahora ser configurada. Previamente, RM/COBOL finalizaba la unidad de ejecución con un mensaje de error apropiado cuando no había un procedimiento USE. Ahora, el comportamiento del runtime puede ser configurado para ocasionar que el programa continúe como si un procedimiento USE vacío fuera aplicado mediante la especificación de DEFAULT-USE-PROCEDURE=CONTINUE en el registro de configuración RUN-FILES-ATTR.

- **Biblioteca de Configuración.** Las bibliotecas de objetos de RM/COBOL y las bibliotecas de subprogramas no-COBOL pueden ahora ser configuradas con la palabra clave L, la cual ha sido agregada al registro de configuración RUN-OPTION (paralelamente a la opción L (Library) del comando de Runtime). Adicionalmente, la palabra clave LIBRARY-PATH ha sido agregada al registro de configuración RUN-OPTION para ocasionar la carga de todas las bibliotecas de objetos en un directorio especificado. Tanto la palabra clave L como LIBRARY-PATH RM/COBOL pueden ser especificadas múltiples veces en la configuración.

- **Configuración del Programa Principal.** La palabra clave MAIN-PROGRAM ha sido agregada al registro de configuración RUN-OPTION. Ella permite especificar un nombre de programa principal para sustituir el nombre de programa especificado en la línea de comando.

- **Mejora al Nombre del Registro de Configuración.** Los nombres del registro de configuración han sido mejorados para permitir el intercambio entre singular y plural. Por ejemplo, los nombres de registro RUN-OPTION y RUN-OPTIONS son ambos permitidos, y cualquiera de ellos soporta el mismo set de palabras claves.

- **COPY-TO-CLIPBOARD Como Valor Agregado para la Palabra Clave ACTION.** En el registro de configuración TERM-INPUT, el valor COPY-TO-CLIPBOARD permite que el texto actual seleccionado sea copiado al portapapeles y en consecuencia proveer la capacidad de configurar una tecla en Windows que lleve a cabo esta acción (por ejemplo, Ctrl+"C").

- **Más Caracteres Equivalentes de TERM-INPUT para Windows.** El poder de configurar teclados para Windows ha sido mejorado para un mejor soporte de teclados internacionales y proveer acceso a teclas que pueden estar en algunos teclados especializados (por ejemplo, el Microsoft Natural Keyboard).

- **Nuevo Agregado Especial de Texto a los Íconos de la Barra de Herramientas.** Esta nueva característica provee la habilidad de especificar texto a los botones de la barra de herramientas que generen pulsaciones configurables.

- **Soporte Euro para Windows.** Un nuevo tipo de registro de configuración, con el identificador INTERNATIONALIZATION, ha sido agregado para permitir la configuración del soporte para el símbolo euro (#) en el ACCEPT, DISPLAY, y las operaciones de impresión en Windows. Además, la palabra clave DATA-CHARACTERS del registro de configuración TERM-ATTR ha sido mejorada para permitir la especificación de múltiples rangos separados bajo Windows para de esta manera emparejar la capacidad ya soportada bajo UNIX. Dado que el símbolo euro no está típicamente en el rango por defecto de los caracteres que son interpretados como de texto, el permitir que el símbolo euro sea ingresado en una declaración ACCEPT implica la modificación del rango de los caracteres de datos. Esto puede hacerse usando múltiples palabras clave DATA-CHARACTERS con separación de rangos o por la modificación del límite superior del rango de caracteres para incluir el símbolo euro.

- **SYSTEM Window Types.** Bajo Windows, los valores que pueden ser especificados para la propiedad SYSTEM Window Type han sido expandidos para incluir MINIMIZEDNOACTIVE y SHOWNOACTIVATE. Estos valores pueden ser almacenados en el registro de Windows mediante **rmconfig** o pueden ser cargados en tiempo de ejecución mediante **C\$GUICFG** .

- **Tooltips en la Barra de Herramientas.** Bajo Windows, la indicación del botón de la barra de herramientas que era desplegada en la barra de estado es ahora también desplegada como un tooltip, el cual es una pequeña ventana pop-up conteniendo un texto cerca del botón cuando el puntero del mouse flota sobre él. Se ha agregado la propiedad Toolbar Prompt para controlar este nuevo comportamiento. Esta nueva propiedad permite escoger el antiguo comportamiento de mostrar el texto sobre la barra de estado. También permite escoger el no desplegar la sugerencia en absoluto, desplegarla solo como un tooltip, y desplegar diferentes valores en la barra de estado y como tooltip (esto último requiere cambiar el set de la sugerencia en el registro de Windows mediante el uso de **rmconfig** o en el programa RM/COBOL mediante una llamada a **C\$TBar**). La propiedad Toolbar Prompt puede ser cargada en el registro con **rmconfig** o temporariamente cambiada en tiempo de ejecución llamando a **C\$GUICFG** .

- **C\$SCRD Soporte para Caracteres de Dibujo de Líneas.** El subprograma **C\$SCRD** ha sido modificado para soportar caracteres de dibujo de líneas por el retorno de guiones, signos de más, y barras verticales. Bajo Windows, se ha agregado la propiedad Screen Read Line Draw para permitir que **C\$SCRD** retorne caracteres de dibujo de líneas DOS (por ejemplo, \$D9, " Û " para el rincón inferior derecho de una caja)

- **P\$GetHandle.** El subprograma **P\$GetHandle** ha sido modificado para proveer el retorno opcional del manipulador de Windows de la impresora P\$ actual. Esto posibilita que un programa no-COBOL adicione información (gráficos especiales o códigos de barras) a la página impresa sobre una impresora P\$.

- **P\$DisableDialog.** El subprograma **P\$DisableDialog** es usado para controlar la invocación automática de la ventana estándar de diálogo de Windows cuando un dispositivo "PRINTER?" es abierto.

- **P\$EnumPrinterInfo.** Es usado para recuperar información detallada acerca de todo de las impresoras del sistema. No es necesario abrir una impresora para obtener esta información.

- **Log de Terminación para UNIX.** El runtime RM/COBOL de UNIX permite ahora registrar los mensajes de error de terminación, incluyendo información de "traceback", usando la clave ENABLE-LOGGING=TERMINATION del registro de configuración RUN-OPTION. El log de terminación permite que los mensajes de error sean coleccionados para posteriores análisis. Los usuarios de UNIX pueden aún redireccionar el "standard error" para coleccionar información de terminación, pero haciéndolo así indicará que la información no será desplegada para el usuario.

- **Creando Archivos sobre un RM/InfoExpress Server.** La nueva palabra clave DISABLE-LOCAL-ACCESS-METHOD del registro de configuración RUN-FILES-ATTR puede ser usada para prevenir archivos con incapacidad, nombres simples creados en el directorio de trabajo actual. Especificar UNQUALIFIED-NAMES para el valor de la nueva palabra clave permitirá nuevos archivos a ser creados sobre un RM/InfoExpress Server. En las versiones previas del runtime system, esto era necesario para la aplicación para especificar la máquina servidora sobre la cual el archivo podía ser creado, ya fuera por la inclusión de una senda completa en el programa COBOL o por el uso de una variable de ambiente que la especificara. Proveer

la especificación de la nueva palabra clave hace ahora posible crear tales archivos en el primer directorio de la variable de ambiente RUNPATH. El especificarla no tiene efecto sobre el hallazgo de los archivos existentes

Nuevas Características del Compilador

El compilador RM/COBOL versión 7.5 para Windows y UNIX ha sido mejorado con las siguientes nuevas características:

- **Palabras Reservadas.** Para soportar las nuevas características del lenguaje mencionadas abajo, la lista de palabras reservadas ha sido ampliada con las nuevas palabras DATA-POINTER, DEFAULT, y LIKE. Asimismo, varias palabras han sido eliminadas de la lista de palabras reservadas y trasladadas a la nueva categoría de palabras de contexto sensitivo.

- **Palabras de Contexto Sensitivo.** Algunas palabras previamente consideradas a ser siempre reservadas han sido cambiadas a ser reservadas solo en ciertos contextos y en consecuencia ahora en la nueva categoría de palabras de contexto sensitivo. Por ejemplo, la palabra UNDERLINE fue previamente una palabra reservada, pero ahora es reservada solo en el contexto de una entrada de descripción de pantalla. Asimismo, varias nuevas palabras han sido agregadas a esta categoría de palabras para soportar las nuevas características más abajo mencionadas.

- **Ítems FILLER Ingresados en la Tabla de Símbolos.** Los ítems de dato FILLER son ahora ingresados en la tabla de símbolos y así ser desplegados en el mapa de asignación del listado. El mantener los ítems de dato FILLER en la tabla de símbolos requiere memoria adicional en la compilación pero permite soporte para la nueva frase WITH FILLER de la declaración INITIALIZE. La nueva opción de compilación, SUPPRESS-FILLER-IN-SYMBOL-TABLE, ha sido agregada para reducir la memoria requerida al compilar un programa con muchos ítems de dato FILLER.

- **Cláusula SELECT Frase NOT OPTIONAL.** Por compatibilidad con los otros dialectos COBOL, la frase NOT OPTIONAL puede ser especificada en la cláusula SELECT para los archivos que deban estar presentes en tiempo de ejecución. En virtud de que RM/COBOL asume que los archivos son requeridos en tiempo de ejecución a menos que la frase OPTIONAL fuera especificada, la frase NOT OPTIONAL no tiene efecto, pero es aceptada en orden de facilitar la conversión de programas originalmente escritos en otros dialectos COBOL.

- **Cláusula PICTURE.** La cláusula PICTURE puede ahora ser omitida para un ítem de dato elemental descrito con una cláusula VALUE que especifique un literal no numérico. En este caso, el ítem definido es como si una cláusula PIC X(*n*) hubiera sido especificada, donde *n* es el largo del literal no numérico especificado en la cláusula VALUE. El compilador RM/COBOL ahora también soporta una cadena de caracteres PICTURE implícita cuando un literal numérico es especificado en la cláusula VALUE.

- **Formato 1 Cláusula VALUE.** El formato 1 de la cláusula VALUE, la cual define la inicialización de valores para los ítems de la Working-Storage, ahora también define valores a ser usados por la nueva frase VALUE de la declaración INITIALIZE. Por consiguiente, la cláusula es ahora permitida en las secciones File, Linkage y Communication y también en las descripciones de registro descriptos con la cláusula EXTERNAL sin la restricciones previas del RM/COBOL en que la cláusula VALUE podía ser usada solo en las situaciones en que ésta era incluida en el programa fuente como parte de un archivo de copia.

- **Formato 2 Cláusula VALUE.** El formato 2 de la cláusula VALUE, la cual define valores de número de nivel 88 para nombres de condición, ha sido ampliado para permitir operadores relacionales. Esto permite, en particular, el uso de la nueva condición LIKE para especificar valores válidos para ítems de datos mediante el uso de una expresión patrón regular.

- **Condición LIKE.** Los operadores relacionales han sido ampliados para incluir el operador

LIKE y entonces proveer el caso especial de condiciones de relación llamada condición LIKE. La condición LIKE especifica un valor basado en verdad sobre un ítem de dato que coincide con un valor patrón. Los valores patrones son como expresiones regulares en la misma forma usada por XML Schema.

· **Mejoras a la Declaración ACCEPT.** La declaración ACCEPT soporta ahora una propuesta de método COBOL estándar de obtener cuatro dígitos para el año mediante el uso de las frases opcionales YYYYMMDD y YYYYDDD en las frases FROM DATE o FROM DAY, respectivamente.

· **Mejoras a la Declaración EXIT.** La declaración EXIT ahora incluye formatos para salir de un párrafo, una sección, o de un PERFORM en-línea.

· **Mejoras a la Declaración INITIALIZE.** La declaración INITIALIZE ahora incluye las frases opcionales FILLER, VALUE y DEFAULT. La frase FILLER causa que los ítems de datos FILLER sean inicializados, en vez de ignorados. La frase VALUE causa la inicialización del valor literal especificado en la cláusula VALUE asociada con un ítem de dato elemental. La frase DEFAULT ocasiona que los ítems sean inicializados a sus valores por defecto (espacios, cero, o nulo) cuando las frases VALUE o REPLACING son especificadas pero ninguno de estos valores es aplicable al ítem de dato elemental a ser inicializado. La declaración INITIALIZE ahora también incluye el nuevo nombre de categoría DATA-POINTER para inicializar ítems punteros de datos y permite la inicialización de ítems de datos con ocurrencias variables.

· **Mejora a la Declaración USE.** La declaración USE ahora permite una serie de modos de OPEN con o sin una serie de nombres de archivos, y de esta manera un simple procedimiento USE puede ser declarado para múltiples modos de apertura o para los archivos especificados.

· **Literales Pasados "BY CONTENT".** El compilador RM/COBOL versión 7.5 ha sido modificado para pasar literales especificados en la frase USING de la declaración CALL como si la frase BY CONTENT fuera aplicada. Este cambio fue realizado para proteger el valor del literal en el programa llamador de inadvertidos cambios realizados al ítem de dato correspondiente de la Linkage Section en el programa llamado. La nueva palabra clave SUPPRESS-LITERAL-BY-CONTENT del registro de configuración COMPILER-OPTIONS ha sido adicionada para sustituir este nuevo comportamiento hasta que el programa fuente COBOL sea corregido.

· **Registración del Compilador.** El compilador RM/COBOL versión 7.5 para Windows consiste ahora de un cliente (ya sea cliente modo-consola, **rmcobolc.exe** , o cliente modo-GUI, **rmcobolg.exe** , uno de los dos llamado **rmcobol.exe**) y un servidor (**rmcbi75c.dll**). Los componentes del cliente son completamente compatibles con las versiones previas del programa **rmcobol.exe** ; esto quiere decir que no se requieren cambios en la línea de comandos. Sin embargo, el servidor DLL debe ser registrado en Windows antes de que los programas RM/COBOL puedan ser compilados. Esto es automáticamente ejecutado durante la instalación y es liberado solo si el compilador es mudado a un directorio diferente sin que sea instalado nuevamente. En este caso, ningún cliente puede ser usado para re-registrar el servidor.

· **Cambio en la Opción Z del Compilador.** La opción Z del compilador puede no ser utilizada para restringir el nivel de versión objeto del código generado a objetos de nivel 6 o por debajo. Objetos nivel 7 corresponden a las versiones 6. *nn* de RM/COBOL. La eliminación de la necesidad del compilador de generar código para estas antiguas versiones perfecciona su eficiencia y confiabilidad y asegura que el mismo no tendrá necesidad de suprimir optimizaciones que el antiguo runtime system no soporta.

· **Los Listados Incluyen Ahora las Declaraciones Remplazadas.** Las líneas fuentes que han sido remplazadas por la declaración REPLACE o la frase REPLACING de la declaración COPY son ahora incluidas como comentarios en el listado de compilación. Estas líneas pueden ser suprimidas del listado mediante la especificación tanto de la opción de compilación C=2 como de C=3, o por el valor de configuración SUPPRESS-REPLACED-LINES para la palabra clave LISTING-ATTRIBUTES del registro de configuración COMPILER-OPTIONS.

· **Salidas de Compilación Más Informativas.** El indicador de nivel de copia en el listado de compilación también ha sido mejorado para proveer información acerca de líneas fuente que han sido modificadas, reemplazadas o insertadas como resultado de la declaración REPLACE o de la frase REPLACING de la declaración COPY. Además, la presentación del reemplazamiento de texto en el listado de compilación ha sido perfeccionada.

· **Listando Solo Errores con Información de Depuración.** La opción E del comando de compilación no es ignorada cuando las opciones Y=2 y Y=3 son especificadas. De esta manera, ahora es posible requerir solo un listado de errores mientras al mismo tiempo se genera una completa información de depuración en el archivo objeto.

Nuevas Características de los Utilitarios

Las nuevas características de los programas utilitarios en RM/COBOL versión 7.5 para Windows y UNIX incluyen:

· **Rápida Conversión al Nuevo Formato de Archivo Indexado.** El utilitario Indexed File Recovery (**recover1**) ha sido mejorado para convertir archivos indexados al nuevo y más confiable formato versión 4. Esta conversión es muy eficiente, tomando solo un poco más de tiempo que una recuperación de archivo realizada para corregir un error. Para convertir un archivo existente al nuevo formato, utilice el utilitario Define Indexed File (**rmdefinix**) para cambiar a la nueva versión de archivo en el cabezal del archivo y luego el **recover1** para ejecutar la conversión. Algunos archivos indexados con muy pequeños tamaños de bloque no pueden ser convertidos al formato de la versión 4. En este caso, cuando usted ejecute **rmdefinix** para cambiar la versión de archivo en el cabezal, recibirá una notificación de que el tamaño del bloque es muy pequeño para que ocurra una conversión exitosa.

· **Modo Batch para Cambiar Archivos Indexados a Versión de Archivo Nivel 4.** Los nuevos parámetros CONVERT4 y ATOMICIO han sido agregados al utilitario Define Indexed File (**rmdefinix**) para permitir que un gran número de archivos sea sencillamente convertido a la versión 4 de archivo ya sea con o sin la capacidad atomic I/O. El uso de uno de estos nuevos parámetros ocasionarán que **rmdefinix** se ejecute en "modo batch" sin la normal e interactiva sugerencia (prompt) al usuario.

Licenciamiento Más Flexible

Ahora, el compilador RM/COBOL, el runtime system de RM/COBOL, y el utilitario Indexed File Recovery de RM/COBOL, requieren la misma clase de certificados de licencia que Cobol-WOW y Relativity han estado usando desde algún tiempo. Estos certificados de licencia son personalizados para cada producto y permiten estandartes (banners) especializados, fechas de expiración de productos, y sencillas actualizaciones

Chequeo Automático de Actualización

El compilador RM/COBOL, el runtime system, y CodeWatch ahora proveen información acerca de actualizaciones disponibles de productos automáticamente. CodeWatch provee la información de actualización cuando comienza, mientras que el compilador y el runtime lo hacen cuando están terminando. La nueva palabra clave DISPLAY-UPDATE-MESSAGES ha sido agregada a los registros de configuración COMPILER-OPTIONS y RUN-OPTION para controlar cuales de los mensajes de actualización deben ser desplegados. Es posible ver todos los mensajes de actualización (por defecto para el sistema de desarrollo RM/COBOL) o los mensajes "urgentes" (por defecto para el runtime system RM/COBOL). Los mensajes urgentes son usados para indicar solo problemas importantes que los

usuarios necesitan enterarse de inmediato.

Nuevas Características en RM/COBOL 7.1 para UNIX

Linkeo en Runtime Eliminado.

Versiones de RM/COBOL anteriores a 7.1 requerían que el runtime estuviese relinkeado para agregar nuevas funcionalidades como el VanGui Interface Builder. También había diferentes versiones del runtime, incluyendo la versión terminfo, termcap, el runtime con el RM/InfoExpress client, y el runtime Flexgen.

La versión de RM/COBOL 7.1 para UNIX elimina la necesidad de relinquear el runtime ya que usa módulos de soporte opcionales para agregarle funcionalidad. Con la versión 7.1, habrá sólo una versión simple del runtime con módulos de soporte para proveer las características adicionales. Estos módulos están implementados como objetos UNIX compartidos. Liant provee módulos de soporte para terminfo y termcap interfaces terminales (seleccionables en la instalación), el RM/InfoExpress client interface, y la rutina de interfase de Flexgen con el runtime RM/COBOL. Otros módulos de soporte, como el del VanGui Interface Builder Server y el Enterprise CodeBench están disponibles por Liant. El CodeBridge también poseerá estas facilidades.

Archivo de configuración automática

Versiones UNIX de RM/COBOL anteriores a la 7.1 permitieron que un archivo de configuración fuese linkeado al runtime, compilador o utilidad de recupero. La versión 7.1 of RM/COBOL para UNIX permite que el archivo de configuración sea automáticamente ubicado por el runtime del sistema, el compilador y la utilidad de recuperación. Este nuevo método es llamado " archvo de configuración automática".

Soporte de UNIX para CodeBridge

CodeBridge está ahora disponible para aquellas plataformas UNIX que soportan la versión del runtime 7.1 de RM/COBOL. El CodeBridge Builder genera módulos fuente C que son independientes de la plataforma. Por ejemplo, usted puede usar el CodeBridge Builder sobre una plataforma Windows para generar archivos con código C que pueden ser usados tanto en Windows como en sistemas UNIX.

Archivo de configuración

La palabra ENABLE-LOGGING del RUN-OPTION en el archivo de configuración ha sido ampliada con nuevos valores para controlar la generación de varios errores y archivos log de información.

Todas las modificaciones pertinentes a v7.0

En adición a los cambios listados arriba, RM/COBOL v7.1 para UNIX incluye todos los v7.0 avances que no eran únicos al sistema operativo Windows. Entre estos encontramos:

Características nuevas del runtime: RM/COBOL versión 7.1 para UNIX incluye dos nuevas keywords de configuración, PRINT-ATTR para asistir en la configuración de LINAGE para impresoras de páginas, como laser printers y ink jet printers. Las nuevas keywords son LINAGE-INITIAL-FORM-POSITION and LINAGE-PAGES-PER-PHYSICAL-PAGE.

Agregados a las bibliotecas de funciones de RM/COBOL

C\$Delay: libera la CPU por un tiempo determinado.

C\$GetEnv y C\$SetEnv recupera y setea respectivamente variables de ambiente.

C\$GetLastFileName recupera el último nombre de archivo y el camino usado en un llamado I/O.

C\$GetLastFileOp devuelve información sobre la última operación de I/O realizada por Cobol.

C\$GetRMInfo devuelve información sobre el runtime.

C\$MemoryAllocate reserva memoria dinámica.

C\$GetSysInfo devuelve información sobre el sistema operativo sobre el cual el runtime de RM/COBOL está corriendo.

C\$MemoryDeallocate libera memoria dinámica.

C\$ShowArgs muestra una lista de argumentos usados para llamar a un subprograma.

Nuevas características del compilador

Aumento de la capacidad. RM/COBOL versión 7.1 para UNIX permite 65535 identificadores en un programa simple (hasta 8192 en la versión 6.61). El máximo espacio para palabras definidas por el usuario ha aumentado, tal que todos los 65535 identificadores pueden tener nombres únicos de 30 caracteres de longitud (hasta 21 en la versión 6.61). Los cambios que soportan el aumento de la capacidad también eliminan el problema que limitó las consecutivas líneas de comentarios a 800. El límite es ahora de 18000 consecutivas.

Nuevas palabras reservadas Para soportar algunas características descritas en los siguiente párrafos, las palabras reservadas han sido extendidas con las siguientes nuevas palabras: ADDRESS, CENTURY-DATE, CENTURY-DAY, COUNT-MAX, COUNT-MIN, DATE-AND-TIME, DAY-AND-TIME, NULL, NULLS, y RETURNING. Nótese que si usamos estas palabras como si fuesen definidas por el usuario, debemos cambiar alguna letra o usar el DERESERVE keyword en el COMPILER-OPTIONS registro de configuración.

ACCEPT Un número de cambios han sido hechos al ACCEPT para mejorar la manera en que las fechas y horas son manejadas. Estos cambios proveen maneras adicionales de escribir Y2K. Nuevas frases incluyen CENTURY-DATE, CENTURY-DAY, DATE-AND-TIME, DATE-COMPILED, y DAY-AND-TIME.

Nombres Constantes Constantes pueden ahora ser definidas con nombres constantes declarados en el nivel 78. Una vez definida, el nombre de la constante puede ser usado en casi todos los contextos donde un literal o un entero es requerido. Cuando se usa correctamente, las constantes simplifican enormemente el mantenimiento de los programas COBOL.

Tipos de punteros

El tipo de puntero ha sido agregado. Punteros pueden ser usados para apuntar a otros datos en un programa o en memoria. Soporte para punteros incluye las NULL y NULLS constantes, las cuales son punteros literales con un valor de puntero null. Hay dos formatos nuevos del SET para manipular los punteros y un ADDRESS OF registro especial para obtener la dirección de un puntero. Un puntero no null se refiere al área de memoria que puede ser accedida en COBOL estableciendo la base de direccionamiento en el nivel 01 o 77 descrito en la sección de linqueos del programa. La reserva de memoria dinámica es soportada por dos nuevos subprogramas en la biblioteca, C\$MemoryAllocate y C\$MemoryDeallocate.

Configuración del tamaño de datos binarios

Nuevas características de configuración permiten que tipos numéricos binarios sean localizados con el mínimo requerimiento de espacio para soportar caracteres o cadenas específicos PICTURE con el tamaño configurado. RM/COBOL ha tradicionalmente localizado tipos numéricos binarios como dos, cuatro, ocho o dieciseis bytes. Las nuevas opciones de configuración permiten que los tipos numéricos binarios descritos con uno o dos dígitos sean localizados como un byte simple.

Override de asignación binaria

Además de la configuración de asignación binaria, el compilador ahora soporta una asignación override en USAGE. Dicha asignación es un entero, encerrado entre paréntesis, que sigue un tipo binario (COMPUTATIONAL-4, COMP-4, o BINARIO). El entero especifica el número de bytes a localizar contrarrestando el número de bytes que deberían ser localizados basados en la configuración del compilador actual. La especificación puede contener menos bytes que los requeridos para soportar la precisión decimal indicada en el PICTURE carácter-string.

COUNT, COUNT-MAX, y COUNT-MIN registros especiales El compilador ahora soporta tres nuevos registros especiales, COUNT OF nombre, COUNT-MIN-OF nombre, y COUNT-MAX-OF nombre, que pueden ser usados para obtener el número de ocurrencias de una tabla. Para una tabla fija, COUNT, COUNT-MAX, y COUNT-MIN todas retornan el número de ocurrencias especificadas en OCCURS. Para una tabla variable, COUNT-MIN retorna el mínimo especificado en OCCURS; COUNT retorna el número corriente de ocurrencias (o sea, el valor de DEPENDING especificado en OCCURS) y COUNT-MAX retorna el máximo número de ocurrencias especificadas en OCCURS.

Registro especial LENGTH

El compilador ahora soporta un registro especial LENGTH OF identificador-1, que puede ser usado para obtener la longitud en bytes de cualquier data. La longitud para la mayoría de los datos es constante.

Registro especial PROGRAM-ID El compilador ahora soporta un nuevo registro especial, PROGRAM-ID, el cual puede ser usado para obtener el nombre del programa de cualquiera que especifique este registro.

Comentarios en líneas

El compilador ahora soporta el símbolo *> como introductor de comentario. Este carácter debe ser precedido por un espacio.

Argumentos OMITTED

El compilador ahora soporta la palabra OMITTED en USING de un CALL. Desde que los argumentos son posicionales, esta característica permite que un argumento sea omitido desde otro del final de la lista USING.

GIVING/RETURNING El compilador ahora soporta GIVING/RETURNING en el procedure Division header y en un CALL. Especifica un argumento adicional tratado como el resultado del programa llamado.

Argumentos formales (USING/GIVING). El compilador ahora maneja la especificación de un argumento formal como un argumento en un CALL o en una referencia a un identificador. En estos dos casos, la referencia es evaluada de acuerdo a la descripción del argumento actual correspondiente al argumento formal más que usar la descripción de la sección Linkage. Esto significa que un programa que es un intermediario entre dos programas no necesita tener una sección descriptora Linkage que describa el tamaño del argumento actual que se está pasando. Por ejemplo, llamamos C\$CARG con un argumento formal, el cual es descrito como más largo que el actual argumento. Esto no arrojará error. En lugar, C\$CARG retornará la longitud correcta del argumento actual, por los cambios introducidos. Esto también significa que un programa puede llamar al subprograma C\$CARG con un argumento que el programa principal ha omitido sin obtener un error. En este caso, la llamada a C\$CARG dará

un resultado correcto y retornará el argumento descriptor que incluye el tipo de OMITTED y una longitud de cero. En este caso de modificar la referencia, un argumento omitido causará un error, pero para un argumento que no es omitido, la modificación de referencia puede usar cualquier desplazamiento y combinación de longitud que sea consistente con el argumento actual. Previo a este avance, modificaciones de referencia que usaban variable implicaban una referencia al ítem descrito en la sección Linkage para el argumento formal e implicaba referencia, si era más largo que el argumento correspondiente actual, lo cual causaría un error ante de que la modificación de referencia fuese aplicada.

Nueva lista de formato de fechas

Ahora puede ser configuradas las fechas para incluir años de cuatro dígitos con el nuevo formato DDMMAAAA, MMDDAAAA, AAAAMMDD, y AAAADDD. La configuración por default ha sido cambiada a MMDDAAAA. Este cambio también incluye que el párrafo DATE-COMPILED use el mismo formato de fecha que el header de lista de compilación.

Recuperación de archivos mejorada

Para largos archivos, el utilitario Indexed File Recovery corre al doble de velocidad que la versión anterior. La opción -m ha sido agregada para permitir al usuario especificar una cantidad más grande de memoria a ser usada por el recovery, lo cual resulta en una recuperación mucho más rápida.

Nueva opción de utilidad Rmpgmcom

Una nueva opción en el rmpgmcom, el STRIP, puede remover tablas COBOL y la información de debug producida por la opción Y de compilación de los

archivos objetos.

Nuevas Características en RM/COBOL 7.0 para Windows

RM/COBOL 7 ofrece mejoras sustanciales a los programadores COBOL que quieren desarrollar en ambiente Microsoft Windows. Algunas de ellas son:

- **RM/COBOL 7 incorpora todas las mejoras que se realizaron para los compiladores UNIX en las versiones 6.60 y 6.61.**

- **Mejoras adicionales del compilador**

La versión 6.61 de RM/COBOL introdujo numerosos cambios en el compilador, y la versión 7 aún más, incluyendo los siguientes: Mayor capacidad. RM/COBOL permite definir hasta 65.535 identificadores en un programa (hasta 8.192 en versión 6.61). El máximo espacio para palabras definidas por el usuario ha crecido de forma que los citados 65.535 identificadores podrían tener 30 caracteres de longitud (21 en la versión 6.61). El número máximo de líneas consecutivas de comentarios ha pasado de 800 a 18.000.

Mejora de la sentencia ACCEPT. Se han añadido nuevas opciones en los ACCEPT para manejar fechas y tiempos (algunas de ellas de cara al año 2000). Nuevas frases: CENTURY-DATE, CENTURY-DAY, DATE-AND-TIME, y DAY-AND-TIME.

Nombres de constantes. Se pueden definir ahora constantes por programa mediante declaraciones de nivel 78, para utilizar en cualquier contexto en el que la sintaxis del lenguaje requiera un literal o un entero. Un uso adecuado de constantes simplifica el mantenimiento de programas COBOL

Tipo de datos POINTER. Se ha añadido este tipo de dato para soportar la gestión dinámica de memoria.

Nuevo tipo de datos BYTE. Una opción de configuración permite definir data ítems con uno o dos dígitos para ser almacenados en un solo byte. RM/COBOL tradicionalmente almacena los data ítems binarios en dos, cuatro, ocho o dieciséis bytes dependiendo del número de dígitos decimales indicados en PICTURE.

La nueva opción permite almacenar en un solo byte los data ítems binarios cuya PICTURE indique como máximo dos dígitos decimales.

- **Log de Terminación**

En Windows se puede usar para crear un log de mensajes de error de terminación, incluyendo información de traceback, y juntar información sobre errores que no son atrapados por las declarativas. Las claves ENABLE-LOGGING=TERMINATION y LOGPATH se han agregado a los registros de configuración RUN-OPTION para permitir que se graben los mensajes de error para su posterior análisis.

- **Recover1 más rápido**

En ficheros grandes, Recover1 se ejecuta casi al doble de velocidad que en las versiones anteriores.

- **Novedades en subprogramas**

C\$CLEARDEVELOPMENTMODE deshabilita la información de errores expandida (conocida como "development mode") en muchos subprogramas C\$ y P\$.

C\$DELAY detiene la CPU durante el periodo de tiempo especificado (en segundos).

C\$GETLASTFILENAME recupera el nombre y path del último fichero empleado en una operación de entrada/salida.

C\$GETLASTFILEOP devuelve qué operación de entrada/salida es la última que se ha ejecutado.

C\$SETENV / C\$GETENV para manejo de variables de entorno.

C\$SETSYN / C\$GETSYN para manejos de sinónimos almacenados en el registry.

C\$GETSYSINFO recupera información sobre el sistema operativo en el que se está ejecutando RM/COBOL.

C\$MEMORYALLOCATE asigna memoria dinámica

C\$MEMORYDELLOCATE libera memoria asignada dinámicamente

C\$PLAYSOUND ejecuta los sonidos predefinidos de Windows para eventos o archivos de sonido.

C\$SETDEVELOPMENTMODE habilita información expandida de errores en muchos subprogramas C\$ y P\$.

C\$SHOW setea el estado show de la window principal de RM/COBOL (rmguife.exe)

C\$SHOWARGS muestra la lista de argumentos empleados para llamar a un subprograma.