

IBM® DB2 Universal Database™



Guía de desarrollo de aplicaciones: Creación y ejecución de aplicaciones

Versión 8.2

IBM® DB2 Universal Database™



Guía de desarrollo de aplicaciones: Creación y ejecución de aplicaciones

Versión 8.2

Antes de utilizar esta información y el producto al que da soporte, asegúrese de leer la información general del apartado *Avisos*.

Esta publicación es la traducción del original inglés *IBM DB2 Universal Database, Application Development Guide: Building and Running Applications, Version 8.2, (SC09-4825-01)*.

Este documento contiene información de IBM patentada. Se proporciona sujeto a un contrato de licencia y está protegido por las leyes de copyright. La información contenida en esta publicación no incluye ninguna garantía de producto, y no se debe interpretar como tal ninguna declaración incluida en este manual.

Puede solicitar publicaciones de IBM en línea o a través del representante local de IBM.

- Para solicitar publicaciones en línea, dirjase al Centro de publicaciones de IBM, en www.ibm.com/shop/publications/order
- Para buscar el representante local de IBM, vaya al Directorio de contactos mundiales de IBM, en www.ibm.com/planetwide

Para solicitar publicaciones de DB2 a DB2 Marketing and Sales, en EEUU o Canadá, telefónee al número 1-800-IBM-4YOU (426-4968).

Cuando se envía información a IBM, se otorga a IBM el derecho no exclusivo a utilizar o distribuir la información de la manera que considere oportuna, sin contraer por ello ninguna obligación con el remitente.

© Copyright International Business Machines Corporation 1993 - 2004. Reservados todos los derechos.

Contenido

Bienvenido al desarrollo de aplicaciones de DB2	ix
Productos de DB2 Developer's Edition	ix
Acerca de este manual	xi

Parte 1. Entorno de desarrollo de aplicaciones 1

Capítulo 1. Soporte del entorno DB2 . . . 3

DB2 Application Development Client	3
Instancias del gestor de bases de datos	5
Servidores DB2 soportados	7
Software de desarrollo soportado por DB2	8
Software de desarrollo soportado por AIX	9
Software de desarrollo soportado por HP-UX	11
Software de desarrollo soportado por Linux	13
Software de desarrollo soportado por Solaris	18
Software de desarrollo soportado por Windows	20

Capítulo 2. Configuración. 25

Información general de configuración.	25
Configuración del entorno de desarrollo de aplicaciones UNIX	25
Reconstrucción de bibliotecas compartidas para rutinas de DB2	27
Actualización del archivo de configuración del gestor de bases de datos	27
Configuración del entorno Java.	28
Configuración de las funciones de DB2 WebSphere MQ	30
UNIX	33
Configuración del entorno de desarrollo de aplicaciones UNIX	34
Valores de variables de entorno de UNIX	34
Configuración del entorno Java de UNIX	35
Configuración del entorno Java de AIX	37
Configuración del entorno Java de HP-UX	38
Configuración del entorno Java de Linux	40
Configuración del entorno Java de Solaris	41
Windows	41
Configuración del entorno de desarrollo de aplicaciones Windows	42
Configuración del entorno Java de Windows	45
Base de datos de ejemplo.	48
Configuración de la base de datos de ejemplo.	48
Creación de la base de datos de ejemplo.	48
Creación de la base de datos de ejemplo en servidores de sistema principal o AS/400 e iSeries	50
Catalogación de la base de datos de ejemplo	50
Vinculación de los programas de utilidad de la base de datos de ejemplo	51
Migración de aplicaciones	53
Migración de aplicaciones a DB2 versión 8	53

Migración de aplicaciones, rutinas y applets Java	54
Migración de aplicaciones desde entornos de 32 bits a entornos de 64 bits	56
Cómo asegurarse de que las aplicaciones se puedan transferir	58
Ejecución de aplicaciones en dos versiones de DB2	59
Dónde ir a continuación	63

Capítulo 3. Programas de ejemplo y archivos relacionados 65

Archivos de ejemplo	65
Programas de ejemplo por lenguaje e interfaz de aplicación	71
Ejemplos de C	71
Ejemplos de C++	75
Ejemplos de C#	78
Ejemplos de la CLI	79
Ejemplos de procesador de línea de mandatos (CLP)	82
Ejemplos de COBOL	82
Ejemplos de reconfiguración dinámica	86
Ejemplos de JDBC	87
Ejemplos de SQLJ	90
Ejemplos de Java WebSphere	92
Ejemplos de plugins de Java.	93
Ejemplos de salidas del usuario para la gestión de anotaciones cronológicas	94
Ejemplos de OLE (Object Linking and Embedding)	96
Ejemplos de funciones de tabla de OLE DB (base de datos de enlace e integración de objetos)	97
Ejemplos de Perl	97
Ejemplos de PHP	98
Ejemplos de REXX	100
Ejemplos de plugins de seguridad	102
Ejemplos de procedimientos de SQL.	102
Ejemplos de Visual Basic	105
Ejemplos de Visual Basic .NET	107
Ejemplos de Visual C++.	108
Ejemplos de Windows Management Instrumentation	109
Archivos de creación, makefiles y programas de utilidad de comprobación de errores.	110
Archivos de creación	110
Makefiles	113
Programas de utilidad de comprobación de errores.	116

Parte 2. Creación y ejecución de aplicaciones independientemente de la plataforma 119

Capítulo 4. Java 121

Programas Java de ejemplo	121
Consideraciones sobre los applets Java	122
JDBC	124
Creación de applets JDBC	124
Creación de aplicaciones JDBC	126
Creación de rutinas JDBC	126
SQLJ	128
Creación de programas SQLJ	128
Creación de applets SQLJ	129
Creación de aplicaciones SQLJ	131
Script de creación de UNIX para aplicaciones y applets SQLJ	132
Opciones de aplicaciones y applets SQLJ para UNIX	133
Script de creación de Windows para aplicaciones y applets SQLJ	134
Opciones de aplicaciones y applets SQLJ para Windows	135
Creación de rutinas SQLJ	136
Script de creación de UNIX para rutinas SQLJ	138
Opciones de rutinas SQLJ para UNIX	139
Archivo de proceso por lotes de Windows para rutinas SQLJ	139
Opciones de rutinas SQLJ para Windows	141
Capítulo 5. El procesador de línea de mandatos	143
Ejecución de scripts del procesador de línea de mandatos (CLP)	143
Llamada a procedimientos desde el Procesador de línea de mandatos (CLP)	144
Capítulo 6. Procedimientos de SQL	147
Creación de procedimientos SQL	147
Invocación de procedimientos SQL mediante aplicaciones cliente	148
Personalización de las opciones de precompilación y vinculación para procedimientos SQL	149
Copia de seguridad y restauración de procedimientos SQL creados antes de DB2 8.2	151
Revinculación de procedimientos SQL	152
Capítulo 7. Perl	153
Creación de aplicaciones Perl	153
Capítulo 8. PHP	155
Creación de aplicaciones PHP	155
<hr/>	
Parte 3. Creación y ejecución de aplicaciones específicas de una plataforma	159
Capítulo 9. UNIX	161
Creación de aplicaciones C de UNIX	161
Creación de aplicaciones C de conexión múltiple de UNIX	163
Creación de rutinas C de UNIX	165
Creación de aplicaciones C++ de UNIX	168

Creación de aplicaciones C++ de conexión múltiple de UNIX	170
Creación de rutinas C++ de UNIX	172
Creación de aplicaciones Micro Focus COBOL de UNIX	176
Creación de rutinas Micro Focus COBOL de UNIX	177
Capítulo 10. AIX	181
Consideraciones importantes	181
Archivos de exportación de AIX para rutinas	181
Las rutinas de AIX y la sentencia CREATE	182
Sustitución de una biblioteca compartida de AIX	183
Consideraciones para la instalación de COBOL en AIX	183
IBM C	184
Script de creación para aplicaciones C	184
Opciones de compilación y enlace para aplicaciones C de AIX	185
Script de creación para rutinas C	186
Opciones de compilación y enlace para rutinas C de AIX	186
Creación de aplicaciones multihebra C en AIX	187
VisualAge C++	188
Script de creación para aplicaciones C++	188
Opciones de compilación y enlace para aplicaciones C++ de AIX	189
Script de creación para rutinas C++	190
Opciones de compilación y enlace para rutinas C++ de AIX	191
Creación de aplicaciones multihebra C++ en AIX	192
Archivos de configuración VisualAge C++.	193
Creación de programas de VisualAge C++ con archivos de configuración	193
Creación de aplicaciones C++ de la API de DB2 con archivos de configuración	194
Creación de aplicaciones C++ de SQL incorporado con archivos de configuración	195
Creación de procedimientos almacenados C++ con archivos de configuración	196
Creación de funciones definidas por el usuario C++ con archivos de configuración	197
IBM COBOL Set para AIX	198
Configuración del compilador IBM COBOL en AIX	198
Creación de aplicaciones IBM COBOL en AIX	199
Script de creación para aplicaciones COBOL DE IBM	201
Opciones de compilación y enlace para aplicaciones IBM COBOL de AIX	201
Creación de rutinas IBM COBOL en AIX	202
Script de creación para rutinas COBOL de IBM	203
Opciones de compilación y enlace para rutinas IBM COBOL de AIX	204
COBOL para Micro Focus	205
Configuración del compilador Micro Focus COBOL en AIX	205
Script de creación para aplicaciones COBOL para Micro Focus	206
Opciones de compilación y enlace para aplicaciones Micro Focus COBOL de AIX	207

Script de creación para rutinas COBOL para Micro Focus	207	Script de creación para aplicaciones COBOL para Micro Focus	238
Opciones de compilación y enlace para rutinas Micro Focus COBOL de AIX	208	Opciones de compilación y enlace para aplicaciones Micro Focus COBOL de Linux	239
REXX	209	Script de creación para rutinas COBOL para Micro Focus	240
Creación de aplicaciones REXX en AIX	209	Opciones de compilación y enlace para rutinas Micro Focus COBOL de Linux	240
Capítulo 11. HP-UX	211	Capítulo 13. Solaris	243
HP-UX C	211	C para Solaris	243
Script de creación para aplicaciones C	211	Script de creación para aplicaciones C	243
Opciones de compilación y enlace para aplicaciones C de HP-UX	212	Opciones de compilación y enlace para aplicaciones C de Solaris	244
Script de creación para rutinas C	214	Script de creación para rutinas C	245
Opciones de compilación y enlace para rutinas C de HP-UX.	215	Opciones de compilación y enlace para rutinas C de Solaris	246
Creación de aplicaciones multihebra C en HP-UX	216	Creación de aplicaciones multihebra C en Solaris.	247
HP-UX C++	216	C++ para Solaris	248
Script de creación para aplicaciones C++	216	Script de creación para aplicaciones C++	248
Opciones de compilación y enlace para aplicaciones C++ de HP-UX	218	Opciones de compilación y enlace para aplicaciones C++ de Solaris.	249
Script de creación para rutinas C++	219	Script de creación para rutinas C++	250
Opciones de compilación y enlace para rutinas C++ de HP-UX.	220	Opciones de compilación y enlace para rutinas C++ de Solaris	251
Creación de aplicaciones multihebra C++ en HP-UX	222	Creación de aplicaciones multihebra C++ en Solaris.	252
COBOL para Micro Focus	222	COBOL para Micro Focus	253
Configuración del compilador Micro Focus COBOL en HP-UX.	223	Configuración del compilador Micro Focus COBOL en Solaris	253
Script de creación para aplicaciones COBOL para Micro Focus	223	Script de creación para aplicaciones COBOL para Micro Focus	254
Opciones de compilación y enlace para aplicaciones Micro Focus COBOL de HP-UX	224	Opciones de compilación y enlace para aplicaciones Micro Focus COBOL de Solaris	254
Script de creación para rutinas COBOL para Micro Focus	224	Script de creación para rutinas COBOL para Micro Focus	255
Opciones de compilación y enlace para rutinas Micro Focus COBOL de HP-UX	225	Opciones de compilación y enlace para rutinas Micro Focus COBOL de Solaris	256
Capítulo 12. Linux	227	Capítulo 14. Windows	257
Linux C	227	La opción de precompilación WCHARTYPE CONVERT	257
Script de creación para aplicaciones C	227	Funciones de tabla de base de datos OLE DB (base de datos de enlace e integración de objetos)	258
Opciones de compilación y enlace para aplicaciones C de Linux	228	Windows Management Instrumentation (WMI)	259
Script de creación para rutinas C	229	Microsoft Visual Basic	260
Opciones de compilación y enlace para rutinas C de Linux	230	Creación de aplicaciones ADO con Visual Basic	260
Creación de aplicaciones multihebra C en Linux C++ para Linux	231	Creación de transacciones emparejadas débilmente con Visual Basic	262
Script de creación para aplicaciones C++	232	Resolución de problemas de un proyecto de transacción emparejada débilmente de Visual Basic	264
Opciones de compilación y enlace para aplicaciones C++ de Linux	233	Creación de aplicaciones RDO con Visual Basic	266
Script de creación para rutinas C++	234	Automatización del enlace e integración de objetos (OLE) con Visual Basic.	267
Opciones de compilación y enlace para rutinas C++ de Linux	235	.NET	268
Creación de aplicaciones multihebra C++ en Linux	236	Creación de aplicaciones C#.NET	268
COBOL para Micro Focus	237	Archivo de proceso por lotes para aplicaciones C#.NET	269
Configuración del compilador Micro Focus COBOL en Linux	237		

	Opciones de compilación y enlace para aplicaciones C# .NET	270
	Creación de aplicaciones Visual Basic .NET	272
	Archivo de proceso por lotes para aplicaciones Visual Basic .NET	273
	Opciones de compilación y enlace para aplicaciones Visual Basic .NET.	274
	Creación de rutinas CLR (Common Language Runtime) .NET	276
	Archivo de proceso por lotes para rutinas C# .NET	279
	Archivo de proceso por lotes para rutinas Visual Basic .NET	279
	Opciones de compilación y enlace para rutinas de CLR .NET	279
	Microsoft Visual C++.	280
	Creación de aplicaciones ADO con Visual C++	281
	Automatización del enlace e integración de objetos (OLE) con Visual C++	282
	Creación de aplicaciones C/C++ en Windows	283
	Archivo de proceso por lotes para aplicaciones C/C++	285
	Opciones de compilación y enlace para aplicaciones C/C++ de Windows.	286
	Creación de rutinas C/C++ en Windows	287
	Archivo de proceso por lotes para rutinas C/C++	290
	Opciones de compilación y enlace para rutinas C/C++ de Windows	291
	Creación de aplicaciones C/C++ de conexión múltiple en Windows.	292
	IBM VisualAge COBOL	295
	Configuración del compilador IBM COBOL en Windows.	295
	Creación de aplicaciones IBM COBOL en Windows.	296
	Archivo de proceso por lotes para aplicaciones COBOL de IBM	298
	Opciones de compilación y enlace para aplicaciones IBM COBOL de Windows	299
	Creación de rutinas IBM COBOL en Windows	300
	Archivo de proceso por lotes para rutinas COBOL de IBM	301
	Opciones de compilación y enlace para rutinas IBM COBOL de Windows	302
	Micro Focus COBOL	303
	Configuración del compilador Micro Focus COBOL en Windows	303
	Creación de aplicaciones Micro Focus COBOL en Windows.	304
	Archivo de proceso por lotes para aplicaciones COBOL para Micro Focus	306
	Opciones de compilación y enlace para aplicaciones Micro Focus COBOL de Windows	306
	Creación de rutinas Micro Focus COBOL en Windows	307
	Archivo de proceso por lotes para rutinas COBOL para Micro Focus	308
	Opciones de compilación y enlace para rutinas Micro Focus COBOL de Windows	309
	Object REXX	309

Creación de aplicaciones Object REXX en Windows.	309
--	-----

Parte 4. Apéndices 311

Apéndice A. Información técnica sobre DB2 Universal Database 313

	Documentación y ayuda de DB2	313
	Actualizaciones de la documentación de DB2	313
	Centro de información de DB2	314
	Escenarios de instalación del Centro de información de DB2	316
	Instalación del Centro de información de DB2 utilizando el asistente de instalación de DB2 (UNIX)	318
	Instalación del Centro de información de DB2 utilizando el asistente de instalación de DB2 (Windows)	321
	Invocación del Centro de información de DB2	323
	Actualización del Centro de información de DB2 instalado en el sistema o en un servidor de intranet	324
	Visualización de temas en el idioma preferido en el Centro de información de DB2	325
	Documentación PDF e impresa de DB2.	326
	Información básica de DB2	326
	Información de administración	327
	Información para el desarrollo de aplicaciones	327
	Información de Business Intelligence	328
	Información de DB2 Connect	328
	Información de iniciación	329
	Información de aprendizaje.	329
	Información sobre componentes opcionales	330
	Notas del release	330
	Impresión de manuales de DB2 desde archivos PDF	331
	Solicitud de manuales de DB2 impresos	332
	Invocación de ayuda según contexto desde una herramienta de DB2	332
	Invocación de la ayuda de mensajes desde el procesador de línea de mandatos.	334
	Invocación de la ayuda de mandatos desde el procesador de línea de mandatos.	334
	Invocación de la ayuda para estados de SQL desde el procesador de línea de mandatos	335
	Guías de aprendizaje de DB2	335
	Información de resolución de problemas de DB2	336
	Accesibilidad	337
	Entrada de teclado y navegación	337
	Pantalla accesible	337
	Compatibilidad con tecnologías de asistencia	338
	Documentación accesible	338
	Diagramas de sintaxis en formato decimal con puntos.	338
	Certificación Common Criteria de productos DB2 Universal Database	340

Apéndice B. Avisos 341

	Marcas registradas.	343
--	-----------------------------	-----

Índice. 345

Cómo ponerse en contacto con IBM	351	Información sobre productos	351
---	------------	---------------------------------------	-----

Bienvenido al desarrollo de aplicaciones de DB2

Productos de DB2 Developer's Edition ix | Acerca de este manual xi

En este prefacio se proporciona información necesaria para iniciar el desarrollo de aplicaciones DB2, específicamente mediante los productos DB2 Developer's Edition. También se brinda una visión general de los tres volúmenes de la Guía para el desarrollo de aplicaciones (Application Development Guide).

Para obtener instrucciones sobre cómo instalar un producto que esté disponible con DB2 Developer's Edition, consulte la publicación *Guía rápida de iniciación* adecuada, que estará disponible en el CD de PDF, o busque las instrucciones de instalación en el propio CD del producto.

Si desea acceder a la documentación de DB2 en el sistema y todavía no ha instalado el Centro de información de DB2, consulte los apartados "Instalación del Centro de información de DB2 utilizando el asistente de instalación de DB2 (UNIX)" en la página 318 o "Instalación del Centro de información de DB2 utilizando el asistente de instalación de DB2 (Windows)" en la página 321. El Centro de información de DB2 contiene documentación para DB2 Universal Database y para los productos relacionados con DB2.

Productos de DB2 Developer's Edition

DB2[®] Universal Database proporciona dos paquetes de productos para el desarrollo de aplicaciones: DB2 Personal Developer's Edition y DB2 Universal Developer's Edition. Personal Developer's Edition proporciona los productos DB2 Universal Database[™] y DB2 Connect[™] Personal Edition que se ejecutan en sistemas operativos Linux y Windows[®]. DB2 Universal Developer's Edition proporciona productos DB2 en estas plataformas, así como en AIX[®], HP-UX y Solaris Operating Environment. Para ver una lista completa de las plataformas soportadas, póngase en contacto con el representante de IBM[®].

Utilizando el software que acompaña a estos productos, podrá desarrollar y crear aplicaciones que se ejecuten en un sistema operativo y acceder a bases de datos que se encuentren en el mismo sistema operativo o en otro. Por ejemplo, puede crear una aplicación que se ejecute en un sistema operativo Windows pero acceda a una base de datos que se encuentre en una plataforma UNIX[®], por ejemplo AIX. Para conocer los términos y condiciones de uso de los productos de Developer's Edition, consulte el Contrato de licencia.

Personal Developer's Edition contiene varios CD-ROM con todo el código necesario para desarrollar y probar las aplicaciones. En cada paquete de soportes, encontrará:

- Los CD-ROM de productos DB2 Universal Database para sistemas operativos Linux y Windows. Cada CD-ROM contiene el servidor DB2 y el Cliente de desarrollo de aplicaciones para el sistema operativo soportado. Estos CD-ROM se le brindan para que pruebe las aplicaciones. Si tiene necesidad de instalar y utilizar una base de datos, tiene que obtener una licencia válida adquiriendo el producto Universal Database.

- DB2 Connect Personal Edition. Si tiene necesidad de instalar y utilizar este producto, tiene que obtener una licencia válida adquiriendo el producto DB2 Connect Personal Edition.
- Un CD-ROM de publicaciones de DB2 que contiene libros de DB2 en formato PDF
- Un CD-ROM de Centro de información de DB2 que contiene documentación de DB2 en formato HTML
- DB2 Net Search Extender (sólo Windows)
- DB2 Spatial Extender (sólo Windows)
- VisualAge[®] para Java[™], Entry Edition

Universal Developer's Edition contiene CD-ROM para todos los sistemas operativos soportados por DB2, que incluyen lo siguiente:

- DB2 Universal Database Personal Edition, Workgroup Server Edition y Enterprise Server Edition
- DB2 Connect Personal Edition y DB2 Connect Enterprise Edition
- Clientes de administración para todas las plataformas. Estos clientes contienen herramientas para administrar bases de datos, como por ejemplo el Centro de control y el Analizador de sucesos. Estos clientes también permiten ejecutar aplicaciones en cualquier sistema.
- Clientes de desarrollo de aplicaciones para todas las plataformas. Estos clientes tienen herramientas para el desarrollo de aplicaciones, programas de ejemplo y archivos de cabecera. Cada cliente AD de DB2 incluye todo lo necesario para desarrollar las aplicaciones.
- Clientes de tiempo de ejecución para todas las plataformas. Una aplicación se puede ejecutar desde un cliente de tiempo de ejecución en cualquier sistema. El cliente de tiempo de ejecución no tiene algunas de las funciones del cliente de administración, como por ejemplo el Analizador de sucesos y el Centro de control de DB2, por lo cual ocupa menos espacio.
- Un CD-ROM de publicaciones de DB2 que contiene libros de DB2 en formato PDF
- Un CD-ROM de Centro de información de DB2 que contiene documentación de DB2 en formato HTML
- DB2 Net Search Extender
- DB2 Spatial Extender

Además, para ambos Developer's Editions se obtienen copias de otro software que puede resultar de utilidad para desarrollar aplicaciones. Este software puede variar de vez en cuando y va acompañado de contratos de licencia para su uso.

Conceptos relacionados:

- "DB2 UDB Enterprise Server Edition" en el manual *Guía rápida de iniciación para servidores DB2*
- "DB2 Run-Time Client" en el manual *Guía rápida de iniciación para clientes DB2*
- "DB2 Administration Client" en el manual *Guía rápida de iniciación para clientes DB2*
- "DB2 Workgroup Server Edition" en el manual *Guía rápida de iniciación para servidores DB2*

Tareas relacionadas:

- “Instalación de DB2 Personal Edition - visión general (Windows)” en el manual *Guía rápida de iniciación para DB2 Personal Edition*
- “Instalación de DB2 Personal Edition - visión general (Linux)” en el manual *Guía rápida de iniciación para DB2 Personal Edition*

Información relacionada:

- “DB2 Application Development Client” en la página 3

Acerca de este manual

El manual *Guía de desarrollo de aplicaciones* es un libro de tres volúmenes que describe lo que tiene que saber sobre codificación, depuración, creación y ejecución de aplicaciones DB2:

- El manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente* contiene lo que tiene que saber para codificar aplicaciones DB2 autónomas que se ejecutan en clientes DB2. Incluye información sobre:
 - Interfaces de programación que reciben soporte de DB2. Se proporcionan descripciones de alto nivel para DB2 Developer’s Edition, interfaces de programación soportadas, recursos para crear aplicaciones Web y funciones de programación proporcionadas por DB2, como rutinas y desencadenantes.
 - La estructura general que debe seguir una aplicación DB2. Se proporcionan recomendaciones sobre cómo mantener valores de datos y relaciones en la base de datos, se describen consideraciones sobre autorización y se proporciona información sobre cómo probar y depurar la aplicación.
 - SQL incorporado, tanto dinámico como estático. Se describen consideraciones generales sobre SQL incorporado, así como temas específicos que se aplican al uso de SQL estático y dinámico en aplicaciones DB2.
 - Lenguajes interpretados y de sistema principal, como C/C++, COBOL, Perl y REXX, y cómo utilizar SQL incorporado en aplicaciones escritas en estos lenguajes.
 - El Proveedor de datos DB2 .NET, así como los proveedores de datos OLE DB .NET y ODBC .NET.
 - Java (tanto JDBC como SQLJ) y consideraciones para crear aplicaciones de Java que se utilicen en los productos WebSphere Application Server.
 - IBM OLE DB Provider para Servidores DB2. Se proporciona información general sobre el soporte de IBM OLE DB para servicios, componentes y propiedades de OLE DB. Asimismo, se brinda información específica sobre aplicaciones de Visual Basic y Visual C++ que utilizan la interfaz OLE DB para ActiveX Data Objects (ADO).
 - Aspectos de soporte de idiomas nacionales. Se describen temas generales, como por ejemplo las secuencias de clasificación, la deducción de páginas de códigos y las conversiones de caracteres. También se describen aspectos más específicos, como por ejemplo las páginas de códigos DBCS, los juegos de caracteres EUC y aspectos que se aplican a los entornos de japonés y chino tradicional EUC y UCS-2.
 - Gestión de transacciones. Se describen aspectos que afectan a las aplicaciones que realizan actualizaciones para varios sitios, así como a las aplicaciones que realizan transacciones simultáneas.
 - Aplicaciones en entornos de base de datos particionada. Se describen DSS dirigidos, elusiones locales, inserciones en almacenamiento intermedio y resolución de problemas en las aplicaciones que se encuentran en entornos de base de datos particionada.

- Técnicas de aplicación utilizadas normalmente. Se proporciona información sobre cómo utilizar columnas generadas y de identidad, tablas temporales declaradas y cómo usar puntos de grabación para gestionar transacciones.
- Las sentencias de SQL cuyo uso se soporta en aplicaciones de SQL incorporado.
- Aplicaciones que acceden a entornos de sistema principal e iSeries. Se describen los aspectos relativos a las aplicaciones de SQL incorporado que acceden a entornos de sistema principal e iSeries.
- La simulación de la clasificación binaria EBCDIC.
- La publicación *Guía de desarrollo de aplicaciones: Programación de aplicaciones de servidor* contiene lo que hay que saber sobre la programación en que se utilizan objetos de la parte del servidor, los cuales incluyen rutinas, objetos grandes, tipos definidos por el usuario y activadores. Incluye información sobre los temas siguientes:
 - Rutinas (procedimientos almacenados, funciones definidas por el usuario y métodos), que incluye:
 - Rendimiento de rutinas, seguridad, consideraciones sobre la gestión de bibliotecas y restricciones.
 - Creación de rutinas, incluidas rutinas externas, y la sentencia CREATE.
 - Modalidades de parámetros de procedimientos y manejo de parámetros.
 - Conjuntos de resultados de procedimientos.
 - Procedimientos de SQL, incluidos la depuración y manejo de condiciones.
 - Funciones de tabla y escalares definidas por el usuario.
 - Llamadas de funciones de tabla y escalares definidas por el usuario (FIRST call, FINAL call,...) y áreas reutilizables.
 - Métodos.
 - Autorizaciones y enlace de rutinas externas.
 - Consideraciones específicas del lenguaje para rutinas de C, de Java, de ejecución en el lenguaje común .NET y de automatización de OLE.
 - Invocación de rutinas.
 - Selección de función.
 - Pase de tipos diferenciados y LOB a las funciones.
 - Páginas de códigos y rutinas.
 - Objetos grandes, que incluyen uso de LOB y localizadores, variables de referencia y datos CLOB.
 - Tipos diferenciados definidos por el usuario, que incluyen tipificación estricta, definición y eliminación de UDT, creación de tablas con tipos estructurados, utilización de tipos diferenciados y tablas con tipo para aplicaciones específicas, manipulación de tipos diferenciados y difusión entre los mismos y realización de comparaciones y asignaciones con tipos diferenciados, que incluyen operaciones UNION sobre columnas con tipo de forma diferenciada.
 - Tipos estructurados definidos por el usuario, que incluyen almacenamiento de instancias y creación de instancias, jerarquías de tipos estructurados, definición del comportamiento de los tipos estructurados, distribución dinámica de métodos, funciones de comparación, difusión y constructor y métodos mutador y observador correspondientes a tipos estructurados.
 - Tablas con tipo, que incluyen creación, eliminación, sustitución y almacenamiento de objetos, definición de identificadores de objetos generados por el sistema y restricciones en las columnas de identificador.

- Tipos de referencia, que incluyen relaciones entre objetos de tablas con tipo, relaciones semánticas con referencias e integridad referencial frente a referencias de ámbito.
- Tablas y vistas con tipo, que incluyen tipos estructurados como tipos de columnas, funciones de transformación y grupos de transformación, correlaciones de programas de lenguaje principal y variables del lenguaje principal de tipos estructurados.
- Desencadenantes, que incluyen los desencadenantes INSERT, UPDATE y DELETE, interacciones con restricciones referenciales, líneas generales de creación, granularidad, hora de activación, tablas y variables de transición, acciones desencadenadas, varios desencadenantes y sinergia entre desencadenantes, restricciones y rutinas.
- La publicación *Guía de desarrollo de aplicaciones: Creación y ejecución de aplicaciones* contiene lo que hay que saber para crear y ejecutar aplicaciones de DB2 en los sistemas operativos soportados por DB2:
 - AIX
 - HP-UX
 - Linux
 - Solaris
 - Windows

Incluye información sobre los temas siguientes:

- Software y servidores para crear aplicaciones soportados por DB2, que incluye los compiladores e intérpretes soportados.
- Los archivos de programas de ejemplo de DB2, makefiles, archivos de creación y archivos de programas de utilidad para la comprobación de errores.
- Cómo configurar el entorno de desarrollo de aplicaciones, lo que incluye instrucciones específicas para funciones de Java y WebSphere MQ.
- Cómo configurar la base de datos de ejemplo.
- Cómo migrar las aplicaciones desde versiones anteriores de DB2.
- Cómo crear y ejecutar applets, aplicaciones y rutinas de Java.
- Cómo crear y ejecutar procedimientos de SQL.
- Cómo crear y ejecutar aplicaciones y rutinas en C/C++.
- Cómo crear y ejecutar aplicaciones y rutinas en COBOL de IBM y de Micro Focus.
- Cómo crear y ejecutar aplicaciones de REXX en AIX y Windows.
- Cómo crear y ejecutar aplicaciones en C# y Visual Basic .NET y rutinas en CLR .NET en Windows.
- Cómo crear y ejecutar aplicaciones con ActiveX Data Objects (ADO) utilizando Visual Basic y Visual C++ en Windows.
- Cómo crear y ejecutar aplicaciones con objetos de datos remotos mediante Visual C++ en Windows.

Parte 1. Entorno de desarrollo de aplicaciones

Capítulo 1. Soporte del entorno DB2

DB2 Application Development Client	3	Software de desarrollo soportado por HP-UX	11
Instancias del gestor de bases de datos	5	Software de desarrollo soportado por Linux	13
Servidores DB2 soportados	7	Software de desarrollo soportado por Solaris	18
Software de desarrollo soportado por DB2	8	Software de desarrollo soportado por Windows	20
Software de desarrollo soportado por AIX	9		

Este volumen de la Guía de desarrollo de aplicaciones describe el soporte de DB2 para el desarrollo de aplicaciones. Proporciona la información que el usuario necesita para configurar el entorno para desarrollar aplicaciones DB2, y da instrucciones paso a paso para compilar, enlazar y ejecutar estas aplicaciones en ese entorno. Explica cómo crear aplicaciones utilizando DB2 Application Development (DB2 AD) Client para DB2 Universal Database Versión 8.2 en las plataformas siguientes:

- AIX
- HP-UX
- Linux
- Solaris Operating Environment
- Windows

DB2 Application Development Client

DB2 Application Development (DB2 AD) Client proporciona las herramientas y el entorno necesarios para desarrollar aplicaciones que acceden a servidores DB2 y servidores de aplicaciones que implantan Distributed Relational Database Architecture (DRDA).

El usuario puede crear y ejecutar aplicaciones DB2 con un DB2 AD Client instalado. También puede ejecutar aplicaciones DB2 en estos clientes DB2:

- DB2 Run-Time Client
- DB2 Administration Client

Los DB2 AD Client para las plataformas soportadas incluyen lo siguiente:

- **Precompiladores para C/C++, COBOL y Fortran** (siempre que el lenguaje esté soportado para esa plataforma).
- **Soporte para aplicaciones de SQL incorporado**, que comprende bibliotecas de programación, archivos de inclusión y ejemplos de código.
- **Soporte para aplicaciones de la CLI de DB2 (Interfaz de Nivel de Llamada de DB2)**, que incluye bibliotecas de programación, archivos de inclusión y ejemplos de código para desarrollar aplicaciones que se pueden transferir fácilmente a ODBC y compilar con un SDK de ODBC. Microsoft proporciona un SDK de ODBC para sistemas operativos Windows; otros fabricantes proporcionan un SDK para muchas de las demás plataformas soportadas. Para los sistemas operativos Windows, los clientes DB2 contienen un controlador ODBC que da soporte a las aplicaciones desarrolladas con el Software Developer's Kit para ODBC de Microsoft. Para todas las demás plataformas, los clientes DB2 contienen un controlador ODBC, de instalación opcional, que da soporte a las aplicaciones que se pueden desarrollar con un SDK de ODBC para esa plataforma, si ese SDK existe. Sólo los Clientes DB2 para sistemas operativos Windows contienen un gestor de controladores ODBC.

- **DB2 Java Enablement**, que incluye soporte de DB2 Java Database Connectivity (DB2 JDBC) para desarrollar aplicaciones y applets Java, y soporte de SQL incorporado para Java de DB2 (DB2 SQLJ) para desarrollar aplicaciones y applets Java de SQL incorporado.
- **Java Development Kit**, o equivalente, que se suministra con DB2 para todos los sistemas operativos soportados. Para conocer detalles específicos de la versión, consulte el apartado sobre "software de desarrollo soportado" correspondiente a su sistema operativo:
 - "Software de desarrollo soportado por AIX" en la página 9
 - "Software de desarrollo soportado por HP-UX" en la página 11
 - "Software de desarrollo soportado por Linux" en la página 13
 - "Software de desarrollo soportado por Solaris" en la página 18
 - "Software de desarrollo soportado por Windows" en la página 20
- **Rutinas (UDF y procedimientos almacenados) de automatización para ActiveX Data Objects (ADO) y Object Linking and Embedding (OLE)** en sistemas operativos Windows, que incluye ejemplos de código implantados en Microsoft Visual Basic y Microsoft Visual C++. También, ejemplos de código con Remote Data Objects (RDO) implementado en Microsoft Visual Basic.
- **Funciones de tabla de Object Linking and Embedding Database (OLE DB)** en sistemas operativos Windows.
- **Aplicaciones en C# y Visual Basic .NET y rutinas en CLR .NET** en sistemas operativos Windows.
- **Centro de Desarrollo de DB2**, aplicación gráfica para el desarrollo rápido de rutinas (procedimientos almacenados y funciones definidas por el usuario), y tipos estructurados. El Centro de Desarrollo proporciona un entorno de desarrollo único que da soporte a la familia completa de productos DB2, desde la estación de trabajo hasta z/OS. Puede iniciar el Centro de Desarrollo como aplicación autónoma o desde un centro de DB2 Universal Database, tal como el Centro de Control, el Editor de mandatos o el Centro de Tareas. El Centro de Desarrollo se implementa con Java, y todas las conexiones de base de datos se gestionan utilizando una API de JDBC (Java Database Connectivity). El Centro de Desarrollo también proporciona una opción de desarrollo DB2 para los siguientes entornos de desarrollo:
 - Microsoft Visual C++, Versión 6
 - Microsoft Visual Basic, Versión 6
 - Microsoft Visual InterDev, Versión 6
- **SQL Interactivo** mediante el Editor de Mandatos o el Procesador de Línea de Mandatos (CLP) para probar sentencias de SQL o realizar consultas ad hoc sobre la base de datos.
- **Un conjunto de Interfaces de Programación de Aplicaciones documentadas** para permitir que otras herramientas de desarrollo de aplicaciones implementen soporte de precompilador para DB2 directamente dentro de sus productos. Por ejemplo, IBM COBOL sobre AIX hace uso de esta interfaz. Puede encontrar información sobre las API de los Servicios de Precompilador en el archivo PDF, prepapi.pdf, situado en este sitio Web de desarrollo de aplicaciones de DB2:
 - <http://www.ibm.com/software/data/db2/udb/ad>
- **Un indicador de conformidad con SQL92 y MVS**, que identifica sentencias de SQL incorporado contenidas en aplicaciones que no cumplen el estándar SQL92 Entry Level de ISO/ANSI, o que no están soportadas por DB2 UDB para z/OS y OS/390. Si el usuario migra aplicaciones desarrolladas en una estación de trabajo hacia otra plataforma, el Indicador le ahorra tiempo al mostrarle las incompatibilidades de sintaxis.

Información relacionada:

- “Mandato PRECOMPILE” en el manual *Consulta de mandatos*
- “Software de desarrollo soportado por AIX” en la página 9
- “Software de desarrollo soportado por HP-UX” en la página 11
- “Software de desarrollo soportado por Linux” en la página 13
- “Software de desarrollo soportado por Solaris” en la página 18
- “Software de desarrollo soportado por Windows” en la página 20

Instancias del gestor de bases de datos

DB2® da soporte a varias instancias del gestor de bases de datos en la misma máquina. Una instancia del gestor de bases de datos tiene sus propios archivos de configuración, directorios y bases de datos.

Cada instancia del gestor de bases de datos puede gestionar varias bases de datos. Sin embargo, una base de datos determinada pertenece a una sola instancia. La figura siguiente muestra esta relación.

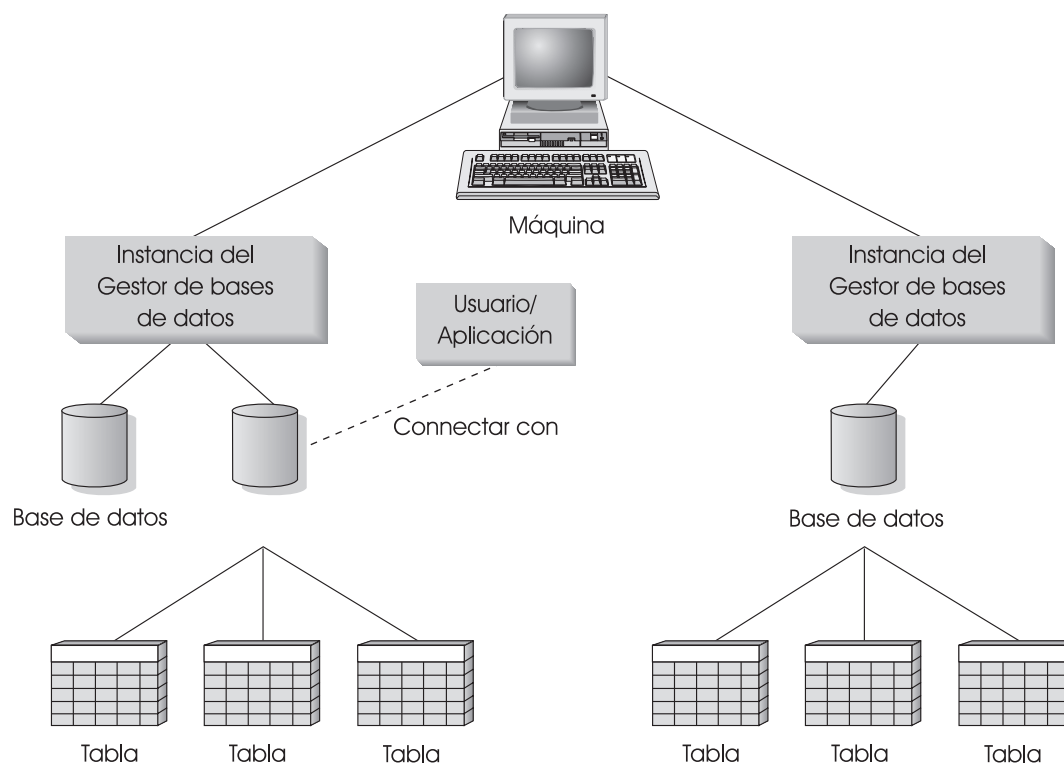


Figura 1. Instancias del gestor de bases de datos

Las instancias del gestor de bases de datos proporcionan al usuario la flexibilidad para disponer de varios entornos de base de datos en la misma máquina. Por ejemplo, puede tener una instancia del gestor de bases de datos para tareas de desarrollo y otra instancia con fines de producción.

Con servidores UNIX® Enterprise Server Edition (ESE) puede tener versiones diferentes de DB2 en distintas instancias del gestor de bases de datos. Por ejemplo, puede tener una instancia del gestor de bases de datos que ejecute DB2 Universal Database Versión 7.1 y otra que ejecute DB2 Universal Database Versión 8.2. En las

versiones anteriores a DB2 Versión 8, dentro de un nivel de versión sólo se da soporte a un único release y nivel de FixPak. Por ejemplo, DB2 Versión 7.1 y DB2 Versión 7.2 no pueden coexistir en un servidor UNIX. Con DB2 Versión 8 pueden coexistir varios niveles de FixPak en el mismo servidor UNIX, como por ejemplo DB2 Versión 8.1 y DB2 Versión 8.2.

Cuando se utilizan servidores de Windows®, cada instancia del gestor de bases de datos debe contener la misma versión, release y nivel de FixPak de DB2. No puede tener una instancia del gestor de bases de datos ejecutando DB2 Universal Database Versión 7.1, y otra instancia ejecutando DB2 Universal Database Versión 8.2.

Es necesario que conozca lo siguiente para cada instancia que utilice:

nombre de instancia

En las plataformas UNIX, esto es un nombre de usuario válido que el usuario especifica al crear la instancia del gestor de bases de datos.

En los sistemas operativos Windows, esto es una cadena alfanumérica de 8 caracteres como máximo. Durante la instalación se crea automáticamente una instancia llamada "DB2".

directorio de instancia

Es el directorio inicial donde reside la instancia.

En las plataformas UNIX, el directorio de instancia es \$HOME/sqllib, donde \$HOME es el directorio inicial del propietario de instancia.

En los sistemas operativos Windows, el directorio de instancia es %DB2PATH%\nombre_instancia. La variable %DB2PATH% determina la ubicación donde se instala DB2. El valor por omisión de la instalación para %DB2PATH% es \Archivos de programa\IBM\SQLLIB, por lo que dependiendo de la unidad donde esté instalado DB2, %DB2PATH% apuntará a *unidad*:\Archivos de programa\IBM\SQLLIB, a menos que se cambie el valor por omisión.

En los servidores Windows, la vía de acceso de la instancia se crea de acuerdo con uno de estos dos elementos:

%DB2PATH%\%DB2INSTANCE%

(por ejemplo, C:\Archivos de programa\IBM\SQLLIB\DB2)

o, si DB2INSTPROF está definido:

%DB2INSTPROF%\%DB2INSTANCE%

(por ejemplo, C:\PROFILES\DB2)

En los servidores Windows, se utiliza la variable de entorno DB2INSTPROF para permitir la ejecución de DB2 en una unidad de red para la que la máquina cliente sólo tiene acceso de lectura. En este caso, DB2 se define de forma que apunte a *unidad*:\Archivos de programa\IBM\SQLLIB, y DB2INSTPROF se define de forma que apunte a una vía de acceso local (por ejemplo, C:\PROFILES) que contendrá toda la información específica de la instancia, tal como los catálogos y las configuraciones, pues DB2 necesita acceso de actualización para estos archivos.

Conceptos relacionados:

- “Multiple instances of the database manager” en el manual *Administration Guide: Implementation*
- “Multiple instances on a UNIX operating system” en el manual *Administration Guide: Implementation*
- “Multiple instances on a Windows operating system” en el manual *Administration Guide: Implementation*

Tareas relacionadas:

- “Actualización del archivo de configuración del gestor de bases de datos” en la página 27
- “Setting the current instance” en el manual *Administration Guide: Implementation*
- “Creating additional instances” en el manual *Administration Guide: Implementation*

Información relacionada:

- “Mandato GET DATABASE MANAGER CONFIGURATION” en el manual *Consulta de mandatos*
- “Mandato UPDATE DATABASE MANAGER CONFIGURATION” en el manual *Consulta de mandatos*

Servidores DB2 soportados

Utilice DB2 AD Client para desarrollar aplicaciones que se ejecutarán en un sistema operativo determinado. Sin embargo, sus aplicaciones pueden acceder a bases de datos remotas situadas en los siguientes servidores de sistema operativo:

- DB2 para AIX
- DB2 para HP-UX
- DB2 para Linux
- DB2 para OS/2
- DB2 para NUMA-Q
- DB2 para Solaris
- DB2 para Windows NT
- DB2 para Windows 2000
- DB2 para Windows XP
- DB2 para Windows Server 2003
- Servidores de aplicaciones compatibles con DRDA (Distributed Relational Database Architecture), tales como:
 - DB2 para z/OS y OS/390
 - DB2 para AS/400 e iSeries
 - DB2 Server para VSE y VM (antes llamado SQL/DS para VM y VSE)
 - Servidores de aplicaciones compatibles con DRDA proporcionados por proveedores de bases de datos distintos de IBM.

Notas:

1. Los servidores HP-UX de 64 bits de DB2 Versión 8 no dan soporte a la ejecución de aplicaciones locales de 64 bits de DB2 Versión 7.
2. DB2 para OS/2 no está disponible para DB2 Versión 8
3. DB2 para NUMA-Q se ejecuta en el sistema operativo PTX y sólo está disponible para DB2 Versión 7

4. Los servidores Windows de 64 bits de DB2 Versión 8 dan soporte a conexiones desde clientes de 32 bits de DB2 Versión 6 y Versión 7 sólo para peticiones de SQL. No se da soporte a las conexiones desde clientes de 64 bits de la Versión 7.

Tareas relacionadas:

- “Configuración del acceso remoto a una base de datos de servidor” en el manual *Suplemento de instalación y configuración*

Información relacionada:

- “Software de desarrollo soportado por DB2” en la página 8

Software de desarrollo soportado por DB2

DB2 Versión 8 da soporte a compiladores, intérpretes y software asociado para los sistemas operativos siguientes:

- AIX
- HP-UX
- Linux
- Solaris Operating Environment
- Windows

DB2 da soporte las versiones de 32 y 64 bits de cada uno de estos sistemas operativos. En la mayoría de estos sistemas operativos existen diferencias respecto a la forma de crear aplicaciones en los entornos de 32 y 64 bits. DB2 no da soporte a la ejecución de aplicaciones o rutinas (procedimientos almacenados y funciones definidas por el usuario) COBOL en entornos de sistema operativo de 64 bits. Para otros idiomas, DB2 da soporte a la ejecución de aplicaciones y rutinas de 32 bits en todos los entornos de sistema operativo de 64 bits soportados excepto Linux IA64 y Linux zSeries.

Con excepción de IBM COBOL en Windows, la información referente a los compiladores para cada uno de estos sistemas operativos da por supuesto que el usuario está utilizando el precompilador DB2 correspondiente al sistema operativo y no el soporte de precompilador que pueda estar incorporado en uno de los compiladores listados. Para IBM COBOL en Windows, DB2 da soporte al precompilador IBM COBOL además del precompilador DB2.

Para obtener la información más reciente sobre compiladores DB2 y actualizaciones del software asociado, consulte el sitio Web sobre desarrollo de aplicaciones DB2, situado en:

<http://www.ibm.com/software/data/db2/udb/ad>

Tenga en cuenta lo siguiente respecto al soporte de software:

- **Fortran y REXX.** DB2 no actualiza el nivel de soporte para Fortran y REXX más allá del nivel de soporte proporcionado para esos lenguajes en DB2 Universal Database versión 5.2.
- **Perl.** En el momento de imprimir el presente manual, el Release 0.76 del controlador de DB2 UDB (DBD::DB2) correspondiente a Perl Database Interface (Perl DBI), Versión 0.93 o posterior, está disponible para AIX, HP-UX, Linux, Solaris y Windows. La versión más reciente del controlador se puede bajar de:

<http://www.ibm.com/software/data/db2/perl>

- **PHP.** PHP se puede utilizar como método para acceder a DB2 desde aplicaciones basadas en la Web o desde la línea de mandatos, y está disponible para AIX, HP-UX, Linux, Solaris y Windows. En el momento de imprimir esta publicación, la última versión es PHP 4.3.4. Puede bajar la versión más reciente de PHP desde:

<http://www.php.net>

Información relacionada:

- “Software de desarrollo soportado por AIX” en la página 9
- “Software de desarrollo soportado por HP-UX” en la página 11
- “Software de desarrollo soportado por Linux” en la página 13
- “Software de desarrollo soportado por Solaris” en la página 18
- “Software de desarrollo soportado por Windows” en la página 20

Software de desarrollo soportado por AIX

DB2 para AIX da soporte a los sistemas operativos siguientes:

AIX Versión 4.3.3 (sólo de 32 bits)

con Nivel de mantenimiento 11

Más el PTF de ejecución en C++ de marzo de 2003 (consulte el apartado sobre C++ siguiente para ver el enlace de descarga) y:

Para sistemas de archivos JFS:

APAR IY49385

Para Java:

- OpenGL.OpenGL_X.rte.base
- OpenGL.OpenGL_X.rte.soft
- X11.adt.lib

AIX Versión 5.1.0 (32 bits y 64 bits)

con Nivel de mantenimiento 5

Más el PTF de ejecución en C++ de marzo de 2003 (consulte el apartado sobre C++ siguiente para ver el enlace de descarga) y:

Para sistemas de archivos JFS:

APAR IY48735

Para sistemas de archivos JFS2:

APAR IY49254

Para Java:

Paquete de mantenimiento recomendado AIX 5100-04 y APAR IY46667

Para ejecutar más de 1000 db2agents:

APAR IY49220, y especifique "vmtune -T 0" delante de db2start o en el arranque de AIX

AIX Versión 5.2.0 (32 bits y 64 bits)

con Nivel de mantenimiento 2, y:

Para el volumen montado de CIO (Concurrent I/O) y DIO (Direct I/O):

Los APAR IY49129 e IY49346

Para sistemas de archivos JFS:

APAR IY48339

| **Para sistemas de archivos JFS2:**

| APAR IY49304

| **Para Java:**

| Paquete de mantenimiento recomendado AIX 5200-01 y APAR
| IY46668

| **Para ejecutar más de 1000 db2agents y utilizar el kernel AIX de 32 bits:**

| APAR IY49885, y especifique "vmo -o pta_balance_threshold=0"
| delante de db2start o en el arranque de AIX

Nota: Puede consultar su sistema para ver si hay una determinada APAR con el mandato siguiente:

```
instfix -v -i  
-k <APAR>
```

Por ejemplo: instfix -v -i -k IY31254

DB2 para AIX da soporte a los lenguajes de programación y compiladores siguientes:

C IBM C para AIX versión 5.0.2.3

| IBM C para AIX versión 6.0

C++ IBM VisualAge C++ versión 5.0.2.3 con el PTF de ejecución en C++ de marzo de 2003:

| <http://www-1.ibm.com/support/docview.wss?rs=0&q=x1C.rte&uid=swg24004427>

| IBM VisualAge C++ versión 6.0 con el PTF de ejecución en C++ de marzo de 2003:

| <http://www-1.ibm.com/support/docview.wss?rs=0&q=x1C.rte&uid=swg24004427>

COBOL

IBM COBOL Set para AIX Versión 1.1

| Micro Focus COBOL Server Express versión 2.2 con Service Pack 1

| **Nota:** El soporte de COBOL sólo es para 32 bits.

Fortran

IBM XL Fortran para AIX, Versiones 4.1 (32 bits solamente) y 5.1.0 (32 y 64 bits)

| **Java** Java Developer Kit versión 1.3.1 y Java Runtime Environment versión 1.3.1 de IBM para AIX de IBM

| IBM Developer Kit para AIX, Java Technology Edition, versión 1.4.1 release de servicio 1 (sólo para AIX 5.1 y 5.2)

| **Nota:** DB2 instalará la última versión soportada del kit de desarrollo, en caso de que aún no esté instalado, a menos que la instalación de DB2 sea una actualización de una instalación previa de DB2 versión 8. Si se está actualizando una instalación previa de DB2 versión 8, el kit de desarrollo soportado se debe instalar forma manual desde el CD-ROM.

| **Perl** Perl 5.004_04 o posterior, DBI 0.93 o posterior

| **PHP** PHP 4.3.4 o posteriores

| **REXX** IBM AIX REXX/6000 AISPO, Número de Producto: 5764-057

IBM Object REXX para AIX Versión 1.1
REXXSAA 4.00

Nota: El soporte de REXX sólo es para 32 bits.

Para conocer la información más reciente sobre el soporte de software de DB2 para AIX, visite el sitio Web de desarrollo de aplicaciones de DB2:

<http://www.ibm.com/software/data/db2/udb/ad>

Información relacionada:

- “Software de desarrollo soportado por DB2” en la página 8
- “Opciones de compilación y enlace para aplicaciones C de AIX” en la página 185
- “Opciones de compilación y enlace para rutinas C de AIX” en la página 186
- “Opciones de compilación y enlace para aplicaciones C++ de AIX” en la página 189
- “Opciones de compilación y enlace para rutinas C++ de AIX” en la página 191
- “Opciones de compilación y enlace para aplicaciones IBM COBOL de AIX” en la página 201
- “Opciones de compilación y enlace para rutinas IBM COBOL de AIX” en la página 204
- “Opciones de compilación y enlace para aplicaciones Micro Focus COBOL de AIX” en la página 207
- “Opciones de compilación y enlace para rutinas Micro Focus COBOL de AIX” en la página 208
- “Requisitos de instalación para servidores DB2 (AIX)” en el manual *Guía rápida de iniciación para servidores DB2*

Software de desarrollo soportado por HP-UX

DB2 para HP-UX da soporte a los sistemas operativos siguientes:

HP-UX en PA-RISC

HP-UX versión 11i(11.11) para PA-RISC 2 con:

Paquetes de junio de 2003 GOLDBASE11i y GOLDAPPS11i, y parches PHSS_26560, PHKL_28489, PHCO_27434 y PHCO_29960

HP-UX en IA64

HP-UX versión 11i Versión 2 (B.11.23) para Intel Itanium con el parche PHKL_30065

Nota: No se soportan los intentos de ejecutar rutinas o aplicaciones DB2 para HP-UX en IA64 que se hayan creado para PA-RISC bajo el conversor de binarios aries.

DB2 para HP-UX en PA-RISC da soporte a los compiladores y lenguajes de programación siguientes:

C Compilador C de HP, Versión B.11.11.02

C++ HP aC++ versión A.03.52

COBOL

Micro Focus COBOL Server Express versión 2.2 con Service Pack 1

Nota: El soporte de COBOL sólo es para 32 bits.

Fortran

HP-UX f90 B.11.01.06

Java HP-UX de 32 bits: Software Developer's Kit y Runtime Environment 1.3.1 y 1.4.2.01 para HP-UX 11.0 y 11i PA-RISC de Hewlett-Packard.

HP-UX de 64 bits: Software Developer's Kit y Runtime Environment 1.4.2.01 para HP-UX 11.0 y 11i PA-RISC de Hewlett-Packard.

Notas:

1. Sólo se da soporte a rutinas Java de 32 bits para Software Developer's Kit.
2. DB2 instalará la última versión soportada del kit de desarrollo, en caso de que aún no esté instalado, a menos que la instalación de DB2 sea una actualización de una instalación previa de DB2 versión 8. Si se está actualizando una instalación previa de DB2 versión 8, el kit de desarrollo se debe instalar forma manual desde el CD-ROM.

Perl Perl 5.004_04 o posterior, DBI 0.93 o posterior

PHP PHP 4.3.4 o posteriores

DB2 para HP-UX en IA64 da soporte a los compiladores y lenguajes de programación siguientes:

C Compilador C de HP versión A.05.52

C++ HP (aCC) aC++ versión A.05.52

Nota: DB2 para HP-UX en IA64 no da soporte a ninguna aplicación C++ ni biblioteca C++ de otras compañías que se haya creado con la opción desaprobada `-AP` o tenga dependencia de `libstd` (en lugar de `libstd_v2`).

Fortran

HP-UX F90 B.11.23

Notas:

1. Para generar código de 64 bits es necesaria la opción de compilación `+DD64`.
2. Para generar código de 32 bits es necesaria la opción de compilación `+DD32`.

Java Software Developer's Kit y Runtime Environment 1.4.2.01 o posteriores para HP-UX 11 PA-RISC y sistemas basados en Itanium.

Nota: DB2 instalará la última versión soportada del kit de desarrollo, en caso de que aún no esté instalado, a menos que la instalación de DB2 sea una actualización de una instalación previa de DB2 versión 8. Si se está actualizando una instalación previa de DB2 versión 8, el kit de desarrollo se debe instalar forma manual desde el CD-ROM en `/opt/java1.4`.

Perl Perl 5.004_04 o posterior, DBI 0.93 o posterior

PHP PHP 4.3.4 o posteriores

Para conocer la información más reciente sobre el soporte de software de DB2 para HP-UX, visite el sitio Web de desarrollo de aplicaciones de DB2:

<http://www.ibm.com/software/data/db2/udb/ad>

Información relacionada:

- “Software de desarrollo soportado por DB2” en la página 8
- “Opciones de compilación y enlace para aplicaciones C de HP-UX” en la página 212
- “Opciones de compilación y enlace para rutinas C de HP-UX” en la página 215
- “Opciones de compilación y enlace para aplicaciones C++ de HP-UX” en la página 218
- “Opciones de compilación y enlace para rutinas C++ de HP-UX” en la página 220
- “Opciones de compilación y enlace para aplicaciones Micro Focus COBOL de HP-UX” en la página 224
- “Opciones de compilación y enlace para rutinas Micro Focus COBOL de HP-UX” en la página 225
- “Requisitos de instalación para servidores DB2 (HP-UX)” en el manual *Guía rápida de iniciación para servidores DB2*

Software de desarrollo soportado por Linux

DB2 para Linux da soporte a las arquitecturas de sistema operativo siguientes:

- **Linux en Intel x86 (32 bits)**
- **Linux en s/390 y zSeries**
- **Linux en AMD64**
- **Linux en PowerPC**
- **Linux en IA64**

Para obtener información sobre las distribuciones soportadas, así como sobre los niveles de kernel y bibliotecas correspondientes a cada una de estas arquitecturas, visite:

<http://www.ibm.com/db2/linux/validate>

Las tablas siguientes describen el soporte de arquitectura de Linux de DB2 en el momento de impresión de esta publicación. Se le insta firmemente a comprobar en el sitio Web de validación (anterior) si existen actualizaciones para este soporte:

Tabla 1. Linux en Intel x86 (32 bits)

Distribuciones	Kernel	Biblioteca	Comentarios
Conectiva Linux Enterprise Edition (CLEE)	2.4.19	glibc 2.2.5	Suministrada por United Linux 1.0
LINX Rocky Secure Server 2.1	2.4.21	glibc 2.2.5	
Red Flag Advanced Server 4.0	2.4.21-as.2	glibc 2.2.93-5	
Red Flag Function Server 4.0	2.4.20-8smp	glibc 2.2.93-5	
Red Hat Enterprise Linux 2.1 AS/ES/WS	2.4.9-e16	glibc 2.2.4	
Red Hat Enterprise Linux (RHEL) 3 AS/ES/WS	2.4.21-7.EL	glibc-2.3.2-95.3	

Tabla 1. Linux en Intel x86 (32 bits) (continuación)

Distribuciones	Kernel	Biblioteca	Comentarios
SCO Linux 4.0	2.4.19	glibc 2.2.5	Suministrada por United Linux 1.0
SuSE Pro 8.0	2.4.18	glibc 2.2.5	
SuSE Pro 8.1	2.4.19	glibc 2.2.5	
SuSE Linux Enterprise Server (SLES) 7	2.4.7	glibc 2.2.2	
SuSE Linux Enterprise Server (SLES) 8	2.4.19	glibc 2.2.5	Validada al nivel del Paquete de servicio 2 de SuSE
Turbolinux 7 Server	2.4.9	glibc 2.2.4	
Turbolinux 8 Server	2.4.18-5	glibc 2.2.5	
Turbolinux Enterprise Server 8	2.4.19	glibc 2.2.5	
United Linux 1.0	2.4.19	glibc 2.2.5	

Tabla 2. Distribuciones no empresariales de Linux en Intel x86 (32 bits) (que el proveedor ya no soporta)

Distribuciones	Kernel	Biblioteca	Comentarios
Red Hat 7.2	2.4.9-34	glibc 2.2.4	
Red Hat 7.3	2.4.18	glibc 2.2.5	
Red Hat 8.0	2.4.18-14	glibc 2.2.93-5	
SuSE 7.3	2.4.10	glibc 2.2.4	

Tabla 3. Linux en s/390 y zSeries (versión de kernel de 31 bits soportada en s/390; 64 bits en zSeries)

Distribuciones	Kernel	Biblioteca	Comentarios
Red Hat 7.2	2.4.9-38	glibc 2.2.4	
SuSE Linux Enterprise Server (SLES) 7	2.4.7-58	glibc 2.2.4	compat.rpm contains libstdc++ 6.1. Utilice JDK 1.3.1 SR 1 para Java
SuSE Linux Enterprise Server (SLES) 8	2.4.19	glibc 2.2.5	Suministrada por United Linux 1.0
Turbo Linux Enterprise Server (TLES) 8	2.4.19	glibc 2.2.5	Suministrada por United Linux 1.0
United Linux 1.0	2.4.19	glibc 2.2.5	

Tabla 4. Linux en AMD64

Distribuciones	Kernel	Biblioteca	Comentarios
Red Hat Enterprise Linux (RHEL) 3 AS/ES/WS	2.4.21-7.EL	glibc-2.3.2-95.3	
SuSE Linux Enterprise Server (SLES) 8.0	2.4.19-SMP	glibc 2.2.5-16	

Tabla 5. Linux en PowerPC (iSeries y pSeries)

Distribuciones	Kernel	Biblioteca	Comentarios
Red Hat Enterprise Linux (RHEL) 3 AS	2.4.21-7.EL	glibc-2.3.2-95.3	
SuSE Enterprise Server (SLES) 8	2.4.19-16	glibc 2.2.5	Suministrada por United Linux 1.0
Turbolinux Enterprise Server 8	2.4.19-16	glibc 2.2.5	Suministrada por United Linux 1.0
United Linux 1.0	2.4.19	glibc 2.2.5	

Tabla 6. Linux en IA64

Distribuciones	Kernel	Biblioteca	Comentarios
Red Hat Enterprise Linux 2.1 AS/ES/WS	2.4.18-e.12smp	glibc	
Red Hat Enterprise Linux (RHEL) 3 AS/ES/WS	2.4.21-7.EL	glibc-2.3.2-95.3	
SuSE Linux Enterprise Server (SLES) 8	2.4.19-SMP	glibc 2.2.5	Suministrada por United Linux 1.0
United Linux 1.0	2.4.19	glibc 2.2.5	

DB2 para Linux para Intel x86 da soporte a los compiladores y lenguajes de programación siguientes:

C gcc de GNU/Linux versiones 2.95.3 y 2.96

C++ g++ de GNU/Linux versiones 2.95.3 y 2.96

COBOL

Micro Focus COBOL Server Express versión 2.2 con Service Pack 1

Java IBM Developer Kit y Runtime Environment para Linux, Java 2 Technology Edition, versiones 1.3.1 y 1.4.1 Release de servicio 1, versión de 32 bits

Nota: DB2 instalará la última versión soportada del kit de desarrollo, en caso de que aún no esté instalado, a menos que la instalación de DB2 sea una actualización de una instalación previa de DB2 versión 8. Si se está actualizando una instalación previa de DB2 versión 8, el kit de desarrollo soportado se debe instalar forma manual desde el CD-ROM.

Perl Perl 5.004_04 o posterior, DBI 0.93 o posterior

PHP PHP 4.3.4 o posteriores

REXX Object REXX Interpreter para Linux versión 2.1

Una instancia de 32 bits en **DB2 para Linux en s/390** o **DB2 para Linux en zSeries** da soporte a los compiladores y lenguajes de programación siguientes:

C gcc de GNU/Linux, versión 2.95.3

C++ g++ de GNU/Linux, versión 2.95.3

COBOL

Micro Focus COBOL Server Express versión 2.2 con Service Pack 1

Java IBM zSeries Developer Kit para Linux, Java 2 Technology Edition (a nivel del SDK 1.3.1 de Sun y Release de servicio 1)

Nota: DB2 instalará la última versión soportada del kit de desarrollo, en caso de que aún no esté instalado, a menos que la instalación de DB2 sea una actualización de una instalación previa de DB2 versión 8. Si se está actualizando una instalación previa de DB2 versión 8, el kit de desarrollo soportado se debe instalar forma manual desde el CD-ROM.

Perl Perl 5.004_04 o posterior, DBI 0.93 o posterior

PHP PHP 4.3.4 o posteriores

REXX Object REXX 2.2.0 para Linux/390

Una instancia de 64 bits en **DB2 para Linux en zSeries** da soporte a los compiladores y lenguajes de programación siguientes:

C GNU/Linux gcc versión 3.2

C++ GNU/Linux g++ versión 3.2

Java IBM zSeries Developer Kit para Linux, Java 2 Technology Edition (a nivel del SDK 1.4.1 de Sun y Release de servicio 1)

Nota: DB2 instalará la última versión soportada del kit de desarrollo, en caso de que aún no esté instalado, a menos que la instalación de DB2 sea una actualización de una instalación previa de DB2 versión 8. Si se está actualizando una instalación previa de DB2 versión 8, el kit de desarrollo soportado se debe instalar forma manual desde el CD-ROM.

Perl Perl 5.8

Una instancia de 32 bits en **DB2 para Linux en AMD64** da soporte a los compiladores y lenguajes de programación siguientes:

C gcc de GNU/Linux versiones 3.2 y 3.3

Nota: Debe utilizarse la opción "-m32" del compilador para generar rutinas o aplicaciones de 32 bits (procedimientos almacenados y funciones definidas por el usuario).

C++ g++ de GNU/Linux versiones 3.2 y 3.3

Notas:

1. Estas versiones del compilador g++ de GNU/Linux no aceptan parámetros enteros para algunas funciones fstream. Consulte la documentación del compilador para obtener más información.
2. Debe utilizarse la opción "-m32" del compilador para generar rutinas o aplicaciones de 32 bits (procedimientos almacenados y funciones definidas por el usuario).

Java IBM Developer Kit y Runtime Environment para Linux x86, Java 2 Technology Edition, versión 1.3.1 y Release de servicio 4, versión de 32 bits, y versión 1.1 Release de servicio 4, versión de 23 bits.

Notas:

1. DB2 instalará la última versión soportada del kit de desarrollo, en caso de que aún no esté instalado, a menos que la instalación de DB2 sea

una actualización de una instalación previa de DB2 versión 8. Si se está actualizando una instalación previa de DB2 versión 8, el kit de desarrollo soportado se debe instalar forma manual desde el CD-ROM.

2. Si ha instalado el Release de servicio de Developer Kit 1.3.1 que SuSE SLES 8 proporciona, debe desinstalarlo antes de instalar DB2; en caso contrario, DB2 no podrá instalar el Developer Kit recomendado. Si DB2 se ha instalado mientras el Developer Kit que SuSE SLES 8 proporciona no estaba desinstalado, consulte las instrucciones del archivo README para actualizar el Developer Kit de forma manual.

Perl Perl 5.8

PHP PHP 4.3.4 o posteriores

Una instancia de 64 bits en **DB2 para Linux en AMD64** da soporte a los compiladores y lenguajes de programación siguientes:

C gcc de GNU/Linux versiones 3.2 y 3.3

C++ g++ de GNU/Linux versiones 3.2 y 3.3

Nota: Estas versiones del compilador g++ de GNU/Linux no aceptan parámetros enteros para algunas funciones fstream. Consulte la documentación del compilador para obtener más información.

Java DB2 no da soporte actualmente a ningún Java Developer Kit para Linux sobre AMD64 de 64 bits.

Perl Perl 5.8

PHP PHP 4.3.4 o posteriores

DB2 para Linux en PowerPC da soporte a los compiladores y lenguajes de programación siguientes:

C GNU/Linux gcc versión 3.2

C++ GNU/Linux g++ versión 3.2

Nota: Esta versión del compilador GNU/Linux g++ no acepta parámetros enteros para algunas funciones fstream. Consulte la documentación del compilador para obtener más información.

Java Para una instancia de 32 bits: IBM Developer Kit y Runtime Environment para Linux Java 2 Technology Edition, versiones 1.3.1 y 1.4.1 Release de servicio 1, versión de 32 bits para PowerPC.

Para una instancia de 64 bits: IBM Developer Kit y Runtime Environment para Linux Java 2 Technology Edition, versión 1.4.1 Release de servicio 1, versión de 64 bits para PowerPC.

Nota: DB2 instalará la última versión soportada del kit de desarrollo, en caso de que aún no esté instalado, a menos que la instalación de DB2 sea una actualización de una instalación previa de DB2 versión 8. Si se está actualizando una instalación previa de DB2 versión 8, el kit de desarrollo soportado se debe instalar forma manual desde el CD-ROM.

Perl Perl 5.8

PHP PHP 4.3.4 o posteriores

DB2 para Linux en IA64 da soporte a los compiladores y lenguajes de programación siguientes:

C GNU/Linux gcc versión 3.2

C++ GNU/Linux g++ versión 3.2

Nota: Esta versión del compilador GNU/Linux g++ no acepta parámetros enteros para algunas funciones `fstream`. Consulte la documentación del compilador para obtener más información.

Java IBM Developer Kit y Runtime Environment para Linux, Java 2 Technology Edition, versión 1.3.1, versión de 64 bits. Para utilizar este Developer Kit, también debe tener instalado gcc 3.2 y las bibliotecas de ejecución de gcc3 libstdc++.

Nota: DB2 instalará el kit de desarrollo, en caso de que aún no esté instalado, a menos que la instalación de DB2 sea una actualización de una instalación previa de DB2 versión 8. Si se está actualizando una instalación previa de DB2 versión 8, el kit de desarrollo se debe instalar forma manual desde el CD-ROM.

Perl Perl 5.8

Nota: Linux IA64 no da soporte a la ejecución de aplicaciones o rutinas (procedimientos almacenados y funciones definidas por el usuario) de DB2 de 32 bits.

Para conocer la información más reciente sobre el soporte de software de DB2 para Linux, visite el sitio Web de desarrollo de aplicaciones de DB2:

<http://www.ibm.com/software/data/db2/udb/ad>

Información relacionada:

- “Software de desarrollo soportado por DB2” en la página 8
- “Opciones de compilación y enlace para aplicaciones C de Linux” en la página 228
- “Opciones de compilación y enlace para rutinas C de Linux” en la página 230
- “Opciones de compilación y enlace para aplicaciones C++ de Linux” en la página 233
- “Opciones de compilación y enlace para rutinas C++ de Linux” en la página 235
- “Requisitos de instalación para DB2 Personal Edition (Linux)” en el manual *Guía rápida de iniciación para DB2 Personal Edition*
- “Requisitos de instalación para servidores DB2 (Linux)” en el manual *Guía rápida de iniciación para servidores DB2*
- “Opciones de compilación y enlace para aplicaciones Micro Focus COBOL de Linux” en la página 239
- “Opciones de compilación y enlace para rutinas Micro Focus COBOL de Linux” en la página 240

Software de desarrollo soportado por Solaris

DB2 para Solaris da soporte al sistema operativo siguiente:

Solaris

DB2 Workgroup Server Edition se puede ejecutar en las siguientes versiones del entorno operativo Solaris:

Solaris 7 (32 bits) con "Parches recomendados y de seguridad" + 107226-17 + 107153-01 + 106327-10

Solaris 8 (32 bits) con "Parches recomendados y de seguridad" + 108921-12 + 108940-24 + 108434-03 + 108528-12

Solaris 9 (32 bits)

DB2 Enterprise Server Edition se puede ejecutar en las siguientes versiones del entorno operativo Solaris:

Solaris 7 (32 bits) con "Parches recomendados y de seguridad" + 107226-17 + 107153-01 + 106327-10

Solaris 7 (64 bits) con "Parches recomendados y de seguridad" + 107226-17 + 107153-01 + 106300-11 + 106327-10

Solaris 8 (32 bits) con "Parches recomendados y de seguridad" + 108921-12 + 108940-24 + 108434-03 + 108528-12

Solaris 8 (64 bits) con "Parches recomendados y de seguridad" + 108921-12 + 108940-24 + 108435-03 + 108434-03 + 108528-12

Solaris 9 (32 bits)

Solaris 9 (64 bits)

Los "Parches recomendados y de seguridad" pueden conseguirse en la siguiente dirección:

<http://sunsolve.sun.com>

En el sitio Web de SunSolve Online, pulse elemento de menú "Patches" del panel izquierdo y seleccione "Recommended & Security Patches" en la sección "Downloads".

También se necesitan los clústeres de parches de J2SE Solaris. Se pueden conseguir en el sitio Web <http://sunsolve.sun.com>. En el sitio Web de SunSolve Online, pulse el elemento de menú "Patches" del panel izquierdo y seleccione "Recommended Patch Clusters" en la sección "Browse & Download Patches".

Para instalar DB2 en Solaris se necesita SUNWlibC.

Para instalar DB2 en sistemas PRIMEPOWER de Fujitsu de 64 bits, también se necesita lo siguiente:

- Parche de actualización del kernel de Solaris 8 108528-16 o posterior para obtener el arreglo para el parche 912040-01
- Parche de actualización del kernel de Solaris 9 112233-01 o posterior para obtener el arreglo para el parche 912041-01

Los parches de Fujitsu PRIMEPOWER para Solaris pueden bajarse de FTSI en el enlace siguiente:

<https://download.ftsi.fujitsu.com/>

DB2 para Solaris da soporte a los lenguajes de programación y compiladores siguientes:

C Forte C Versiones 5.0, 6, 6.1 y 6.2

| Sun ONE Studio 7 y 8, Compiler Collection
| **C++** Forte C++ Versiones 5.0, 6, 6.1 y 6.2
| Sun ONE Studio 7 y 8, Compiler Collection

COBOL

| Micro Focus COBOL Server Express versión 2.2 con Service Pack 1

| **Nota:** El soporte de COBOL sólo es para 32 bits.

Fortran

| SPARCompiler Fortran Versiones 4.2 y 5.0

| **Java** Solaris de 32 bits: Java Development Kit (JDK) versiones 1.3.1 y 1.4.2 para
| Solaris de Sun Microsystems

| Solaris de 64 bits: Java Development Kit (JDK) versión 1.4.2 para Solaris de
| Sun Microsystems

| **Nota:** DB2 instalará la última versión soportada de JDK, en caso de que
| aún no esté instalado, a menos que la instalación de DB2 sea una
| actualización de una instalación previa de DB2 versión 8. Si se está
| actualizando una instalación previa de DB2 versión 8, se debe
| instalar JDK de forma manual desde el CD-ROM.

| **Perl** Perl 5.004_04 o posterior, DBI 0.93 o posterior

| **PHP** PHP 4.3.4 o posterior

Para conocer la información más reciente sobre el soporte de software de DB2 para Solaris, visite el sitio Web de desarrollo de aplicaciones de DB2:

<http://www.ibm.com/software/data/db2/udb/ad>

Información relacionada:

- “Software de desarrollo soportado por DB2” en la página 8
- “Opciones de compilación y enlace para aplicaciones C de Solaris” en la página 244
- “Opciones de compilación y enlace para rutinas C de Solaris” en la página 246
- “Opciones de compilación y enlace para aplicaciones C++ de Solaris” en la página 249
- “Opciones de compilación y enlace para rutinas C++ de Solaris” en la página 251
- “Opciones de compilación y enlace para aplicaciones Micro Focus COBOL de Solaris” en la página 254
- “Opciones de compilación y enlace para rutinas Micro Focus COBOL de Solaris” en la página 256
- “Requisitos de instalación para servidores DB2 (Entorno operativo Solaris)” en el manual *Guía rápida de iniciación para servidores DB2*

Software de desarrollo soportado por Windows

DB2 para sistemas operativos Windows de 32 bits da soporte a lo siguiente:

Microsoft Windows XP

Microsoft Windows Server 2003

Microsoft Windows 2000

Para Windows Terminal Server se necesita el Service Pack 2.

Microsoft Windows NT

Versión 4.0 con Service Pack 6a o posterior.

Microsoft Windows ME

Microsoft Windows 98

DB2 para sistemas operativos Windows de 32 bits da soporte a los lenguajes de programación siguientes:

Basic Microsoft Visual Basic 6.0 Professional Edition

Microsoft Visual Basic .NET 7.0 y 7.1 para Microsoft .NET Framework versiones 1.0 y 1.1 respectivamente

Nota: .NET Framework debe estar instalado antes de utilizar el programa de instalación de DB2 para instalar .NET Data Provider de DB2.

C# Compilador Microsoft Visual C# .NET versiones 7.0 y 7.1 para Microsoft .NET Framework versiones 1.0 y 1.1 respectivamente

Nota: .NET Framework debe estar instalado antes de utilizar el programa de instalación de DB2 para instalar .NET Data Provider de DB2.

C/C++ Microsoft Visual C++ versión 6.0

Microsoft Visual C++ .NET 2002 y 2003

Compilador Intel C++ para aplicaciones de 32 bits, versión 6 o posterior

COBOL

Micro Focus COBOL Versión 4.0.20

Micro Focus COBOL Net Express, Versión 3.1.0

IBM VisualAge COBOL versión 3.0.4 o posterior

Java IBM Developer Kit y Runtime Environment para Windows, Java 2 Technology Edition, versiones 1.3.1 y 1.4.1 con release de servicio 1, versión de 32 bits

Nota: DB2 instalará la última versión soportada del Developer Kit, en caso de que aún no esté instalado.

Java Development Kit (JDK) 1.3.1 para Win32 de Sun Microsystems

Perl Perl 5.004_04, DBI 0.93

PHP PHP 4.3.4 o posterior

REXX IBM Object REXX para Windows NT/95, Versión 1.1

Para conocer información sobre la obtención de IBM Object REXX para Windows, visite este sitio Web:

<http://www.ibm.com/software/ad/obj-rexx/>

Microsoft Windows Scripting Host

Versión 5.1

Nota: Para Windows 2000 y Windows XP: para ejecutar aplicaciones de DB2 que tengan objetos COM+ que utilicen ODBC o aplicaciones que usen OLE DB Provider para ODBC con la agrupación de recursos OLE DB inhabilitada,

debe utilizar Windows 2000 Service Pack 3 o Windows XP Service Pack 1. Si no está seguro de que su entorno de aplicación esté calificado, le recomendamos que instale el nivel de servicio de Windows adecuado. Para obtener más información, consulte el siguiente artículo de Microsoft Knowledge Base:

<http://support.microsoft.com/default.aspx?scid=KB;EN-US;306414>

Estos Service Pack no son necesarios para el servidor DB2 propiamente dicho o para las aplicaciones suministradas como parte de los productos de DB2.

DB2 para sistemas operativos Windows de 64 bits da soporte a lo siguiente:

Microsoft Windows XP, Edición de 64 bits

Microsoft Windows Server 2003

DB2 para sistemas operativos Windows de 64 bits da soporte a los lenguajes de programación siguientes:

C/C++ Compilador Intel C++ para Itanium versión 7.1 con Microsoft Software Developer's Kit 3790 o posterior.

Compilador C/C++ de Microsoft para la arquitectura Intel Itanium (disponible con Microsoft Software Developer's Kit 3790)

Java IBM Developer Kit y Runtime Environment para Windows, Java 2 Technology Edition, versión 1.4.1 con release de servicio 1, versión de 64 bits

Nota: DB2 instalará el Developer Kit, en caso de que aún no esté instalado.

Perl Perl 5.004_04, DBI 0.93

PHP PHP 4.3.4 o posterior

Microsoft Windows Scripting Host

Versión 5.1

Nota: Windows Server 2003 da soporte a lo siguiente:

- Windows Server 2003, Standard Edition
- Windows Server 2003, Enterprise Edition
- Windows Server 2003, Datacenter Edition

Para conocer la información más reciente sobre el soporte de software de DB2 para Windows, visite el sitio Web de desarrollo de aplicaciones de DB2:

<http://www.ibm.com/software/data/db2/udb/ad>

Información relacionada:

- "Software de desarrollo soportado por DB2" en la página 8
- "Requisitos de instalación para servidores DB2 (Windows)" en el manual *Guía rápida de iniciación para servidores DB2*
- "Opciones de compilación y enlace para aplicaciones C/C++ de Windows" en la página 286
- "Opciones de compilación y enlace para rutinas C/C++ de Windows" en la página 291
- "Opciones de compilación y enlace para aplicaciones IBM COBOL de Windows" en la página 299

- “Opciones de compilación y enlace para rutinas IBM COBOL de Windows” en la página 302
- “Opciones de compilación y enlace para aplicaciones Micro Focus COBOL de Windows” en la página 306
- “Opciones de compilación y enlace para rutinas Micro Focus COBOL de Windows” en la página 309
- “Requisitos de instalación para DB2 Personal Edition (Windows)” en el manual *Guía rápida de iniciación para DB2 Personal Edition*
- “Opciones de compilación y enlace para aplicaciones Visual Basic .NET” en la página 274
- “Opciones de compilación y enlace para aplicaciones C# .NET” en la página 270
- “Opciones de compilación y enlace para rutinas de CLR .NET” en la página 279

Capítulo 2. Configuración

Información general de configuración.	25	Configuración del entorno Java de Windows	45
Configuración del entorno de desarrollo de aplicaciones UNIX	25	Base de datos de ejemplo.	48
Reconstrucción de bibliotecas compartidas para rutinas de DB2	27	Configuración de la base de datos de ejemplo	48
Actualización del archivo de configuración del gestor de bases de datos	27	Creación de la base de datos de ejemplo.	48
Configuración del entorno Java.	28	Creación de la base de datos de ejemplo en servidores de sistema principal o AS/400 e iSeries	50
Configuración de las funciones de DB2		Catalogación de la base de datos de ejemplo	50
WebSphere MQ	30	Vinculación de los programas de utilidad de la base de datos de ejemplo.	51
UNIX	33	Migración de aplicaciones	53
Configuración del entorno de desarrollo de aplicaciones UNIX	34	Migración de aplicaciones a DB2 versión 8	53
Valores de variables de entorno de UNIX	34	Migración de aplicaciones, rutinas y applets Java	54
Configuración del entorno Java de UNIX	35	Migración de aplicaciones desde entornos de 32 bits a entornos de 64 bits	56
Configuración del entorno Java de AIX	37	Cómo asegurarse de que las aplicaciones se puedan transferir	58
Configuración del entorno Java de HP-UX	38	Ejecución de aplicaciones en dos versiones de DB2	59
Configuración del entorno Java de Linux	40	Dónde ir a continuación	63
Configuración del entorno Java de Solaris	41		
Windows	41		
Configuración del entorno de desarrollo de aplicaciones Windows	42		

Información general de configuración

Para ver la información de configuración de la CLI de DB2, vea el manual *CLI Guide and Reference*.

Configuración del entorno de desarrollo de aplicaciones UNIX

Para crear y ejecutar aplicaciones de DB2, debe utilizar un compilador o intérprete para uno de los lenguajes de programación soportados en su sistema operativo, a menos que utilice el script del procesador de línea de mandatos (CLP) o procedimientos de SQL (véase más adelante). Debe configurar el entorno DB2 para el desarrollo de aplicaciones. Debe seguir determinados procedimientos para migrar aplicaciones de DB2 desde una versión anterior de DB2. Asimismo, puede crear la base de datos de ejemplo de DB2 para utilizarla con fines de prueba.

Prerrequisitos:

Compruebe que el entorno DB2 esté configurado correctamente para el compilador o intérprete de DB2 que piensa utilizar; para ello cree primero una aplicación no perteneciente a DB2. Si encuentra algún problema, consulte la documentación que se proporciona con el compilador o intérprete.

Instale el cliente de Desarrollo de Aplicaciones en la estación de trabajo cliente o servidor que está utilizando. Si está desarrollando aplicaciones desde un cliente remoto, compruebe que la máquina cliente puede acceder a la máquina donde reside el servidor de bases de datos de DB2. Compruebe también que el cliente puede conectar satisfactoriamente con la base de datos. Puede utilizar el procesador de línea de mandatos (CLP) o el asistente de configuración del cliente (CCA) para probar la conectividad.

Procedimiento:

Para configurar el entorno de desarrollo de aplicaciones:

1. A menos que se puedan aceptar los valores por omisión, siga las instrucciones indicadas en el apartado “Actualización del archivo de configuración del gestor de bases de datos” en la página 27
2. Si piensa programar con las funciones de la interfaz CLI de DB2, Java o procedimientos de SQL, debe configurar el entorno. Antes de hacer cambios específicos de la plataforma, siga las instrucciones de estas secciones:
 - Configuración del entorno CLI
 - “Configuración del entorno Java” en la página 28
 - “Configuración de las funciones de DB2 WebSphere MQ” en la página 30
3. Configure el entorno del sistema operativo siguiendo las instrucciones de estas secciones:
 - “Configuración del entorno de desarrollo de aplicaciones UNIX” en la página 34
 - “Configuración del entorno de desarrollo de aplicaciones Windows” en la página 42
4. Opcional: “Configuración de la base de datos de ejemplo” en la página 48

Procedimientos de SQL

A partir de DB2 versión 8.2, la creación de procedimientos de SQL no requiere un compilador C o C++ en el servidor, por lo que no se precisa ninguna configuración de compilador C o C++. Cuando se crea un procedimiento de SQL, las sentencias procesales del mismo se convierten en una representación nativa que se almacena en los catálogos de la base de datos, tal como se hace con otras sentencias de SQL. Cuando se llama a un procedimiento de SQL, se carga esta representación desde los catálogos y el motor de DB2 la ejecuta.

Conceptos relacionados:

- “Instancias del gestor de bases de datos” en la página 5
- “Migración de aplicaciones a DB2 versión 8” en la página 53

Tareas relacionadas:

- “Actualización del archivo de configuración del gestor de bases de datos” en la página 27
- “Setting up the CLI environment” en el manual *CLI Guide and Reference, Volume 1*
- “Configuración del entorno Java” en la página 28
- “Configuración de las funciones de DB2 WebSphere MQ” en la página 30
- “Configuración del entorno de desarrollo de aplicaciones UNIX” en la página 34
- “Configuración del entorno de desarrollo de aplicaciones Windows” en la página 42
- “Configuración de la base de datos de ejemplo” en la página 48

Información relacionada:

- “DB2 Application Development Client” en la página 3
- “Servidores DB2 soportados” en la página 7
- “Software de desarrollo soportado por DB2” en la página 8
- “Software de desarrollo soportado por AIX” en la página 9

- “Software de desarrollo soportado por HP-UX” en la página 11
- “Software de desarrollo soportado por Linux” en la página 13
- “Software de desarrollo soportado por Solaris” en la página 18
- “Software de desarrollo soportado por Windows” en la página 20

Reconstrucción de bibliotecas compartidas para rutinas de DB2

DB2® situará en la antememoria, una vez cargadas, las bibliotecas compartidas utilizadas para procedimientos almacenados y funciones definidas por el usuario. Si está desarrollando una rutina, puede que desee cargar varias veces la misma biblioteca, y esta utilización de la antememoria le impedirá obtener la última versión de una biblioteca compartida. La manera de evitar problemas derivados de la antememoria depende del tipo de rutina:

1. **Rutinas delimitadas, no protegidas por hebras.** La palabra clave `KEEPFENCED` de configuración del gestor de bases de datos tiene un valor por omisión de `YES`. Este valor mantiene vivo el proceso en modalidad delimitada. Este valor por omisión puede interferir con la recarga de la biblioteca. Es preferible cambiar el valor de esta palabra clave por `NO` mientras esté desarrollando rutinas delimitadas, sin protección por hebras, y luego revertir al valor `YES` cuando esté preparado para cargar la versión final de la biblioteca compartida. Para obtener más información, consulte el apartado “Actualización del archivo de configuración del gestor de bases de datos” en la página 27.
2. **Rutinas fiables o protegidas por hebras.** A excepción de las rutinas de SQL (incluidos los procedimientos de SQL), la única manera de asegurarse que se obtenga una versión actualizada de la biblioteca de rutinas de DB2 cuando se utilice dicha biblioteca para rutinas fiables o protegidas por hebras, consiste en reciclar la instancia de DB2 entrando `db2stop` seguido de `db2start` en la línea de mandatos. Esto no es necesario para una rutina de SQL puesto que, cuando se recrea, el compilador utiliza un nuevo nombre de biblioteca exclusivo para evitar posibles conflictos.

Para las rutinas que no son de SQL, también puede evitar los problemas derivados de la antememoria creando la nueva versión de la rutina con una biblioteca cuyo nombre sea distinto (por ejemplo, `foo.a` pasa a ser `foo.1.a`) y utilizando después las sentencias `ALTER PROCEDURE` o `ALTER FUNCTION` de SQL con la nueva biblioteca.

Tareas relacionadas:

- “Actualización del archivo de configuración del gestor de bases de datos” en la página 27

Información relacionada:

- “Sentencia `ALTER FUNCTION`” en el manual *Consulta de SQL, Volumen 2*
- “Sentencia `ALTER PROCEDURE`” en el manual *Consulta de SQL, Volumen 2*

Actualización del archivo de configuración del gestor de bases de datos

Este archivo contiene valores importantes para el desarrollo de aplicaciones.

La palabra clave `KEEPFENCED` tiene el valor por omisión `YES`. En el caso de las rutinas (procedimientos almacenados y UDF) sin protección de hebras, este valor

l mantiene vivo el proceso de la rutina. Es preferible cambiar el valor de esta
l palabra clave por NO mientras esté desarrollando estas rutinas, y luego revertir al
l valor YES cuando esté preparado para cargar la versión final de la biblioteca
l compartida. Para obtener más información, consulte el apartado “Reconstrucción
l de bibliotecas compartidas para rutinas de DB2” en la página 27.

Nota: KEEPFENCED se conocía como KEEPDAI en versiones anteriores de DB2.

Para el desarrollo de aplicaciones Java, es necesario actualizar la palabra clave JDK_PATH con la vía de acceso donde está instalado el Java Development Kit.

Nota: JDK_PATH se conocía como JDK11_PATH en versiones anteriores de DB2.

Procedimiento:

Para cambiar estos valores, especifique:

```
db2 update dbm cfg using <palabra_clave> <valor>
```

Por ejemplo, para asignar el valor NO a la palabra clave KEEPFENCED:

```
db2 update dbm cfg using KEEPFENCED NO
```

Para establecer que el valor de la palabra clave JDK_PATH sea el directorio /home/db2inst/jdk13:

```
db2 update dbm cfg using JDK_PATH /home/db2inst/jdk13
```

Para ver los valores actuales contenidos en el archivo de configuración del gestor de bases de datos, especifique:

```
db2 get dbm cfg
```

Nota: En Windows, es necesario entrar estos mandatos en una ventana de mandatos de DB2.

Conceptos relacionados:

- “Reconstrucción de bibliotecas compartidas para rutinas de DB2” en la página 27
- “Instancias del gestor de bases de datos” en la página 5

Tareas relacionadas:

- “Configuración del entorno Java” en la página 28

Información relacionada:

- “Sentencia CREATE FUNCTION” en el manual *Consulta de SQL, Volumen 2*
- “Sentencia CREATE PROCEDURE” en el manual *Consulta de SQL, Volumen 2*
- “Mandato GET DATABASE MANAGER CONFIGURATION” en el manual *Consulta de mandatos*
- “Mandato RESET DATABASE MANAGER CONFIGURATION” en el manual *Consulta de mandatos*

Configuración del entorno Java

l Puede desarrollar programas Java para acceder a bases de datos DB2 utilizando el
l Java Developer Kit correspondiente a su plataforma. El Developer Kit incluye Java
l Database Connectivity (JDBC), una API de SQL dinámico para Java.

El soporte para JDBC de DB2 se proporciona como parte de la opción Java Enablement en clientes y servidores DB2. Mediante este soporte, puede crear y ejecutar aplicaciones y applets de JDBC. Estos programas sólo contienen SQL dinámico y utilizan una interfaz de llamada de Java para pasar sentencias de SQL a DB2.

Formando parte de Java Enablement también se proporciona SQL incorporado de DB2 para el soporte de Java (SQLJ). Mediante el soporte de SQLJ para DB2, junto con el soporte de JDBC para DB2, puede crear y ejecutar aplicaciones y applets de SQLJ. Estos programas contienen SQL estático y utilizan sentencias de SQL que están vinculadas a una base de datos DB2.

El soporte de SQLJ proporcionado por el Cliente de Desarrollo de Aplicaciones de DB2 incluye:

- El conversor SQLJ de DB2, `sqlj`, que sustituye las sentencias de SQL incorporado del programa SQLJ por sentencias fuente de Java, y genera un perfil serializado que contiene información sobre las operaciones de SQL contenidas en el programa SQLJ.
- El personalizador de perfiles SQLJ para DB2, `db2sqljcustomize`, que precompila las sentencias SQL almacenadas en el perfil serializado, las convierte en llamadas de función y genera un paquete en la base de datos DB2.

Nota: El personalizador de perfiles SQLJ para DB2 se había denominado `db2profc` en versiones anteriores de DB2.

- El impresor de perfiles SQLJ para DB2, `db2sqljprint`, que imprime en forma de texto plano el contenido de una versión personalizada para DB2 correspondiente a un perfil.

Nota: La impresora de perfiles SQLJ para DB2 se había denominado `db2profp` en versiones anteriores de DB2.

- El vinculador de perfiles SQLJ para DB2, `db2sqljbind`, que genera paquetes a partir de un programa SQLJ que se haya personalizado previamente.

Nota: Los controladores JDBC de tipos 2 y 3 basados en la CLI se desaprueban. No se añadirán funciones ni mejoras a estos controladores, que no estarán disponibles en releases futuros de DB2. Se proporciona el controlador Universal JDBC, completamente rediseñado, para sustituir estos controladores heredados con funciones más ampliadas. Se alienta al usuario a que migre las aplicaciones para utilizar el nuevo controlador lo antes posible.

Procedimiento:

Para crear aplicaciones con conectividad JDBC universal de tipo 2 o JDBC universal de tipo 4 y crear applets con conectividad JDBC universal de tipo 4, el oyente de TCP/IP debe estar en ejecución. Para garantizar esto, realice lo siguiente:

1. Establezca la variable de entorno DB2COMM en TCPIP de la forma siguiente:

```
db2set DB2COMM=TCPIP
```

2. Actualice el archivo de configuración del gestor de bases de datos con el nombre del servicio de TCP/IP que se haya especificado en el archivo de servicios:

```
db2 update dbm cfg using SVCENAME <nombre del servicio de TCP/IP>
```

Debe realizar una operación "db2stop" y "db2start" para que este valor entre en vigor.

Nota: El número de puerto utilizado para los applets y los programas de SQLJ necesita ser el mismo que el número de SVCENAME de TCP/IP utilizado en el archivo de configuración del gestor de bases de datos.

Para ejecutar aplicaciones Java de DB2, debe instalar e invocar una Máquina Virtual Java (JVM), que proporciona soporte para hebras nativas. Para ejecutar una aplicación Java utilizando hebras nativas, puede utilizar la opción -native en el mandato. Por ejemplo, para ejecutar la aplicación Java de ejemplo, DbInfo.class, puede utilizar este mandato:

```
java -native DbInfo
```

Es posible especificar hebras nativas como valor por omisión para el soporte de hebras para algunas Máquinas Virtuales Java asignando el valor "native" a la variable de entorno THREADS_FLAG. En la presente documentación se considera que el soporte de hebras nativas es el valor por omisión. Consulte la documentación de JVM para conocer cómo definir las hebras nativas como valor por omisión del sistema.

Para ejecutar applets Java de DB2, puede invocar a una Máquina Virtual Java que proporcione soporte para hebras nativas o hebras "green".

Una vez instalados y en estado de ejecución los elementos anteriores, puede configurar el entorno Java correspondiente a su sistema operativo siguiendo los pasos de una estas secciones:

- "Configuración del entorno Java de UNIX" en la página 35
- "Configuración del entorno Java de Windows" en la página 45

Para conocer la información más reciente sobre el desarrollo de aplicaciones Java de DB2, consulte la página Web situada en:

```
http://www.ibm.com/software/data/db2/udb/ad/v8/java
```

Tareas relacionadas:

- "Configuración del entorno Java de UNIX" en la página 35
- "Configuración del entorno Java de Windows" en la página 45
- "Instalación del controlador Universal JDBC de DB2" en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente*

Información relacionada:

- "db2sqljcustomize - Mandato Personalizador de perfiles SQLJ de DB2" en el manual *Consulta de mandatos*
- "db2sqljprint - Mandato Impresora de perfiles SQLJ de DB2" en el manual *Consulta de mandatos*
- "Mandato db2sqljbind - DB2 SQLJ Profile Binder" en el manual *Consulta de mandatos*

Configuración de las funciones de DB2 WebSphere MQ

DB2 y WebSphere MQ se pueden utilizar para construir aplicaciones que combinen funciones de mensajes y acceso a bases de datos. Las funciones de MQ son parecidas a las UDF (funciones definidas por el usuario) y, opcionalmente, se pueden habilitar en DB2. Utilizando estas funciones básicas es posible dar soporte a

a un amplio rango de aplicaciones, desde simples notificaciones de sucesos a la gestión de depósitos de datos, pasando por la actualización de fuentes de datos federadas.

Procedimiento:

Para configurar funciones de DB2 WebSphere MQ:

1. Instale WebSphere MQ en cada máquina física.

Asegúrese que en el servidor DB2 Universal Database se ha instalado, por lo menos, WebSphere MQ versión 5.1 con el último FixPak. Si ya está instalada esta versión de WebSphere MQ, pase al siguiente paso, “Instalar WebSphere MQ AMI.” DB2 versión 8 incluye una copia del servidor WebSphere MQ para utilizarlo con DB2. Puede encontrar instrucciones específicas de la plataforma para instalar WebSphere MQ, o para actualizar una instalación de WebSphere MQ existente, en una publicación de Guía rápida de iniciación específica de la plataforma en <http://www.ibm.com/software/ts/mqseries/library/manuals>. Asegúrese de configurar un gestor de colas por omisión a medida que avance por el proceso de instalación.

2. Instale la AMI (Application Messaging Interface) de WebSphere MQ en cada máquina física.

Se trata de una ampliación de las interfaces de programación de WebSphere MQ que proporciona una separación limpia de las tareas de administración y programación. Las funciones de DB2 WebSphere MQ requieren la instalación de esta interfaz. Si la AMI de WebSphere MQ ya está instalada en el servidor DB2, pase al siguiente paso, “Habilitar y configurar las funciones definidas por el usuario de DB2 WebSphere MQ.” Si la AMI de WebSphere MQ no está instalada, la puede instalar desde el paquete de instalación suministrado con DB2 o bajando una copia de la AMI desde el sitio Web de WebSphere MQ, en <http://www.ibm.com/software/ts/mqseries/txppacs>. Puede encontrar la AMI en “Categoría 3 – Extensiones de productos”. A efectos de comodidad, con DB2 se proporciona una copia de la AMI de WebSphere MQ. Este archivo está ubicado en el directorio `sqlib/cfg/mq`.

El nombre del archivo depende del sistema operativo:

Tabla 7. AMI de WebSphere MQ

Sistema operativo	Nombre de archivo
AIX versión 4.3 y posteriores	ma0f_ax.tar.Z
HP-UX	ma0f_hp.tar.Z
Solaris Operating Environment	ma0f_sol7.tar.Z o mq0f_sol26.tar.Z
Windows	ma0f_win.zip

Nota: Las funciones de DB2 WebSphere MQ no se soportan en Linux

Siga el proceso normal de instalación de la AMI perfilado en el archivo Readme de la AMI contenido en la imagen de instalación comprimida.

3. Habilite y configure las funciones definidas por el usuario de DB2 WebSphere MQ.

El programa de utilidad **enable_MQFunctions** es un mandato flexible que realiza las acciones siguientes:

- a. Comprueba que se ha configurado el entorno de WebSphere MQ adecuado
- b. Instala y crea una configuración por omisión para las funciones de DB2 WebSphere MQ

- c. Habilita la base de datos especificada con estas funciones
- d. Confirma que la configuración funciona

En UNIX de 64 bits, la vía de acceso de bibliotecas en tiempo de ejecución se debe modificar para incluir \$HOME/sqllib/lib32 a fin de ejecutar enable/disable_MQFunctions. Lo hacen los valores siguientes.

AIX

```
LIBPATH=$HOME/sqllib/lib32 enable_MQFunctions -n nombrebd \
-u IDusuario -p contraseña -v 0pc [-q qMgr -force -noValidate]
LIBPATH=$HOME/sqllib/lib32 disable_MQFunctions -n nombrebd \
-u IDusuario -p contraseña -v 0pc
```

HP-UX

```
SHLIB_PATH=$HOME/sqllib/lib32 enable_MQFunctions -n nombrebd \
-u IDusuario -p contraseña -v 0pc [-q qMgr -force -noValidate]
SHLIB_PATH=$HOME/sqllib/lib32 disable_MQFunctions -n nombrebd \
-u IDusuario -p contraseña -v 0pc
```

Solaris

```
LD_LIBRARY_PATH=$HOME/sqllib/lib32 enable_MQFunctions -n nombrebd \
-u IDusuario -p contraseña -v 0pc [-q qMgr -force -noValidate]
LD_LIBRARY_PATH=$HOME/sqllib/lib32 disable_MQFunctions -n nombrebd \
-u IDusuario -p contraseña -v 0pc
```

Durante el paso de habilitación se configura y habilita una base de datos para las funciones de DB2 WebSphere MQ con los pasos siguientes:

- a. Para Windows, vaya al paso 'c'.
- b. Habilite las funciones de WebSphere MQ en UNIX añadiendo el propietario de la instancia de DB2 (muchas veces db2inst1) y el ID de usuario asociado a las funciones definidas por el usuario delimitadas (muchas veces db2fenc1) al grupo mqm de WebSphere MQ. Esto es necesario para que las funciones de DB2 accedan a WebSphere MQ.
- c. Añada la variable de entorno AMT_DATA_PATH a la lista implícita para DB2. Puede editar los archivos \$HOME/sqllib/profile.env (UNIX) o %DB2PATH%\profile.env (Windows) y añadir AMT_DATA_PATH a DB2ENVLIST. También puede utilizar el mandato **db2set**.
- d. Reinicie la instancia de la base de datos para que surtan efecto los cambios realizados en la variable de entorno.
- e. Pase a los directorios \$HOME/sqllib/cfg (UNIX) o %DB2PATH%\cfg (Windows).
- f. Ejecute el mandato **enable_MQFunctions** para configurar y habilitar una base de datos para las funciones de DB2 WebSphere MQ. En un entorno DB2 ESE, sólo debe realizar este paso en el nodo de catálogo. Consulte el apartado "enable_MQFunctions" para ver una descripción completa de este mandato. A continuación se proporcionan algunos ejemplos comunes. Tras una realización satisfactoria, la base de datos especificada estará habilitada y se comprobará la configuración.
- g. Para comprobar estas funciones mediante el Procesador de línea de mandatos, emita los mandatos siguientes después de conectar con la base de datos habilitada:

```
values DB2MQ.MQSEND('una prueba')
values DB2MQ.MQRECEIVE()
```

La primera sentencia envía el mensaje "una prueba" a la cola DB2MQ_DEFAULT_Q y la segunda la recibe de vuelta.

Nota: Después de ejecutar enable_MQFunctions, el programa de utilidad establece un entorno de WebSphere MQ por omisión. Asimismo, el

programa de utilidad crea el gestor de colas de WebSphere MQ DB2MQ_DEFAULT_MQM, y la cola por omisión DB2MQ_DEFAULT_Q. El programa de utilidad instala los archivos amt.xml, amthost.xml y amt.dtd, en caso de que todavía no existan en el directorio al que apunta AMT_DATA_PATH. Si existe un archivo amthost.xml, y si éste no contiene una definición para la conexión DB2MQ, se añade al archivo una línea con la información adecuada. Se guarda una copia de archivo original con el nombre DB2MQSAVE.amthost.xml.

4. Si desea utilizar UDF de MQ transaccionales, asegúrese que la base de datos esté configurada para operaciones federadas. Hágalo mediante el mandato siguiente:

```
update dbm cfg using federated yes
```

5. Para asegurarse que las funciones de DB2 WebSphere MQ brindan la posibilidad de publicar/suscribir, también debe instalar las extensiones MQSeries Integrator o WebSphere MQ Publish/Subscribe del producto en cada máquina física. Puede encontrar información acerca de MQSeries Integrator en <http://www.ibm.com/software/ts/mqseries/integrator>

Puede encontrar información sobre la característica WebSphere MQ Publish/Subscribe en

<http://www.ibm.com/software/ts/mqseries/txppacs>

Conceptos relacionados:

- “Habilitación de MQSeries” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente*
- “Visión general funcional de WebSphere MQ” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente*
- “Mensajería de WebSphere MQ” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente*
- “Envío de mensajes con las funciones de WebSphere MQ” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente*
- “Recuperación de mensajes con las funciones de WebSphere MQ” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente*
- “Conectividad de aplicación a aplicación de WebSphere MQ” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente*
- “Comunicaciones de petición/respuesta con las funciones de WebSphere MQ” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente*
- “Publicar/suscribir con las funciones de WebSphere MQ” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente*
- “Cómo utilizar funciones de WebSphere MQ en DB2” en el manual *IBM DB2 Information Integrator Application Developer’s Guide*

Información relacionada:

- “Mandato db2mqdsn - MQ Listener” en el manual *Consulta de mandatos*
- “enable_MQFunctions” en el manual *Consulta de mandatos*
- “disable_MQFunctions” en el manual *Consulta de mandatos*

UNIX

Para ver la información de configuración de la CLI de DB2 para UNIX, vea el manual *CLI Guide and Reference*.

Configuración del entorno de desarrollo de aplicaciones UNIX

Es necesario que defina variables de entorno para la instancia de base de datos. Cada instancia del gestor de bases de datos tiene dos archivos: `db2profile` y `db2cshrc`, que son scripts (secuencias de mandatos) para definir las variables de entorno para la instancia.

Procedimiento:

Ejecute el script correspondiente al shell que está utilizando:

Para el shell bash o Korn:

```
. $HOME/sql1lib/db2profile
```

Para el shell C:

```
source $HOME/sql1lib/db2cshrc
```

donde `$HOME` es el directorio inicial del propietario de la instancia.

Si incluye este mandato en el archivo `.profile` o `.login`, el mandato se ejecutará automáticamente cuando inicie la sesión.

Si piensa utilizar ODBC, CLI de DB2 o Java, siga los pasos descritos en una de las secciones siguientes:

- Configuración del entorno ODBC de UNIX
- Configuración del entorno Java de UNIX

Conceptos relacionados:

- “Valores de variables de entorno de UNIX” en la página 34

Tareas relacionadas:

- “Setting up the UNIX ODBC environment” en el manual *CLI Guide and Reference, Volume 1*
- “Configuración del entorno Java de UNIX” en la página 35

Información relacionada:

- “Software de desarrollo soportado por AIX” en la página 9
- “Software de desarrollo soportado por HP-UX” en la página 11
- “Software de desarrollo soportado por Linux” en la página 13
- “Software de desarrollo soportado por Solaris” en la página 18

Valores de variables de entorno de UNIX

En función de la plataforma UNIX[®] que esté utilizando, se establecen valores para las variables de entorno siguiente, ya sea en `db2profile` (para los shell bash o korn) o en `db2cshrc` (para el shell C), y se coloca una llamada a estos archivos en los archivos `.profile` (shell bash o korn) o `.login` (shell C) del propietario de la instancia.

AIX[®]:

- `PATH`, incluye varios directorios de DB2[®] tales como `sql1lib/bin`
- `LIBPATH`, incluye el directorio `sql1lib/lib` (ver nota más abajo)

HP-UX:

- PATH, incluye varios directorios de DB2 tales como sqllib/bin
- SHLIB_PATH (32 bits y 64 bits) o LD_LIBRARY_PATH (64 bits), incluye el directorio sqllib/lib (ver nota más abajo)

Linux y Solaris:

- PATH, incluye varios directorios de DB2 tales como sqllib/bin
- LD_LIBRARY_PATH, incluye el directorio sqllib/lib (ver nota más abajo)

Nota: Si está ejecutando una aplicación de 32 bits local en una instancia de DB2 de 64 bits, consulte el apartado “Migración de aplicaciones desde entornos de 32 bits a entornos de 64 bits” en la página 56.

Durante la creación de instancias se crean los archivos vacíos sqllib/userprofile y sqllib/usercshrc para que el usuario coloque sus propios valores de entorno para las instancias. Estos archivos no se modifican durante una actualización de instancias (db2iupdt) en ningún FixPak de DB2 o instalación posterior de versiones. Si no desea tener los nuevos valores de entorno en los scripts db2profile o db2cshrc, puede anularlos utilizando el correspondiente script de usuario, que se invoca al final del script db2profile o db2cshrc. Durante una migración de instancias (db2imigr), estos scripts de usuario se copian encima, por lo que las modificaciones del entorno seguirán vigentes.

Tareas relacionadas:

- “Configuración del entorno de desarrollo de aplicaciones UNIX” en la página 34
- “Migración de aplicaciones desde entornos de 32 bits a entornos de 64 bits” en la página 56

Configuración del entorno Java de UNIX

Para ejecutar programas JDBC y SQLJ en UNIX con soporte JDBC de DB2, los archivos db2profile y db2cshrc del gestor de bases de datos contienen mandatos para actualizar el entorno Java. Cuando se crea una instancia de DB2, se modifican los archivos .bashrc, .profile y .cshrc, de forma que:

1. THREADS_FLAG se establece en “native” (sólo en Solaris)
2. CLASSPATH incluye:
 - “.” (el directorio actual)
 - el archivo sqllib/java/db2java.zip
 - el archivo sqllib/java/db2jcc.jar
 - el archivo sqllib/java/db2jcc_license_cu.jar

Nota: db2jcc_license_cisuz.jar también se incluye en CLASSPATH para DB2 Connect Personal Edition, DB2 Connect Enterprise Edition y DB2 ESE. Este archivo proporciona conectividad adicional a DB2 para z/OS y OS/390, DB2 para AS/400 e iSeries y DB2 para VSE y VM.

Para crear programas SQLJ, CLASSPATH también se actualiza para que incluya el archivo:

```
sqllib/java/sqlj.zip
```

Para crear programas de fuentes de datos con Java Developer Kit 1.3 o 1.4, también debe conseguir e instalar lo siguiente:

| **Bibliotecas de clase JNDI 1.2.1 (jndi.jar y providerutil.jar)**

| <http://java.sun.com/products/jndi/#download>

| **File System Service Provider 1.2 (fscontext.jar)**

| <http://java.sun.com/products/jndi/#download>

| Para Java Developer Kit 1.3, debe conseguir e instalar además lo siguiente:

| **Paquete opcional de JDBC 2.0**

| <http://java.sun.com/products/jdbc/download.html#spec>

| **Nota:** El Paquete opcional de JDBC 2.0 no es necesario para crear programas de
| fuentes de datos con Java Developer Kit 1.4.

| Para los programas de fuentes de datos, también debe actualizar CLASSPATH para
| que incluya los archivos siguientes:

- | • jndi.jar
- | • fscontext.jar
- | • providerutil.jar

| Para Java Developer Kit 1.3, también debe actualizar CLASSPATH para que
| incluya uno de los archivos siguientes:

- | • jdbc2_0-stdext.jar
- | • j2ee.jar

| **Notas:**

- | 1. Para Java Developer Kit 1.3, si ya ha actualizado CLASSPATH con j2ee.jar, no
| necesita jdbc2_0-stdext.jar.
- | 2. Para Java Developer Kit 1.4, no necesita jdbc2_0-stdext.jar ni j2ee.jar en
| CLASSPATH.

| Los programas de ejemplo de fuentes de datos se incluyen en el directorio
| sql11ib/samples/java/sqlj. Para obtener detalles, consulte el archivo README de
| ejemplo en sql11ib/samples/java.

| **Notas:**

- | 1. Si incluye otros archivos en CLASSPATH, asegúrese de especificar en primer
| lugar los archivos citados anteriormente.
- | 2. Debe eliminar cualquier referencia directa al directorio sql11ib/java12 ya que
| este directorio no existirá en futuras versiones de DB2. En su lugar, consulte el
| directorio sql11ib/java.
- | 3. DB2 Java Enablement incluye el programa de utilidad vinculador de paquetes
| de JDBC, db2jdbcbind. En los servidores DB2 versión 8, los paquetes de JDBC
| se vinculan automáticamente. El programa de utilidad sirve para crear los
| paquetes de JDBC en servidores de versiones anteriores como, por ejemplo las
| versiones 6 y 7 de DB2.

| **Procedimiento:**

| Para ejecutar rutinas Java de DB2 (procedimientos almacenados y funciones
| definidas por el usuario), debe actualizar la configuración del gestor de bases de
| datos de DB2 en el servidor para incluir la vía de acceso donde Java Developer Kit
| esté instalado en esa máquina. Puede hacer esto entrando lo siguiente en la línea
| de mandatos del servidor:

```
db2 update dbm cfg using JDK_PATH /home/db2inst/jdk13
```

donde `/home/db2inst/jdk13` es la vía de acceso donde está instalado Java Developer.

Para comprobar el valor del campo `JDK_PATH` en la configuración el gestor de bases de datos de DB2, entre este mandato en el servidor:

```
db2 get dbm cfg
```

Es posible que desee redirigir la salida a un archivo, para tener una visión más fácil. El campo `JDK_PATH` aparece cerca del comienzo de los datos de salida del mandato.

Una vez instalados y en estado de ejecución los elementos anteriores, puede configurar el entorno UNIX correspondiente a su sistema operativo siguiendo los pasos de una estas secciones:

- Configuración del entorno Java de AIX
- Configuración del entorno Java de HP-UX
- Configuración del entorno Java de Linux
- Configuración del entorno Java de Solaris

Tareas relacionadas:

- “Configuración del entorno Java de AIX” en la página 37
- “Configuración del entorno Java de HP-UX” en la página 38
- “Configuración del entorno Java de Linux” en la página 40
- “Configuración del entorno Java de Solaris” en la página 41
- “Actualización del archivo de configuración del gestor de bases de datos” en la página 27
- “Instalación del controlador Universal JDBC de DB2” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente*

Información relacionada:

- “Mandato GET DATABASE MANAGER CONFIGURATION” en el manual *Consulta de mandatos*
- “Mandato RESET DATABASE MANAGER CONFIGURATION” en el manual *Consulta de mandatos*
- “db2jdbcbind: Mandato vinculador de paquetes de JDBC de DB2” en el manual *Consulta de mandatos*

Configuración del entorno Java de AIX

Antes de implantar estas instrucciones, realice la configuración indicada en el apartado “Configuración del entorno Java de UNIX” en la página 35.

Procedimiento:

Para crear aplicaciones Java en AIX con soporte JDBC para DB2, necesita estos elementos:

1. Uno de los Developer Kit soportados que se listan en el apartado “Software de desarrollo soportado por AIX” en la página 9.
2. DB2 Java Enablement, incluido en DB2 Universal Database Versión 8 para clientes y servidores AIX.

Conceptos relacionados:

- “Programas Java de ejemplo” en la página 121

- “Consideraciones sobre los applets Java” en la página 122

Tareas relacionadas:

- “Configuración del entorno Java de UNIX” en la página 35
- “Configuración de la base de datos de ejemplo” en la página 48

Información relacionada:

- “Software de desarrollo soportado por AIX” en la página 9

Configuración del entorno Java de HP-UX

Antes de implantar estas instrucciones, realice la configuración indicada en el apartado “Configuración del entorno Java de UNIX” en la página 35.

Procedimiento:

Para crear aplicaciones Java en HP-UX con soporte JDBC para DB2, necesita instalar y configurar estos elementos en la máquina de desarrollo:

1. Uno de los Developer Kit soportados que se listan en el apartado “Software de desarrollo soportado por HP-UX” en la página 11.
2. DB2 Java Enablement, incluido en DB2 Universal Database Versión 8 para clientes y servidores HP-UX.

Para rutinas Java (procedimientos almacenados y funciones definidas por el usuario) de 32 bits en HP-UX, el JAVA_HEAP_SZ mínimo es 2048.

Para 64 bits en HP-UX, DB2 codifica el valor mínimo de pila igual al valor máximo de pila. En DB2 para HP-UX en IA64, la variable JAVA_HEAP_SZ de configuración del gestor de bases de datos se debe establecer, como mínimo, en 4096.

Para ejecutar rutinas Java en HP-UX de 32 bits, la vía de acceso de la biblioteca compartida debe ser similar a lo siguiente:

```
export SHLIB_PATH=$JAVADIR/jre/lib/PA_RISC:\
    $JAVADIR/jre/lib/PA_RISC/classic:\
    $HOME/sql1lib/lib:\
    /usr/lib:$SHLIB_PATH
```

donde \$JAVADIR normalmente se establece en /opt/java1.3 (la ubicación por omisión del SDK de Java para HP-UX de 32 bits).

Para ejecutar rutinas Java en HP-UX de 64 bits, habilite la herramienta db2hpjv emitiendo este mandato en la línea de mandatos:

```
db2hpjv -e
```

El mandato siguiente inhabilita este soporte:

```
db2hpjv -d
```

Debe efectuar un db2stop y db2start después de emitir db2hpjv -e o db2hpjv -d para que los cambios surtan efecto. Como valor predeterminado, el soporte de la rutina Java está inhabilitado.

Nota: DB2 para HP-UX no se ejecutará si el soporte de la rutina de Java está habilitado con db2hpjv -e pero Java se ha desinstalado en el sistema.

Para HP-UX de 64 bits en PA-RISC, los enlaces simbólicos a las bibliotecas siguientes deberían crearse en /usr/lib/pa20_64, en caso contrario es posible que se encuentre con un error SQL4301N:

```
/opt/java1.4/jre/lib/PA_RISC2.0W/libnet.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libzip.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/librmi.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libnio.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libverify.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libmlib_image.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libhprof.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libjaas_unix.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libawt.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libcmm.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libdcp.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libdt_socket.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libfontmanager.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libioser12.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libmawt.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libjsound.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libjava.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libjawt.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libjcov.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libjcpm.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libjdpw.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/libjpeg.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/hotspot/libjsig.sl
/opt/java1.4/jre/lib/PA_RISC2.0W/hotspot/libjvm.sl
```

Si los enlaces no existen, pueden crearse con los mandatos siguientes (que requieren autorización root para ejecutarse):

```
ln -s /opt/java1.4/jre/lib/PA_RISC2.0W/*.sl /usr/lib/pa20_64
ln -s /opt/java1.4/jre/lib/PA_RISC2.0W/hotspot/*.sl /usr/lib/pa20_64
```

Para HP-UX en IA64, los enlaces simbólicos a las bibliotecas siguientes deberían crearse en /usr/lib/hpux64, en caso contrario es posible que se encuentre con un error SQL4301N:

```
/opt/java1.4/jre/lib/IA64W/hotspot/libjunwind.so
/opt/java1.4/jre/lib/IA64W/hotspot/libjvm.so
/opt/java1.4/jre/lib/IA64W/hotspot/libjsig.so
/opt/java1.4/jre/lib/IA64W/libJdbcOdbc.so
/opt/java1.4/jre/lib/IA64W/libverify.so
/opt/java1.4/jre/lib/IA64W/librmi.so
/opt/java1.4/jre/lib/IA64W/libzip.so
/opt/java1.4/jre/lib/IA64W/libawt.so
/opt/java1.4/jre/lib/IA64W/libcmm.so
/opt/java1.4/jre/lib/IA64W/libmawt.so
/opt/java1.4/jre/lib/IA64W/libjava.so
/opt/java1.4/jre/lib/IA64W/libjcov.so
/opt/java1.4/jre/lib/IA64W/libjcpm.so
/opt/java1.4/jre/lib/IA64W/libjdpw.so
/opt/java1.4/jre/lib/IA64W/libjpeg.so
/opt/java1.4/jre/lib/IA64W/libjsound.so
/opt/java1.4/jre/lib/IA64W/libmlib_image.so
/opt/java1.4/jre/lib/IA64W/libnet.so
/opt/java1.4/jre/lib/IA64W/libnio.so
/opt/java1.4/jre/lib/IA64W/libjaas_unix.so
/opt/java1.4/jre/lib/IA64W/libioser12.so
/opt/java1.4/jre/lib/IA64W/libhprof.so
/opt/java1.4/jre/lib/IA64W/libfontmanager.so
/opt/java1.4/jre/lib/IA64W/libdt_socket.so
/opt/java1.4/jre/lib/IA64W/libdcp.so
/opt/java1.4/jre/lib/IA64W/libjawt.so
```

Si los enlaces no existen, pueden crearse con los mandatos siguientes (que requieren autorización root para ejecutarse):

```
ln -s /opt/java1.4/jre/lib/IA64W/*.so /usr/lib/hpux64
ln -s /opt/java1.4/jre/lib/IA64W/hotspot/*.so /usr/lib/hpux64
```

Conceptos relacionados:

- “Programas Java de ejemplo” en la página 121
- “Consideraciones sobre los applets Java” en la página 122

Tareas relacionadas:

- “Configuración del entorno Java de UNIX” en la página 35
- “Configuración de la base de datos de ejemplo” en la página 48

Información relacionada:

- “Software de desarrollo soportado por HP-UX” en la página 11

Configuración del entorno Java de Linux

Antes de implantar estas instrucciones, realice la configuración indicada en el apartado “Configuración del entorno Java de UNIX” en la página 35.

Procedimiento:

Para crear aplicaciones Java en Linux con soporte JDBC para DB2, necesita instalar y configurar lo siguiente en la máquina de desarrollo:

1. Uno de estos elementos:

- Uno de los Developer Kit soportados que se listan en el apartado “Software de desarrollo soportado por Linux” en la página 13.
- DB2 Java Enablement, incluido en DB2 Universal Database Versión 8 para clientes y servidores Linux.

Para ejecutar procedimientos almacenados Java o funciones definidas por el usuario, el editor de enlaces en tiempo de ejecución de Linux debe poder acceder a determinadas bibliotecas compartidas de Java, y DB2 debe poder cargar estas bibliotecas y la Máquina Virtual Java. Debido a que el programa que realiza esta carga se ejecuta con privilegios setuid, sólo buscará las bibliotecas dependientes en /usr/lib.

Cree enlaces simbólicos en /usr/lib para apuntar a las bibliotecas compartidas de Java. A continuación mostramos las bibliotecas compartidas esenciales con las que necesitará enlazar. Según las aplicaciones que esté creando y ejecutando, es posible que tenga que enlazar con bibliotecas compartidas adicionales.

Para el IBM Developer Kit 1.3, son necesarios enlaces simbólicos con libjava.so, libjvm.so y libhpi.so. Puede crear los enlaces simbólicos ejecutando los mandatos siguientes como usuario root:

```
cd /usr/lib
ln -fs $JAVAHOME/jre/bin/libjava.so .
ln -fs $JAVAHOME/jre/bin/classic/libjvm.so .
ln -fs $JAVAHOME/jre/bin/libhpi.so .
```

donde *JAVAHOME* es el directorio base del IBM Developer Kit. Si DB2 no puede encontrar estas bibliotecas, el usuario recibirá un error -4301 cuando intente

ejecutar una rutina Java y el archivo de anotaciones de administración contendrá mensajes referentes a bibliotecas no encontradas.

Nota: Una alternativa consiste en añadir las bibliotecas compartidas e de Java a `/etc/ld.so.conf` en lugar de crear enlaces en `/usr/lib`. Si lo hace así, debe ejecutar `ldconfig` como usuario root después de cambiar `/etc/ld.so.conf`, de lo contrario no funcionará, puesto que la llamada a la rutina se colgará (no se completará). Este método alternativo puede no funcionar en instancias específicas, dando también como resultado que se cuelgue la rutina (sin completarse). Si se encuentra con esta situación, cree los enlaces en el directorio `/usr/lib` tal como se ha indicado anteriormente.

Conceptos relacionados:

- “Programas Java de ejemplo” en la página 121
- “Consideraciones sobre los applets Java” en la página 122

Tareas relacionadas:

- “Configuración del entorno Java de UNIX” en la página 35
- “Configuración de la base de datos de ejemplo” en la página 48

Información relacionada:

- “Software de desarrollo soportado por Linux” en la página 13

Configuración del entorno Java de Solaris

Antes de implantar estas instrucciones, realice la configuración indicada en el apartado “Configuración del entorno Java de UNIX” en la página 35.

Procedimiento:

Para crear aplicaciones Java en el entorno operativo Solaris con soporte JDBC para DB2, necesita instalar y configurar lo siguiente en la máquina de desarrollo:

1. Uno de los Developer Kit soportados que se listan en el apartado “Software de desarrollo soportado por Solaris” en la página 18.
2. DB2 Java Enablement, incluido en DB2 Universal Database Versión 8 para clientes y servidores Solaris.

Conceptos relacionados:

- “Programas Java de ejemplo” en la página 121
- “Consideraciones sobre los applets Java” en la página 122

Tareas relacionadas:

- “Configuración del entorno Java de UNIX” en la página 35
- “Configuración de la base de datos de ejemplo” en la página 48

Información relacionada:

- “Software de desarrollo soportado por Solaris” en la página 18

Windows

Para ver la información de configuración de la CLI de DB2 para Windows, vea el manual *CLI Guide and Reference*.

Configuración del entorno de desarrollo de aplicaciones Windows

Cuando se instala DB2 AD Client en Windows NT, Windows 2000, Windows XP o Windows Server 2003, el programa de instalación actualiza el registro de configuración con las variables de entorno INCLUDE, LIB y PATH. La instalación establece la variable de entorno para todo el sistema, DB2INSTANCE, en la instancia por omisión creada, denominada DB2. DB2PATH se establece dentro de una ventana de mandatos de DB2 cuando se abre la ventana. Cuando instala DB2 AD Client en Windows 98 o Windows ME, el programa de instalación actualiza el archivo autoexec.bat.

Puede alterar estas variables de entorno para definir valores para la máquina o usuario conectado actualmente. Tenga precaución cuando modifique estas variables de entorno. No modifique la variable de entorno DB2PATH. DB2INSTANCE se define como variable de entorno a nivel de sistema. No es necesario hacer uso de la variable de registro DB2INSTDEF de DB2, que define el nombre de instancia por omisión que se debe utilizar si no se establece DB2INSTANCE.

Procedimiento:

Para modificar los valores de variables de entorno, utilice cualquiera de estos elementos:

- El panel de control de Windows XP
- El panel de control de Windows Server 2003
- El panel de control de Windows NT
- El panel de control de Windows 2000
- La ventana de mandatos de Windows 98 o Windows ME
- El archivo autoexec.bat de Windows 98 o Windows ME

Si utiliza la variable %DB2PATH% en un mandato, coloque la vía de acceso completa entre comillas, de esta forma: set LIB="%DB2PATH%\lib";%LIB%. El valor de instalación por omisión para esta variable es \Archivos de programa\IBM\SQLLIB, que contiene un espacio, por lo que es necesario utilizar comillas para no ocasionar un error.

Además, debe seguir los pasos específicos siguientes para ejecutar aplicaciones de DB2:

- Cuando cree programas C o C++, la variable de entorno INCLUDE debe incluir %DB2PATH%\INCLUDE como primer directorio.

Para hacerlo, actualice el archivo de configuración del entorno para el compilador:

Microsoft Visual C++ 6.0

```
"C:\Archivos de programa\Microsoft Visual  
Studio\VC98\bin\vcvars32.bat"
```

Microsoft Visual C++ .NET

```
"C:\Archivos de programa\Microsoft Visual Studio  
.NET\Common7\Tools\vsvars32.bat"
```

Estos archivos tienen los mandatos siguientes:

Microsoft Visual C++ 6.0

```
set INCLUDE=%MSVCDir%\ATL\INCLUDE;%MSVCDir%\INCLUDE;  
%MSVCDir%\MFC\INCLUDE;%INCLUDE%
```

Microsoft Visual C++ .NET

```
@set INCLUDE=%MSVCDir%\ATLMFC\INCLUDE;...;  
%FrameworkSDKDir%\include;%INCLUDE%
```

Para utilizar cualquiera de estos archivos con DB2, primero traslade %INCLUDE%, que define la vía %DB2PATH%\INCLUDE, desde el final de la lista hasta el comienzo, de esta manera:

Microsoft Visual C++ 6.0

```
set INCLUDE=%INCLUDE%;%MSVCDir%\ATL\INCLUDE;  
%MSVCDir%\INCLUDE;%MSVCDir%\MFC\INCLUDE
```

Microsoft Visual C++ .NET

```
@set INCLUDE=%INCLUDE%;%MSVCDir%\ATLMFC\INCLUDE;...;  
%FrameworkSDKDir%\include
```

- Cuando cree programas Micro Focus COBOL, la variable de entorno COBCPY debe apuntar a %DB2PATH%\INCLUDE\cobol_mf.
- Cuando cree programas IBM COBOL, la variable de entorno SYSLIB debe apuntar a %DB2PATH%\INCLUDE\cobol_a.
- Asegúrese de que la variable de entorno LIB apunta a %DB2PATH%\lib utilizando este mandato:

```
set LIB="%DB2PATH%\lib";%LIB%
```

Nota: Para permitir el desarrollo cruzado de aplicaciones de 64 bits desde un entorno de 32 bits, consulte el apartado "Migración de aplicaciones desde entornos de 32 bits a entornos de 64 bits" en la página 56.

- Asegúrese de que la variable de entorno DB2COMM esté definida en el servidor de una base de datos remota.
- Asegúrese que el servicio de seguridad se ha iniciado en el servidor para la autenticación SERVER, y en el cliente, cuando se utilice la autenticación CLIENT.

Nota: Puesto que la autenticación CLIENT se produce en la parte de cliente en lugar de la parte de servidor, la aplicación cliente se ejecuta bajo el contexto del usuario. La API de autenticación de Win32 requiere determinados privilegios que el usuario puede tener o no tener. Para asegurarse que la autenticación CLIENT tiene lugar satisfactoriamente, las solicitudes de autenticación se pasan de la aplicación cliente al servidor de seguridad (que se ejecuta, por omisión, bajo una sistema local de cuenta privilegiada y tiene el derecho de llamar a la API de autenticación).

Para iniciar manualmente el servicio de seguridad, utilice el mandato NET START DB2NTSECSERVER.

Normalmente, la única situación en que deseará iniciar automáticamente el servicio de seguridad será cuando la estación de trabajo actúe como cliente DB2 que se conecta a un servidor que está configurado para la autenticación de clientes. Para hacer que el servicio de seguridad se inicie automáticamente, siga estos pasos:

Windows NT

1. Pulse el botón Inicio.
2. Pulse Configuración.
3. Pulse Panel de Control.
4. En el Panel de Control, pulse Servicios.

5. En la ventana Servicios, resalte Servidor de Seguridad DB2.
6. Si esta opción no muestra los valores "Iniciado" y "Automático", pulse "Arranque".
7. Pulse Automático.
8. Pulse Bien.
9. Rearranque la máquina para que los valores seleccionados sean efectivos.

Windows 2000 y Windows Server 2003

1. Pulse el botón Inicio.
2. Para Windows 2000, pulse "Configuración" y luego pulse "Panel de control".
Para Windows Server 2003, pulse "Panel de control".
3. Pulse Herramientas Administrativas.
4. Pulse Servicios.
5. En la ventana Servicios, resalte Servidor de Seguridad DB2.
6. Si esta opción no muestra los valores "Iniciado" y "Automático", pulse Acción en el menú superior.
7. Pulse Propiedades.
8. Vaya a la página General.
9. Seleccione "Automático" en el menú desplegable Tipo de Arranque.
10. Pulse Bien.
11. Rearranque la máquina para que los valores seleccionados sean efectivos.

Windows XP

1. Pulse el botón Inicio.
2. Pulse Configuración.
3. Pulse Panel de Control.
4. Pulse Rendimiento y Mantenimiento.
5. Pulse Herramientas Administrativas.
6. Pulse Servicios.
7. En la ventana Servicios, resalte Servidor de Seguridad DB2.
8. Si esta opción no muestra los valores "Iniciado" y "Automático", pulse Acción en el menú superior.
9. Pulse Propiedades.
10. Vaya a la página General.
11. Seleccione "Automático" en el menú desplegable Tipo de Arranque.
12. Pulse Bien.
13. Rearranque la máquina para que los valores seleccionados sean efectivos.

El gestor de bases de datos en un entorno Windows XP, Windows Server 2003, Windows NT o Windows 2000 se implanta como servicio y, por tanto, no devuelve errores ni avisos cuando se inicia el servicio, aunque se puedan haber producido problemas. Esto supone que cuando ejecuta el mandato db2start o NET START, no se emiten avisos si el subsistema de comunicaciones no arranca. Por consiguiente, el usuario debe siempre examinar los registros de sucesos o el registro de notificaciones de Administración de DB2 para conocer los errores que se puedan haber producido al ejecutar estos mandatos.

- Si piensa utilizar la CLI de DB2 o Java, vaya a la tarea apropiada:
- Configuración del entorno CLI de Windows
 - "Configuración del entorno Java de Windows" en la página 45

Tareas relacionadas:

- "Migración de aplicaciones desde entornos de 32 bits a entornos de 64 bits" en la página 56
- "Setting up the Windows CLI environment" en el manual *CLI Guide and Reference, Volume 1*
- "Configuración del entorno Java de Windows" en la página 45

Información relacionada:

- "DB2 Application Development Client" en la página 3
- "Software de desarrollo soportado por Windows" en la página 20

Configuración del entorno Java de Windows

Esta sección proporciona la información necesaria para crear y ejecutar programas Java de DB2 en un entorno Windows.

Para ejecutar programas JDBC y SQLJ en una plataforma Windows[®] soportada con soporte de JDBC para DB2[®], CLASSPATH se actualiza automáticamente cuando se instala DB2 para que esa variable incluya lo siguiente:

- "." (el directorio actual)
- el archivo `sql1lib\java\db2java.zip`
- el archivo `sql1lib\java\db2jcc.jar`
- el archivo `sql1lib\java\db2jcc_license_cu.jar`

Nota: `db2jcc_license_cisuz.jar` también se incluye en CLASSPATH para DB2 Connect Personal Edition, DB2 Connect Enterprise Edition y DB2 ESE. Este archivo proporciona conectividad adicional a DB2 para z/OS y OS/390, DB2 para AS/400 e iSeries y DB2 para VSE y VM.

Para crear programas SQLJ, CLASSPATH también se actualiza para que incluya el archivo:

`sql1lib\java\sqlj.zip`

Para crear programas de fuentes de datos con Java Developer Kit 1.3 o 1.4, también debe conseguir e instalar lo siguiente:

Bibliotecas de clase JNDI 1.2.1 (`jndi.jar` y `providerutil.jar`)

<http://java.sun.com/products/jndi/#download>

File System Service Provider 1.2 (`fscontext.jar`)

<http://java.sun.com/products/jndi/#download>

Para Java Developer Kit 1.3, debe conseguir e instalar además lo siguiente:

Paquete opcional de JDBC 2.0 (`jdbc2_0-stdext.jar`)

<http://java.sun.com/products/jdbc/download.html#spec>

Nota: El Paquete opcional de JDBC 2.0 no es necesario para crear programas de fuentes de datos con Java Developer Kit 1.4. Para los programas de fuentes de datos, también debe actualizar CLASSPATH para que incluya los archivos siguientes:

- jndi.jar
- fscontext.jar
- providerutil.jar

Para Java Developer Kit 1.3, también debe actualizar CLASSPATH para que incluya uno de los archivos siguientes:

- jdbc2_0-stdext.jar
- j2ee.jar

Notas:

1. Para Java Developer Kit 1.3, si ya ha actualizado CLASSPATH con j2ee.jar, no necesita jdbc2_0-stdext.jar.
2. No necesita jdbc2_0-stdext.jar ni j2ee.jar en CLASSPATH si utiliza Java Developer Kit 1.4.

Los programas de ejemplo de fuentes de datos se incluyen en el directorio `sql11ib\samples\java\sqlj`. Para obtener detalles, consulte el archivo README de ejemplo en `sql11ib\samples\java`.

Notas:

1. Si incluye otros archivos en CLASSPATH, asegúrese de especificar en primer lugar los archivos citados anteriormente.
2. Debe eliminar cualquier referencia directa al directorio `sql11ib\java12` ya que este directorio no existirá en futuras versiones de DB2. En su lugar, haga referencia al directorio `sql11ib\java`.
3. DB2 Java Enablement incluye el programa de utilidad vinculador de paquetes de JDBC, `db2jdbcbind`. En los servidores DB2 Versión 8, los paquetes de JDBC se vinculan automáticamente. El programa de utilidad sirve para crear los paquetes de JDBC en servidores de versiones anteriores como, por ejemplo las versiones 6 y 7 de DB2.
4. El Software Developer's Kit de Microsoft para Java no está soportado en DB2 Versión 8. No puede utilizarse para la personalización de SQLJ ni para ejecutar aplicaciones JDBC de tipo 2.

Procedimiento:

Para crear aplicaciones Java en un sistema operativo Windows con soporte JDBC para DB2, necesita instalar y configurar lo siguiente en la máquina de desarrollo:

1. Uno de los kits de desarrollo soportados que se listan en el apartado "Software de desarrollo soportado por Windows" en la página 20.
2. DB2 Java Enablement, incluido en DB2 Universal Database Versión 8 para clientes y servidores Windows.

Para ejecutar rutinas Java de DB2 (procedimientos almacenados y funciones definidas por el usuario), debe actualizar la configuración del gestor de bases de datos de DB2 en el servidor para incluir la vía de acceso donde Java Developer Kit esté instalado en esa máquina. Puede hacer esto entrando lo siguiente en la línea de mandatos del servidor:

```
db2 update dbm cfg using JDK_PATH c:\jdk13
```

donde `c:\jdk13` es la vía de acceso donde está instalado el Java Developer Kit.

Si la vía de acceso donde está instalado el Java Developer Kit contiene un nombre de directorio que tiene algún espacio en blanco, puede colocar la vía de acceso entre comillas simples. Por ejemplo:

```
db2 update dbm cfg using JDK_PATH 'c:\Archivos de programa\jdk13'
```

o bien utilizar la forma abreviada del nombre de directorio, que no contiene ningún espacio en blanco:

```
db2 update dbm cfg using JDK_PATH c:\progra~1\jdk13
```

Para comprobar el valor del campo `JDK_PATH` en la configuración el gestor de bases de datos de DB2, entre este mandato en el servidor:

```
db2 get dbm cfg
```

Es posible que desee redirigir la salida a un archivo, para tener una visión más fácil. El campo `JDK_PATH` aparece cerca del comienzo de los datos de salida del mandato.

Puede colocar los mandatos siguientes en un archivo de proceso por lotes para definir el entorno Java para los Java Developer Kit de IBM. El archivo de proceso por lotes se debe ejecutar desde una ventana de mandatos de DB2. Haga todos los cambios necesarios en las vías de acceso para adaptarlas a su entorno. Puede utilizar mandatos similares para otros Java Developer Kit soportados.

El ejemplo siguiente muestra los mandatos de un archivo de proceso por lotes pensado para configurar el entorno JDK 1.3.1 de Sun:

```
set JDKPATH=D:\JAVA\SUNjdk131
set PATH=%JDKPATH%\bin;%PATH%
set CLASSPATH=%CLASSPATH%;%JDKPATH%\lib\jdbc2_0-stxext.jar
db2 update dbm cfg using JDK_PATH %JDKPATH%
db2 terminate
db2stop
db2start
```

El archivo de proceso por lotes se debe ejecutar desde una ventana de mandatos de DB2.

Conceptos relacionados:

- “Programas Java de ejemplo” en la página 121
- “Consideraciones sobre los applets Java” en la página 122

Tareas relacionadas:

- “Configuración de la base de datos de ejemplo” en la página 48
- “Instalación del controlador Universal JDBC de DB2” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente*

Información relacionada:

- “Software de desarrollo soportado por Windows” en la página 20

Base de datos de ejemplo

Configuración de la base de datos de ejemplo

Para utilizar los programas de ejemplo proporcionados con DB2, debe crear la base de datos `sample` en una estación de trabajo servidor. Se necesitarán, aproximadamente, 23,5 megabytes de espacio en la unidad de disco duro. Una base de datos DB2 vacía requiere, aproximadamente, 21,5 megabytes de espacio en la unidad de disco duro.

Además, si para acceder a la base de datos `sample` de un servidor utilizará un cliente remoto donde se ejecuta una versión diferente de DB2 o un sistema operativo diferente, es necesario vincular los programas de utilidad de la base de datos, incluidos los archivos de programas de utilidad de la CLI de DB2, con la base de datos `sample`.

Procedimiento:

Estos son los pasos para configurar la base de datos `sample`:

1. “Creación de la base de datos de ejemplo” en la página 48
2. “Catalogación de la base de datos de ejemplo” en la página 50
3. “Vinculación de los programas de utilidad de la base de datos de ejemplo” en la página 51

Tareas relacionadas:

- “Creación de la base de datos de ejemplo” en la página 48
- “Catalogación de la base de datos de ejemplo” en la página 50
- “Vinculación de los programas de utilidad de la base de datos de ejemplo” en la página 51

Información relacionada:

- “La base de datos `SAMPLE`” en el manual *Consulta de SQL, Volumen 1*
- “`db2sampl` - Mandato Crear base de datos de ejemplo” en el manual *Consulta de mandatos*

Creación de la base de datos de ejemplo

Para crear la base de datos `sample`, ejecute el mandato `db2sampl` desde la línea de mandatos.

Prerrequisitos:

Debe tener autorización de Administrador del Sistema (SYSADM) o de Control del Sistema (SYSCTRL) para crear una base de datos. SYSADM y SYSCTRL son, respectivamente, el primer y el segundo nivel de autorización más alto para DB2.

Procedimiento:

Para crear la base de datos, siga estos pasos en el servidor:

1. Compruebe que el programa `db2sampl` (utilizado para crear la base de datos `sample`) se encuentra en la vía de acceso actual. El archivo `db2profile` o

db2cshrc coloca db2samp1 en la vía de acceso utilizada por el usuario, por lo que permanecerá allí a menos que el usuario lo cambie.

- En los servidores UNIX, db2samp1 está situado en:

```
$HOME/sql1lib/bin
```

donde \$HOME es el directorio inicial del propietario de la instancia de DB2.

- En Windows, db2samp1 está situado en:

```
%DB2PATH%\bin
```

donde %DB2PATH% es el lugar donde está instalado DB2.

2. Compruebe que el valor de la variable de entorno DB2INSTANCE es igual al nombre de la instancia donde desea crear la base de datos sample. Si la variable no está definida, puede definirla utilizando estos mandatos:

- En UNIX:

para el shell bash o shell Korn:

```
DB2INSTANCE=nombre_instancia  
export DB2INSTANCE
```

y para el shell C:

```
setenv DB2INSTANCE nombre_instancia
```

- En Windows, especifique:

```
set DB2INSTANCE=nombre_instancia
```

donde *nombre_instancia* es el nombre de la instancia de base de datos.

3. Cree la base de datos sample entrando db2samp1 seguido de la ubicación donde desea crear la base de datos. En las plataformas UNIX, la ubicación es una *vía*, por ejemplo: "\$HOME", y se especificaría así:

```
db2samp1 vía
```

Por ejemplo:

```
db2samp1 $HOME
```

En Windows, la ubicación es una *unidad*, por ejemplo: "C:", y se especificaría así:

```
db2samp1 unidad
```

Por ejemplo:

```
db2samp1 C:
```

Si no especifica la vía ni la unidad, el programa de instalación instala las tablas de sample en la vía o unidad por omisión especificada por el parámetro DFTDBPATH en el archivo de configuración del gestor de bases de datos. El tipo de autenticación para la base de datos es el mismo que el de la instancia en la que se crea la base de datos.

Tareas relacionadas:

- "Creación de la base de datos de ejemplo en servidores de sistema principal o AS/400 e iSeries" en la página 50
- "Catalogación de la base de datos de ejemplo" en la página 50
- "Vinculación de los programas de utilidad de la base de datos de ejemplo" en la página 51

Creación de la base de datos de ejemplo en servidores de sistema principal o AS/400 e iSeries

Si desea ejecutar los programas de ejemplo en un servidor de sistema principal tal como DB2 UDB para z/OS y OS/390, o en un servidor AS/400 e iSeries, necesita crear una base de datos que contenga las tablas de ejemplo descritas en la publicación Consulta de SQL.

Nota: Es necesario DB2 Connect para conectar con un servidor de sistema principal.

Restricciones:

Existen algunas diferencias en la sintaxis de SQL y en los mandatos de DB2 entre el DB2 instalado en una estación de trabajo y el DB2 instalado en sistemas principales. Cuando acceda a bases de datos en DB2 UDB para z/OS y OS/390 o DB2 para AS/400 e iSeries, compruebe que sus programas utilizan sentencias de SQL y opciones de precompilación/vinculación que estén soportadas en estos sistemas de base de datos.

Procedimiento:

Para crear la base de datos:

1. Cree la base de datos `sample` en una instancia de servidor de estación de trabajo DB2 utilizando `db2samp1`.
2. Conéctese a la base de datos `sample`.
3. Exporte los datos de las tablas de ejemplo a un archivo.
4. Conéctese a la base de datos del sistema principal.
5. Cree las tablas de ejemplo.
6. Importe los datos de las tablas de ejemplo desde el archivo al que exportó los datos en el servidor de estación de trabajo.

Conceptos relacionados:

- “Export Overview” en el manual *Data Movement Utilities Guide and Reference*
- “Import Overview” en el manual *Data Movement Utilities Guide and Reference*

Tareas relacionadas:

- “Catalogación de la base de datos de ejemplo” en la página 50
- “Vinculación de los programas de utilidad de la base de datos de ejemplo” en la página 51

Ejemplos relacionados:

- “`expsamp.sqb` -- Export and import tables with table data to a DRDA database (IBM COBOL)”
- “`tbmove.sqc` -- How to move table data (C)”
- “`tbmove.sqC` -- How to move table data (C++)”

Catalogación de la base de datos de ejemplo

Para acceder a la base de datos `sample` del servidor desde un cliente remoto, es necesario catalogar la base de datos `sample` en la estación de trabajo cliente.

No es necesario catalogar la base de datos `sample` en la estación de trabajo servidor porque se catalogó en el servidor cuando creó la base de datos.

La catalogación hace que se actualice el directorio de base de datos de la estación de trabajo cliente con el nombre de la base de datos que la aplicación cliente desea acceder. Cuando se procesan las peticiones del cliente, el gestor de bases de datos utiliza el nombre catalogado para encontrar y conectar con la base de datos.

Procedimiento:

Para catalogar la base de datos `sample` en la estación de trabajo cliente remota, especifique:

```
db2 catalog database sample as sample at node nombreNodo
```

donde *nombreNodo* es el nombre del nodo servidor.

Debe también catalogar el nodo remoto para poder conectar con la base de datos.

Tareas relacionadas:

- “Catalogación de un nodo TCP/IP desde el cliente DB2” en el manual *Suplemento de instalación y configuración*
- “Catalogación de una base de datos desde un cliente DB2 mediante el CLP” en el manual *Suplemento de instalación y configuración*
- “Catalogación de un nodo NetBIOS desde el cliente DB2” en el manual *Suplemento de instalación y configuración*
- “Catalogación de un nodo de conexiones con nombre desde el cliente” en el manual *Suplemento de instalación y configuración*
- “Vinculación de los programas de utilidad de la base de datos de ejemplo” en la página 51

Vinculación de los programas de utilidad de la base de datos de ejemplo

Si para acceder a la base de datos `sample` del servidor desde un cliente remoto donde se ejecuta una versión diferente de DB2, es necesario vincular los programas de utilidad de la base de datos, incluidos los programas de utilidad de la CLI de DB2, con la base de datos `sample`.

La vinculación da como resultado la creación del paquete que el gestor de bases de datos necesita para acceder a la base de datos cuando se ejecuta una aplicación. La vinculación se puede realizar explícitamente ejecutando el mandato `BIND` con el archivo de vinculación creado durante la precompilación.

Procedimiento:

La vinculación de los programas de utilidad de la base de datos se realiza de modo diferente dependiendo de la plataforma de la estación de trabajo cliente que esté utilizando.

En una estación de trabajo cliente UNIX:

1. Conéctese a la base de datos `sample` especificando:

```
db2 connect to sample user IDusuario using contraseña
```

donde *IDusuario* y *contraseña* son el ID de usuario y la contraseña de la instancia donde reside la base de datos *sample*.

2. Vincule los programas de utilidad con la base de datos, especificando:

```
db2 bind BNDPATH/@db2ubind.lst blocking all sqlerror continue \  
messages bind.msg grant public
```

```
db2 bind BNDPATH/@db2cli.lst blocking all sqlerror continue \  
messages cli.msg grant public
```

donde *BNDPATH* es la vía donde residen los archivos de vinculación, tal como *\$HOME/sql1lib/bnd*, donde *\$HOME* es el directorio inicial del propietario de la instancia de DB2.

3. Para comprobar que la vinculación se realizó satisfactoriamente, examine los archivos de mensajes de la vinculación: *bind.msg* y *cli.msg*.

En una estación de trabajo cliente donde se ejecuta un sistema operativo Windows:

1. En el menú Inicio, seleccione Programas.
2. En el menú Programas (o "Todos los programas" en Windows XP), seleccione IBM DB2.
3. En el menú IBM DB2, seleccione "Herramientas de la Línea de Mandatos".
4. En el menú "Herramientas de la Línea de Mandatos", seleccione la Ventana de Mandatos de DB2.

Se abrirá la ventana de mandatos.

5. Conéctese a la base de datos *sample* especificando:

```
db2 connect to sample user IDusuario using contraseña
```

donde *IDusuario* y *contraseña* son el ID de usuario y la contraseña de la instancia donde reside la base de datos *sample*.

6. Vincule los programas de utilidad con la base de datos, especificando:

```
db2 bind "%DB2PATH%\bnd\@db2ubind.lst" blocking all  
sqlerror continue messages bind.msg grant public
```

```
db2 bind "%DB2PATH%\bnd\@db2cli.lst" blocking all  
sqlerror continue messages cli.msg grant public
```

donde *%DB2PATH%* es la vía donde está instalado DB2.

7. Salga de la ventana de mandatos y compruebe que la vinculación se realizó satisfactoriamente examinando los archivos de mensajes de la vinculación: *bind.msg* y *cli.msg*.

Para todos los clientes que acceden a servidores de sistema principal, especifique uno de los siguientes archivos *.lst* en lugar de *db2ubind.lst*:

ddcsmvs.lst

para DB2 para z/OS y OS/390

ddcsvm.lst

para DB2 para VM

ddcsvse.lst

para DB2 para VSE

ddcs400.lst

para DB2 para AS/400 e iSeries

Por ejemplo:

- Si accede a un servidor de DB2 para z/OS y OS/390 desde un cliente UNIX, especifique:

```
db2 bind BNDPATH/@ddcsmvs.lst blocking all sqlerror continue \
messages bind.msg grant public
```
- Si accede a un servidor de DB2 para z/OS y OS/390 desde un cliente Windows, especifique:

```
db2 bind "%DB2PATH%\bnd\@ddcsmvs.lst" blocking all
sqlerror continue messages bind.msg grant public
```

Tareas relacionadas:

- “Creación de la base de datos de ejemplo” en la página 48
- “Creación de la base de datos de ejemplo en servidores de sistema principal o AS/400 e iSeries” en la página 50
- “Catalogación de la base de datos de ejemplo” en la página 50

Información relacionada:

- “Mandato BIND” en el manual *Consulta de mandatos*

Migración de aplicaciones

Migración de aplicaciones a DB2 versión 8

DB2® versión 8 da soporte a las versiones de DB2 siguientes para la migración:

- DB2 Versión 6
- DB2 Versión 7.1
- DB2 Versión 7.2
- DataJoiner® Versión 2.1.1

Cuando migra hacia una versión posterior de DB2, los directorios de bases de datos y de nodos se migran automáticamente. Para migrar desde cualquier otra versión anterior de DB2, antes debe migrar a una de las versiones soportadas indicadas anteriormente y luego migrar desde esa versión a DB2 versión 8.

HP-UX

Si está migrando DB2 desde HP-UX Versión 10 (o anterior) a HP-UX Versión 11, debe volver a precompilar sus programas DB2 en HP-UX Versión 11 (si los programas contienen SQL incorporado) y volver a compilarlos. Esto incluye todas las aplicaciones DB2, procedimientos almacenados, funciones definidas por el usuario y programas de salida de usuario. Además, los programas DB2 que se compilan en HP-UX versión 11 no se pueden ejecutar en HP-UX versión 10 o anteriores. Los programas DB2 que se compilan y ejecutan en HP-UX versión 10 se pueden conectar de forma remota con servidores HP-UX versión 11.

Micro Focus COBOL

Las aplicaciones existentes precompiladas con DB2 Versión 2.1.1 (o anterior) y compiladas con Micro Focus COBOL se deben volver a precompilar con la versión actual de DB2, y luego volver a compilar con Micro Focus COBOL. Si estas aplicaciones creadas con versiones anteriores del precompilador de IBM® no se vuelven a precompilar, existe la posibilidad de que la base de datos se corrompa si se produce una terminación anómala.

Conceptos relacionados:

- “Recomendaciones para la migración” en el manual *Guía rápida de iniciación para servidores DB2*
- “Migración de aplicaciones, rutinas y applets Java” en la página 54
- “Ejecución de aplicaciones en dos versiones de DB2” en la página 59

Tareas relacionadas:

- “Migración de bases de datos” en el manual *Guía rápida de iniciación para servidores DB2*
- “Migración de instancias (UNIX)” en el manual *Guía rápida de iniciación para servidores DB2*
- “Migración de DB2 UDB (Windows)” en el manual *Guía rápida de iniciación para servidores DB2*
- “Migración de DB2 UDB (UNIX)” en el manual *Guía rápida de iniciación para servidores DB2*
- “Migración de aplicaciones desde entornos de 32 bits a entornos de 64 bits” en la página 56
- “Cómo asegurarse de que las aplicaciones se puedan transferir” en la página 58

Información relacionada:

- “Administrative APIs and application migration” en el manual *Administrative API Reference*
- “Restricciones de migración” en el manual *Guía rápida de iniciación para servidores DB2*
- “Version 8 incompatibilities with previous releases” en el manual *Administration Guide: Planning*

Migración de aplicaciones, rutinas y applets Java

Aplicaciones y rutinas SQLJ

En DB2® versión 8, SQLJ se basa en una nueva estructura, independiente de la plataforma, que da soporte a un rendimiento muy mejorado y la posibilidad de transferir rutinas y aplicaciones SQLJ personalizadas a través de los sistemas operativos soportados por DB2. Se han realizado varios cambios desde el soporte de SQLJ de DB2 versión 7. Es posible que estos cambios requieran alguna modificación de las rutinas y aplicaciones SQLJ, y que se precise una nueva conversión y personalización de las rutinas y aplicaciones SQLJ que se migren a DB2 versión 8.

Para migrar rutinas y aplicaciones SQLJ existentes desde la versión 7 a la versión 8, se deben llevar a cabo los pasos siguientes:

1. Cambie todas las sentencias VALUES por una selección ficticia. Por ejemplo:

```
#sql [ctxt] hv = {VALUES (DUMMY(1))}
#sql [ctxt] {SELECT DUMMY(1) INTO :hv FROM SYSIBM.SYSDUMMY1}
```

Nota: En DB2 versión 8, la sentencia VALUES y el SQL compuesto y ano se soportan con SQLJ.

2. Elimine todas las sentencias BLOCK. Cámbielas por sentencias SQL individuales. Si las dos únicas sentencias bloqueadas eran una sentencia de SQL ejecutable y una COMMIT, se puede utilizar una sola sentencia con la confirmación automática (autocommit) activada para la conexión.

Opcionalmente, sustituya el uso de sentencias BLOCK por llamadas a las API de proceso por lotes de SQLJ para lograr sus objetivos de bloqueo.

3. Vuelva a convertir las rutinas y aplicaciones utilizando los programas de utilidad de SQLJ de la versión 8 con el mandato `sqlj`, y vuelva a personalizar estas rutinas y aplicaciones utilizando el mandato `db2sqljcustomize`. Este paso se tiene que llevar siempre a cabo, aunque no se produzca ningún cambio en el código fuente, para que el nuevo soporte de plataforma cruzada que acompaña a la versión 8 pueda ejecutar los binarios.

Applets de Java™

El controlador JDBC de tipo 3, antes conocido como controlador "net", ya no se utiliza. Los applets Java de DB2 se deben migrar al controlador JDBC universal para DB2, que contiene la conectividad de tipo 4. Para convertir un applet JDBC de tipo 3 y utilizar el nuevo controlador JDBC universal para DB2, haga estos cambios:

1. El archivo del controlador JDBC universal es `db2jcc.jar`. En el archivo `.html` del applet, cambie el nombre de archivo `db2java.zip` a `db2jcc.jar`. Copie `db2jcc.jar` en el servidor Web.
2. El nombre de clase del controlador de JDBC universal para DB2 es `com.ibm.db2.jcc.DB2Driver`. En los archivos `.java` del applet, cambie el nombre de clase del controlador JDBC de tipo 3, `COM.ibm.db2.jdbc.net.DB2Driver`, por el nombre de clase del controlador JDBC universal para DB2. Puede que no exista una referencia a la clase de controlador JDBC si el applet utiliza `javax.sql.DataSource` para obtener conexiones.
3. Tanto el controlador de tipo 3 como el controlador JDBC universal para DB2 utilizan un URL de fuente de datos con el mismo formato: `jdbc:db2://servidor:númeroPuerto/nombreBaseDatos`. Sin embargo, las tres partes: `servidor`, `númeroPuerto` y `nombreBaseDatos` tienen un significado diferente en los dos controladores.

El controlador JDBC de tipo 3 es un modelo de tres niveles formado por un cliente (el navegador donde se ejecuta el applet), un servidor JDBC Applet y un servidor DB2. El servidor y el númeroPuerto contenidos en el URL hacen referencia al servidor JDBC Applet. El nombreBaseDatos es el alias de base de datos catalogado en el sistema donde se ejecuta el servidor JDBC Applet.

El cliente del controlador JDBC universal para DB2 conecta directamente con el servidor DB2 por lo que el servidor y el númeroPuerto hacen referencia al oyente TCP/IP del servidor DB2. El nombreBaseDatos es el alias de base de datos catalogado en el sistema donde se ejecuta el servidor DB2.

Si el applet utiliza `DriverManager.getConnection` para conectar con DB2, actualice los archivos `.java` y (si es necesario) el archivo `.html` con el nuevo URL del controlador JDBC universal para DB2.

4. Si el applet utiliza `COM.ibm.db2.jdbc.DB2DataSource`, se deben crear nuevos objetos `javax.sql.DataSource` de la clase `com.ibm.jcc.db2.DB2SimpleDataSource`. Es necesario actualizar el applet para que utilice esta nueva clase.

Conceptos relacionados:

- "Migración de aplicaciones a DB2 versión 8" en la página 53
- "Consideraciones sobre los applets Java" en la página 122

Tareas relacionadas:

- "Creación de aplicaciones SQLJ" en la página 131

- “Creación de rutinas SQLJ” en la página 136
- “Creación de programas SQLJ” en la página 128

Migración de aplicaciones desde entornos de 32 bits a entornos de 64 bits

Las aplicaciones Windows de 32 bits se pueden ejecutar como en Windows de 64 bits, sin cambios en el entorno de 64 bits. En UNIX, en todas las instancias de 64 bits de DB2 a excepción de Linux para IA64 y Linux para zSeries, puede migrar las aplicaciones locales de 32 bits existentes revinculando la aplicación y ejecutándola con la configuración correcta de vías de acceso a las bibliotecas. En HP-UX, sólo se puede hacer esto si se ha enlazado la aplicación con la opción +s. Si la aplicación no se ha enlazado con la opción +s, debe reconstruir la aplicación con la opción +s o con la vía de acceso de tiempo de ejecución incorporada para incluir la biblioteca de 32 bits de DB2 (vea a continuación).

Procedimiento:

En UNIX, la vía de acceso de bibliotecas correcta para las aplicaciones de 32 bits en un entorno de 64 bits es lib32. Probablemente no deseará cambiar el valor de la variable de entorno por lib32, puesto que esto afectaría a todas las aplicaciones del entorno de la instancia (de 32 y de 64 bits). Para evitarlo, puede utilizar un script reiniciador para establecer la variable de entorno solamente para la aplicación que también ejecuta desde el script.

He aquí un ejemplo de un script reiniciador que podría utilizar:

```
#!/bin/sh
echo <VALOR_VAR_ENTORNO>
export <VALOR_VAR_ENTORNO>
rm -rf $HOME/sqllib/db2dump/* > /dev/null 2>&1
echo
echo Ejecutando aplicación...
$1
echo ...Aplicación ejecutada.
```

donde <VALOR_VAR_ENTORNO> es el valor de la variable de entorno para la plataforma:

AIX: LIBPATH=\$HOME/sqllib/lib32:\$LIBPATH

HP-UX (una de entre):

SHLIB_PATH=\$HOME/sqllib/lib32:\$SHLIB_PATH

LD_LIBRARY_PATH=\$HOME/sqllib/lib32:\$LD_LIBRARY_PATH

Nota: El reiniciador sólo funcionará en HP-UX si se ha enlazado la aplicación con la opción +s.

Linux: LD_LIBRARY_PATH=\$HOME/sqllib/lib32:\$LD_LIBRARY_PATH

Solaris:

LD_LIBRARY_PATH=\$HOME/sqllib/lib32:\$LD_LIBRARY_PATH

Después de revincular la aplicación, puede ejecutar este programa reiniciador entrando el nombre del script reiniciador en la línea de mandatos, seguido del nombre del ejecutable:

```
<script_reiniciador> <ejecutable>
```


Es posible que un cambio de las variables de entorno dentro de un reiniciador no funcione para las aplicaciones que invocan otros ejecutables (como, por ejemplo, con una llamada `system()` de C) si la vía de acceso de bibliotecas del script reiniciador no es compatible con el ejecutable invocado. Para migrar estas aplicaciones, se tiene que volver a enlazar el archivo objeto y luego volver a vincular la aplicación.

Se debe utilizar la vía de acceso de bibliotecas de tiempo de ejecución con `lib32` para enlazar el archivo objeto, en lugar de la variable de entorno para la plataforma. Los scripts de creación en C, C++ y CLI para los programas de ejemplo utilizan la vía de acceso de tiempo de ejecución apropiada para permitir que las nuevas aplicaciones se transfieran fácilmente a entornos de 64 bits.

Se deben utilizar las mismas opciones de enlace para enlazar archivos objeto de aplicaciones de 32 bits existentes en entornos de 64 bits (vea los scripts de creación en los enlaces relacionados con los ejemplos siguientes). Se pueden utilizar los distintivos siguientes para incluir la biblioteca `db2` de 32 bits en la vía de acceso de bibliotecas de tiempo de ejecución:

AIX: `-L$DB2PATH/lib32`

Alternativamente se puede utilizar la opción `-blibpath` del enlazador para especificar una vía de acceso de bibliotecas de tiempo de ejecución completa. Los scripts de creación de ejemplo para AIX utilizan el método anterior.

HP-UX:

`-Wl,+b$DB2PATH/lib32`

Linux:

`-Wl,-rpath,$DB2PATH/lib32`

Solaris:

`-R$DB2PATH/lib32`

Notas:

1. Cada uno de estos mandatos asume que se utiliza el compilador para el enlace, en oposición a la realización del enlace directamente con `ld`.
2. En Solaris, se deben desestablecer `LD_LIBRARY_PATH` y `LD_LIBRARY_PATH_32` antes de enlazar la aplicación con la vía de acceso de tiempo de ejecución. De no hacerlo, se utilizará el valor de `LD_LIBRARY_PATH` o de `LD_LIBRARY_PATH_32` en lugar del valor de la vía de acceso de tiempo de ejecución.
3. En Linux, si utiliza la opción de enlace `--enable-new-dtags`, desestablezca `LD_LIBRARY_PATH` antes de ejecutar el ejecutable de 32 bits. Si no lo hace, se utilizará el valor de `LD_LIBRARY_PATH` en lugar del valor de la vía de acceso de tiempo de ejecución.

Desarrollo cruzado en Windows

Para desarrollar aplicaciones de 64 bits en entornos Windows de 32 bits, asegúrese que la variable de entorno `LIB` apunta a `%DB2PATH%\lib\Win64`. Por omisión, se añadirá `%DB2PATH%\lib` (vía de acceso de 32 bits) a la vía de acceso `LIB`, por lo que se debe asegurar que la vía de acceso `%DB2PATH%\lib\Win64` está delante de la vía de acceso de 32 bits por omisión cuando realice un desarrollo cruzado de aplicaciones de 64 bits en entornos de 32 bits. `%DB2PATH%\lib` es para desarrollar aplicaciones de 32 bits en entornos de 32 bits, o aplicaciones de 64 bits en entornos de 64 bits.

Cambio de tipos de datos largos

Si desea migrar una aplicación de 32 bits para utilizarla en un entorno operativo de 64 bits y seguir ejecutándola en un servidor de 32 bits, utilice la opción de precompilación LONGERROR para preparar la transferencia de la aplicación. Asigne el valor YES a LONGERROR en el entorno de 32 bits para que el precompilador devuelva un error cada vez que encuentre una variable de lenguaje principal del tipo largo. Luego, siga estos pasos:

1. Evite utilizar tipos largos para variables de lenguaje principal, a menos que sean necesarios. En su lugar utilice las nuevas variables de lenguaje principal transferibles, `sqlint32` o `sqluint32`. Por ejemplo:

```
EXEC SQL BEGIN DECLARE SECTION;
long y; /* esta declaración genera un error para 64 bits */
sqlint32 x; /* esta declaración es aceptable para 64 bits */
EXEC SQL END DECLARE SECTION;
```
2. Precompile la aplicación con una base de datos situada en un servidor de 64 bits. Como resultado se creará un nuevo paquete para la aplicación que se está transfiriendo.
3. Compile la aplicación en la modalidad de 64 bits.
4. Enlace la aplicación con las nuevas bibliotecas de DB2 de 64 bits.
5. Vincule la aplicación con una base de datos situada en un servidor de 64 bits.

Nota: DB2 no da soporte a la ejecución de aplicaciones de 64 bits en una instancia de 32 bits.

Conceptos relacionados:

- “Valores de variables de entorno de UNIX” en la página 34
- “Migración de aplicaciones a DB2 versión 8” en la página 53

Ejemplos relacionados:

- “`bldapp -- Builds AIX C application programs (C)`”
- “`bldapp -- Builds HP-UX C applications (C)`”
- “`bldapp -- Builds Linux C applications (C)`”
- “`bldapp -- Builds Solaris C applications (C)`”

Cómo asegurarse de que las aplicaciones se puedan transferir

Tenga en cuenta lo siguiente cuando desarrolle aplicaciones. Estas consideraciones le ayudarán a hacer que las aplicaciones se puedan transferir.

Procedimiento:

- En UNIX, utilice en sus aplicaciones la vía de búsqueda de bibliotecas por omisión: `/usr/lib:/lib`. En los sistemas operativos Windows[®] asegúrese de que la variable de entorno LIB apunta a `%DB2PATH%\lib` utilizando este mandato:

```
set LIB=%DB2PATH%\lib;%LIB%
```

Además, cree enlaces simbólicos entre la vía de acceso por omisión y la versión de DB2 que está utilizando. Asegúrese de que el enlace está el nivel mínimo de DB2 necesario para sus aplicaciones. Consulte el manual *Guía rápida de iniciación*, o el apartado sobre instalación, correspondiente a su plataforma para obtener información sobre la definición de enlaces.

- Si su aplicación necesita una versión determinada de DB2, utilice la vía de acceso que especifica la versión de DB2 en la aplicación. Por ejemplo, si su aplicación AIX® necesita la versión 5 de DB2, utilice /usr/lpp/db2_05_00/lib. Normalmente no necesitará hacer esto.
- Cuando crea una aplicación con fines de producción, en lugar de para desarrollo interno, la vía de acceso especificada en la aplicación no debe apuntar a la copia, perteneciente al propietario de la instancia, del directorio sql1ib/lib en UNIX, ni al directorio sql1ib\lib en los sistemas operativos Windows. Esto hace que la aplicación sea altamente dependiente de determinados nombres de usuario y entornos.
- En general, no debe utilizar las variables de entorno siguientes para modificar vías de búsqueda en un entorno determinado: LIBPATH (AIX), SHLIB_PATH (HP-UX 32 bits), LD_LIBRARY_PATH (HP-UX 64 bits, Linux y Solaris) y LIB (Windows). Estas variables prevalecen sobre las vías de búsqueda especificadas en las aplicaciones que se ejecutan en el entorno, por lo que las aplicaciones pueden no ser capaces de encontrar las bibliotecas o archivos que necesitan.
- En DB2 Universal Database™ Versiones 6, 7 y 8, todos los elementos de matrices de caracteres con semántica de cadena pertenecen al tipo CHAR, en lugar de utilizar otras variaciones, tales como CHAR sin signo. Las aplicaciones que codifique con DB2 Universal Database versión 6, versión 7 o versión 8 deben seguir esa práctica.

Si tiene aplicaciones de DB2 versión 1 que hacen uso del tipo CHAR sin signo, el compilador puede emitir avisos o errores debido a conflictos entre el tipo CHAR sin signo de las aplicaciones de la versión 1 y el tipo CHAR de los prototipos de función de las Versiones 6, 7 u 8. Si esto ocurre, utilice la opción de compilador -DSQLOLDCHAR para evitar este problema.

Conceptos relacionados:

- “Valores de variables de entorno de UNIX” en la página 34
- “Migración de aplicaciones a DB2 versión 8” en la página 53

Tareas relacionadas:

- “Configuración del entorno de desarrollo de aplicaciones Windows” en la página 42
- “Migración de aplicaciones desde entornos de 32 bits a entornos de 64 bits” en la página 56

Ejecución de aplicaciones en dos versiones de DB2

En las plataformas UNIX®, si tiene aplicaciones pertenecientes a una versión de release anterior de DB2® y desea ejecutarlas en una instancia de base de datos de la versión anterior y también en una instancia de DB2 versión 8 en la misma máquina, puede ser necesario que haga algunos cambios en el entorno. Para determinar qué cambios debe hacer, responda a las preguntas siguientes y luego repase la sección “Condiciones” para ver si cualquiera de las condiciones corresponde a su situación.

Se utiliza un sistema AIX® para explicar los puntos surgidos. Los mismos conceptos son aplicables a otras plataformas UNIX, pero los detalles pueden diferir, como por ejemplo las variables de entorno y mandatos específicos.

Preguntas

Pregunta 1: ¿Cómo estaba enlazada la aplicación en la versión anterior de DB2 con la biblioteca de ejecución del cliente DB2, por ejemplo, ibdb2.a en AIX?

Para determinar la vía de búsqueda de la biblioteca compartida incorporada para un ejecutable, utilice uno de los siguientes mandatos del sistema en el directorio donde reside el ejecutable (que puede ser /usr/bin o el directorio de la instancia):

AIX dump -H *nombreArchivo_ejecutable*

HP-UX

chatr *nombreArchivo_ejecutable*

Linux objdump -p *nombreArchivo_ejecutable*

Solaris

dump -Lv *nombreArchivo_ejecutable*

donde *nombreArchivo_ejecutable* es el nombre del archivo ejecutable correspondiente a la aplicación.

Lo siguientes es un vuelco de la aplicación C de ejemplo de DB2 Versión 7.2 para AIX, dbcat, que se ha obtenido en el subdirectorio "samples" de la instancia de DB2, /home/dbinst/samples/c:

Vuelco de ejemplo de la aplicación C dbcat			

dbcat:			
Loader Section			
Loader Header Information			
VERSION#	#SYMTABLEENT	#RELOCENT	LENIDSTR
0x00000001	0x0000000f	0x00000015	0x00000047
#IMPFILEID	OFFIDSTR	LENSTRTBL	OFFSTRTBL
0x00000003	0x00000284	0x0000007f	0x000002cb
Import File Strings			
INDEX	PATH	BASE	MEMBER
0	/home/db2inst/sql1lib/lib:/usr/lib:/lib		
1		libc.a	shr.o
2		libdb2.a	shr.o

La línea 0 (cero) muestra las vías de directorios en donde el ejecutable busca las bibliotecas compartidas a las que está enlazado. Las líneas 1 y 2 muestran las bibliotecas compartidas a las que está enlazada la aplicación.

En función del modo en que se haya creado la aplicación, puede ver las vías de acceso siguientes: /usr/lpp/db2_07_01_0000/lib, *INSTHOME*/sql1lib/lib (donde *INSTHOME* es el directorio inicial del propietario de la instancia de la base de datos), o simplemente la combinación /usr/lib:/lib.

Pregunta 2: ¿Cómo están configuradas las bibliotecas de ejecución de DB2 en el sistema?

Si está instalado DB2 versiones 1, 2, 5, 6, 7 ó 8, existe un paso opcional que crea enlaces simbólicos entre la vía de la biblioteca compartida por omisión, `/usr/lib`, y la vía de instalación de DB2 que contiene las bibliotecas de ejecución del cliente DB2.

Las vías de instalación correspondientes a las diferentes versiones de DB2 son las siguientes:

Versión 1

`/usr/lpp/db2_01_01_0000/lib`

Versión 2

`/usr/lpp/db2_02_01/lib`

Versión 5

`/usr/lpp/db2_05_00/lib`

Versión 6.1

`/usr/lpp/db2_06_01/lib`

Versión 7

`/usr/lpp/db2_07_01/lib`

Versión 8

`/usr/opt/db2_08_01/lib`

En todos los casos, las bibliotecas compartidas de ejecución se denominan `libdb2.a`.

Sólo una única versión de estas bibliotecas puede ser la biblioteca por omisión en un momento determinado cualquiera. DB2 proporciona este valor por omisión para que cuando cree una aplicación, ésta no dependa de una versión determinada de DB2.

Pregunta 3: ¿Especifica vías de búsqueda diferentes en su entorno?

Puede alterar la vía de búsqueda de biblioteca compartida que está codificada en su aplicación utilizando la variable de entorno `LIBPATH` en AIX, `SHLIB_PATH` en HP-UX de 32 bits, `SHLIB_PATH` o `LD_LIBRARY_PATH` en HP-UX de 64 bits, o `LD_LIBRARY_PATH` en Linux y Solaris. En Solaris, también puede utilizar `LD_LIBRARY_PATH_32` para aplicaciones de 32 bits y `LD_LIBRARY_PATH_64` para aplicaciones de 64 bits. Puede visualizar la vía de búsqueda de bibliotecas utilizando el mandato correspondiente a su plataforma, tal como se describe en la respuesta a la Pregunta 1.

Condiciones

Una vez obtenidas las respuestas a las preguntas anteriores, puede que necesite efectuar cambios en el entorno. Lea las condiciones listadas a continuación. Si una de las condiciones es aplicable a su situación, haga los cambios necesarios.

Condición 1: Si una aplicación de la Versión 7 carga una biblioteca compartida contenida en la vía de bibliotecas compartidas por omisión de AIX, `/usr/lib/libdb2.a`, y

- Si existe un enlace simbólico entre `/usr/lib/libdb2.a` y `/usr/lpp/db2_07_01/lib/libdb2.a`, y el servidor de bases de datos es DB2 Universal Database versión 8 para AIX, siga uno de estos pasos:
 - Cambie el enlace simbólico para que apunte a:

```
/usr/opt/db2_08_01/lib/libdb2.a
```

Como usuario root, puede cambiar enlaces utilizando el mandato "db2ln", de esta manera:

```
/usr/opt/db2_08_01/cfg/db2ln
```

- Defina la variable de entorno LIBPATH de forma que apunte a /usr/opt/db2_08_01/lib o *INSTHOME*/sql1lib/lib, donde *INSTHOME* es el directorio inicial del propietario de la instancia de DB2 Versión 8.
- Configure una conexión TCP/IP entre la instancia de la aplicación cliente y la instancia de servidor.
- Si existe un enlace simbólico entre /usr/lib/libdb2.a y /usr/opt/db2_08_01/lib/libdb2.a, y el servidor de bases de datos es DB2 versión 7, configure una conexión TCP/IP entre la instancia de la aplicación cliente y la instancia de servidor.

Condición 2: Si una aplicación de la versión 7 carga una biblioteca compartida de la vía \$HOME de un propietario de instancia de DB2 versión 7 (\$HOME/sql1lib/lib/libdb2.a) y el servidor de bases de datos es DB2 Universal Database™ versión 8 para AIX, siga uno de estos pasos:

- Migre la instancia de la aplicación hacia la misma versión a la que pertenece la instancia del servidor de bases de datos.
- Defina la variable de entorno LIBPATH de forma que apunte a /usr/opt/db2_08_01/lib o *INSTHOME*/sql1lib/lib, donde *INSTHOME* es el directorio inicial del propietario de la instancia de la Versión 8.
- Configure una conexión TCP/IP entre la instancia de la aplicación cliente y la instancia de servidor.

Condición 3: Si una aplicación de la Versión 7 carga una biblioteca compartida que reside en la vía de instalación de DB2 Versión 7 (/usr/lpp/db2_07_01/lib/libdb2.a) y el servidor de bases de datos es DB2 Universal Database versión 8 para AIX, siga uno de estos pasos:

- Defina la variable de entorno LIBPATH de forma que apunte a /usr/opt/db2_08_01/lib o *INSTHOME*/sql1lib/lib, donde *INSTHOME* es el directorio inicial del propietario de la instancia de base de datos.
- Configure una conexión TCP/IP entre la instancia de la aplicación cliente y la instancia de servidor.

Condición 4: Si una aplicación de la Versión 7 carga una biblioteca compartida que reside en la vía de instalación de DB2 Universal Database Versión 8 para AIX (/usr/opt/db2_08_01/lib/libdb2.a), y el servidor de bases de datos es DB2 versión 7, configure una conexión TCP/IP entre la instancia de la aplicación cliente y la instancia de servidor.

Conceptos relacionados:

- "Valores de variables de entorno de UNIX" en la página 34
- "Instancias del gestor de bases de datos" en la página 5
- "Migración de aplicaciones a DB2 versión 8" en la página 53

Tareas relacionadas:

- "Configuración del entorno de desarrollo de aplicaciones Windows" en la página 42

Dónde ir a continuación

Una vez que se haya configurado el entorno, el usuario está preparado para crear aplicaciones de DB2. En el capítulo siguiente se explican los programas de ejemplo y los archivos relacionados, incluyendo los archivos de creación. Los capítulos siguientes a éste utilizan los archivos de creación y los ejemplos para mostrarle cómo compilar, enlazar y ejecutar las aplicaciones en el entorno de aplicación. Lea el capítulo específico para las necesidades de desarrollo de la aplicación determinada.

Capítulo 3. Programas de ejemplo y archivos relacionados

Archivos de ejemplo	65	Ejemplos de funciones de tabla de OLE DB (base de datos de enlace e integración de objetos) . . .	97
Programas de ejemplo por lenguaje e interfaz de aplicación	71	Ejemplos de Perl	97
Ejemplos de C	71	Ejemplos de PHP	98
Ejemplos de C++	75	Ejemplos de REXX	100
Ejemplos de C#	78	Ejemplos de plugins de seguridad	102
Ejemplos de la CLI	79	Ejemplos de procedimientos de SQL.	102
Ejemplos de procesador de línea de mandatos (CLP)	82	Ejemplos de Visual Basic	105
Ejemplos de COBOL	82	Ejemplos de Visual Basic .NET	107
Ejemplos de reconfiguración dinámica	86	Ejemplos de Visual C++.	108
Ejemplos de JDBC	87	Ejemplos de Windows Management Instrumentation	109
Ejemplos de SQLJ	90	Archivos de creación, makefiles y programas de utilidad de comprobación de errores.	110
Ejemplos de Java WebSphere	92	Archivos de creación	110
Ejemplos de plugins de Java.	93	Makefiles.	113
Ejemplos de salidas del usuario para la gestión de anotaciones cronológicas	94	Programas de utilidad de comprobación de errores.	116
Ejemplos de OLE (Object Linking and Embedding)	96		

En este capítulo se describen los programas de ejemplo y los archivos relacionados para los lenguajes de programación para todas las plataformas soportadas por DB2. Este capítulo presenta el diseño de los ejemplos en función de la estructura de componentes de DB2, y proporciona una lista de los ejemplos de DB2 con una descripción de cada uno. También describe los usos de los archivos de creación, los makefiles y los programas de utilidad de comprobación de errores que se proporcionan con DB2.

Archivos de ejemplo

Los programas de ejemplo se incluyen con DB2® Application Development (DB2 AD) Client. No todos los programas de ejemplo están disponibles en todas las plataformas o lenguajes de programación soportados. Puede utilizar los programas de ejemplo como plantillas para crear sus propias aplicaciones, y como herramienta de aprendizaje para comprender la funcionalidad de DB2.

Los programas de ejemplo de DB2 se proporcionan "tal cual", sin garantía de ninguna clase. Corresponde al usuario, y no a IBM®, asumir todo el riesgo respecto a la calidad, funcionamiento y reparación de los defectos.

Además de los archivos de los programas de ejemplo, DB2 proporciona otros archivos de ejemplo en los directorios `sql1lib/samples` (UNIX) y `sql1lib\samples` (Windows). Estos archivos son archivos de creación y makefiles para compilar y enlazar los programas de ejemplo, programas de utilidad de comprobación de errores (con enlaces hacia la mayoría de los programas de ejemplo), y diversos archivos de script para el desarrollo de aplicaciones. Por ejemplo, en varios subdirectorios correspondientes a diversos lenguajes de programación donde se proporcionan scripts para catalogar y descatalogar procedimientos almacenados y funciones definidas por el usuario. Cada directorio "samples" ("ejemplos") incluye un archivo README que describe los archivos contenidos en el directorio.

Se proporcionan versiones HTML de la mayoría de los archivos fuente de programas de ejemplo, a los que se puede acceder en la documentación en línea. Los ejemplos en HTML están enlazados con temas de la documentación para mostrar la funcionalidad descrita por ellos. Los elementos clave, tales como las sentencias de SQL y las API de DB2, tienen enlaces dinámicos dentro de los ejemplos en HTML para que el usuario pueda ir directamente a la documentación donde están descritos. La mayoría de los ejemplos en HTML tienen un enlace en la sección de comentarios, al comienzo del archivo, que conduce a un archivo de salida que muestra los resultados normales de ejecutar el programa de ejemplo compilado. Observe que la salida real depende, en muchos casos, de la máquina y la plataforma, por lo que la salida que obtendrá al ejecutar un mismo programa puede variar.

La tabla siguiente muestra los directorios de ejemplos y los archivos README para los principales lenguajes de programación e interfaces soportados en cada plataforma. Los archivos README están enlazados dinámicamente dentro de la documentación en línea, y los listados de ejemplo contenidos en ellos tienen enlaces dinámicos con el código fuente del archivo de ejemplo. Puede también acceder a los archivos de ejemplo contenidos en los directorios "samples" listados. En las vías de directorios se utiliza el estilo de barra inclinada propio de UNIX[®], como en `samples/c`, excepto cuando el directorio sólo es para Windows[®], como en `samples\VB\ADO`.

Tabla 8. Archivos README de ejemplo por plataforma

Plataforma → Lenguaje	AIX®	HP-UX	Linux	Solaris	Windows
C samples/c	README	README	README	README	README
C++ samples/cpp	README	README	README	README	README
C# samples\ .NET\cs	n/a	n/a	n/a	n/a	README
CLI samples/cli	README	README	README	README	README
CLP samples/clp	README	README	README	README	README
IBM COBOL samples/cobol	README	n/a	n/a	n/a	README
Micro Focus COBOL samples/cobol_mf	README	README	README	README	README
JDBC samples/java/jdbc	README	README	README	README	README
SQLJ samples/java/sqlj	README	README	README	README	README
Perl samples/perl	README	README	README	README	README
PHP samples/php	README	README	README	README	README
Procedimientos de SQL samples/sqlproc	README	README	README	README	README
Visual Basic samples\VB\ADO	n/a	n/a	n/a	n/a	ReadMe.txt
Visual Basic. NET samples\ .NET\vb	n/a	n/a	n/a	n/a	README

La extensión de archivo de los programas de ejemplo es diferente para cada lenguaje soportado, y dentro de cada lenguaje, según sea un programa de SQL incorporado o de SQL no incorporado. Las extensiones de archivo pueden también diferir para grupos de programas dentro de un lenguaje. Las tablas siguientes muestran las distintas clases de extensiones de los archivos de ejemplo:

Extensiones de archivos de ejemplo por lenguaje

Tabla 9 en la página 68.

Extensiones de archivos de ejemplo por grupo de programas

Tabla 10 en la página 68.

Tabla 9. Extensiones de archivos de ejemplo por lenguaje

Lenguaje	Directorio	Programas de SQL incorporado	Programas de SQL no incorporado
C	samples/c samples/cli (progr. CLI)	.sqc	.c
C++	samples/cpp	.sqc (UNIX) .sqx (Windows)	.c (UNIX) .cxx (Windows)
C#	samples\.NET\cs		.cs
COBOL	samples/cobol samples/cobol_mf	.sqb	.cbl
Java™	samples/java/jdbc samples/java/sqlj samples/java/WebSphere samples/java/plugin	.sqlj	.java
REXX	samples/rexx	.cmd	.cmd
Visual Basic	samples\VB\ADO samples\VB\MTS samples\VB\RDO		.bas .frm .vbp
Visual Basic. NET	samples\.NET\vb		.vb
Visual C++	samples\VC\ADO		.cpp .dsp .dsw

Tabla 10. Extensiones de archivos de ejemplo por grupo de programas

Grupo de programas	Directorio	Extensión de archivo
CLP	samples/clp	.db2
OLE	samples\ole\msvb (Visual Basic) samples\ole\msvc (Visual C++)	.bas .vbp (Visual Basic) .cpp (Visual C++)
OLE DB	samples\oledb	.db2
procedimientos de SQL	samples/sqlproc	.db2 (Scripts de procedimiento SQL) .c (Aplicaciones cliente CLI) .sqc (Aplicaciones cliente en C intercaladas) .java (Aplicaciones cliente JDBC)
Salida de usuario	samples/c	.ctsm (UNIX & Windows) .cdisk (UNIX & Windows) .ctape (UNIX) .cxbsa (UNIX)

Nota:

Delimitadores de directorios

En UNIX el delimitador de directorios es una barra inclinada (/). En Windows es una barra inclinada invertida (\). En las tablas, se utilizan los delimitadores de UNIX a menos que el directorio sólo exista en Windows.

Programas de SQL incorporado

Requieren precompilación, excepto para los programas REXX de SQL incorporado, en los que las sentencias SQL se interpretan al ejecutar el programa.

Ejemplos de IBM COBOL

Sólo se proporcionan para sistemas operativos AIX y Windows de 32 bits del subdirectorio cobol.

Ejemplos de Micro Focus COBOL

Sólo se proporcionan para AIX, HP-UX, el entorno operativo Solaris y los sistemas operativos Windows de 32 bits; se encuentran en el subdirectorio cobol_mf.

Ejemplos de Java

Comprenden applets, aplicaciones y rutinas de JDBC (Java Database Connectivity) y de SQLJ. También ejemplos de WebSphere® y archivos de ejemplo de plugin para el Centro de Control de DB2. Existen programas Java de ejemplo para todas las plataformas DB2 soportadas.

Ejemplos de REXX

Sólo se proporcionan para AIX y sistemas operativos Windows de 32 bits.

Ejemplos de CLP

Son scripts del Procesador de Línea de Mandatos que ejecutan sentencias SQL.

Ejemplos de OLE

Están pensados para OLE (Object Linking and Embedding) cuando se utiliza Microsoft® Visual Basic y Microsoft Visual C++; se proporcionan sólo para sistemas operativos Windows.

Ejemplos de Visual Basic

Comprenden ejemplos de ActiveX Data Objects, Remote Data Objects y Microsoft Transaction Server; se proporcionan sólo para sistemas operativos Windows.

Ejemplos de Visual C++

Comprenden ejemplos de ActiveX Data Object; se proporcionan sólo para sistemas operativos Windows.

Ejemplos de salidas de usuario

Son programas de salida de usuario para la gestión de archivos de anotaciones; se utilizan para archivar y acceder a archivos de anotaciones de bases de datos. La extensión de los archivos se debe cambiar para que sea .c y se deben compilar como programas de lenguaje C.

En la mayoría de las plataformas, el directorio de los programas de ejemplo es normalmente de sólo lectura. Para modificar o crear los programas de ejemplo, debe copiarlos en un directorio de trabajo.

Estructura y diseño

La mayoría de los programas de ejemplo de DB2 en C, CLI, C++, C#, Java, Perl, PHP, Visual Basic ADO y Visual Basic .NET se organizan de forma que reflejen un modelo de diseño basado en objetos de los componentes de la base de datos.. Los programas de ejemplos están agrupados en categorías que representan diversos niveles de DB2. Un prefijo de dos caracteres situado al comienzo del nombre del ejemplo indica el nivel al que pertenece un ejemplo. No todos los niveles están representados en los programas para cada API. Los niveles están representados de esta manera para el conjunto de programas de ejemplo:

prefijo

	Nivel de DB2
il	Nivel de imagen de instalación
cl	Nivel de cliente
in	Nivel de instancia
db	Nivel de base de datos
ts	Nivel de espacio de tablas
tb	Nivel de tabla
dt	Nivel de tipo de datos

Los niveles muestran una estructura jerárquica. El nivel de imagen de instalación es el nivel superior de DB2. Debajo de este nivel, una aplicación a nivel de cliente puede acceder a diversas instancias; una instancia puede tener una o más bases de datos; una base de datos tiene espacios de tablas donde existen las tablas, y que a su vez contienen datos de diferentes tipos.

Este diseño no incluye todos los programas de ejemplo de DB2. La finalidad de algunos programas de ejemplo es mostrar diversos métodos de acceder a los datos. Estos métodos son la finalidad principal de estos programas, por lo que están representados por estos métodos de una manera similar a la descrita anteriormente:

prefijo

	Método de programación
fn	Función de SQL
sp	Procedimiento almacenado
ud	Función definida por el usuario

Además de estas categorías, existe un conjunto de programas de aprendizaje que describen conceptos básicos de programación de bases de datos. Estos programas utilizan algunas de las funciones más simples representadas en el diseño de los programas de ejemplo, y comienzan con los caracteres "tut" (de "tutorial").

Existen otros programas de ejemplo no incluidos en este diseño, tales como los programas de salida de usuario para la gestión de archivos de anotaciones, programas en COBOL, Visual C++ y REXX, programas de Object Linking and Embedding (OLE), scripts de CLP y procedimientos SQL.

Nota: Los nombres de los programas Java, C# y Visual Basic .NET tienen el primer carácter, y a veces otros caracteres, en mayúsculas. Los programas Java no tienen caracteres de subrayado en los nombres de los programas de ejemplo. Los ejemplos de Visual Basic ADO tienen algunos caracteres en mayúsculas (pero no el primer carácter).

Para encontrar las actualizaciones de ejemplos más recientes, visite la página Web sobre ejemplos de desarrollo de aplicaciones DB2:

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

Conceptos relacionados:

- "Archivos de creación" en la página 110

- “Makefiles” en la página 113
- “Programas de utilidad de comprobación de errores” en la página 116

Información relacionada:

- “Ejemplos de C” en la página 71
- “Ejemplos de la CLI” en la página 79
- “Ejemplos de JDBC” en la página 87
- “Ejemplos de SQLJ” en la página 90
- “Ejemplos de procedimientos de SQL” en la página 102
- “Ejemplos de Visual Basic” en la página 105
- “Ejemplos de Visual C++” en la página 108
- “Ejemplos de OLE (Object Linking and Embedding)” en la página 96
- “Ejemplos de funciones de tabla de OLE DB (base de datos de enlace e integración de objetos)” en la página 97
- “Ejemplos de procesador de línea de mandatos (CLP)” en la página 82
- “Ejemplos de salidas del usuario para la gestión de anotaciones cronológicas” en la página 94
- “Ejemplos de COBOL” en la página 82
- “Ejemplos de Java WebSphere” en la página 92
- “Ejemplos de plugins de Java” en la página 93
- “Ejemplos de Windows Management Instrumentation” en la página 109
- “Ejemplos de REXX” en la página 100
- “Ejemplos de reconfiguración dinámica” en la página 86
- “Ejemplos de C#” en la página 78
- “Ejemplos de Visual Basic .NET” en la página 107
- “Ejemplos de Perl” en la página 97
- “Ejemplos de PHP” en la página 98
- “Ejemplos de plugins de seguridad” en la página 102

Programas de ejemplo por lenguaje e interfaz de aplicación

Ejemplos de C

Directorio UNIX: sqllib/samples/c. Directorio Windows: sqllib\samples\c.

Extensiones de archivos: .c (sin SQL incorporado); .sqc (SQL incorporado)

Tabla 11. Archivos de programas de ejemplo de C

Nombre del programa de ejemplo	Descripción del programa
Ejemplos de aprendizaje - Programas que muestran cómo realizar operaciones básicas con bases de datos.	
tut_mod.sqc	Cómo modificar datos de tablas.
tut_read.sqc	Cómo leer tablas.
tut_use.sqc	Cómo utilizar una base de datos.
Nivel de cliente - Ejemplos referentes al nivel de cliente de DB2.	

Tabla 11. Archivos de programas de ejemplo de C (continuación)

Nombre del programa de ejemplo	Descripción del programa
cli_info.c	Cómo obtener y definir información a nivel de cliente.
clisnap.c	Cómo capturar una instantánea a nivel de cliente.
Nivel de instancia - Ejemplos referentes al nivel de instancia de DB2.	
inattach.c	Cómo conectar/desconectar respecto a una instancia.
inauth.sqc	Cómo visualizar autorizaciones a nivel de instancia.
ininfo.c	Cómo obtener y definir información a nivel de instancia.
insnap.c	Cómo capturar una instantánea a nivel de instancia.
instart.c	Cómo detener e iniciar la instancia local actual.
Nivel de base de datos - Ejemplos referentes a objetos de bases de datos en DB2.	
dbauth.sqc	Cómo visualizar/otorgar/revocar autorizaciones a nivel de base de datos.
dbcfg.sqc	Cómo configurar parámetros de base de datos y de gestor de bases de datos.
dbconn.sqc	Cómo conectar y desconectar con respecto de una base de datos.
dbcreate.c	Cómo crear y eliminar bases de datos.
dbhistfile.sqc	Cómo leer y actualizar una entrada de archivo de historia de recuperación de base de datos.
dbinfo.c	Cómo obtener y definir información a nivel de base de datos.
dbinline.sqc	Cómo utilizar el lenguaje de procedimientos SQL incorporados.
dbinspect.sqc	Cómo verificar la integridad arquitectural con la API de DB2 db2Inspect.
dblogconn.sqc	Cómo leer archivos de registro de base de datos de forma asíncrona con una conexión de base de datos.
dblognoconn.sqc	Cómo leer archivos de registro de base de datos de forma asíncrona sin ninguna conexión de base de datos.
dbmcon.sqc	Cómo conectar y desconectar con respecto a varias bases de datos.
dbmcon1.h	Archivo de cabecera de dbmcon1.sqc.
dbmcon1.sqc	Archivo de soporte de dbmcon.sqc.
dbmcon2.h	Archivo de cabecera de dbmcon2.sqc.
dbmcon2.sqc	Archivo de soporte de dbmcon.sqc.
dbmigrat.c	Cómo migrar una base de datos.
dbpkg.sqc	Cómo trabajar con paquetes.
dbrec.sqc	Cómo utilizar la API db2GetRecommendations.
dbrecov.sqc	Cómo recuperar una base de datos.
dbredirect.sqc	Cómo realizar una Restauración redirigida de una base de datos.
dbrestore.sqc	Cómo restaurar una base de datos a partir de una copia de seguridad.
dbrollfwd.sqc	Cómo realizar una retrotracción después de una restauración de una base de datos.
dbsample.sqc	Cómo crear la base de datos de ejemplo de forma que incluya tablas y vistas de Host y AS/400.
dbsnap.c	Cómo capturar una instantánea a nivel de base de datos.

Tabla 11. Archivos de programas de ejemplo de C (continuación)

Nombre del programa de ejemplo	Descripción del programa
dbthrds.sqc	Cómo utilizar las API de múltiples contextos en UNIX.
dbthrds.sqc	Cómo utilizar las API de múltiples contextos en Windows.
dbuse.sqc	Cómo utilizar objetos de base de datos.
Nivel de espacio de tablas - Ejemplos referentes al nivel de espacio de tablas de DB2.	
tscreate.sqc	Cómo crear y eliminar agrupaciones de almacenamientos intermedios y espacios de tablas.
tsinfo.sqc	Cómo obtener información a nivel de espacio de tablas.
Nivel de tabla - Ejemplos referentes a objetos de tabla de DB2.	
tbast.sqc	Cómo utilizar una tabla de etapas para actualizar una Tabla de resumen automático diferida.
tbcompress.sqc	Cómo crear tablas con opciones de compresión de valores por omisión y nulos.
tbconstr.sqc	Cómo trabajar con restricciones de tabla.
tbcreate.sqc	Cómo crear, modificar y eliminar tablas.
tbident.sqc	Cómo utilizar columnas de identidad.
tbinfo.sqc	Cómo obtener y definir información a nivel de tabla.
tbintrig.sqc	Cómo utilizar un desencadenante 'INSTEAD OF' en una vista.
tbload.sqc	Cómo cargar en una base de datos particionada.
tbmerge.sqc	Cómo utilizar la sentencia MERGE.
tbmod.sqc	Cómo modificar información de una tabla.
tbmove.sqc	Cómo mover datos de tablas.
tbonlineinx.sqc	Cómo crear y reorganizar índices en una tabla.
tbpriv.sqc	Cómo visualizar/otorgar/revocar privilegios a nivel de tabla.
tbread.sqc	Cómo leer información de una tabla.
tbreorg.sqc	Cómo reorganizar una tabla.
tbrunstats.sqc	Cómo ejecutar runstats sobre una tabla.
tbsavept.sqc	Cómo utilizar puntos de salvar externos.
tbse1.sqc	Cómo seleccionar de entre: inserción, actualización, supresión.
tbse1create.db2	Cómo crear las tablas para el programa tbse1.
tbse1drop.db2	Cómo eliminar las tablas del programa tbse1.
tbtemp.sqc	Cómo utilizar una tabla temporal declarada.
tbtrig.sqc	Cómo utilizar un desencadenante en una tabla.
tbumqt.sqc	Cómo utilizar tablas de consultas materializadas (tablas de resumen).
tbunion.sqc	Cómo insertar utilizando una vista UNION ALL.
tbxload.sqc	Cómo devolver datos simultáneamente desde una sentencia SELECT y cargarlos en una tabla.
Nivel de tipo de datos - Ejemplos referentes a tipos de datos.	
dtformat.sqc	Cómo utilizar extensiones de formato de datos de carga e importación.
dtlob.sqc	Cómo leer y escribir datos de LOB.

Tabla 11. Archivos de programas de ejemplo de C (continuación)

Nombre del programa de ejemplo	Descripción del programa
dtudt.sqc	Cómo crear, utilizar y eliminar tipos diferenciados definidos por el usuario.
Nivel de función de DB2	
fnuse.sqc	Cómo utilizar funciones de SQL.
Nivel de procedimiento almacenado - Ejemplos que muestran cómo trabajar con procedimientos almacenados.	
spcat	Script del catálogo de procedimientos almacenados para el programa spserver. Este script llama a spdrop.db2 y spcreate.db2.
spcreate.db2	Script de CLP para emitir sentencias CREATE PROCEDURE.
spdrop.db2	Script de CLP para eliminar procedimientos almacenados del catálogo.
spclient.sqc	Programa cliente utilizado para llamar a las rutinas de servidor declaradas en spserver.sqc.
spserver.sqc	Rutinas de procedimiento almacenado creadas y ejecutadas en el servidor.
Nivel de UDF - Ejemplos que muestran cómo trabajar con funciones definidas por el usuario.	
udfcli.sqc	Aplicación cliente que sirve para llamar a la función definida por el usuario contenida en udfsrv.c, udfsrv.C.
udfsrv.c	Función definida por el usuario ScalarUDF invocada por udfcli.sqc
udfemcli.sqc	Aplicación cliente que sirve para llamar a la biblioteca de funciones definidas por el usuario de SQL incorporado, udfemsrv.
udfemsrv.sqc	Biblioteca de funciones definidas por el usuario de SQL incorporado invocada por udfemcli.
Otros	
evm.sqc	Cómo crear y analizar supervisores de sucesos de archivos, conexiones y tablas.
utilrecov.c	Programas de utilidad para los archivos de copia de seguridad, restauración y registro.
utilsnap.c	Programas de utilidad para los programas de ejemplo de supervisión de capturas.

Para encontrar las actualizaciones de ejemplos más recientes, visite la página Web sobre ejemplos de desarrollo de aplicaciones DB2:

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

Conceptos relacionados:

- “Archivos de creación” en la página 110
- “Makefiles” en la página 113
- “Programas de utilidad de comprobación de errores” en la página 116
- “Archivos de ejemplo” en la página 65

Ejemplos de C++

Directorio UNIX: sql11ib/samples/cpp. Directorio Windows: sql11ib\samples\cpp.

Extensiones de archivos UNIX: .c (sin SQL incorporado); .sqc (SQL incorporado)

Extensiones de archivos Windows: .cxx (sin SQL incorporado); .sqx (SQL incorporado)

Tabla 12. Archivos de programas de ejemplo C++

Nombre del programa de ejemplo	Descripción del programa
Ejemplos de aprendizaje - Programas que muestran cómo realizar operaciones básicas con bases de datos.	
tut_mod.sqc	Cómo modificar datos de tablas.
tut_read.sqc	Cómo leer tablas.
tut_use.sqc	Cómo utilizar una base de datos.
Nivel de cliente - Ejemplos referentes al nivel de cliente de DB2.	
cli_info.c	Cómo obtener y definir información a nivel de cliente.
clisnap.c	Cómo capturar una instantánea a nivel de cliente.
Nivel de instancia - Ejemplos referentes al nivel de instancia de DB2.	
inattach.c	Cómo conectar/desconectar respecto a una instancia.
inauth.sqc	Cómo visualizar autorizaciones a nivel de instancia.
ininfo.c	Cómo obtener y definir información a nivel de instancia.
insnap.c	Cómo capturar una instantánea a nivel de instancia.
instart.c	Cómo detener e iniciar la instancia local actual.
Nivel de base de datos - Ejemplos referentes a objetos de bases de datos en DB2.	
dbauth.sqc	Cómo visualizar/otorgar/revocar autorizaciones a nivel de base de datos.
dbcfg.sqc	Cómo configurar parámetros de base de datos y de gestor de bases de datos.
dbconn.sqc	Cómo conectar y desconectar con respecto de una base de datos.
dbcreate.c	Cómo crear y eliminar bases de datos.
dbhistfile.sqc	Cómo leer y actualizar una entrada de archivo de recuperación de base de datos.
dbinfo.c	Cómo obtener y definir información a nivel de base de datos.
dbinline.sqc	Cómo utilizar el lenguaje de procedimientos SQL incorporados.
dbinspec.sqc	Cómo verificar la integridad arquitectural con la API de DB2 db2Inspect.
dblogconn.sqc	Cómo leer archivos de registro de base de datos de forma asíncrona con una conexión de base de datos.
dblognoconn.sqc	Cómo leer archivos de registro de base de datos de forma asíncrona sin ninguna conexión de base de datos.
dbmcon.sqc	Cómo conectar y desconectar con respecto a varias bases de datos.
dbmcon1.h	Archivo de cabecera de dbmcon1.sqc
dbmcon1.sqc	Archivo de soporte de dbmcon.sqc.

Tabla 12. Archivos de programas de ejemplo C++ (continuación)

Nombre del programa de ejemplo	Descripción del programa
dbmcon2.h	Archivo de cabecera de dbmcon2.sqlC
dbmcon2.sqlC	Archivo de soporte de dbmcon.sqlC.
dbmigrat.C	Cómo migrar una base de datos.
dbpkg.sqlC	Cómo trabajar con paquetes.
dbrec.sqlC	Cómo utilizar la API db2GetRecommendations.
dbrecov.sqlC	Cómo recuperar una base de datos.
dbredirect.sqlC	Cómo recuperar una base de datos utilizando una restauración redirigida.
dbrestore.sqlC	Cómo recuperar una base de datos.
dbrollfwd.sqlC	Cómo recuperar una base de datos utilizando una restauración retrotraída.
dbsample.sqlC	Cómo crear la base de datos de ejemplo de forma que incluya tablas y vistas de Host y AS/400.
dbsnap.C	Cómo capturar una instantánea a nivel de base de datos.
dbthrds.sqlC	Cómo utilizar las API de múltiples contextos en UNIX.
dbthrds.sqlC	Cómo utilizar las API de múltiples contextos en Windows.
dbuse.sqlC	Cómo utilizar objetos de base de datos.
Nivel de espacio de tablas - Ejemplos referentes al nivel de espacio de tablas de DB2.	
tscreate.sqlC	Cómo crear y eliminar agrupaciones de almacenamientos intermedios y espacios de tablas.
tsinfo.sqlC	Cómo obtener información a nivel de espacio de tablas.
Nivel de tabla - Ejemplos referentes a objetos de tabla de DB2.	
tbconstr.sqlC	Cómo trabajar con restricciones de tabla.
tbcreate.sqlC	Cómo crear, modificar y eliminar tablas.
tbident.sqlC	Cómo utilizar columnas de identidad.
tbinfo.sqlC	Cómo obtener y definir información a nivel de tabla.
tbintrig.sqlC	Cómo utilizar un desencadenante 'INSTEAD OF' en una vista.
tbmerge.sqlC	Cómo utilizar la sentencia MERGE.
tbmod.sqlC	Cómo modificar información de una tabla.
tbmove.sqlC	Cómo mover datos de tablas.
tbpriv.sqlC	Cómo visualizar/otorgar/revocar privilegios a nivel de tabla.
tbread.sqlC	Cómo leer información de una tabla.
tbreorg.sqlC	Cómo reorganizar una tabla.
tbsavept.sqlC	Cómo utilizar puntos de salvar externos. También muestra cómo cambiar el valor por omisión de una columna.
tbssel.sqlC	Cómo seleccionar de entre: inserción, actualización, supresión.
tbsselcreate.db2	Cómo crear las tablas para el programa tbssel.
tbsseldrop.db2	Cómo eliminar las tablas del programa tbssel.
tbtemp.sqlC	Cómo utilizar una tabla temporal declarada.
tbtrig.sqlC	Cómo utilizar un desencadenante en una tabla.

Tabla 12. Archivos de programas de ejemplo C++ (continuación)

Nombre del programa de ejemplo	Descripción del programa
tbunion.sqC	Cómo insertar utilizando una vista UNION ALL.
tbxload.sqC	Cómo devolver datos simultáneamente desde una sentencia SELECT y cargarlos en una tabla.
Nivel de tipo de datos - Ejemplos referentes a tipos de datos.	
dtformat.sqC	Cómo utilizar extensiones de formato de datos de carga e importación.
dtlob.sqC	Cómo leer y escribir datos de LOB.
dtstruct.sqC	Cómo crear, utilizar y eliminar una jerarquía de tipos estructurados y tablas tipificadas.
dtudt.sqC	Cómo crear, utilizar y eliminar tipos diferenciados definidos por el usuario.
Nivel de función de DB2	
fnuse.sqC	Cómo utilizar funciones de SQL.
Nivel de procedimiento almacenado - Ejemplos que muestran cómo trabajar con procedimientos almacenados.	
spscat	Script del catálogo de procedimientos almacenados para el programa spserver. Este script llama a spdrop.db2 y spcreate.db2.
spcreate.db2	Script de CLP para emitir sentencias CREATE PROCEDURE.
spdrop.db2	Script de CLP para eliminar procedimientos almacenados del catálogo.
spclient.sqC	Programa cliente utilizado para llamar a las rutinas de servidor declaradas en spserver.sqC, spserver.sqC.
spserver.sqC	Rutinas de procedimiento almacenado creadas y ejecutadas en el servidor.
Nivel de UDF - Ejemplos que muestran cómo trabajar con funciones definidas por el usuario.	
udfcli.sqC	Aplicación cliente que sirve para llamar a la función definida por el usuario contenida en udfsrv.c, udfsrv.C.
udfsrv.C	Función definida por el usuario ScalarUDF invocada por udfcli.sqC, udfcli.sqC.
udfemcli.sqC	Aplicación cliente que sirve para llamar a la biblioteca de funciones definidas por el usuario de SQL incorporado, udfemsrv.
udfemsrv.sqC	Biblioteca de funciones definidas por el usuario de SQL incorporado invocada por udfemcli.
Otros	
evm.sqC	Cómo crear y analizar supervisores de sucesos de archivos, conexiones y tablas.
utilrecov.C	Programas de utilidad para los archivos de copia de seguridad, restauración y registro.
utilsnap.C	Programas de utilidad para los programas de ejemplo de supervisión de capturas.

Para encontrar las actualizaciones de ejemplos más recientes, visite la página Web sobre ejemplos de desarrollo de aplicaciones DB2:

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

Conceptos relacionados:

- “Archivos de creación” en la página 110
- “Makefiles” en la página 113
- “Programas de utilidad de comprobación de errores” en la página 116
- “Archivos de ejemplo” en la página 65

Ejemplos de C#

Directorio: `sql11ib\samples\ .NET\cs.`

Tabla 13. Archivos de programas de ejemplo de C# .NET

Nombre del programa de ejemplo	Descripción del programa
Nivel de base de datos - Programas que tratan con objetos de bases de datos de DB2.	
DbAuth.cs	Cómo visualizar/otorgar/revocar autorizaciones a nivel de base de datos.
DbDatAdp.cs	Cómo utilizar DB2DataAdapter.
DbDatMap.cs	Cómo configurar las correlaciones de DataTable y DataColumn.
DbDsetCn.cs	Cómo añadir restricciones existentes a DataSet.
DbEvent.cs	Cómo gestionar eventos de DB2DataAdapter.
DbUse.cs	Cómo utilizar objetos de base de datos.
DbValue.cs	Cómo obtener un solo valor de la base de datos.
Nivel de tabla - Programas que tratan con objetos de tabla de DB2.	
TbConstr.cs	Cómo trabajar con restricciones de tabla.
TbInfo.cs	Cómo obtener y definir información a nivel de tabla.
TbPriv.cs	Cómo visualizar/otorgar/revocar privilegios a nivel de tabla.
TbSel.cs	Cómo seleccionar de entre: inserción, actualización, supresión.
TbTrig.cs	Cómo utilizar un desencadenante en una tabla.
TbUse.cs	Cómo manipular los datos de la tabla y conectarse a una base de datos o desconectarse de la misma.
Nivel de tipo de datos - Programas de ejemplo que tratan con tipos de datos.	
DtLob.cs	Cómo utilizar el tipo de datos LOB.
Procedimientos almacenados - Programas de ejemplo que muestran cómo trabajar con procedimientos almacenados.	
SpCat.db2	Elimina y crea los procedimientos implantados en SpServer.cs.
SpClient.cs	Programa cliente utilizado para llamar a los procedimientos almacenados en SpServer.cs.
SpCreate.db2	Crea los procedimientos externos implantados en SpServer.cs.
SpDrop.db2	Elimina los procedimientos externos creados en SpCreate.db2 para C#.
SpReturn.cs	Aplicación cliente que llama al procedimiento almacenado EMP_DETAILS y obtiene su valor de retorno.
SpServer.cs	Implantación de código C# externo de los procedimientos creados en SpCat.db2.

Tabla 13. Archivos de programas de ejemplo de C# .NET (continuación)

Nombre del programa de ejemplo	Descripción del programa
EmpDetails.db2	Script de CLP que crea un procedimiento almacenado denominado EMP_DETAILS.
Funciones definidas por el usuario - Ejemplos que muestran cómo trabajar con UDF (funciones definidas por el usuario).	
UdfCat.db2	Elimina y crea las UDF externas implantadas en UDFSrv.cs.
UdfCli.cs	Aplicación cliente que llama a las funciones definidas por el usuario en UdfSrv.cs.
UdfCreate.db2	Crea las UDF externas implantadas en UdfSrv.cs.
UdfDrop.db2	Elimina las UDF externas creadas en udfcreate.db2 para C#.
UdfSrv.cs	Funciones escalares definidas por el usuario llamadas por UDFCli.cs.
Transacciones emparejadas débilmente	
empcat.bat	Cataloga el procedimiento almacenado EMP_DETAILS para el programa cliente C#, SpReturn.
LCTrans.cs	Demuestra las transacciones emparejadas débilmente.
regCOM.bat	Registra objetos COM+ para el programa C# LCTrans.
RootCOM.cs	Este archivo se utiliza para crear un conjunto de bibliotecas RootCOM.dll. LCTrans.cs hace referencia a las clases y los métodos que se definen en este archivo.
SubCOM.cs	Este archivo se utiliza para crear un conjunto de bibliotecas SubCOM.dll. LCTrans.cs hace referencia a las clases y los métodos que se definen en este archivo.

Para encontrar las actualizaciones de ejemplos más recientes, visite la página Web sobre ejemplos de desarrollo de aplicaciones DB2:

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

Conceptos relacionados:

- “Archivos de ejemplo” en la página 65

Tareas relacionadas:

- “Creación de aplicaciones C# .NET” en la página 268
- “Creación de rutinas CLR (Common Language Runtime) .NET” en la página 276

Ejemplos de la CLI

Directorio UNIX: sqllib/samples/cli. Directorio Windows: sqllib\samples\cli.

Tabla 14. Archivos de programas de ejemplo de la CLI

Nombre del programa de ejemplo	Descripción del programa
Ejemplos de aprendizaje - Programas que muestran cómo realizar operaciones básicas con bases de datos.	
tut_mod.c	Cómo modificar datos de tablas.

Tabla 14. Archivos de programas de ejemplo de la CLI (continuación)

Nombre del programa de ejemplo	Descripción del programa
tut_read.c	Cómo leer tablas.
tut_use.c	Cómo utilizar una base de datos.
Nivel de imagen de instalación - Ejemplos referentes al nivel de imagen de instalación de DB2 y CLI.	
ilinfo.c	Cómo obtener y definir información a nivel de instalación (tal como la versión del controlador CLI).
Nivel de cliente - Ejemplos referentes al nivel de cliente de DB2.	
cli_info.c	Cómo obtener y definir información a nivel de cliente.
clihandl.c	Cómo asignar y liberar descriptores.
clisqlca.c	Cómo trabajar con datos de la SQLCA.
Nivel de instancia - Ejemplos referentes al nivel de instancia de DB2.	
ininfo.c	Cómo obtener y definir información a nivel de instancia.
Nivel de base de datos - Ejemplos referentes a objetos de bases de datos en DB2.	
dbcongui.c	Cómo conectar con una base de datos utilizando una Interfaz Gráfica de Usuario (GUI).
dbconn.c	Cómo conectar y desconectar con respecto de una base de datos.
dbinfo.c	Cómo obtener y definir información a nivel de base de datos.
dbmcon.c	Cómo conectar y desconectar con respecto a varias bases de datos.
dbmconx.c	Cómo conectar y desconectar con respecto a varias bases de datos mediante SQL incorporado.
dbmconx1.h	Archivo de cabecera de dbmconx1.sqc.
dbmconx1.sqc	Archivo de SQL incorporado del programa dbmconx.
dbmconx2.h	Archivo de cabecera de dbmconx2.sqc.
dbmconx2.sqc	Archivo de SQL incorporado del programa dbmconx.
dbnative.c	Cómo convertir una sentencia que contiene una cláusula de escape ODBC en un formato específico de la fuente de datos.
dbuse.c	Cómo utilizar objetos de base de datos.
dbusemx.sqc	Cómo utilizar objetos de base de datos que contienen SQL incorporado.
dbxamon.c	Cómo mostrar y retrotraer transacciones dudosas.
Nivel de tabla - Ejemplos referentes a objetos de tabla de DB2.	
tbconstr.c	Cómo trabajar con restricciones de tabla.
tbcreate.c	Cómo crear, modificar y eliminar tablas.
tbinfo.c	Cómo obtener y definir información a nivel de tabla.
tbload.c	Cómo insertar datos utilizando el programa de utilidad LOAD de CLI.
tbmod.c	Cómo modificar información de una tabla.
tbread.c	Cómo leer información de una tabla.
Nivel de tipo de datos - Ejemplos referentes a tipos de datos.	
dtinfo.c	Cómo obtener información sobre tipos de datos.
dtlob.c	Cómo leer y escribir datos de LOB.

Tabla 14. Archivos de programas de ejemplo de la CLI (continuación)

Nombre del programa de ejemplo	Descripción del programa
dtudt.c	Cómo crear, utilizar y eliminar tipos diferenciados definidos por el usuario.
Nivel de procedimiento almacenado - Ejemplos que muestran cómo trabajar con procedimientos almacenados.	
spcat	Script del catálogo de procedimientos almacenados para el programa spserver. Este script llama a spdrop.db2 y spcreate.db2.
spcreate.db2	Script de CLP para emitir sentencias CREATE PROCEDURE.
spdrop.db2	Script de CLP para eliminar procedimientos almacenados del catálogo.
spclient.c	Programa cliente utilizado para llamar a las funciones de servidor declaradas en spserver.c.
spserver.c	Funciones de procedimiento almacenado creadas y ejecutadas en el servidor.
spclires.c	Aplicación cliente que muestra la diferencia entre SQLMoreResults y SQLNextResults para varios conjuntos de resultados.
spcall.c	Programa cliente para llamar a un procedimiento almacenado cualquiera.
Nivel de UDF - Ejemplos que muestran cómo trabajar con funciones definidas por el usuario.	
udfcli.c	Aplicación cliente que llama a la función definida por el usuario contenida en udfsrv.c.
udfsrv.c	Función definida por el usuario ScalarUDF invocada por udfcli.c.
Archivos de programas de utilidad comunes	
utilcli.c	Funciones de programa de utilidad utilizadas en programas de ejemplo de CLI.
utilcli.h	Archivo de cabecera para funciones de programa de utilidad utilizadas en programas de ejemplo de CLI.

Para encontrar las actualizaciones de ejemplos más recientes, visite la página Web sobre ejemplos de desarrollo de aplicaciones DB2:

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

Conceptos relacionados:

- “Archivos de creación” en la página 110
- “Makefiles” en la página 113
- “Programas de utilidad de comprobación de errores” en la página 116
- “Archivos de ejemplo” en la página 65

Ejemplos de procesador de línea de mandatos (CLP)

Directorio UNIX: sql11ib/samples/clp. Directorio Windows: sql11ib\samples\clp.

Tabla 15. Scripts de ejemplo de Procesador de línea de mandatos (CLP).

Nombre del archivo de ejemplo	Descripción del archivo
autocfg.db2	Cómo configurar automáticamente los parámetros de configuración de la base de datos y del gestor de bases de datos basándose en las recomendaciones del asistente de Configuración del rendimiento.
const.db2	Crea una tabla con una cláusula CHECK CONSTRAINT.
cte.db2	Muestra el uso de una expresión de tabla común.
flt.db2	Muestra el uso de una consulta recursiva.
healthmon.db2	Cómo utilizar Funciones de tabla para la Instantánea del Supervisor de salida.
join.db2	Muestra el uso de una unión externa de tablas.
onlineoad.db2	Cómo realizar una carga en línea utilizando la opción do ALLOW READ ACCESS.
stock.db2	Muestra el uso de desencadenantes.
testdata.db2	Utiliza funciones internas de DB2, tales como RAND() y TRANSLATE(), para llenar una tabla con datos de prueba generados aleatoriamente.

Para encontrar las actualizaciones de ejemplos más recientes, visite la página Web sobre ejemplos de desarrollo de aplicaciones DB2:

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

Conceptos relacionados:

- “Archivos de ejemplo” en la página 65

Tareas relacionadas:

- “Llamada a procedimientos desde el Procesador de línea de mandatos (CLP)” en la página 144
- “Ejecución de scripts del procesador de línea de mandatos (CLP)” en la página 143

Ejemplos de COBOL

Directorios de UNIX. IBM COBOL: sql11ib/samples/cobol; Micro Focus COBOL: sql11ib/samples/cobol_mf.

Directorios de Windows. IBM COBOL: sql11ib\samples\cobol; Micro Focus COBOL: sql11ib\samples\cobol_mf.

Nota: Los ejemplos de COBOL no están estructurados según el diseño de niveles de DB2 utilizado para los ejemplos de C, CLI, C++, C#, Java, Perl, PHP, Visual Basic ADO y Visual Basic .NET.

Tabla 16. Programas COBOL de ejemplo de la API de DB2 sin SQL incorporado

Programa de ejemplo	API incluida
checkerr.cbl	<ul style="list-style-type: none"> • sqlaintp - Obtener mensaje de error • sqllogstt - Obtener mensaje de SQLSTATE
client.cbl	<ul style="list-style-type: none"> • sqleqryc - Consultar cliente • sqleetc - Definir cliente
d_dbconf.cbl	<ul style="list-style-type: none"> • sqleatin - Conectar • sqledtin - Desconectar • sqlfddb - Obtener valores por omisión de configuración de base de datos
d_dbmcon.cbl	<ul style="list-style-type: none"> • sqleatin - Conectar • sqledtin - Desconectar • sqlfdfs - Obtener valores por omisión de configuración de gestor de bases de datos
db_udcs.cbl	<ul style="list-style-type: none"> • sqleatin - Conectar • sqlecrea - Crear base de datos • sqledrpd - Eliminar base de datos
dbcatt.cbl	<ul style="list-style-type: none"> • sqlecadb - Catalogar base de datos • db2DbDirCloseScan - Cerrar exploración de directorio de base de datos • db2DbDirGetNextEntry - Obtener entrada siguiente del directorio de base de datos • db2DbDirOpenScan - Abrir exploración de directorio de base de datos • sqleuncd - Descatalogar base de datos
dbcmt.cbl	<ul style="list-style-type: none"> • sqledcgd - Cambiar comentario de base de datos • db2DbDirCloseScan - Cerrar exploración de directorio de base de datos • db2DbDirGetNextEntry - Obtener entrada siguiente del directorio de base de datos • db2DbDirOpenScan - Abrir exploración de directorio de base de datos • sqleisig - Instalar gestor de señales
dbconf.cbl	<ul style="list-style-type: none"> • sqleatin - Conectar • sqlecrea - Crear base de datos • sqledrpd - Eliminar base de datos • sqlfrdb - Restaurar configuración de base de datos • sqlfudb - Actualizar configuración de base de datos • sqlfxdb - Obtener configuración de base de datos
dbinst.cbl	<ul style="list-style-type: none"> • sqleatcp - Conectar y cambiar contraseña • sqleatin - Conectar • sqledtin - Desconectar • sqlegins - Obtener instancia

Tabla 16. Programas COBOL de ejemplo de la API de DB2 sin SQL incorporado (continuación)

Programa de ejemplo	API incluida
dbmconf.cbl	<ul style="list-style-type: none"> • sqleatin - Conectar • sqledtin - Desconectar • sqlfrsys - Restaurar configuración de gestor de bases de datos • sqlfusys - Actualizar configuración de gestor de bases de datos • sqlfxsys - Obtener configuración de gestor de bases de datos
dbsnap.cbl	<ul style="list-style-type: none"> • sqleatin - Conectar • sqlmonss - Obtener instantánea
dbstart.cbl	<ul style="list-style-type: none"> • sqlepstart - Iniciar gestor de bases de datos
dbstop.cbl	<ul style="list-style-type: none"> • sqlefrce - Forzar aplicación • sqlepstp - Detener gestor de bases de datos
dcscat.cbl	<ul style="list-style-type: none"> • sqlegdad - Catalogar bases de datos DCS • sqlegdcl - Cerrar exploración de directorio DCS • sqlegdel - Descatalogar base de datos DCS • sqlegdge - Obtener entrada de directorio DCS para base de datos • sqlegdgt - Obtener entradas de directorio DCS • sqlegdsc - Abrir exploración de directorio DCS
ebcdicdb.cbl	<ul style="list-style-type: none"> • sqleatin - Conectar • sqlecrea - Crear base de datos • sqledrpd - Eliminar base de datos
migrate.cbl	<ul style="list-style-type: none"> • sqlemgdb - Migrar base de datos
monreset.cbl	<ul style="list-style-type: none"> • sqleatin - Conectar • sqlmrset - Restaurar monitor
monsz.cbl	<ul style="list-style-type: none"> • sqleatin - Conectar • sqlmonss - Obtener instantánea • sqlmonsz - Calcular tamaño necesario para almacenamiento intermedio de salida de sqlmonss()
nodecat.cbl	<ul style="list-style-type: none"> • sqlectnd - Catalogar nodo • sqlencls - Cerrar exploración de directorio de nodo • sqlengne - Obtener entrada siguiente del directorio de nodo • sqlenops - Abrir exploración de directorio de nodo • sqleuncn - Descatalogar nodo
restart.cbl	<ul style="list-style-type: none"> • sqlerstd - Reiniciar base de datos
setact.cbl	<ul style="list-style-type: none"> • sqlesact - Definir cadena de contabilidad
sws.cbl	<ul style="list-style-type: none"> • sqleatin - Conectar • sqlmon - Obtener/Actualizar conmutadores de monitor

Tabla 17. Programas COBOL de ejemplo de la API de DB2 con SQL incorporado

Programa de ejemplo	API incluida
dbauth.sqb	<ul style="list-style-type: none"> • sqluadau - Obtener autorizaciones
dbstat.sqb	<ul style="list-style-type: none"> • db2Reorg - Reorganizar tabla • db2Runstats - Ejecutar estadísticas
expsamp.sqb	<ul style="list-style-type: none"> • db2Export - Exportar • sqluimpr - Importar
impexp.sqb	<ul style="list-style-type: none"> • db2Export - Exportar • sqluimpr - Importar
loadqry.sqb	<ul style="list-style-type: none"> • db2LoadQuery - Cargar consulta
rebind.sqb	<ul style="list-style-type: none"> • sqlarbnd - Volver a vincular
tabscont.sqb	<ul style="list-style-type: none"> • sqlbctcq - Cerrar consulta del contenedor del espacio de tablas • sqlbftcq - Recuperar consulta del contenedor del espacio de tablas • sqlbotcq - Abrir consulta del contenedor del espacio de tablas • sqlbctcq - Consulta del contenedor del espacio de tablas • sqlefmem - Liberar memoria
tabspace.sqb	<ul style="list-style-type: none"> • sqlbctsq - Cerrar consulta del espacio de tablas • sqlbftpq - Recuperar consulta del espacio de tablas • sqlbgtss - Obtener estadísticas del espacio de tablas • sqlbmtsq - Consulta del espacio de tablas • sqlbotsq - Abrir consulta del espacio de tablas • sqlbstpq - Consulta individual del espacio de tablas • sqlefmem - Liberar memoria
tload.sqb	<ul style="list-style-type: none"> • db2Export - Exportar • sqluload - Cargar • sqluvqdp - Inmovilizar espacios de tablas para tabla
tspace.sqb	<ul style="list-style-type: none"> • sqlbctcq - Cerrar consulta del contenedor del espacio de tablas • sqlbctsq - Cerrar consulta del espacio de tablas • sqlbftcq - Recuperar consulta del contenedor del espacio de tablas • sqlbftpq - Recuperar consulta del espacio de tablas • sqlbgtss - Obtener estadísticas del espacio de tablas • sqlbmtsq - Consulta del espacio de tablas • sqlbotcq - Abrir consulta del contenedor del espacio de tablas • sqlbotsq - Abrir consulta del espacio de tablas • sqlbstpq - Consulta individual del espacio de tablas • sqlbstsc - Definir contenedores del espacio de tablas • sqlbctcq - Consulta del contenedor del espacio de tablas • sqlefmem - Liberar memoria

Tabla 18. Programas COBOL de ejemplo con SQL incorporado sin ninguna API de DB2

Nombre del programa de ejemplo	Descripción del programa
advsql.sqb	Muestra el uso de expresiones complejas de SQL, tales como CASE, CAST, y selecciones completas escalares.
cursor.sqb	Muestra el uso de un cursor que utiliza SQL estático.
delet.sqb	Muestra el uso de SQL estático para suprimir elementos de una base de datos.
dynamic.sqb	Muestra el uso de un cursor que utiliza SQL dinámico.
joinsql.sqb	Muestra el uso de expresiones de unión complejas de SQL.
lobeval.sqb	Muestra el uso de localizadores de LOB y difiere la evaluación de los datos LOB propiamente dichos.
lobfile.sqb	Muestra el uso de los descriptores de archivo de LOB.
lobloc.sqb	Muestra el uso de los localizadores de LOB.
openftch.sqb	Muestra cómo recuperar, actualizar y suprimir filas utilizando SQL estático.
static.sqb	Muestra el uso de SQL estático para recuperar información.
tabsql.sqb	Muestra el uso de expresiones complejas de tabla de SQL.
trigsq1.sqb	Muestra el uso de desencadenantes y restricciones complejos de SQL.
updat.sqb	Muestra el uso de SQL estático para actualizar una base de datos.
varinp.sqb	Muestra el uso de datos de entrada variables para llamadas a sentencias de SQL dinámico incorporadas utilizando marcadores de parámetros.

Para encontrar las actualizaciones de ejemplos más recientes, visite la página Web sobre ejemplos de desarrollo de aplicaciones DB2:

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

Conceptos relacionados:

- “Archivos de creación” en la página 110
- “Makefiles” en la página 113
- “Programas de utilidad de comprobación de errores” en la página 116
- “Archivos de ejemplo” en la página 65

Ejemplos de reconfiguración dinámica

Directorio (AIX y Solaris): `sql1ib/samples/DLPAR`.

Tabla 19. Scripts de ejemplo de reconfiguración dinámica

Nombre del script de ejemplo	Descripción del archivo
ibm_db2_sl1n	Este script de Reconfiguración dinámica (script DR) del shell Korn para AIX facilita el uso de las posibilidades de Particionamiento lógico dinámico (DLPAR) proporcionadas con AIX versión 5.2 en ejecución en sistemas pSeries basados en POWER4, como por ejemplo p690 y p670. Estas posibilidades permiten la adición y eliminación dinámicas de recursos como unidades centrales de proceso y memoria de las particiones lógicas activas sin necesidad de rearranque. Se puede llamar a este script DR durante los sucesos de reconfiguración dinámica para asegurarse que los sucesos se producen sin afectar al funcionamiento de DB2. La configuración de DB2 se modifica dinámicamente para dar cuenta del suceso. Se puede encontrar información adicional en el propio script DR.
IBM,DB2	Este script Perl para Solaris Operating Environment lo utilizan las herramientas de reconfiguración y coordinación dinámicas para proporcionar la interfaz entre DB2 y el Gestor de reconfiguración y coordinación (RCM). Estas herramientas se proporcionan con Solaris 9 o posteriores en ejecución en una máquina de serie 3800 o posteriores. Un uso correcto de este script asegurará que DB2 sigue funcionando aún en el caso de que se eliminen recursos de hardware. Se puede encontrar información adicional en el propio script.

El ejemplo `ibm_db2_sl1n` está en el directorio `sql11ib/samples/DLPAR` de DB2 para AIX, y el ejemplo `IBM,DB2` está en el directorio `sql11ib/samples/DLPAR` de DB2 para Solaris.

Para encontrar las actualizaciones de ejemplos más recientes, visite la página Web sobre ejemplos de desarrollo de aplicaciones DB2:

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

Conceptos relacionados:

- “Archivos de ejemplo” en la página 65

Tareas relacionadas:

- “Configuring parameters dynamically” en el manual *Administration Guide: Performance*

Información relacionada:

- “Software de desarrollo soportado por AIX” en la página 9
- “Software de desarrollo soportado por Solaris” en la página 18

Ejemplos de JDBC

Directorio UNIX: `sql11ib/samples/java/jdbc`.

Directorio Windows: sql1lib\samples\java\jdbc.

Tabla 20. archivos de programas de ejemplo JDBC

Nombre del programa de ejemplo	Descripción del programa
Ejemplos de aprendizaje - Programas que muestran cómo realizar operaciones básicas con bases de datos.	
TutMod.java	Cómo modificar datos de tablas.
TutRead.java	Cómo leer tablas.
Nivel de imagen de instalación - Ejemplos referentes al nivel de imagen de instalación de DB2.	
IlInfo.java	Cómo obtener y definir información a nivel de instalación.
Nivel de base de datos - Ejemplos referentes a objetos de bases de datos en DB2.	
DbAuth.java	Cómo visualizar/otorgar/revocar autorizaciones a nivel de base de datos.
DbConn.java	Cómo conectar y desconectar con respecto de una base de datos.
DbInfo.java	Cómo obtener y definir información a nivel de base de datos.
DbMCon.java	Cómo conectar y desconectar con respecto a varias bases de datos.
DbNative.java	Cómo convertir una sentencia que contiene una cláusula de escape ODBC en un formato específico de la fuente de datos.
DbRsHold.java	Cómo utilizar la posibilidad de contención del cursor del conjunto de resultados en los controladores JDBC Legacy de tipo 2 y JDBC universal. Para compilar este ejemplo, necesita Java Developer Kit 1.4 o superior. Para ejecutar este ejemplo, necesita Java Runtime Environment 1.4 o superior.
DbSeq.java	Cómo crear, modificar y eliminar una secuencia en una base de datos.
DbUse.java	Cómo utilizar objetos de base de datos.
Nivel de tabla - Ejemplos referentes a objetos de tabla de DB2.	
TbConstr.java	Cómo trabajar con restricciones de tabla.
TbCreate.java	Cómo crear, modificar y eliminar tablas.
TbGenCol.java	Cómo utilizar columnas generadas.
TbIdent.java	Cómo utilizar columnas de identidad.
TbInfo.java	Cómo obtener y definir información a nivel de tabla.
TbInTrig.java	Cómo utilizar un desencadenante 'INSTEAD OF' en una vista.
TbMerge.java	Cómo utilizar la sentencia MERGE.
TbMod.java	Cómo modificar información de una tabla.
TbPriv.java	Cómo visualizar/otorgar/revocar privilegios a nivel de tabla.
TbRead.java	Cómo leer información de una tabla.
TbSel.java	Cómo seleccionar de entre: inserción, actualización, supresión.
TbTemp.java	Cómo utilizar tablas temporales declaradas.
TbTrig.java	Cómo utilizar un desencadenante en una tabla.
TbUnion.java	Cómo insertar utilizando una vista UNION ALL.
Nivel de tipo de datos - Ejemplos referentes a tipos de datos.	
DtInfo.java	Cómo obtener información sobre tipos de datos.
DtLob.java	Cómo leer y escribir datos de LOB.

Tabla 20. archivos de programas de ejemplo JDBC (continuación)

Nombre del programa de ejemplo	Descripción del programa
DtUdt.java	Cómo crear, utilizar y eliminar tipos diferenciados definidos por el usuario.
Applets - Programas de ejemplo que muestran cómo trabajar con applets.	
App1t.java	Cómo utilizar applets.
Procedimientos almacenados - Ejemplos que muestran cómo trabajar con procedimientos almacenados.	
spscat	Script del catálogo de procedimientos almacenados para el programa spserver. Este script llama a SpDrop.db2 y SpCreate.db2.
SpCreate.db2	Script de CLP para emitir sentencias CREATE PROCEDURE.
SpDrop.db2	Script de CLP para eliminar procedimientos almacenados del catálogo.
SpClient.java	Programa cliente utilizado para llamar a las funciones de servidor declaradas en SpServer.java.
SpServer.java	Funciones de procedimiento almacenado creadas y ejecutadas en el servidor.
UDF - Programas de ejemplo que muestran cómo trabajar con funciones definidas por el usuario.	
UDFcli.java	Aplicación cliente que llama a la biblioteca de funciones definidas por el usuario, UDFsrv.
UDFsrv.java	Funciones definidas por el usuario llamadas por UDFcli.java.
udfcat	Script de catálogo de funciones definidas por el usuario para el programa UDFsrv. Este script llama a UDFDrop.db2 y UDFCreate.db2.
UDFDrop.db2	Script de CLP para eliminar funciones definidas por el usuario (UDF) del catálogo.
UDFCreate.db2	Script de CLP para emitir sentencias CREATE PROCEDURE.
UDFjcli.java	Aplicación cliente que llama a la biblioteca de funciones definidas por el usuario, UDFjsrv.
UDFjsrv.java	Funciones definidas por el usuario llamadas por UDFjcli.java.
udfjcat	Script de catálogo de funciones definidas por el usuario (UDF) para el programa UDFjsrv. Este script llama a UDFjDrop.db2 y UDFjCreate.db2.
UDFjDrop.db2	Script de CLP para eliminar funciones definidas por el usuario (UDF) del catálogo.
UDFjCreate.db2	Script de CLP para emitir sentencias CREATE PROCEDURE.
UDFsCreate.db2	Cómo catalogar las UDF contenidas en UDFsqlsv.java
UDFsDrop.db2	Cómo descatalogar las UDF contenidas en UDFsqlsv.java
UDFsq1cl.java	Llamar a las UDF contenidas en UDFsqlsv.java
UDFsq1sv.java	Funciones definidas por el usuario que contienen sentencias de SQL y son llamadas por UDFsq1cl.java
Beans Java - Ejemplos que muestran cómo trabajar con clases de beans Java.	
CreateEmployee.java	Cómo crear un registro de empleado.
GeneratePayroll.java	Cómo crear informes de cálculo de nóminas para cada departamento.

Tabla 20. archivos de programas de ejemplo JDBC (continuación)

Nombre del programa de ejemplo	Descripción del programa
Otros	
Util.java	Programas de utilidad para programas de ejemplo de JDBC.

Para encontrar las actualizaciones de ejemplos más recientes, visite la página Web sobre ejemplos de desarrollo de aplicaciones DB2:

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

Conceptos relacionados:

- “Programas Java de ejemplo” en la página 121
- “Archivos de ejemplo” en la página 65

Información relacionada:

- “Ejemplos de SQLJ” en la página 90
- “Ejemplos de Java WebSphere” en la página 92
- “Ejemplos de plugins de Java” en la página 93

Ejemplos de SQLJ

Directorio UNIX: sql1lib/samples/java/sqlj.

Directorio Windows: sql1lib\samples\java\sqlj.

Tabla 21. archivos de programas de ejemplo SQLJ

Nombre del programa de ejemplo	Descripción del programa
Ejemplos de aprendizaje - Programas que muestran cómo realizar operaciones básicas con bases de datos.	
TutMod.sqlj	Cómo modificar datos de tablas.
TutRead.sqlj	Cómo leer tablas.
Nivel de base de datos - Ejemplos referentes a objetos de bases de datos en DB2.	
DbAuth.sqlj	Cómo visualizar/otorgar/revocar autorizaciones a nivel de base de datos.
DbConn.sqlj	Cómo conectar y desconectar con respecto de una base de datos.
DbMCon.sqlj	Cómo conectar y desconectar con respecto a varias bases de datos.
DbUse.sqlj	Cómo utilizar objetos de base de datos.
Nivel de tabla - Ejemplos referentes a objetos de tabla de DB2.	
TbConstr.sqlj	Cómo trabajar con restricciones de tabla.
TbCreate.sqlj	Cómo crear, modificar y eliminar tablas.
TbIdent.sqlj	Cómo utilizar columnas de identidad.
TbInfo.sqlj	Cómo obtener y definir información a nivel de tabla.
TbMod.sqlj	Cómo modificar información de una tabla.
TbPriv.sqlj	Cómo visualizar/otorgar/revocar privilegios a nivel de tabla.

Tabla 21. archivos de programas de ejemplo SQLJ (continuación)

Nombre del programa de ejemplo	Descripción del programa
TbRead.sqlj	Cómo leer información de una tabla.
TbSel.sqlj	Cómo seleccionar de entre: inserción, actualización, supresión.
TbTrig.sqlj	Cómo utilizar un desencadenante en una tabla.
Nivel de tipo de datos - Ejemplos referentes a tipos de datos.	
DtUdt.sqlj	Cómo crear, utilizar y eliminar tipos diferenciados definidos por el usuario.
Nivel de applet - Ejemplos que muestran cómo trabajar con applets.	
Appl t.sqlj	Cómo utilizar applets.
Nivel de procedimiento almacenado - Ejemplos que muestran cómo trabajar con procedimientos almacenados.	
spcat	Script del catálogo de procedimientos almacenados para el programa SpServer. Este script llama a SpDrop.db2 y SpCreate.db2.
SpCreate.db2	Script de CLP para emitir sentencias CREATE PROCEDURE.
SpDrop.db2	Script de CLP para eliminar procedimientos almacenados del catálogo.
SpClient.sqlj	Programa cliente utilizado para llamar a las funciones de servidor declaradas en SpServer.sqlj.
SpServer.sqlj	Funciones de procedimiento almacenado creadas y ejecutadas en el servidor.
SpIterat.sqlj	Archivo de clase de iterador para SpServer.sqlj.
Nivel de UDF - Ejemplos que muestran cómo trabajar con funciones definidas por el usuario.	
UDFcli.sqlj	Aplicación cliente que llama a la biblioteca de funciones definidas por el usuario, UDFsrv.
UDFsrv.java	Funciones definidas por el usuario llamadas por UDFcli.
udfcat	Script de catálogo de funciones definidas por el usuario para el programa UDFsrv. Este script llama a UDFDrop.db2 y UDFCreate.db2.
UDFDrop.db2	Script de CLP para eliminar funciones definidas por el usuario (UDF) del catálogo.
UDFCreate.db2	Script de CLP para emitir sentencias CREATE PROCEDURE.
UDFjcli.sqlj	Aplicación cliente que llama a la biblioteca de funciones definidas por el usuario, UDFjsrv.
UDFjsrv.java	Funciones definidas por el usuario llamadas por UDFjcli.
udfjcat	Script de catálogo de funciones definidas por el usuario (UDF) para el programa UDFjsrv. Este script llama a UDFjDrop.db2 y UDFjCreate.db2.
UDFjDrop.db2	Script de CLP para eliminar funciones definidas por el usuario (UDF) del catálogo.
UDFjCreate.db2	Script de CLP para emitir sentencias CREATE PROCEDURE.
Beans Java - Ejemplos que muestran cómo trabajar con clases de beans Java.	
CreateEmployee.sqlj	Cómo crear un registro de empleado.

Tabla 21. archivos de programas de ejemplo SQLJ (continuación)

Nombre del programa de ejemplo	Descripción del programa
GeneratePayroll.sqlj	Cómo crear informes de cálculo de nóminas para cada departamento.
DataSource - Ejemplos que demuestran las fuentes de datos.	
Batch1Demo.sqlj	Proceso por lotes SQLJ - Cómo funciona el proceso por lotes SQLJ.
Batch2Demo.sqlj	Proceso por lotes SQLJ - Asociación de ExecutionContext con BatchContext.
Batch3Demo.sqlj	Proceso por lotes SQLJ - Cuándo es necesario ejecutar un proceso por lotes implícitamente.
BlobClobDemo.sqlj	Cómo acceder a campos Blob o Clob en las tablas de DB2.
createRegisterDS.java	Crear y registrar fuentes de datos tal como especifican los archivos de propiedades DataSource.
CreateDemoSchema.sqlj	Este programa crea el esquema para los programas de demostración de DataSource.
DbConnDataSource.sqlj	Cómo conectar con una base de datos utilizando DataSource con el controlador JDBC universal para DB2.
DbConnMDataSources.sqlj	Cómo conectar con una base de datos utilizando Multiple DataSources con el controlador JDBC universal para DB2.
ScrollIterDemo.sqlj	Cómo utilizar iteradores con nombre y posicionales desplazables en SQLJ.
Otros	
Util.sqlj	Programas de utilidad para programas de ejemplo de SQLJ.

Para encontrar las actualizaciones de ejemplos más recientes, visite la página Web sobre ejemplos de desarrollo de aplicaciones DB2:

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

Conceptos relacionados:

- “Programas Java de ejemplo” en la página 121
- “Archivos de ejemplo” en la página 65

Información relacionada:

- “Ejemplos de JDBC” en la página 87
- “Ejemplos de Java WebSphere” en la página 92
- “Ejemplos de plugins de Java” en la página 93

Ejemplos de Java WebSphere

Directorio UNIX: `sql1lib/samples/java/WebSphere`.

Directorio Windows: sql1lib\samples\java\WebSphere.

Tabla 22. Archivos de ejemplo de Java WebSphere

Nombre del programa de ejemplo	Descripción del programa
AccessEmployee.ear	Este archivo de Enterprise ARchive (.EAR) consta de cuatro módulos que contienen 32 archivos diferentes .class, .JSP y .HTML. Este archivo EAR, que se despliega con facilidad utilizando IBM WebSphere Application Server, muestra cómo los clientes Java pueden interactuar con Enterprise Java Beans(EJB) para acceder a datos almacenados en DB2.

Para encontrar las actualizaciones de ejemplos más recientes, visite la página Web sobre ejemplos de desarrollo de aplicaciones DB2:

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

Conceptos relacionados:

- “Programas Java de ejemplo” en la página 121
- “Archivos de ejemplo” en la página 65

Información relacionada:

- “Ejemplos de JDBC” en la página 87
- “Ejemplos de SQLJ” en la página 90
- “Ejemplos de plugins de Java” en la página 93

Ejemplos de plugins de Java

Directorio UNIX: sql1lib/samples/java/plugin.

Directorio Windows: sql1lib\samples\java\plugin.

Tabla 23. Archivos de ejemplo de plugins del Centro de control de Java

Nombre del programa de ejemplo	Descripción del programa
Example1.java	Cómo añadir un nuevo botón a la barra de herramientas del Centro de Control.
Example2.java	Cómo añadir nuevas acciones de menú a objetos de base de datos del Centro de Control.
Example3.java	Cómo añadir nuevos objetos de base de datos al árbol del Centro de Control.
Example3Child.java	Cómo añadir objetos de plugin bajo los objetos de Base de datos del árbol del Centro de Control.
Example3Folder.java	Cómo añadir nuevos objetos de base de datos al árbol del Centro de Control.

Conceptos relacionados:

- “Introducing the plug-in architecture for the Control Center” en el manual *Administration Guide: Implementation*

- “Compiling and running the example plugins” en el manual *Administration Guide: Implementation*
- “Programas Java de ejemplo” en la página 121
- “Archivos de ejemplo” en la página 65
- “Writing plugins as Control Center extensions” en el manual *Administration Guide: Implementation*

Tareas relacionadas:

- “Creating a plugin that adds a toolbar button” en el manual *Administration Guide: Implementation*
- “Setting attributes for a plugin tree object” en el manual *Administration Guide: Implementation*

Información relacionada:

- “Ejemplos de JDBC” en la página 87
- “Ejemplos de SQLJ” en la página 90
- “Ejemplos de Java WebSphere” en la página 92

Para encontrar las actualizaciones de ejemplos más recientes, visite la página Web sobre ejemplos de desarrollo de aplicaciones DB2:

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

Ejemplos de salidas del usuario para la gestión de anotaciones cronológicas

Directorio UNIX: `sql1lib/samples/c`. Directorio Windows: `sql1lib\samples\c`.

Nota: Los archivos fuente de la tabla siguiente contienen, al comienzo de ellos, instrucciones para compilar los programas de salida de usuario para la gestión de archivos de anotaciones.

Tabla 24. Archivos de ejemplo de salidas del usuario para la gestión de anotaciones cronológicas.

Nombre del archivo de ejemplo	Descripción del archivo
db2uext2.ctsm	<p>Este programa hace uso de las API de Tivoli Storage Manager (TSM) para archivar y recuperar archivos de anotaciones de base de datos. El programa proporciona un seguimiento de auditoría de llamadas (almacenado en un archivo separado para cada opción) que incluye una indicación de fecha y hora, y los parámetros recibidos. También proporciona un seguimiento de llamadas erróneas junto con una indicación de fecha y hora, e información para la determinación de problemas. Estas opciones se pueden inhabilitar. El nombre del archivo se debe cambiar a <code>db2uext2.c</code> y se debe compilar como un programa en C. Está disponible para UNIX y los sistemas operativos Windows.</p> <p>Nota: Las aplicaciones sobre AIX que hacen uso de la API Cliente de TSM se deben crear mediante el compilador <code>xlC_r</code> o <code>xlC_r</code>, no con <code>xlC</code> ni <code>xlC</code>, aunque las aplicaciones sean de una sola hebra. Esto asegura que la protección de las hebras para las bibliotecas. Si tiene aplicación que está compilada con una biblioteca sin hebras protegidas, puede aplicar el <code>fixtest IC21925E</code> o consultar al proveedor de la aplicación. El <code>fixtest</code> se puede obtener en el servidor <code>ftp</code> anónimo <code>index.storsys.ibm.com</code>.</p>

Tabla 24. Archivos de ejemplo de salidas del usuario para la gestión de anotaciones cronológicas. (continuación)

Nombre del archivo de ejemplo	Descripción del archivo
db2uext2.cdisk	Este programa utiliza el mandato de copia del sistema correspondiente a la plataforma para la cual se proporciona. El programa archiva y recupera archivos de anotaciones de base de datos, y proporciona un seguimiento de auditoría de llamadas (almacenado en un archivo separado para cada opción) que incluye una indicación de fecha y hora, y los parámetros recibidos. También proporciona un seguimiento de llamadas erróneas junto con una indicación de fecha y hora, e información para la determinación de problemas. Estas opciones se pueden inhabilitar. El nombre del archivo se debe cambiar a db2uext2.c y se debe compilar como un programa en C. Está disponible para UNIX y los sistemas operativos Windows.
db2uext2.ctape	Este programa utiliza mandatos de cinta del sistema correspondientes a la plataforma UNIX para la cual se proporciona. El programa archiva y recupera archivos de anotaciones de base de datos. Esta salida de usuario tiene todas las limitaciones propias de los mandatos de cinta del sistema. El programa proporciona un seguimiento de auditoría de llamadas (almacenado en un archivo separado para cada opción) que incluye una indicación de fecha y hora, y los parámetros recibidos. También proporciona un seguimiento de llamadas erróneas junto con una indicación de fecha y hora, e información para la determinación de problemas. Estas opciones se pueden inhabilitar. El nombre del archivo se debe cambiar a db2uext2.c y se debe compilar como un programa en C. Sólo está disponible para las plataformas UNIX.
db2uext2.cxbsa	Este programa hace uso de las API de XBSA para archivar y recuperar archivos de anotaciones de base de datos. El programa proporciona un seguimiento de auditoría de llamadas (almacenado en un archivo separado para cada opción) que incluye una indicación de fecha y hora, y los parámetros recibidos. También proporciona un seguimiento de llamadas erróneas junto con una indicación de fecha y hora, e información para la determinación de problemas. Estas opciones se pueden inhabilitar. El nombre del archivo se debe cambiar a db2uext2.c y se debe compilar como un programa en C. Sólo está disponible en plataformas UNIX.

Para encontrar las actualizaciones de ejemplos más recientes, visite la página Web sobre ejemplos de desarrollo de aplicaciones DB2:

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

Conceptos relacionados:

- “Managing log files through log archiving” en el manual *Data Recovery and High Availability Guide and Reference*
- “Archivos de ejemplo” en la página 65

Información relacionada:

- “Tivoli Storage Manager” en el manual *Data Recovery and High Availability Guide and Reference*

Ejemplos de OLE (Object Linking and Embedding)

Directorios. Visual Basic: `sql1lib\samples\ole\msvb`; Visual C++:
`sql1lib\samples\ole\msvc`.

Tabla 25. Programas de ejemplo de OLE (Object Linking and Embedding)

Nombre del programa de ejemplo	Descripción del programa
sales	Muestra cómo utilizar consultas remontadas en una hoja de cálculo de ventas de Microsoft Excel (implementada en Visual Basic).
names	Consulta un libro de direcciones de Lotus Notes (implementado en Visual Basic).
inbox	Consulta los mensajes entrantes de correo electrónico de Queries Microsoft Exchange utilizando OLE/Messaging (implementado in Visual Basic).
invoice	Es una función definida por el usuario de OLE Automation que envía facturas creadas por Microsoft Word en forma de mensajes anexos de correo electrónico (implementado en Visual Basic).
bcounter	Función definida por el usuario de OLE Automation que muestra la utilización de una memoria de trabajo que hace uso de variables de instancia (implementado en Visual Basic).
ccount	Función definida por el usuario de OLE Automation que se utiliza como contador (implementado en Visual C++).
salsrv	Procedimiento almacenado de OLE Automation que calcula el salario medio de la tabla STAFF perteneciente a la base de datos sample (implementado en Visual Basic).
salcltvc	Ejemplo de la CLI de DB2 para Visual C++ que llama al procedimiento almacenado de Visual Basic, salsrv.
salcltvb	Ejemplo de la CLI de DB2 para Visual Basic que llama al procedimiento almacenado de Visual Basic, salsrv.
salsvado	Procedimiento almacenado de OLE Automation, implementado en Visual Basic y ADO de 32 bits, que muestra el uso de parámetros de salida calculando el salario medio en la tabla recién creada, STAFF2, y muestra el uso de conjuntos de resultados obteniendo salarios a partir de la tabla.
salclado	Programa cliente de Visual Basic que llama al procedimiento almacenado de Visual Basic, salsvado.
testcli	Aplicación cliente de SQL incorporado de OLE Automation que llama al procedimiento almacenado tstsrv (implementado en Visual C++).
tstsrv	Procedimiento almacenado de OLE Automation que muestra cómo pasar diversos tipos entre el cliente y el procedimiento almacenado (implementado en Visual C++).

Para encontrar las actualizaciones de ejemplos más recientes, visite la página Web sobre ejemplos de desarrollo de aplicaciones DB2:

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

Conceptos relacionados:

- “Archivos de ejemplo” en la página 65

Información relacionada:

- “Ejemplos de funciones de tabla de OLE DB (base de datos de enlace e integración de objetos)” en la página 97

Ejemplos de funciones de tabla de OLE DB (base de datos de enlace e integración de objetos)

Directorio: sql11ib\samples\oledb.

Tabla 26. Funciones de tabla de OLE DB (base de datos de enlace e integración de objetos)

Nombre del programa de ejemplo	Descripción del programa
inora.db2	INTERSOLV Oracle8 OLE DB Provider
jet.db2	Proveedor Microsoft.Jet.OLEDB.3.51
jetsrv.db2	Funcionamiento de base de datos federada con Microsoft.Jet.OLEDB.4.0 Provider.
mapi.db2	INTERSOLV Connect OLE DB para MAPI
msdaora.db2	Proveedor Microsoft OLE DB para Oracle
msdasql.db2	Proveedor Microsoft OLE DB para controladores ODBC
msidxs.db2	Proveedor Microsoft OLE DB Index Server
notes.db2	INTERSOLV Connect OLE DB para Notes
sampprov.db2	Proveedor de ejemplo Microsoft OLE DB
sqloledb.db2	Proveedor Microsoft OLE DB para SQL Server

Conceptos relacionados:

- “Archivos de ejemplo” en la página 65

Información relacionada:

- “Ejemplos de OLE (Object Linking and Embedding)” en la página 96

Para encontrar las actualizaciones de ejemplos más recientes, visite la página Web sobre ejemplos de desarrollo de aplicaciones DB2:

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

Ejemplos de Perl

Directorio UNIX: sql11ib/samples/perl.

Directorio Windows: sql11ib\samples\perl.

Tabla 27. Archivos de programas de ejemplo de Perl

Nombre del programa de ejemplo	Descripción del programa
Nivel de base de datos - Programas que tratan con objetos de bases de datos de DB2.	
dbauth.pl	Cómo visualizar/otorgar/revocar autorizaciones a nivel de base de datos.

Tabla 27. Archivos de programas de ejemplo de Perl (continuación)

Nombre del programa de ejemplo	Descripción del programa
dbuse.pl	Cómo utilizar una base de datos.
Nivel de tabla - Programas que tratan con objetos de tabla de DB2.	
tbconstr.pl	Cómo crear, utilizar y eliminar restricciones.
tbinfo.pl	Cómo obtener información sobre una tabla a nivel de la tabla.
tbpriv.pl	Cómo otorgar, visualizar y revocar privilegios sobre una tabla.
tbse1.pl	Cómo seleccionar de entre: inserción, actualización, supresión.
tbse1create.pl	Cómo crear las tablas para el programa tbse1.
tbse1drop.pl	Cómo eliminar las tablas del programa tbse1.
tbtrig.pl	Cómo utilizar un desencadenante en una tabla.
tbuse.pl	Cómo realizar operaciones básicas de base de datos y conectarse a una base de datos o desconectarse de la misma.
Nivel de tipo de datos - Programas de ejemplo que tratan con tipos de datos.	
dtlob.pl	Cómo utilizar el tipo de datos LOB.
Procedimientos almacenados - Programas de ejemplo que muestran cómo trabajar con procedimientos almacenados.	
spclient.pl	Programa cliente que contiene diez funciones para llamar procedimientos almacenados.
Otros archivos de ejemplo	
DB2SampUtil.pm	Define funciones comunes como la comprobación de argumentos de la línea de mandatos. También define funciones para preparar y ejecutar una sentencia SQL, así como para retrotraerla si se produce un error.

Para encontrar las actualizaciones de ejemplos más recientes, visite la página Web sobre ejemplos de desarrollo de aplicaciones DB2:

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

Conceptos relacionados:

- “Consideraciones sobre la programación en Perl” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente*
- “Archivos de ejemplo” en la página 65

Tareas relacionadas:

- “Creación de aplicaciones Perl” en la página 153

Ejemplos de PHP

Directorio UNIX: `sql1lib/samples/php`.

Directorio Windows: sqllib\samples\php.

Tabla 28. Archivos de programas de ejemplo de PHP

Nombre del programa de ejemplo	Descripción del programa
Nivel de base de datos - Programas que tratan con objetos de bases de datos de DB2.	
dbauth.php	Cómo visualizar/otorgar/revocar autorizaciones a nivel de base de datos.
dbuse.php	Cómo utilizar una base de datos.
Nivel de tabla - Programas que tratan con objetos de tabla de DB2.	
tbconstr.php	Cómo crear, utilizar y eliminar restricciones.
tbinfo.php	Cómo obtener información sobre una tabla a nivel de la tabla.
tbpriv.php	Cómo otorgar, visualizar y revocar privilegios sobre una tabla.
tbset.php	Cómo seleccionar de entre: inserción, actualización, supresión.
tbsetcreate.db2	Cómo crear las tablas para el programa tbset.
tbsetdrop.db2	Cómo eliminar las tablas del programa tbset.
tbtrig.php	Cómo utilizar un desencadenante en una tabla.
tbuse.php	Cómo realizar operaciones básicas de base de datos y conectarse a una base de datos o desconectarse de la misma.
Nivel de tipo de datos - Programas de ejemplo que tratan con tipos de datos.	
dtlob.php	Cómo utilizar el tipo de datos LOB.
Funciones definidas por el usuario - Ejemplos que muestran cómo trabajar con funciones definidas por el usuario.	
udfcli.php	Programa cliente que llama a varios tipos de funciones definidas por el usuario.
Otros archivos de ejemplo	
util_funcs.php	Define funciones comunes como la comprobación de argumentos de la línea de mandatos. También define funciones para preparar y ejecutar una sentencia SQL, así como para retrotraerla si se produce un error.

Para encontrar las actualizaciones de ejemplos más recientes, visite la página Web sobre ejemplos de desarrollo de aplicaciones DB2:

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

Conceptos relacionados:

- “Archivos de ejemplo” en la página 65

Tareas relacionadas:

- “Creación de aplicaciones PHP” en la página 155

Ejemplos de REXX

Directorio AIX: `sqllib/samples/rexx`. Directorio Windows: `sqllib\samples\rexx`.

Tabla 29. Archivos de programas de ejemplo de REXX.

Nombre del archivo de ejemplo	Descripción del archivo
<code>blobfile.cmd</code>	Muestra el manejo de los BLOB (Binary Large Object).
<code>chgisl.cmd</code>	Muestra el uso de la API: CHANGE ISOLATION LEVEL.
<code>client.cmd</code>	Muestra el uso de las API: SET CLIENT y QUERY CLIENT.
<code>d_dbconf.cmd</code>	Muestra el uso de la API: GET DATABASE CONFIGURATION DEFAULTS
<code>d_dbmcon.cmd</code>	Muestra el uso de la API: GET DATABASE MANAGER CONFIGURATION DEFAULTS
<code>db_udcs.cmd</code>	Muestra el uso de las API CREATE DATABASE y DROP DATABASE para simular el comportamiento de una secuencia de clasificación de DB2 para MVS/ESA CCSID 500 (EBCDIC International).
<code>dbauth.cmd</code>	Muestra el uso de la API: GET AUTHORIZATIONS
<code>dbcatt.cmd</code>	Muestra el uso de las API siguientes: CATALOG DATABASE CLOSE DATABASE DIRECTORY SCAN GET NEXT DATABASE DIRECTORY ENTRY OPEN DATABASE DIRECTORY SCAN UNCATALOG DATABASE
<code>dbcmt.cmd</code>	Muestra el uso de las API siguientes: CHANGE DATABASE COMMENT GET ERROR MESSAGE INSTALL SIGNAL HANDLER
<code>dbconf.cmd</code>	Muestra el uso de las API siguientes: CREATE DATABASE DROP DATABASE GET DATABASE CONFIGURATION RESET DATABASE CONFIGURATION UPDATE DATABASE CONFIGURATION
<code>dbinst.cmd</code>	Muestra el uso de las API siguientes: ATTACH TO INSTANCE DETACH FROM INSTANCE GET INSTANCE
<code>dbmconf.cmd</code>	Muestra el uso de las API siguientes: GET DATABASE MANAGER CONFIGURATION RESET DATABASE MANAGER CONFIGURATION UPDATE DATABASE MANAGER CONFIGURATION
<code>dbstart.cmd</code>	Muestra el uso de la API: START DATABASE MANAGER
<code>dbstat.cmd</code>	Muestra el uso de las API siguientes: REORGANIZE TABLE RUN STATISTICS
<code>dbstop.cmd</code>	Muestra el uso de las API siguientes: FORCE USERS STOP DATABASE MANAGER

Tabla 29. Archivos de programas de ejemplo de REXX. (continuación)

Nombre del archivo de ejemplo	Descripción del archivo
dcscat.cmd	Muestra el uso de las API siguientes: ADD DCS DIRECTORY ENTRY CLOSE DCS DIRECTORY SCAN GET DCS DIRECTORY ENTRY FOR DATABASE GET DCS DIRECTORY ENTRIES OPEN DCS DIRECTORY SCAN UNCATALOG DCS DIRECTORY ENTRY
dynamic.cmd	Muestra el uso de un cursor que utiliza SQL dinámico
ebcdicdb.cmd	Muestra el uso de las API CREATE DATABASE y DROP DATABASE para simular el comportamiento de una secuencia de clasificación de DB2 para MVS/ESA CCSID 037 (EBCDIC - inglés americano)
impexp.cmd	Muestra el uso de las API: EXPORT e IMPORT.
lobeval.cmd	Muestra cómo diferir la evaluación de un LOB dentro de una base de datos
lobfile.cmd	Muestra el uso de los descriptores de archivo de LOB
lobloc.cmd	Muestra el uso de los localizadores de LOB
lobval.cmd	Muestra el uso de los LOB
migrate.cmd	Muestra el uso de la API: MIGRATE DATABASE
nodecat.cmd	Muestra el uso de las API siguientes: CATALOG NODE CLOSE NODE DIRECTORY SCAN GET NEXT NODE DIRECTORY ENTRY OPEN NODE DIRECTORY SCAN UNCATALOG NODE
quitab.cmd	Muestra el uso de la API: QUIESCE TABLESPACES FOR TABLE
rechist.cmd	Muestra el uso de las API siguientes: CLOSE RECOVERY HISTORY FILE SCAN GET NEXT RECOVERY HISTORY FILE ENTRY OPEN RECOVER HISTORY FILE SCAN PRUNE RECOVERY HISTORY FILE ENTRY UPDATE RECOVERY HISTORY FILE ENTRY
restart.cmd	Muestra el uso de la API: RESTART DATABASE
sqlcsrx.cmd	Es un ejemplo de secuencia de clasificación
updat.cmd	Utiliza SQL dinámico para actualizar una base de datos

Para encontrar las actualizaciones de ejemplos más recientes, visite la página Web sobre ejemplos de desarrollo de aplicaciones DB2:

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

Conceptos relacionados:

- “Archivos de ejemplo” en la página 65

Tareas relacionadas:

- “Creación de aplicaciones REXX en AIX” en la página 209
- “Creación de aplicaciones Object REXX en Windows” en la página 309

Ejemplos de plugins de seguridad

Directorio UNIX: `sqllib/samples/security/plugins`.

Directorio Windows: `sqllib\samples\security\plugins`.

Tabla 30. Archivos de programas de ejemplo de plugins de seguridad

Nombre del programa de ejemplo	Descripción del programa
<code>combined.c</code>	Ejemplo de autenticación combinada de IDusuario/contraseña y búsqueda de grupo.
<code>group_file.c</code>	Ejemplo de plugin de gestión de grupo simple basada en el archivo.
<code>gssapi_simple.c</code>	Ejemplo de plugin de autenticación GSS-API básica (tanto de cliente como de servidor).
<code>IBMkrb5.c</code>	Código fuente para el plugin de seguridad Kerberos suministrado por IBM para UNIX.

Para encontrar las actualizaciones de ejemplos más recientes, visite la página Web sobre ejemplos de desarrollo de aplicaciones DB2:

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

Conceptos relacionados:

- “Archivos de ejemplo” en la página 65
- “Conectores de seguridad” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente*

Ejemplos de procedimientos de SQL

Directorio UNIX: `sqllib/samples/sqlproc`.

Directorio Windows: `sqllib\samples\sqlproc`.

Tabla 31. Archivos de programas de ejemplo de procedimientos de SQL

Nombre del programa de ejemplo	Descripción del programa
<code>basecase.db2</code>	El procedimiento <code>UPDATE_SALARY</code> aumenta el salario del empleado identificado por el parámetro de entrada "empno" en la tabla "staff" de la base de datos "sample". El procedimiento determina el aumento de salario de acuerdo con una sentencia <code>CASE</code> que hace uso del parámetro de entrada "rating".
<code>basecase.sqc</code>	Llama al procedimiento <code>UPDATE_SALARY</code> .
<code>baseif.db2</code>	El procedimiento <code>UPDATE_SALARY_IF</code> aumenta el salario del empleado identificado por el parámetro de entrada "empno" en la tabla "staff" de la base de datos "sample". El procedimiento determina el aumento de salario de acuerdo con una sentencia <code>IF</code> que hace uso del parámetro de entrada "rating".
<code>baseif.sqc</code>	Llama al procedimiento <code>UPDATE_SALARY_IF</code> .

Tabla 31. Archivos de programas de ejemplo de procedimientos de SQL (continuación)

Nombre del programa de ejemplo	Descripción del programa
dynamic.db2	El procedimiento CREATE_DEPT_TABLE utiliza un DDL dinámico para crear una nueva tabla. El nombre de la tabla está basado en el valor del parámetro de entrada del procedimiento.
dynamic.sqc	Llama al procedimiento CREATE_DEPT_TABLE.
iterate.db2	El procedimiento ITERATOR utiliza un bucle FETCH para obtener datos de la tabla "department". Si el valor de la columna "deptno" no es 'D11', se insertan datos modificados en la tabla "department". Si el valor de la columna "deptno" es 'D11', una sentencia ITERATE vuelve a pasar el flujo de control al comienzo de la sentencia LOOP.
iterate.sqc	Llama al procedimiento ITERATOR.
leave.db2	El procedimiento LEAVE_LOOP cuenta el número de operaciones de búsqueda (FETCH) realizadas en una sentencia LOOP necesarias para que el gestor de condiciones "not_found" llame a una sentencia LEAVE. La sentencia LEAVE hace que el flujo de control abandone el bucle y finalice el procedimiento almacenado.
leave.sqc	Llama al procedimiento LEAVE_LOOP.
loop.db2	El procedimiento LOOP_UNTIL_SPACE cuenta el número de operaciones FETCH realizadas en una sentencia LOOP hasta que el cursor recupera una fila con un valor en blanco (' ') para la columna "midinit". La sentencia LOOP hace que el flujo de control abandone el bucle y finalice el procedimiento almacenado.
loop.sqc	Llama al procedimiento LOOP_UNTIL_SPACE.
nestcase.db2	El procedimiento BUMP_SALARY utiliza sentencias CASE anidadas para aumentar los salarios de los empleados de un departamento identificado por el parámetro de entrada "dept" de la tabla "staff", perteneciente a la base de datos "sample".
nestcase.sqc	Llama al procedimiento BUMP_SALARY.
nestif.db2	El procedimiento BUMP_SALARY-IF utiliza sentencias IF anidadas para aumentar los salarios de los empleados de un departamento identificado por el parámetro de entrada "dept" de la tabla "staff", perteneciente a la base de datos "sample".
nestif.sqc	Llama al procedimiento BUMP_SALARY_IF.
nestedsp.db2	Los procedimientos OUT_AVERAGE, OUT_MEDIAN y MAX_SALARY obtienen los valores promedio, medio y máximo de la tabla "staff" de la base de datos "sample".
nestedspdrop.db2	Elimina los procedimientos OUT_AVERAGE, OUT_MEDIAN y MAX_SALARY de SQL que se crean con el script nestedsp.db2.
NestedSP.java	Llama al procedimiento OUT_AVERAGE.
repeat.db2	El procedimiento REPEAT_STMT cuenta el número de operaciones FETCH realizadas en una sentencia REPEAT hasta que el cursor no puede recuperar más filas. El gestor de condiciones hace que el flujo de control abandone el bucle de repetición y finalice el procedimiento almacenado.
repeat.sqc	Llama al procedimiento REPEAT_STMT.
rsultset.c	Llama al procedimiento MEDIAN_RESULT_SET, visualiza el salario medio y luego muestra el conjunto de resultados generado por el procedimiento SQL. Este programa cliente se escribe utilizando la API de CLI, la cual puede aceptar conjuntos de resultados.

Tabla 31. Archivos de programas de ejemplo de procedimientos de SQL (continuación)

Nombre del programa de ejemplo	Descripción del programa
rsultset.db2	El procedimiento MEDIAN_RESULT_SET obtiene el salario medio de los empleados de un departamento identificado por el parámetro de entrada "dept" en la tabla "staff" de la base de datos "sample". El valor medio se asigna al parámetro de salida "salary" y se devuelve a la aplicación cliente "rsultset". Luego, el procedimiento abre un cursor WITH RETURN para devolver el conjunto de los empleados cuyo salario es mayor que la media. El procedimiento devuelve el resultado a la aplicación cliente.
spserver.db2	Los procedimientos SQL de este script de CLP muestran cómo realizar operaciones básicas de manejo de errores, cómo llamar a procedimientos almacenados y devolver los resultados a la aplicación cliente o aplicación solicitante. El usuario puede invocar los procedimientos utilizando la aplicación "spcall", contenida en el directorio "samples" de CLI. Puede también utilizar la aplicación "spclient", contenida en los directorios "samples" de C y CLP, para invocar los procedimientos que no devuelven conjuntos de resultados.
tbfn.db2	Crea las tablas y funciones de tabla utilizadas en el ejemplo de tbfuse. Después de que se ejecuta el script tbfuse, se retrotraen todos los cambios y se eliminan las tablas y funciones creadas en este archivo.
tbfuse.db2	Muestra el uso de las funciones de tabla creadas en el ejemplo de tbfuse. Al final de este script se retrotraen las sentencias y se eliminan las tablas y funciones creadas en tbfn.db2.
tbssel.sqc	Cómo seleccionar de entre: inserción, actualización, supresión. En este ejemplo se llama a un procedimiento de SQL, BUY_COMPANY, creado a partir de tbsselcreate.db2. BUY_COMPANY contiene ejemplos de uso de una cláusula SELECT de una sentencia de cambio de datos.
tbsselcreate.db2	Cómo crear las tablas y el procedimiento utilizados en el programa tbssel.
tbsseldrop.db2	Cómo eliminar las tablas y el procedimiento utilizados en el programa tbssel.
whiles.db2	El procedimiento DEPT_MEDIAN obtiene el salario medio de los empleados de un departamento identificado por el parámetro de entrada "dept" en la tabla "staff" de la base de datos "sample". El valor medio se asigna al parámetro de salida "salary" y se devuelve a la aplicación cliente "whiles". Luego, la aplicación cliente "whiles" muestra el salario medio.
whiles.sqc	Llama al procedimiento DEPT_MEDIAN.

Para encontrar las actualizaciones de ejemplos más recientes, visite la página Web sobre ejemplos de desarrollo de aplicaciones DB2:

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

Conceptos relacionados:

- "Archivos de ejemplo" en la página 65

Tareas relacionadas:

- "Creación de procedimientos SQL" en la página 147

Ejemplos de Visual Basic

Directorios. Objetos de datos ActiveX: sql1lib\samples\VB\ADO; Microsoft Transaction Server: sql1lib\samples\VB\MTS; Objetos de datos remotos: sql1lib\samples\VB\RDO.

Tabla 32. Archivos de programas de ejemplo de objetos de datos ActiveX de Visual Basic

Nombre del programa de ejemplo	Descripción del programa
Nivel de cliente	
cliExeSQL.bas	Cómo ejecutar sentencias SQL.
cli_Info.bas	Cómo obtener/definir información a nivel de cliente.
Nivel de base de datos	
dbConn.bas	Cómo conectar/desconectar respecto de una base de datos.
dbInfo.bas	Cómo obtener y definir información a nivel de base de datos.
dbCommit.bas	Cómo controlar dinámicamente la confirmación automática a nivel de base de datos.
Nivel de tipo de datos	
dtHier.bas	Cómo recuperar datos jerárquicos.
dtLob.bas	Cómo obtener datos LOB.
Procedimientos almacenados	
spCall.bas	Cómo invocar procedimientos almacenados.
Funciones definidas por el usuario	
udfUse.bas	Cómo crear y utilizar tipos definidos por el usuario y funciones definidas por el usuario.

Tabla 33. Archivos de programas de ejemplo de Microsoft Transaction Server de Visual Basic

Nombre del programa de ejemplo	Descripción del programa
db2com.vbp	Este proyecto de Visual Basic muestra cómo actualizar una base de datos utilizando el servidor Microsoft Transaction Server. Crea una DLL de servidor que es utilizada por el programa cliente, db2mts.vbp, y tiene cuatro módulos de clase: <ul style="list-style-type: none"> • UpdateNumberColumn.cls • UpdateRow.cls • UpdateStringColumn.cls • VerifyUpdate.cls Para este programa se crea la tabla temporal DB2MTS en la base de datos sample.
db2mts.vbp	Este proyecto de Visual Basic es para un programa cliente que hace uso del servidor Microsoft Transaction Server para llamar a la DLL de servidor creada a partir de db2com.vbp.
LCTransTest.vbp	Este proyecto Visual Basic muestra una transacción emparejada débilmente en una base de datos DB2. Crea una DLL de servidor que el programa cliente utiliza, main.vbp, y tiene un módulo de clase, TestClass.cls. En la base de datos de ejemplo se crea una tabla temporal, LCTEST.

Tabla 33. Archivos de programas de ejemplo de Microsoft Transaction Server de Visual Basic (continuación)

Nombre del programa de ejemplo	Descripción del programa
main.vbp	Éste es un proyecto Visual Basic para un programa cliente que hace uso de Microsoft Transaction Server para llamar a la DLL de servidor creada a partir de LCTransTest.vbp.

Tabla 34. Archivos de programas de ejemplo de objetos de datos remotos de Visual Basic

Nombre del programa de ejemplo	Descripción del programa
Bank.vbp	Este programa RDO sirve para crear y mantener datos de sucursales bancarias, y permite realizar transacciones sobre cuentas de clientes. El programa puede utilizar cualquier base de datos especificada por el usuario, pues contiene la DDL para crear las tablas necesarias para que la aplicación almacene datos.

Para encontrar las actualizaciones de ejemplos más recientes, visite la página Web sobre ejemplos de desarrollo de aplicaciones DB2:

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

Conceptos relacionados:

- “Archivos de ejemplo” en la página 65

Tareas relacionadas:

- “Creación de aplicaciones ADO con Visual Basic” en la página 260
- “Creación de aplicaciones RDO con Visual Basic” en la página 266
- “Creación de transacciones emparejadas débilmente con Visual Basic” en la página 262
- “Resolución de problemas de un proyecto de transacción emparejada débilmente de Visual Basic” en la página 264

Información relacionada:

- “Ejemplos de Windows Management Instrumentation” en la página 109

Para encontrar las actualizaciones de ejemplos más recientes, visite la página Web sobre ejemplos de desarrollo de aplicaciones DB2:

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

Ejemplos de Visual Basic .NET

Directorio: sql11ib\samples\.NET\vb.

Tabla 35. Archivos de programas de ejemplo de Visual Basic .NET

Nombre del programa de ejemplo	Descripción del programa
Nivel de base de datos - Ejemplos referentes a objetos de bases de datos en DB2.	
DbAuth.vb	Cómo visualizar/otorgar/revocar autorizaciones a nivel de base de datos.
DbDatAdp.vb	Cómo utilizar DB2DataAdapter.
DbDatMap.vb	Cómo configurar las correlaciones de DataTable y DataColumn.
DbDsetCn.vb	Cómo añadir restricciones existentes a DataSet.
DbEvent.vb	Cómo gestionar eventos de DB2DataAdapter.
DbUse.vb	Cómo utilizar objetos de base de datos.
DbValue.vb	Cómo obtener un solo valor de la base de datos.
Nivel de tabla - Ejemplos referentes a objetos de tabla de DB2.	
TbConstr.vb	Cómo trabajar con restricciones de tabla.
TbInfo.vb	Cómo obtener y definir información a nivel de tabla.
TbPriv.vb	Cómo visualizar/otorgar/revocar privilegios a nivel de tabla.
TbSel.vb	Cómo seleccionar de entre: inserción, actualización, supresión.
TbTrig.vb	Cómo utilizar un desencadenante en una tabla.
TbUse.vb	Cómo manipular los datos de la tabla y conectarse a una base de datos o desconectarse de la misma.
Nivel de tipo de datos - Ejemplos referentes a tipos de datos.	
DtLob.vb	Cómo utilizar el tipo de datos LOB.
Procedimientos almacenados - Ejemplos que muestran cómo trabajar con procedimientos almacenados.	
SpCat.db2	Elimina y crea los procedimientos implantados en SpServer.vb.
SpClient.vb	Programa cliente utilizado para llamar a los procedimientos almacenados en SpServer.vb.
SpCreate.db2	Crea los procedimientos externos implantados en SpServer.vb.
SpDrop.db2	Elimina los procedimientos externos creados en SpCreate.db2.
SpReturn.vb	Aplicación cliente que llama al procedimiento almacenado EMP_DETAILS y obtiene su valor de retorno.
SpServer.vb	Implantación de código Visual Basic .NET externo de los procedimientos creados en SpCat.db2.
EmpDetails.db2	Script de CLP que crea un procedimiento almacenado denominado EMP_DETAILS.
Funciones definidas por el usuario - Ejemplos que muestran cómo trabajar con UDF (funciones definidas por el usuario).	
UdfCat.db2	Elimina y crea las UDF externas implantadas en UdfSrv.vb.
UdfCli.vb	Aplicación cliente que llama a las funciones definidas por el usuario en UdfSrv.vb.
UdfCreate.db2	Crea las UDF externas implantadas en UdfSrv.vb.

Tabla 35. Archivos de programas de ejemplo de Visual Basic .NET (continuación)

Nombre del programa de ejemplo	Descripción del programa
UdfDrop.db2	Elimina las UDF externas creadas en udfcreate.db2.
UdfSrv.vb	Funciones escalares definidas por el usuario llamadas por UDFCli.
Transacciones emparejadas débilmente	
empcat.bat	Cataloga el procedimiento almacenado EMP_DETAILS para el programa cliente, SpReturn.
LCTrans.vb	Demuestra las transacciones emparejadas débilmente.
regCOM.bat	Registra objetos COM+ para el programa LCTrans.
RootCOM.vb	Este archivo se utiliza para crear un conjunto de bibliotecas RootCOM.dll. LCTrans.vb hace referencia a las clases y los métodos que se definen en este archivo.
SubCOM.vb	Este archivo se utiliza para crear un conjunto de bibliotecas SubCOM.dll. LCTrans.vb hace referencia a las clases y los métodos que se definen en este archivo.

Conceptos relacionados:

- “Archivos de ejemplo” en la página 65

Tareas relacionadas:

- “Creación de aplicaciones Visual Basic .NET” en la página 272
- “Creación de rutinas CLR (Common Language Runtime) .NET” en la página 276

Para encontrar las actualizaciones de ejemplos más recientes, visite la página Web sobre ejemplos de desarrollo de aplicaciones DB2:

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

Ejemplos de Visual C++

Directorio: sql11ib\samples\VC\ADO.

Tabla 36. archivos de programas de ejemplo Visual C++

Nombre del programa de ejemplo	Descripción del programa
BLOBAccess.dsw	Este programa de ejemplo muestra cómo acceder a ADO/Blob utilizando Microsoft Visual C++. Es similar al programa de ejemplo de Visual Basic, Blob.vbp. El programa de ejemplo de BLOB tiene dos funciones principales: <ol style="list-style-type: none"> 1. Leer un BLOB en la base de datos "sample" y visualizarlo en la pantalla. 2. Leer un BLOB en un archivo e insertarlo en la base de datos. (Importar)
VarChar.dsp	Este programa Visual C++ utiliza ADO para acceder a datos VarChar como campos de texto. Proporciona una interfaz gráfica de usuario para permitir al usuario visualizar y actualizar datos en la tabla ORG de la base de datos sample.

Conceptos relacionados:

- “Automatización del enlace e integración de objetos (OLE) con Visual C++” en la página 282
- “Archivos de ejemplo” en la página 65

Tareas relacionadas:

- “Creación de aplicaciones ADO con Visual C++” en la página 281

Para encontrar las actualizaciones de ejemplos más recientes, visite la página Web sobre ejemplos de desarrollo de aplicaciones DB2:

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

Ejemplos de Windows Management Instrumentation

Directorio: sql1lib\samples\wmi.

Tabla 37. Archivos de programas de ejemplo de Windows Management Instrumentation.

Nombre del archivo de ejemplo	Descripción del archivo
backupdb.vbs	Cómo copiar una base de datos.
createdb.vbs	Cómo crear y eliminar una base de datos.
listsvr.vbs	Cómo listar instancias de servidor e iniciar/detener una instancia de DB2.
perfmon.mof	Archivo MOF para perfmon.vbs.
perfmon.vbs	Cómo obtener un contador de operaciones de DB2. Nota: primero debe ejecutar “mofcomp perfmon.mof”.
regvar.mof	Archivo MOF para regvar.vbs.
regvar.vbs	Cómo obtener una variable del registro de DB2. Nota: primero debe ejecutar “mofcomp regvar.mof”.
restoredb.vbs	Cómo restaurar una base de datos.
rollfwddb.vbs	Cómo actualizar una base de datos por aplicación de cambios.
updatedbcfg.vbs	Cómo obtener y actualizar la configuración de la base de datos.
updatedbmcfg.vbs	Cómo obtener y actualizar la configuración del gestor de bases de datos.

Conceptos relacionados:

- “Archivos de ejemplo” en la página 65
- “Windows Management Instrumentation (WMI)” en la página 259

Información relacionada:

- “Ejemplos de Visual Basic” en la página 105

Para encontrar las actualizaciones de ejemplos más recientes, visite la página Web sobre ejemplos de desarrollo de aplicaciones DB2:

<http://www.ibm.com/software/data/db2/udb/ad/v8/samples.html>

Archivos de creación, makefiles y programas de utilidad de comprobación de errores

Archivos de creación

Los archivos utilizados para mostrar la creación de programas de ejemplo se denominan archivos de script en UNIX® y archivos de proceso por lotes en Windows®. Nosotros los llamaremos genéricamente archivos de creación. Estos archivos contienen los mandatos de compilación y enlace recomendados para los compiladores de las plataformas soportadas.

En el mismo directorio que los programas de ejemplo correspondientes a cada lenguaje, DB2® proporciona archivos de creación para cada lenguaje en las plataformas soportadas donde están disponibles los tipos de programas que dichos archivos generan. La tabla siguiente muestra los diferentes tipos de archivos de creación para crear diversos tipos de programas. A menos que se indique otra cosa, estos archivos de creación están pensados para los lenguajes soportados en todas las plataformas soportadas. Los archivos de creación tienen la extensión `.bat` (batch) en Windows, la cual no se muestra en la tabla. No existe extensión de archivo para las plataformas UNIX.

Tabla 38. Archivos de creación de DB2

Archivo de creación	Tipos de programas creados
<code>bldapp</code>	Programas de aplicación
<code>bldrtn</code>	Rutinas (procedimientos almacenados y funciones definidas por el usuario)
<code>bldsqlj</code>	Aplicaciones SQLJ de Java™
<code>bldsqljs</code>	Rutinas SQLJ de Java (procedimientos almacenados y funciones definidas por el usuario)
<code>bldmc</code>	Aplicaciones C/C++ de conexión múltiple
<code>bldmt</code>	Aplicaciones C/C++ multihebra
<code>bldcli</code>	Aplicaciones cliente de CLI para procedimientos SQL contenidos en el subdirectorio <code>sqlproc</code> de <code>samples</code> .

Nota: El archivo `bldcli` es el mismo que el archivo `bldapp` del directorio `samples/cli`. Se le ha dado un nombre diferente porque el archivo de C incluido, `bldapp`, también está dentro del directorio `samples/sqlproc`.

La tabla siguiente lista los archivos de creación correspondientes a cada plataforma y lenguaje de programación, y los directorios donde están situados. En la documentación en línea, los nombres de los archivos de creación están enlazados dinámicamente con los archivos fuente de HTML. El usuario puede también acceder a los archivos de texto contenidos en los directorios `samples` correspondientes.

Tabla 39. Archivos de creación por lenguaje y plataforma

Plataforma —> Idioma	AIX®	HP-UX	Linux	Solaris	Windows
C samples/c	bldapp bldrtn bldmt bldmc	bldapp bldrtn bldmt bldmc	bldapp bldrtn bldmt bldmc	bldapp bldrtn bldmt bldmc	bldapp.bat bldrtn.bat bldmt.bat bldmc.bat
C++ samples/cpp	bldapp bldrtn bldmt bldmc	bldapp bldrtn bldmt bldmc	bldapp bldrtn bldmt bldmc	bldapp bldrtn bldmt bldmc	bldapp.bat bldrtn.bat bldmt.bat bldmc.bat
CLI samples/cli	bldapp bldrtn bldmc	bldapp bldrtn bldmc	bldapp bldrtn bldmc	bldapp bldrtn bldmc	bldapp.bat bldrtn.bat bldmc.bat
SQLJ samples/java/sqlj	bldsqlj bldsqljs	bldsqlj bldsqljs	bldsqlj bldsqljs	bldsqlj bldsqljs	bldsqlj.bat bldsqljs.bat
IBM® COBOL samples/cobol	bldapp bldrtn	n/a	n/a	n/a	bldapp.bat bldrtn.bat
Micro Focus COBOL samples/cobol_mf	bldapp bldrtn	bldapp bldrtn	bldapp bldrtn	bldapp bldrtn	bldapp.bat bldrtn.bat
C# samples\.NET\cs	n/a	n/a	n/a	n/a	bldapp.bat bldrtn.bat
Visual Basic. NET samples\.NET\vb	n/a	n/a	n/a	n/a	bldapp.bat bldrtn.bat

Los archivos de creación se utilizan en la documentación sobre la creación de aplicaciones y rutinas porque muestran muy claramente las opciones de compilación y enlace que DB2 recomienda para los compiladores soportados. Generalmente existen otras muchas opciones de compilación y enlace, que el usuario puede probar. Consulte la documentación del compilador para conocer todas las opciones de compilación y enlace proporcionadas. Además de crear los programas de ejemplo, el programador puede también crear sus propios programas mediante los archivos de creación. Los programas de ejemplo se pueden utilizar como modelos que el usuario puede modificar al desarrollar sus programas.

Una característica práctica de los archivos de creación es que están diseñados para crear un archivo fuente cuyo nombre puede ser cualquier nombre de archivo permitido por el compilador. En cambio, en los makefiles, los nombres de programa están codificados en el archivo. Los makefiles acceden a los archivos de creación para compilar y enlazar los programas que crean. Los archivos de creación utilizan la variable \$1 en UNIX y la variable %1 en los sistemas operativos Windows para que internamente se utilice la variable en lugar del nombre de programa. Mediante un mayor número de estos nombres de variable se sustituyen otros argumentos que puedan ser necesarios.

Los archivos de creación permiten experimentar de forma rápida y sencilla, pues cada archivo está pensado para una clase específica de creación de programas, tales como la creación de programas autónomos, rutinas (procedimientos almacenados y funciones definidas por el usuario) o tipos de programa más especializados, tales como los programas de conexión múltiple o los programas

multihebra. Se proporciona cada tipo de archivo de creación siempre que el compilador da soporte a la clase determinada de programa para el que está diseñado el archivo de creación.

El archivo objeto y el archivo ejecutable creados por un archivo de creación se sobrescriben cada vez que se crea un programa, aunque el archivo fuente no se modifique. No ocurre así cuando se utiliza un makefile. Por tanto, el programador puede volver a crear un programa existente sin tener que suprimir un archivo objeto o archivo ejecutable anterior, ni modificar el archivo fuente.

Los archivos de creación contienen un valor por omisión para la base de datos "sample" (base de datos de ejemplo). Si el usuario desea acceder a otra base de datos, puede simplemente proporcionar otro parámetro para sobrescribir el valor por omisión. Si utilizan regularmente la otra base de datos, pueden codificar el nombre de esa base de datos, en lugar de sample, dentro del propio archivo de creación.

Para programas de SQL incorporado, salvo cuando se utilice el precompilador COBOL de IBM en Windows, los archivos de creación llamar a otro archivo, embprep, que contiene los pasos de precompilación y enlace para los programas de SQL incorporado. Para estos pasos pueden ser necesarios los parámetros opcionales correspondientes al ID de usuario y la contraseña, según el lugar donde se cree el programa de SQL incorporado.

Excepto para SQLJ, si un programador crea el programa en una instancia de servidor donde reside la base de datos, el ID de usuario y la contraseña serán comunes a ambos y, por tanto, no es necesario proporcionarlos. En cambio, si el programa se crea en una instancia diferente, por ejemplo, en una máquina cliente que accede a un servidor remoto de bases de datos, se deberán proporcionar dichos parámetros.

Los archivos de creación de SQLJ necesitan un ID de usuario y una contraseña para el personalizador db2sqljcustomize, incluso para acceder a una base de datos local. Esto sigue los convenios del controlador JDBC universal para DB2.

Por último, el programador puede modificar los archivos de creación de acuerdo con sus necesidades. Además de cambiar el nombre de la base de datos dentro del archivo de creación (tal como se describió anteriormente), el programador puede codificar fácilmente otros parámetros dentro del archivo, cambiar opciones de compilación y enlace, o modificar la vía de instancia predefinida de DB2. El carácter simple, directo y específico del archivo de creación hace que el usuario pueda adaptarlo fácilmente a sus necesidades.

Conceptos relacionados:

- "Makefiles" en la página 113
- "Programas de utilidad de comprobación de errores" en la página 116
- "Archivos de ejemplo" en la página 65

Información relacionada:

- "Software de desarrollo soportado por AIX" en la página 9
- "Software de desarrollo soportado por HP-UX" en la página 11
- "Software de desarrollo soportado por Linux" en la página 13
- "Software de desarrollo soportado por Solaris" en la página 18
- "Software de desarrollo soportado por Windows" en la página 20

Makefiles

El directorio "samples" (ejemplos) correspondiente a un compilador soportado contiene un archivo "makefile" para crear la mayoría de los programas de ejemplo proporcionados en el directorio. El makefile llama a los archivos de creación para compilar y enlazar cada programa. La sintaxis de los makefiles y los datos resultantes de sus mandatos difieren en algunos aspectos importantes respecto a los archivos de creación. Sin embargo, el usuario puede utilizar el makefile como interfaz para los archivos de creación y así explotar los potentes y sencillos mandatos del makefile:

make <nombre_programa>

Compila y enlaza el programa especificado.

make all

Compila y enlaza todos los programas listados en el makefile.

make clean

Suprime todos los archivos intermedios, tales como archivos objeto, correspondientes a todos los programas listados en el makefile.

make cleanall

Suprime todos los archivos intermedios y archivos ejecutables correspondientes a todos los programas listados en el makefile.

| Java™ no utiliza habitualmente makefiles y no se proporcionan ejecutables make
| con los Java Developer Kits. En cambio, DB2® proporciona makefiles como opción
| para los ejemplos de Java, por si el usuario desea hacer uso de los mandatos make.
| Para utilizar los makefiles de Java, debe tener disponible un ejecutable make que
| se utilice normalmente con el compilador de otro lenguaje.

La tabla siguiente lista los makefiles de cada plataforma proporcionados por DB2 para los lenguajes de programación y las API principales, y los directorios de ejemplos donde están situados los makefiles. En la documentación en línea, los makefiles están enlazados dinámicamente con los programas de ejemplo resultantes de los makefiles. También puede acceder a estos archivos en los directorios de ejemplos.

Tabla 40. Makefiles de ejemplo por plataforma

Plataforma → Idioma	AIX®	HP-UX	Linux	Solaris	Windows®
C samples/c	makefile	makefile	makefile	makefile	makefile
C++ samples/cpp	makefile	makefile	makefile	makefile	makefile
CLI samples/cli	makefile	makefile	makefile	makefile	makefile
JDBC samples/java/jdbc	makefile	makefile	makefile	makefile	makefile
SQLJ samples/java/sqlj	makefile	makefile	makefile	makefile	makefile
IBM® COBOL samples/cobol	makefile	n/a	n/a	n/a	makefile
Micro Focus COBOL samples/cobol_mf	makefile	makefile	makefile	makefile	makefile
Procedimientos de SQL samples/sqlproc	makefile	makefile	makefile	makefile	makefile
C# samples\.NET\cs	n/d	n/d	n/d	n/d	makefile
Visual Basic. NET samples\.NET\vb	n/d	n/d	n/d	n/d	makefile

A diferencia de los archivos de creación, los makefiles no sobrescriben los archivos intermedios y archivos ejecutables correspondientes a los programas listados en el makefile. Si se utiliza el mandato `make all`, es más rápida la creación de ejecutables para algunos de los archivos si ya existen ejecutables para otros archivos, pues `make all` pasará por alto estos archivos. Pero ello implica también tener que utilizar los mandatos `make clean` y `make cleanall` para eliminar los archivos objeto y archivos ejecutables que ya no sean necesarios.

Los makefiles se pueden utilizar para el desarrollo de programas. Debido a que necesitan que el nombre del programa se codifique dentro del propio archivo, le pueden parecer menos prácticos de utilizar que los archivos de creación, pero son una opción a tener en cuenta si desea disponer de la potencia y comodidad de los mandatos `make`.

Los makefiles organizan los programas que invocan de acuerdo con varias categorías de programas cliente y servidor representadas por variables (vea los makefiles para conocer detalles). Cuando añada un programa a un makefile, compruebe que se pueda acceder a él mediante las variables correctas. Por ejemplo, un programa que se puede ejecutar en cualquier cliente (local respecto al servidor o remoto) se coloca dentro de la variable `client_run`.

También es necesario que especifique el nombre de programa en la variable `cleanall` para asegurarse de que el ejecutable producido se pueda suprimir mediante el mandato `make cleanall`. Además, si es un programa de SQL incorporado, especifique el archivo de SQL no incorporado resultante de la precompilación en la variable `clean` para que el mandato `make clean` (así como el mandato `make cleanall` que sirve para invocarlo) suprima el archivo de SQL no incorporado.

Además, es necesario que especifique el nuevo archivo con la sintaxis correcta para invocar el archivo de creación apropiado para compilar y enlazar el programa.

Para apreciar dónde es necesario añadir un nuevo archivo a uno de los makefiles de ejemplo, a continuación se indican los lugares donde está situado el programa de SQL incorporado, dbauth, dentro del makefile C de AIX:

```

*****
#                               2f - make client_run
*****

client_run : \
    cli_info clisnap \
    dbauth dbconn dbcreate dbinfo dbmcon \
. . .
*****
#                               2g - make clean
*****

clean :
    $(ERASE) *.o
    $(ERASE) *.DEL *.TXT *.MSG
    $(ERASE) dbauth.c dbcfg.c dbconn.c dbmcon.c dbmcon1.c dbmcon2.c
. . .
*****
#                               2h - make cleanall
*****

cleanall : \
    clean
    $(ERASE) *.bnd
    $(ERASE) cli_info clisnap
    $(ERASE) dbauth dbcfg dbconn dbcreate dbinfo dbmcon dbmcon1 dbmcon2
. . .
*****
#                               3b - regular samples, embedded SQL
*****

dbauth :
    $(BLDAPP) dbauth $(ALIAS) $(UID) $(PWD)

```

Las tres variables que siguen al nombre del programa en la última línea anterior, ALIAS, UID y PWD, representan, respectivamente, el alias de base de datos, el ID de usuario y la contraseña de la base de datos. Estas variables se pasan al archivo de creación (en este caso, el archivo de creación bldapp representado por la variable BLDAPP). Si el programa utiliza SQL incorporado, ALIAS, UID y PWD se pasan a su vez al script de precompilación y vinculación embprep, el cual es invocado por el archivo de creación. Antes de utilizar el makefile, puede ser necesario cambiar los valores de estas variables. Por omisión, el valor de ALIAS es la base de datos sample, mientras que UID y PWD no tienen ningún valor definido.

UID y PWD son parámetros opcionales que no es necesario definirlos si el usuario ya está trabajando en la misma instancia que el servidor de bases de datos. En otro caso, por ejemplo, si el usuario se conecta de forma remota al servidor desde una máquina cliente, el usuario debe modificar el makefile para asignar los valores correctos a las variables UID y PWD a fin de acceder a la base de datos.

Para los programas de conexión múltiple, los makefiles para C, CLI y C++ tienen un segundo alias de base de datos, ALIAS2, cuyo valor por omisión es la base de datos sample2. Las variables correspondientes del ID de usuario y la contraseña,

UID2 y PWD2, no tienen ningún valor definido. Estas variables, al igual que las variables UID y PWD, no necesitan un valor si a la segunda base de datos se accede localmente.

Los makefiles también definen una variable ERASE para suprimir archivos cuando se invocan los mandatos `make clean` y `make cleanall`. En UNIX®, se establece en `rm -f`; en Windows se establece en `del`.

Conceptos relacionados:

- “Archivos de creación” en la página 110
- “Programas de utilidad de comprobación de errores” en la página 116
- “Archivos de ejemplo” en la página 65

Información relacionada:

- “Software de desarrollo soportado por AIX” en la página 9
- “Software de desarrollo soportado por HP-UX” en la página 11
- “Software de desarrollo soportado por Linux” en la página 13
- “Software de desarrollo soportado por Solaris” en la página 18
- “Software de desarrollo soportado por Windows” en la página 20

Programas de utilidad de comprobación de errores

DB2® AD Client proporciona varios archivos de programa de utilidad. Estos archivos contienen funciones para la comprobación de errores y la impresión de información de errores. Los archivos de programa de utilidad correspondientes a cada lenguaje se encuentran en el directorio “samples”. Cuando se utilizan junto con un programa de aplicación, los archivos de los programas de utilidad de comprobación de errores proporcionan información útil sobre errores y facilitan la depuración de los programas DB2. La mayoría de los programas de comprobación de errores utilizan las API de DB2 `GET SQLSTATE MESSAGE (sqllogstt)` y `GETERROR MESSAGE (sqlaintp)` para obtener la información pertinente de `SQLSTATE` y `SQLCA` referente a problemas encontrados al ejecutar el programa. El archivo de programa de utilidad de la CLI de DB2, `utilcli.c`, no hace uso de estas API de DB2; en su lugar utiliza sentencias equivalentes de la CLI de DB2. Todos los programas de utilidad de comprobación de errores imprimen mensajes de error descriptivos para ayudar al programador a comprender rápidamente el problema.

Algunos programas DB2, tales como las rutinas (procedimientos almacenados y funciones definidas por el usuario), no necesitan hacer uso de los programas de utilidad. Tampoco son necesarios para Java™, pues si se produce una excepción, se lanza el objeto `SQLException`.

A continuación se indican los archivos de los programas de utilidad de comprobación de errores utilizados por los compiladores soportados por DB2 para los diversos lenguajes de programación:

Tabla 41. Archivos del programa de utilidad de comprobación de errores por idioma

Idioma	Archivo fuente de SQL no incorporado	Archivo de cabecera de SQL no incorporado	Archivo fuente de SQL incorporado	Archivo de cabecera de SQL incorporado
C samples/c	utilapi.c	utilapi.h	utilemb.sqc	utilemb.h
C++ samples/cpp	utilapi.C	utilapi.h	utilemb.sqC	utilemb.h
CLI samples/cli	utilcli.c	utilcli.h	n/a	n/a
IBM® COBOL samples/cobol	checkerr.cb1	n/a	n/a	n/a
Micro Focus COBOL samples/cobol_mf	checkerr.cb1	n/a	n/a	n/a

Para poder utilizar las funciones del programa de utilidad, se debe primero compilar el archivo del programa de utilidad y luego enlazar su archivo objeto durante la creación del archivo ejecutable del programa destino. Tanto el makefile como los archivos de creación contenidos en los directorios samples realizan esas acciones para los programas que necesitan hacer uso de los programas de utilidad de comprobación de errores.

El ejemplo siguiente muestra cómo se utilizan los programas de utilidad de comprobación de errores en los programas DB2. El archivo de cabecera utilemb.h define la macro EMB_SQL_CHECK, para las funciones SqlInfoPrint() y TransRollback():

```
/* macro para la comprobación del SQL incorporado */
#define EMB_SQL_CHECK( MSG_STR ) \
SqlInfoPrint(MSG_STR, &sqlca, __LINE__, __FILE__); \
if (sqlca.sqlcode < 0) \
{ \
    TransRollback(); \
    return 1; \
}
```

SqlInfoPrint() comprueba el indicador SQLCODE. Esta función imprime la información disponible referente al error específico denotado por este indicador. También indica el lugar donde se produjo el error en el código fuente. TransRollback() permite que el archivo de programa de utilidad retrotraiga sin peligro una transacción donde se ha producido un error. Esta función hace que sentencias de SQL incorporado conecten con la base de datos y ejecuten una retrotracción. El ejemplo siguiente muestra cómo el programa de C dbuse invoca las funciones del programa de utilidad utilizando la macro, y proporciona el valor "Delete with host variables -- Execute" para el parámetro MSG_STR de la función SqlInfoPrint():

```
EXEC SQL DELETE FROM org
WHERE deptnumb = :hostVar1 AND
division = :hostVar2;
EMB_SQL_CHECK("Delete with host variables -- Execute");
```

La macro EMB_SQL_CHECK hace que si la sentencia DELETE falla, la transacción se retrotraiga sin peligro, y se imprima el mensaje de error apropiado.

El programador puede utilizar estos programas de utilidad de comprobación de errores y utilizarlos como modelo al crear sus propios programas DB2.

Conceptos relacionados:

- “Archivos de creación” en la página 110
- “Makefiles” en la página 113
- “Archivos de ejemplo” en la página 65

Parte 2. Creación y ejecución de aplicaciones independientemente de la plataforma

Capítulo 4. Java

Programas Java de ejemplo	121		Opciones de aplicaciones y applets SQLJ para UNIX	133
Consideraciones sobre los applets Java	122		Script de creación de Windows para aplicaciones y applets SQLJ	134
JDBC	124		Opciones de aplicaciones y applets SQLJ para Windows	135
Creación de applets JDBC	124		Creación de rutinas SQLJ	136
Creación de aplicaciones JDBC	126		Script de creación de UNIX para rutinas SQLJ	138
Creación de rutinas JDBC	126		Opciones de rutinas SQLJ para UNIX	139
SQLJ	128		Archivo de proceso por lotes de Windows para rutinas SQLJ.	139
Creación de programas SQLJ	128		Opciones de rutinas SQLJ para Windows	141
Creación de applets SQLJ	129			
Creación de aplicaciones SQLJ.	131			
Script de creación de UNIX para aplicaciones y applets SQLJ	132			

Este capítulo proporciona información detallada para crear applets y aplicaciones de Java. Para conocer la información más reciente sobre el desarrollo de aplicaciones Java de DB2, consulte la página Web situada en:

<http://www.ibm.com/software/data/db2/udb/ad/v8/java>

Programas Java de ejemplo

DB2[®] proporciona programas de ejemplo para mostrar la creación y ejecución de programas JDBC que usan exclusivamente SQL dinámico, y programas SQLJ que utilizan SQL estático. Existen directorios separados para los ejemplos de JDBC y SQLJ dentro del directorio java de "samples". A continuación se muestra la estructura del directorio samples de Java[™] en los sistemas operativos UNIX[®] y Windows[®]:

- En UNIX:

sql1ib/samples/java

Contiene un archivo README para todos los programas Java de ejemplo contenidos en todos los subdirectorios.

sql1ib/samples/java/jdbc

Contiene archivos de programas de ejemplo JDBC.

sql1ib/samples/java/sqlj

Contiene programas de ejemplo SQLJ.

sql1ib/samples/java/Websphere

Contiene programas de ejemplo Websphere.

sql1ib/samples/java/plugin

Contiene archivos de ejemplo para plugins del Centro de Control de DB2.

sql1ib/samples/java/plugin/doc

Contiene archivos javadoc para las interfaces de plugin.

- En Windows:

sql1ib\samples\java

Contiene un archivo README para todos los programas Java de ejemplo contenidos en todos los subdirectorios.

sql1ib\samples\java\jdbc

Contiene programas de ejemplo JDBC.

sql1ib\samples\java\sqlj

Contiene programas de ejemplo SQLJ.

sql1ib\samples\java\Websphere

Contiene programas de ejemplo Websphere.

sql1ib\samples\java\plugin

Contiene archivos de ejemplo para plugins del Centro de Control de DB2.

sql1ib\samples\java\plugin\doc

Contiene archivos javadoc para las interfaces de plugin.

El directorio samples de SQLJ contiene archivos de creación (archivos de script en UNIX, archivos de proceso por lotes en Windows) para crear el SQL incorporado de los programas Java. El directorio JDBC no contiene archivos de creación porque la creación de programas JDBC utilizando javac desde la línea de mandatos es tan sencilla que no son necesarios archivos de creación.

Los directorios samples de JDBC y SQLJ también contienen makefiles opcionales. Los makefiles no se utilizan de forma habitual en Java, y los Java Development Kits (JDK) no se proporcionan con archivos ejecutables make. DB2 proporciona makefiles de ejemplo para Java por si el usuario desea aprovechar la comodidad que su uso proporciona. Cada makefile de Java crear todos los programas de ejemplo proporcionados en el directorio samples de JDBC o SQLJ. Puede utilizar un programa make, tal como gnumake, que se utilice con un compilador de otro lenguaje.

Se proporcionan dos archivos de creación de SQLJ: bldsqlj en UNIX, o bldsqlj.bat en Windows, que crean applets y aplicaciones SQLJ, y bldsqljs en UNIX, o bldsqljs.bat en Windows, que crean rutinas SQLJ (procedimientos almacenados y funciones definidas por el usuario).

Tareas relacionadas:

- “Configuración del entorno Java” en la página 28
- “Creación de applets JDBC” en la página 124
- “Creación de aplicaciones JDBC” en la página 126
- “Creación de rutinas JDBC” en la página 126
- “Creación de applets SQLJ” en la página 129
- “Creación de aplicaciones SQLJ” en la página 131
- “Creación de rutinas SQLJ” en la página 136
- “Creación de programas SQLJ” en la página 128

Información relacionada:

- “Ejemplos de JDBC” en la página 87
- “Ejemplos de SQLJ” en la página 90
- “Ejemplos de Java WebSphere” en la página 92
- “Ejemplos de plugins de Java” en la página 93

Consideraciones sobre los applets Java

Se puede acceder a las bases de datos DB2[®] utilizando applets Java[™]. Tenga en cuenta lo siguiente cuando utilice applets Java:

1. Si está utilizando el controlador de tipo 3 ahora en desuso (también conocido como controlador "net"), es esencial que el archivo db2java.zip utilizado por el applet Java esté al mismo nivel de FixPak que el servidor de applets JDBC. En circunstancias normales, db2java.zip se carga desde el servidor Web donde se está ejecutando el servidor de applets JDBC. Esto asegura una correspondencia. En cambio, si el applet Java carga db2java.zip desde una ubicación diferente, se puede producir una discrepancia. La correspondencia entre los niveles de FixPak de los dos archivos se hace cumplir estrictamente durante la conexión. Si se detecta una falta de correspondencia, se rechaza la conexión y el cliente recibe una de las excepciones siguientes:

- Si db2java.zip está al nivel de DB2 Versión 7 FixPak 2 o posterior:

```
COM.ibm.db2.jdbc.DB2Exception: [IBM][Controlador JDBC]
  CLI0621E Configuración de servidor JDBC no soportada.
```

- Si db2java.zip es anterior a DB2 versión 7 FixPak 2:

```
COM.ibm.db2.jdbc.DB2Exception: [IBM][Controlador JDBC]
  CLI0601E Se ha cerrado la sentencia o descriptor de contexto de
  sentencia no válidos. SQLSTATE=S1000
```

Si se produce una discrepancia, el servidor de applets JDBC registra uno de los mensajes siguientes en el archivo jdbcerr.log:

- Si el servidor de applets JDBC está al nivel de DB2 Versión 7 FixPak 2 o posterior:

```
jdbcFSQLConnect: Las versiones del servidor de applets JDBC y el cliente
(db2ja a.zip) no coinciden. La conexión no puede proseguir., einfo= -111
```

- Si el servidor de applets JDBC es anterior a DB2 versión 7 FixPak 2:

```
jdbcServiceConnection(): Se ha recibido una petición no válida., einfo= 0
```

Nota: El archivo de ejemplo db2JDBCVersion.java de sqllib\samples\java (Windows) o sqllib/samples/java (UNIX) no debe utilizarse con DB2 versión 8. Este programa puede utilizarse con DB2 versión 7 para comprobar qué versión del controlador JDBC para DB2 se está utilizando actualmente y si el entorno JDBC está configurado correctamente para la misma.

Es muy recomendable que el usuario migre sus applets hacia el controlador de JDBC universal para DB2.

2. En el caso de un applet JDBC o SQLJ de mayor tamaño que conste de varias clases Java, puede elegir empaquetar todas sus clases en un solo archivo JAR. Para un applet SQLJ, también tendría que empaquetar sus perfiles serializados junto con sus clases. Si decide hacer esto, añada el archivo JAR al parámetro archive en el código "applet". Para obtener detalles, consulte la documentación de Java Developer Kit versión 1.3.

Para los applets SQLJ, algunos navegadores no tienen todavía soporte para cargar un objeto serializado desde un archivo de recursos asociado al applet. Por ejemplo, obtendrá el mensaje de error siguiente si intenta cargar el applet de ejemplo proporcionado App1t en esos navegadores:

```
java.lang.ClassNotFoundException: App1t_SJProfile0
```

Para evitar este problema, existe un programa de utilidad que convierte un perfil serializado en un perfil que está almacenado según el formato de una clase Java. El programa de utilidad es una clase Java llamada sqlj.runtime.profile.util.SerProfileToClass. Este programa utiliza como

entrada un archivo de recursos de un perfil serializado y produce como resultado una clase Java donde está contenido el perfil. El perfil se puede convertir utilizando uno de estos mandatos:

```
profconv Applet_SJProfile0.ser
```

o

```
java sqlj.runtime.profile.util.SerProfileToClass Applet_SJProfile0.ser
```

Como resultado se crea la clase `Applet_SJProfile0.class`. Normalmente el problema se resuelve sustituyendo todos los perfiles con formato `.ser` utilizados por el applet por perfiles con formato `.class`.

3. Puede colocar los archivos `db2java.zip` o `db2jcc.jar`, o ambos, en un directorio compartido por varios applets que se pueden cargar desde su sitio Web. `db2java.zip` es para los applets que utilizan el controlador de JDBC tipo 3; `db2jcc.jar` es para los applets que utilizan el controlador de JDBC universal para DB2 o para cualquier applet SQLJ. Estos archivos están en el directorio `sqllib\java` en los sistemas operativos Windows®, y en el directorio `sqllib/java` en UNIX®. Puede ser necesario añadir un parámetro `codebase` al código "applet" del archivo HTML para identificar el directorio. Para obtener detalles, consulte la documentación de Java Developer Kit versión 1.3.
4. A partir de DB2 Versión 5.2, se ha añadido el manejo de señales al servidor de applets JDBC (oyente), `db2jd`, para hacerlo más robusto. Como resultado, no se puede utilizar el mandato `Control-C` para matar el proceso de `db2jd`. Por tanto, la única forma de cerrar el oyente consiste en matar el proceso utilizando `kill -9` (en Unix) o el Administrador de Tareas (en Windows).
5. Para obtener información sobre la ejecución de applets Java de DB2 en un servidor Web, específicamente el servidor Web Domino™ Go, visite este sitio Web:

<http://www.ibm.com/software/data/db2/db2lotus/gojava.htm>

Tareas relacionadas:

- "Configuración del entorno Java" en la página 28
- "Creación de applets JDBC" en la página 124
- "Creación de applets SQLJ" en la página 129

JDBC

Creación de applets JDBC

`Applet` muestra el uso de un applet Java de SQL dinámico para acceder a una base de datos DB2.

Procedimiento:

Puede utilizar el controlador de tipo 3, ahora en desuso (conocido también como controlador "net"), o el controlador universal JDBC, que se instala con DB2 Java Enablement. Más abajo se describe la conexión mediante estos dos controladores. Es muy recomendable que migre sus applets al controlador JDBC universal.

Para crear y ejecutar el applet JDBC, `Applet`, mediante mandatos entrados en la línea de mandatos, asegúrese de tener un servidor Web instalado y en ejecución en

su máquina DB2 (servidor o cliente), o utilice el visor de applets incluido con el Java Development Kit entrando el mandato siguiente desde el directorio de trabajo de la máquina cliente:

```
appletviewer App1t.html
```

Conexión mediante el controlador de tipo 3 (controlador "net")

Para conectar mediante el controlador de tipo 3, modifique primero el archivo App1t.html de acuerdo con las instrucciones contenidas en el archivo. Luego, inicie el servidor de applets JDBC en el puerto TCP/IP especificado en App1t.html. Por ejemplo, si ha especificado param name=port value='6789' en App1t.html, entraría lo siguiente:

```
db2jstrt 6789
```

Compruebe que el número de puerto JDBC utilizado para la conexión es el valor por omisión recomendado, "6789". Cambie este valor solamente si está seguro de que el nuevo valor no entrará en conflicto con otro número de puerto. No utilice el número de puerto de la base de datos, "50000".

Conexión mediante el controlador JDBC universal

Para conectar mediante el controlador JDBC universal, modifique el archivo App1t.html según las instrucciones contenidas en el mismo. Como número de puerto TCP/IP, puede utilizar el número de puerto de la base de datos, "50000".

Creación del applet

1. Compile App1t.java para obtener el archivo App1t.class mediante este mandato:

```
javac App1t.java
```

2. Compruebe que su navegador Web puede acceder a su directorio de trabajo. Si el directorio no es accesible, copie App1t.class y App1t.html en un directorio que sea accesible.
3. Si está utilizando un controlador de tipo 3, copie sqllib\java\db2java.zip (para Windows) o sqllib/java/db2java.zip (para UNIX) en el mismo directorio que App1t.class y App1t.html.

Si utiliza el controlador JDBC universal, copie sqllib\java\db2jcc.jar en Windows o sqllib/java/db2jcc.jar en UNIX, en el mismo directorio que App1t.class y App1t.html.

4. En la máquina cliente, inicie el navegador Web (que debe dar soporte a Java 1.3) y cargue App1t.html.

Puede también utilizar el makefile de Java para crear este programa.

Conceptos relacionados:

- "Consideraciones sobre los applets Java" en la página 122

Tareas relacionadas:

- "Creación de aplicaciones JDBC" en la página 126
- "Creación de rutinas JDBC" en la página 126
- "Creación de applets SQLJ" en la página 129

Información relacionada:

- "Ejemplos de JDBC" en la página 87

Ejemplos relacionados:

- “Applt.java -- A Java applet that use JDBC applet driver to access a database (JDBC)”

Creación de aplicaciones JDBC

DbInfo muestra el uso de una aplicación Java de SQL dinámico para acceder a una base de datos DB2.

Procedimiento:

Para crear y ejecutar esta aplicación mediante mandatos entrados en la línea de mandatos:

1. Compile DbInfo.java para obtener el archivo DbInfo.class mediante este mandato:

```
javac DbInfo.java
```

2. Ejecute el intérprete java para la aplicación mediante este mandato:

```
java DbInfo
```

Puede también utilizar el makefile de Java para crear este programa.

Nota: Si está ejecutando una aplicación Java en Unix en una instancia de DB2 de 64 bits, pero el Java Developer Kit es de 32 bits, debe cambiar la vía de bibliotecas de DB2 antes de ejecutar la aplicación. Por ejemplo, en AIX:

- En el shell bash o Korn:

```
export LIBPATH=$HOME/sqllib/lib32
```

- En el shell C:

```
setenv LIBPATH $HOME/sqllib/lib32
```

Tareas relacionadas:

- “Creación de applets JDBC” en la página 124
- “Creación de rutinas JDBC” en la página 126
- “Creación de aplicaciones SQLJ” en la página 131

Información relacionada:

- “Ejemplos de JDBC” en la página 87

Ejemplos relacionados:

- “DbInfo.java -- How to get/set info in a database (JDBC)”

Creación de rutinas JDBC

DB2 proporciona programas de ejemplo para mostrar el uso de rutinas JDBC (procedimientos almacenados y funciones definidas por el usuario) contenidas en el directorio `samples/java/jdbc` (para UNIX) o en el directorio `samples\java\jdbc` (para Windows). Las rutinas se compilan y almacenan en un servidor. Cuando una aplicación cliente llama a una rutina, ésta accede al servidor de bases de datos y devuelve datos a la aplicación cliente.

Procedimiento:

Los ejemplos siguientes muestran cómo crear rutinas, las cuales pueden ser:

- procedimientos almacenados
- funciones definidas por el usuario sin sentencias de SQL
- funciones definidas por el usuario con sentencias de SQL

Procedimientos almacenados

SpServer muestra procedimientos almacenados PARAMETER STYLE JAVA de SQL dinámico.

Para crear y ejecutar este programa desde la línea de mandatos del servidor:

1. Compile SpServer.java para obtener el archivo SpServer.class, mediante este mandato:

```
javac SpServer.java
```

2. Copie SpServer.class en el directorio sqllib\function (en sistemas operativos Windows) o en el directorio sqllib/function (en UNIX).
3. A continuación, catalogue las rutinas ejecutando el script spcat en el servidor. Escriba:

```
spcat
```

Este script conecta con la base de datos "sample", descataloga mediante SpDrop.db2 las rutinas que se hubieran catalogado previamente, luego las cataloga llamando a SpCreate.db2, y finalmente desconecta de la base de datos. También puede ejecutar los scripts SpDrop.db2 y SpCreate.db2 por separado.

4. A continuación, detenga y reorganice la base de datos para que se reconozca el nuevo archivo de clase. Si es necesario, defina la modalidad de archivo del archivo de clase como "read" para que pueda ser leído por el usuario.
5. Compile y ejecute la aplicación cliente SpClient para acceder a la clase del procedimiento almacenado.

Funciones definidas por el usuario sin sentencias de SQL

UDFsrv es una biblioteca de funciones definidas por el usuario que no contiene sentencias de SQL. DB2 proporciona una aplicación cliente JDBC, UDFcli, y una aplicación cliente SQLJ, UDFcli, que pueden acceder a la biblioteca UDFsrv.

Para crear y ejecutar el programa UDF desde la línea de mandatos del servidor:

1. Compile UDFsrv.java para obtener el archivo UDFsrv.class, mediante este mandato:

```
javac UDFsrv.java
```

2. Copie UDFsrv.class en el directorio sqllib\function (para sistemas operativos Windows) o en el directorio sqllib/function (para UNIX).
3. Para acceder a la biblioteca UDFsrv, puede utilizar aplicaciones cliente JDBC o SQLJ. Ambas versiones del programa cliente contienen la sentencia de SQL CREATE FUNCTION, que puede utilizar para registrar en la base de datos las UDF (funciones definidas por el usuario) contenidas en UDFsrv. Los programas cliente también contienen sentencias de SQL que hacen uso de las UDF, una vez que éstas se han registrado.

Funciones definidas por el usuario con sentencias de SQL

UDFsqlsv es una biblioteca de funciones definidas por el usuario que contiene sentencias de SQL. DB2 proporciona una aplicación cliente JDBC, UDFsqlcl, para acceder a la biblioteca UDFsqlsv.

Para crear y ejecutar el programa UDF desde la línea de mandatos del servidor:

1. Compile `UDFsqlsv.java` para obtener el archivo `UDFsqlsv.class`, mediante este mandato:

```
javac UDFsqlsv.java
```

2. Copie `UDFsqlsv.class` en el directorio `sqllib\function` (para sistemas operativos Windows) o en el directorio `sqllib/function` (para UNIX).
3. Para acceder a la biblioteca `UDFsqlsv`, utilice el programa cliente, `UDFsqlcl`, el cual contiene la sentencia de SQL `CREATE FUNCTION`, que puede utilizar para registrar en la base de datos las UDF (funciones definidas por el usuario) contenidas en `UDFsqlsv`. El programa cliente también contiene sentencias de SQL que hacen uso de las UDF, una vez que éstas se han registrado.

Puede también utilizar el `makefile` de Java para crear los programas anteriores.

Tareas relacionadas:

- “Creación de applets JDBC” en la página 124
- “Creación de aplicaciones JDBC” en la página 126
- “Creación de rutinas SQLJ” en la página 136

Información relacionada:

- “Ejemplos de JDBC” en la página 87

Ejemplos relacionados:

- “`spcat -- To catalog SQLj stored procedures on UNIX`”
- “`SpClient.java -- Call a variety of types of stored procedures from SpServer.java (JDBC)`”
- “`SpCreate.db2 -- How to catalog the stored procedures contained in SpServer.java`”
- “`SpDrop.db2 -- How to uncatalog the stored procedures contained in SpServer.java`”
- “`SpServer.java -- Provide a variety of types of stored procedures to be called from (JDBC)`”
- “`UDFcli.java -- Call the UDFs in UDFsrv.java (JDBC)`”
- “`UDFCreate.db2 -- How to catalog the Java UDFs contained in UDFsrv.java`”
- “`UDFDrop.db2 -- How to uncatalog the Java UDFs contained in UDFsrv.java`”
- “`UDFsCreate.db2 -- How to catalog the UDFs contained in UDFsqlsv.java`”
- “`UDFsDrop.db2 -- How to uncatalog the UDFs contained in UDFsqlsv.java`”
- “`UDFsqlcl.java -- Call the UDFs in UDFsqlsv.java (JDBC)`”
- “`UDFsqlsv.java -- Provide UDFs to be called by UDFsqlcl.java (JDBC)`”
- “`UDFsrv.java -- Provide UDFs to be called by UDFcli.java (JDBC)`”

SQLJ

Creación de programas SQLJ

DB2 proporciona archivos de creación para crear los programas de ejemplo SQLJ. Para los applets y aplicaciones, puede utilizar el script `bldsqj` en UNIX, o el archivo de proceso por lotes `bldsqj.bat` en Windows. Para las rutinas

(procedimientos almacenados y funciones definidas por el usuario), puede utilizar el script `blsqlj` en UNIX, o el archivo de proceso por lotes `blsqlj.bat` en Windows.

El programa traductor de SQLJ que se incluye con DB2 llama al compilador de Java para compilar los archivos `.java` convertidos y generar archivos `.class`. Por lo tanto, los archivos de creación utilizan el mandato `sqlj` para hacer ambas cosas.

Notas:

1. En versiones anteriores de DB2, el mandato `db2prof` utilizaba un URL de la forma `-url=jdbc:db2:nombreBD`, donde `nombreBD` era el alias de base de datos catalogado localmente. El mandato nuevo `db2sqljcustomize` sigue los convenios del controlador universal JDBC para DB2: `-url jdbc:db2://sistpral:número_puerto/nombreBD` donde `sistpral` es el nombre del servidor DB2, `número_puerto` es el número del puerto TCP/IP de escucha del servidor DB2 y `nombreBD` es el alias de base de datos catalogado en el servidor DB2. Esto supone que el servidor DB2 debe estar configurado para conexiones TCP/IP.
2. Los programas SQLJ convertidos con versiones anteriores del mandato `sqlj` se deben volver a convertir con el mandato `sqlj` de DB2 versión 8 y personalizar con el mandato `db2sqljcustomize`.
3. La impresora de perfiles de SQLJ de DB2, `db2sqljprint`, imprime el contenido de un perfil de DB2 en texto plano.

Procedimiento:

Para crear los diversos tipos de programas SQLJ de DB2, consulte estos temas:

- “Creación de applets SQLJ” en la página 129
- “Creación de aplicaciones SQLJ” en la página 131
- “Creación de rutinas SQLJ” en la página 136

Conceptos relacionados:

- “Programas Java de ejemplo” en la página 121
- “Consideraciones sobre los applets Java” en la página 122

Tareas relacionadas:

- “Creación de applets SQLJ” en la página 129
- “Creación de aplicaciones SQLJ” en la página 131
- “Creación de rutinas SQLJ” en la página 136

Información relacionada:

- “`db2sqljcustomize` - Mandato Personalizador de perfiles SQLJ de DB2” en el manual *Consulta de mandatos*
- “`db2sqljprint` - Mandato Impresora de perfiles SQLJ de DB2” en el manual *Consulta de mandatos*
- “Ejemplos de SQLJ” en la página 90
- “`sqlj` - Mandato Conversor SQLJ de DB2” en el manual *Consulta de mandatos*

Creación de applets SQLJ

Los pasos siguientes describen cómo crear el programa de ejemplo `App1t` que muestra el uso de un applet SQLJ para acceder a una base de datos DB2. Estos

pasos hacen uso del archivo de creación, `bldsqlj` (UNIX) o `bldsqlj.bat` (Windows), el cual contiene mandatos para crear un applet o aplicación SQLJ.

El archivo de creación utiliza como entrada un máximo de seis parámetros: \$1, \$2, \$3, \$4, \$5 y \$6 en UNIX, y %1, %2, %3, %4, %5 y %6 en Windows. El primer parámetro especifica el nombre del programa. El segundo parámetro especifica el ID de usuario correspondiente a la instancia de base de datos; el tercer parámetro especifica la contraseña. El cuarto parámetro especifica el nombre del servidor. El quinto parámetro especifica el número de puerto. Finalmente, el sexto parámetro especifica el nombre de la base de datos. Se pueden utilizar valores por omisión para todos los parámetros, excepto para el primero, el nombre del programa. Consulte el archivo de creación para conocer detalles sobre la utilización de valores de parámetro por omisión.

Procedimiento:

| Puede utilizar el controlador de tipo 3, ahora en desuso (conocido también como
| controlador "net"), o el controlador universal JDBC, que se instala con DB2 Java
| Enablement. Más abajo se describe la conexión mediante estos dos controladores.
| Es muy recomendable que migre sus applets al controlador JDBC universal.

Para ejecutar este applet, asegúrese de tener un servidor Web instalado y en ejecución en su máquina DB2 (servidor o cliente), o utilice el visor de applets incluido con el Java Development Kit entrando el mandato siguiente desde el directorio de trabajo de la máquina cliente:

```
appletviewer App1t.html
```

Conexión mediante el controlador de tipo 3 (controlador "net")

Para conectar mediante el controlador de tipo 3, modifique primero el archivo `App1t.html` de acuerdo con las instrucciones contenidas en el archivo. Luego, inicie el servidor de applets JDBC en el puerto TCP/IP especificado en `App1t.html`. Por ejemplo, si ha especificado `param name=port value='6789'` en `App1t.html`, entraría lo siguiente:

```
db2jstrt 6789
```

Compruebe que el número de puerto JDBC utilizado para la conexión es el valor por omisión recomendado, "6789". Cambie este valor solamente si está seguro de que el nuevo valor no entrará en conflicto con otro número de puerto. No utilice el número de puerto de la base de datos, "50000".

Conexión mediante el controlador JDBC universal

| Para conectar mediante el controlador JDBC universal, modifique el archivo
| `App1t.html` según las instrucciones contenidas en el mismo. Como número de
| puerto TCP/IP debe utilizar el número de puerto de la base de datos, "50000".

Creación del applet

1. Cree el applet, utilizando este mandato:

```
bldsqlj App1t <IDusuario> <contraseña> <nombre_servidor> <número_puerto> <nombre_bd>
```

donde todos los parámetros, salvo el nombre del programa, pueden utilizar valores por omisión, tal como se explica en el archivo de creación.

2. Compruebe que su navegador Web o visor de applets, si lo utiliza, puede acceder a su directorio de trabajo. Si su directorio no es accesible, copie los archivos siguientes en un directorio que sea accesible:

Applt.html	Applt.class
Applt_Cursor1.class	Applt_Cursor2.class
Applt_SJProfileKeys.class	Applt_SJProfile0.ser

3. Si está utilizando un controlador de tipo 3, copie `sql1lib\java\db2jcc.jar` y `sql1lib\java\db2java.zip` en Windows, o `sql1lib/java/db2jcc.jar` y `sql1lib/java/db2java.zip` en UNIX, en el mismo directorio que `Applt.class` y `Applt.html`.

Si utiliza el controlador JDBC universal, copie `sql1lib\java\db2jcc.jar` en Windows o `sql1lib/java/db2jcc.jar` en UNIX, en el mismo directorio que `Applt.class` y `Applt.html`.

4. En la máquina cliente, inicie el navegador Web (que debe dar soporte al Java Developer Kit 1.3) o el visor de applets, y cargue `Applt.html`.

Puede también utilizar el `makefile` de Java para crear este programa.

Conceptos relacionados:

- “Consideraciones sobre los applets Java” en la página 122

Tareas relacionadas:

- “Creación de applets JDBC” en la página 124
- “Creación de aplicaciones SQLJ” en la página 131
- “Creación de rutinas SQLJ” en la página 136

Información relacionada:

- “Opciones de aplicaciones y applets SQLJ para UNIX” en la página 133
- “Opciones de aplicaciones y applets SQLJ para Windows” en la página 135
- “Ejemplos de SQLJ” en la página 90

Ejemplos relacionados:

- “`Applt.sqlj` -- An SQLJ applet that uses a JDBC applet driver to access a database (SQLj)”
- “`bldsqlj.bat` -- Builds a Java embedded SQL (SQLJ) application or applet on Windows”
- “`bldsqlj` -- Builds Java embedded SQL (SQLJ) applications and applets on UNIX”

Creación de aplicaciones SQLJ

Los pasos siguientes describen cómo crear el programa de ejemplo `TbMod` que muestra el uso de una aplicación SQLJ para acceder a una base de datos DB2. Estos pasos hacen uso del archivo de creación, `bldsqlj` (UNIX) o `bldsqlj.bat` (Windows), el cual contiene mandatos para crear un applet o aplicación SQLJ.

El archivo de creación utiliza como entrada un máximo de seis parámetros: `$1`, `$2`, `$3`, `$4`, `$5` y `$6` en UNIX, y `%1`, `%2`, `%3`, `%4`, `%5` y `%6` en Windows. El primer parámetro especifica el nombre del programa. El segundo parámetro especifica el ID de usuario correspondiente a la instancia de base de datos; el tercer parámetro especifica la contraseña. El cuarto parámetro especifica el nombre del servidor. El quinto parámetro especifica el número de puerto. Finalmente, el sexto parámetro especifica el nombre de la base de datos. Se pueden utilizar valores por omisión

para todos los parámetros, excepto para el primero, el nombre del programa. Consulte el archivo de creación para conocer detalles sobre la utilización de valores de parámetro por omisión.

Procedimiento:

Para crear TbMod mediante el archivo de creación bldsqlj (UNIX) o bldsqlj.bat (Windows), entre este mandato:

```
bldsqlj TbMod <IDusuario> <contraseña> <nombre_servidor> <número_puerto> <nombre_bd>
```

donde todos los parámetros, salvo el nombre del programa, pueden utilizar valores por omisión, tal como se explica en el archivo de creación.

Ejecute el intérprete Java para la aplicación mediante este mandato:

```
java TbMod
```

Puede también utilizar el makefile de Java para crear este programa.

Nota: Si está ejecutando una aplicación Java en Unix en una instancia de DB2 de 64 bits, pero el Java Developer Kit es de 32 bits, debe cambiar la vía de bibliotecas de DB2 antes de ejecutar la aplicación. Por ejemplo, en AIX:

- En el shell bash o Korn:

```
export LIBPATH=$HOME/sql1lib/lib32
```
- En el shell C:

```
setenv LIBPATH $HOME/sql1lib/lib32
```

Tareas relacionadas:

- “Creación de aplicaciones JDBC” en la página 126
- “Creación de applets SQLJ” en la página 129
- “Creación de rutinas SQLJ” en la página 136

Información relacionada:

- “Opciones de aplicaciones y applets SQLJ para UNIX” en la página 133
- “Opciones de aplicaciones y applets SQLJ para Windows” en la página 135
- “Ejemplos de SQLJ” en la página 90

Ejemplos relacionados:

- “bldsqlj.bat -- Builds a Java embedded SQL (SQLJ) application or applet on Windows”
- “bldsqlj -- Builds Java embedded SQL (SQLJ) applications and applets on UNIX”
- “TbMod.sqlj -- How to modify table data (SQLj)”

Script de creación de UNIX para aplicaciones y applets SQLJ

```
#!/bin/sh
# SCRIPT: bldsqlj
# Crea aplicaciones y applets de SQL incorporado de Java (SQLJ) en UNIX
# Uso: bldsqlj nombre_prog (requiere contraseña e ID de usuario codificados)
#     bldsqlj nombre_prog IDusuario contraseña
#     bldsqlj nombre_prog IDusuario contraseña nombre_servidor
#     bldsqlj nombre_prog IDusuario contraseña nombre_servidor número_puerto
#     bldsqlj nombre_prog IDusuario contraseña nombre_servidor número_puerto nombre_bd
#
# Valores por omisión:
#     IDusuario      = la variable $USER requiere actualización si se usa
#     contraseña     = la variable $PSWD requiere actualización si se usa
```

```

# nombre_servidor = variable $SERVER establecida en nombre sist.princ. local
# número_puerto = variable $PORTNUM establecida en 50000
# nombre_bd = variable $DB establecida en "sample"

# Para codificar el ID de usuario (USER) y la contraseña (PSWD)
# Sustituya "NULL" por los valores correctos entre comillas
USER="NULL"
PSWD="NULL"
# Puede sustituir cada uno de los valores por omisión siguientes por
# un nuevo valor. Observe que el número de PORTNUM no puede corresponder
# a un puerto que ya utilice otro proceso.
SERVER="nombresistprinc"
PORTNUM=50000
DB="sample"

# Traducir y compilar el archivo fuente SQLJ
# y vincular el paquete con la base de datos.
if ( ( [ $# -eq 1 ] && [ $USER != "NULL" ] && [ $PSWD != "NULL" ] ) || \
( [ $# -ge 3 ] && [ $# -le 6 ] ) )
then
# Eliminar la extensión .sqlj
nombreprog=${1%.sqlj}

sqlj "${nombreprog}.sqlj"

if [ $# -eq 1 ]
then
db2sqljcustomize -url jdbc:db2://$SERVER:$PORTNUM/$DB \
-user $USER -password $PSWD "${nombreprog}_SJProfile0"
elif [ $# -eq 3 ]
then
db2sqljcustomize -url jdbc:db2://$SERVER:$PORTNUM/$DB -user $2 -password $3 \
"${nombreprog}_SJProfile0"
elif [ $# -eq 4 ]
then
db2sqljcustomize -url jdbc:db2://$4:$PORTNUM/$DB -user $2 -password $3 \
"${nombreprog}_SJProfile0"
elif [ $# -eq 5 ]
then
db2sqljcustomize -url jdbc:db2://$4:$5/$DB -user $2 -password $3 \
"${nombreprog}_SJProfile0"
else
db2sqljcustomize -url jdbc:db2://$4:$5/$6 -user $2 -password $3 \
"${nombreprog}_SJProfile0"
fi
else
echo 'Uso: bldsqlj nombre_prog (requiere contraseña e ID de usuario codificados)'
echo ' bldsqlj nombre_prog IDusuario contraseña'
echo ' bldsqlj nombre_prog IDusuario contraseña nombre_servidor'
echo ' bldsqlj nombre_prog IDusuario contraseña nombre_servidor número_puerto'
echo ' bldsqlj nombre_prog IDusuario contraseña nombre_servidor número_puerto nombre_bd'
echo ''
echo ' Valores por omisión:'
echo ' IDusuario = '$USER
echo ' contraseña = '$PSWD
echo ' nombre_servidor = '$SERVER
echo ' número_puerto = '$PORTNUM
echo ' nombre_bd = '$DB
fi

```

Opciones de aplicaciones y applets SQLJ para UNIX

La tabla siguiente contiene las opciones de conversión y personalización de SQLJ utilizadas en el script de creación bldsqlj en UNIX. Estas son las opciones que DB2 recomienda que utilice para crear aplicaciones y applets de SQLJ en las plataformas UNIX.

Opciones de conversión y personalización para bldsqlj

sqlj	El conversor SQLJ (también compila el programa).
"\${nombreprog}.sqlj"	El archivo fuente de SQLJ. El mandato progname=\${1%.sqlj} elimina la extensión si se ha incluido en el nombre de archivo de entrada, por lo que cuando se vuelve a añadir la extensión no está duplicado.
db2sqljcustomize	El personalizador de perfiles de DB2 para Java.
-url	Especifica un URL de JDBC para establecer una conexión de base de datos, tal como jdbc:db2://nombreServidor:50000/sample.
-user	Especifica un ID de usuario.
-password	Especifica una contraseña.
"\${nombreprog}_SJProfile0"	Especifica un perfil serializado para el programa.

Tareas relacionadas:

- "Creación de applets SQLJ" en la página 129
- "Creación de aplicaciones SQLJ" en la página 131

Información relacionada:

- "Opciones de rutinas SQLJ para UNIX" en la página 139

Ejemplos relacionados:

- "bldsqlj -- Builds Java embedded SQL (SQLJ) applications and applets on UNIX"

Script de creación de Windows para aplicaciones y applets SQLJ

```
@echo off
rem BATCH FILE: bldsqlj.bat
rem Crea una aplicación o applet de SQL incorporado de Java (SQLJ) en Windows

rem Para añadir los valores por omisión para el ID de usuario (USER) y contraseña (PSWD)
rem Descomentar lo siguiente y añadir los valores apropiados
rem set USR=
rem set PSWD=

rem Puede sustituir cada uno de los valores por omisión siguientes por
rem un nuevo valor. Observe que el número de PORTNUM no puede corresponder
rem a un puerto que ya utilice otro proceso.
set SERVER=%COMPUTERNAME%
set PORTNUM=50000
set DB=sample

goto start
:usage
echo Uso: bldsqlj nombre_prog (requiere contraseña e ID de usuario codificados)
echo      bldsqlj nombre_prog IDusuario contraseña
echo      bldsqlj nombre_prog IDusuario contraseña nombre_servidor
echo      bldsqlj nombre_prog IDusuario contraseña nombre_servidor número_puerto
echo      bldsqlj nombre_prog IDusuario contraseña nombre_servidor número_puerto nombre_bd
echo.
echo Valores por omisión:
echo      IDusuario      = %USR%
echo      contraseña     = %PSWD%
echo      nombre_servidor = %SERVER%
echo      número_puerto  = %PORTNUM%
echo      nombre_bd      = %DB%
```

```

goto exit

:start
rem Traducir y compilar el archivo fuente SQLJ
rem y vincular el paquete con la base de datos.
if "%1" == "" goto usage
if "%2" == "" goto case1
if "%3" == "" goto usage
if "%4" == "" goto case3
if "%5" == "" goto case4
if "%6" == "" goto case5
if "%7" == "" goto case6
goto usage

:case1
if "%USR%" == "" goto usage
if "%PSWD%" == "" goto usage
if "%SERVER%" == "" goto nohostname
sqlj %1.sqlj
db2sqljcustomize -url jdbc:db2://%SERVER%:%PORTNUM%/%DB% -user %USR%
                -password %PSWD% %1_SJProfile0
goto continue

:case3
if "%SERVER%" == "" goto nohostname
sqlj %1.sqlj
db2sqljcustomize -url jdbc:db2://%SERVER%:%PORTNUM%/%DB% -user %2
                -password %3 %1_SJProfile0
goto continue

:case4
sqlj %1.sqlj
db2sqljcustomize -url jdbc:db2://%4:%PORTNUM%/%DB% -user %2
                -password %3 %1_SJProfile0
goto continue

:case5
sqlj %1.sqlj
db2sqljcustomize -url jdbc:db2://%4:%5/%DB% -user %2
                -password %3 %1_SJProfile0
goto continue

:case6
sqlj %1.sqlj
db2sqljcustomize -url jdbc:db2://%4:%5/%6 -user %2
                -password %3 %1_SJProfile0
goto continue

:continue
goto exit

:nohostname
echo No se ha podido determinar el nombre del servidor (hostname).
echo.
goto usage

:exit
@echo on

```

Opciones de aplicaciones y applets SQLJ para Windows

La tabla siguiente contiene las opciones de conversión y personalización de SQLJ utilizadas en el archivo de proceso por lotes b1dsq1j.bat en los sistemas operativos Windows. Éstas son las opciones que DB2 recomienda que utilice para crear aplicaciones y applets de SQLJ en Windows.

Opciones de conversión y personalización para bldsqlj.bat	
sqlj	El conversor SQLJ (también compila el programa).
%1.sqlj	El archivo fuente de SQLJ.
db2sqljcustomize	El personalizador de perfiles de DB2 para Java.
-url	Especifica un URL de JDBC para establecer una conexión de base de datos, tal como jdbc:db2://nombreServidor:50000/sample.
-user	Especifica un ID de usuario.
-password	Especifica una contraseña.
%1_SJProfile0	Especifica un perfil serializado para el programa.

Tareas relacionadas:

- “Creación de applets SQLJ” en la página 129
- “Creación de aplicaciones SQLJ” en la página 131

Información relacionada:

- “Opciones de rutinas SQLJ para Windows” en la página 141

Ejemplos relacionados:

- “bldsqlj.bat -- Builds a Java embedded SQL (SQLJ) application or applet on Windows”

Creación de rutinas SQLJ

DB2 proporciona programas de ejemplo para mostrar el uso de rutinas SQLJ (procedimientos almacenados y funciones definidas por el usuario) contenidas en el directorio `samples/java/sqlj` (para UNIX) o en el directorio `samples\java\sqlj` (para Windows). Las rutinas se compilan y almacenan en un servidor. Cuando una aplicación cliente llama a una rutina, ésta accede al servidor de bases de datos y devuelve datos a la aplicación cliente.

En el mismo directorio, DB2 también proporciona el archivo de creación `bldsqljs` (UNIX) o `bldsqljs.bat` (Windows), que contiene mandatos para crear rutinas.

El archivo de creación utiliza como entrada un máximo de seis parámetros: \$1, \$2, \$3, \$4, \$5 y \$6 en UNIX, y %1, %2, %3, %4, %5 y %6 en Windows. El primer parámetro especifica el nombre del programa. El segundo parámetro especifica el ID de usuario correspondiente a la instancia de base de datos; el tercer parámetro especifica la contraseña. El cuarto parámetro especifica el nombre del servidor. El quinto parámetro especifica el número de puerto. Finalmente, el sexto parámetro especifica el nombre de la base de datos. Se pueden utilizar valores por omisión para todos los parámetros, excepto para el primero, el nombre del programa. Consulte el archivo de creación para conocer detalles sobre la utilización de valores de parámetro por omisión.

Procedimiento:

El ejemplo siguiente muestra cómo crear un archivo de clase con procedimientos almacenados.

SpServer muestra procedimientos almacenados PARAMETER STYLE JAVA que hacen uso del controlador de aplicaciones JDBC para acceder a una base de datos DB2.

Para crear esta clase de procedimiento almacenado con el archivo de creación bldsqljs (UNIX) o bldsqljs.bat (Windows):

1. Entre el mandato siguiente:

```
bldsqljs SpServer <IDusuario> <contraseña> <nombre_servidor> \  
    <número_puerto> <nombre_bd>
```

donde todos los parámetros, salvo el nombre del programa, pueden utilizar valores por omisión, tal como se explica en el archivo de creación.

2. A continuación, catalogue las rutinas ejecutando el script spcat en el servidor. Escriba:

```
spcat
```

Este script conecta con la base de datos "sample", descataloga mediante SpDrop.db2 las rutinas que se hubieran catalogado previamente, luego las cataloga llamando a SpCreate.db2, y finalmente desconecta de la base de datos. También puede ejecutar los scripts SpDrop.db2 y SpCreate.db2 por separado.

3. A continuación, detenga y rearranque la base de datos para que se reconozca el nuevo archivo de clase. Si es necesario, defina la modalidad de archivo del archivo de clase como "read" para que pueda ser leído por el usuario.
4. Cree y ejecute la aplicación cliente SpClient para invocar los procedimientos almacenados. Puede crear SpClient mediante el archivo de creación de aplicaciones bldsqlj (UNIX) o bldsqlj.bat (Windows).

Puede también utilizar el makefile de Java para crear los programas anteriores.

Tareas relacionadas:

- "Creación de rutinas JDBC" en la página 126
- "Creación de applets SQLJ" en la página 129
- "Creación de aplicaciones SQLJ" en la página 131

Información relacionada:

- "Opciones de rutinas SQLJ para UNIX" en la página 139
- "Opciones de rutinas SQLJ para Windows" en la página 141
- "Ejemplos de SQLJ" en la página 90

Ejemplos relacionados:

- "bldsqljs.bat -- Builds a Java embedded SQL (SQLJ) stored procedure on Windows"
- "bldsqljs -- Builds Java embedded SQL (SQLJ) stored procedures on UNIX"
- "spcat -- To catalog SQLJ stored procedures on UNIX"
- "SpClient.sqlj -- Call a variety of types of stored procedures from SpServer.sqlj (SQLJ)"
- "SpCreate.db2 -- How to catalog the stored procedures contained in SpServer.sqlj"

- "SpDrop.db2 -- How to uncatlog the stored procedures contained in SpServer.sqlj"
- "SpIterat.sqlj -- Iterator class file for SpServer.sqlj (SQLj)"
- "SpServer.sqlj -- Provide a variety of types of stored procedures to be called from (SQLj)"

Script de creación de UNIX para rutinas SQLJ

```

#!/bin/sh
# SCRIPT: blsqljs
# Crea procedimientos almacenados de SQL incorporado de Java (SQLJ) en UNIX
# Uso: blsqljs nombre_prog (requiere contraseña e ID de usuario codificados)
#     blsqljs nombre_prog IDusuario contraseña
#     blsqljs nombre_prog IDusuario contraseña nombre_servidor
#     blsqljs nombre_prog IDusuario contraseña nombre_servidor número_puerto
#     blsqljs nombre_prog IDusuario contraseña nombre_servidor número_puerto nombre_bd
#
# Valores por omisión:
#     IDusuario      = la variable $USER requiere actualización si se usa
#     contraseña     = la variable $PSWD requiere actualización si se usa
#     nombre_servidor = variable $SERVER establecida en nombre sist.princ. local
#     número_puerto  = variable $PORTNUM establecida en 50000
#     nombre_bd      = variable $DB establecida en "sample"

# Para codificar el ID de usuario (USER) y la contraseña (PSWD)
# Sustituya "NULL" por los valores correctos entre comillas
USER="NULL"
PSWD="NULL"
# Puede sustituir cada uno de los valores por omisión siguientes por
# un nuevo valor. Observe que el número de PORTNUM no puede corresponder
# a un puerto que ya utilice otro proceso.
SERVER="nombresistprinc"
PORTNUM=50000
DB="sample"

# Traducir y compilar el archivo fuente SQLJ
# y vincular el paquete con la base de datos.
if ( ( [ $# -eq 1 ] && [ $USER != "NULL" ] && [ $PSWD != "NULL" ] ) \
    || ( [ $# -ge 3 ] && [ $# -le 6 ] ) )
then
    # Eliminar la extensión .sqlj
    nombreprog=${1%.sqlj}

    sqlj "${nombreprog}.sqlj"

    if [ $# -eq 1 ]
    then
        db2sqljcustomize -url jdbc:db2://$SERVER:$PORTNUM/$DB \
            -user $USER -password $PSWD "${nombreprog}_SJProfile0"
    elif [ $# -eq 3 ]
    then
        db2sqljcustomize -url jdbc:db2://$SERVER:$PORTNUM/$DB -user $2 \
            -password $3 "${nombreprog}_SJProfile0"
    elif [ $# -eq 4 ]
    then
        db2sqljcustomize -url jdbc:db2://$4:$PORTNUM/$DB -user $2 -password $3 \
            "${nombreprog}_SJProfile0"
    elif [ $# -eq 5 ]
    then
        db2sqljcustomize -url jdbc:db2://$4:$5/$DB -user $2 -password $3 \
            "${nombreprog}_SJProfile0"
    else
        db2sqljcustomize -url jdbc:db2://$4:$5/$6 -user $2 -password $3 \
            "${nombreprog}_SJProfile0"
    fi

    # Copiar los archivos *.class y *.ser en el directorio 'function'.
    rm -f "$DB2PATH/function/${nombreprog}*.class"
    rm -f "$DB2PATH/function/${nombreprog}*.ser"
    cp "${nombreprog}*.class" "$DB2PATH/function"
    cp "${nombreprog}*.ser" "$DB2PATH/function"

    else
    echo 'Uso: blsqljs nombre_prog (requiere contraseña e ID de usuario codificados)'
    echo '     blsqljs nombre_prog IDusuario contraseña'

```

```

echo '      bldsqljs nombre_prog IDusuario contraseña nombre_servidor'
echo '      bldsqljs nombre_prog IDusuario contraseña nombre_servidor número_puerto'
echo '      bldsqljs nombre_prog IDusuario contraseña nombre_servidor número_puerto nombre_bd'
echo ''
echo '      Valores por omisión:'
echo '      IDusuario      = '$USER
echo '      contraseña     = '$PSWD
echo '      nombre_servidor = '$SERVER
echo '      número_puerto  = '$PORTNUM
echo '      nombre_bd      = '$DB
fi

```

Opciones de rutinas SQLJ para UNIX

La tabla siguiente contiene las opciones de conversión y personalización de SQLJ utilizadas en el script de creación `bldsqljs` en UNIX. Estas son las opciones que DB2 recomienda que utilice para crear rutinas SQLJ (procedimientos almacenados y funciones definidas por el usuario) en las plataformas UNIX.

Opciones de conversión y personalización para <code>bldsqljs</code>	
<code>sqlj</code>	El conversor SQLJ (también compila el programa).
<code>"\${nombreprog}.sqlj"</code>	El archivo fuente de SQLJ. El mandato <code>progname=\${1%.sqlj}</code> elimina la extensión si se ha incluido en el nombre de archivo de entrada, por lo que cuando se vuelve a añadir la extensión no está duplicado.
<code>db2sqljcustomize</code>	El personalizador de perfiles de DB2 para Java.
<code>-url</code>	Especifica un URL de JDBC para establecer una conexión de base de datos, tal como <code>jdbc:db2://nombreServidor:50000/sample</code> .
<code>-user</code>	Especifica un ID de usuario.
<code>-password</code>	Especifica una contraseña.
<code>"\${nombreprog}_SJProfile0"</code>	Especifica un perfil serializado para el programa.

Tareas relacionadas:

- “Creación de rutinas SQLJ” en la página 136

Información relacionada:

- “Opciones de aplicaciones y applets SQLJ para UNIX” en la página 133

Ejemplos relacionados:

- “`bldsqljs -- Builds Java embedded SQL (SQLJ) stored procedures on UNIX`”

Archivo de proceso por lotes de Windows para rutinas SQLJ

```

@echo off
rem BATCH FILE: bldsqljs.bat
rem Crea un procedimiento almacenado de SQL incorporado de Java (SQLJ) en Windows

rem Para añadir valores por omisión para ID de usuario (USER) y contraseña (PSWD)
rem Descomentar lo siguiente y añadir los valores apropiados
rem set USR=
rem set PSWD=
rem Puede sustituir cada uno de los valores por omisión siguientes por
rem un nuevo valor. Observe que el número de PORTNUM no puede corresponder
rem a un puerto que ya utilice otro proceso.
set SERVER=%COMPUTERNAME%
set PORTNUM=50000

```

```

set DB=sample

goto start
:usage
echo Uso: bldsqljs nombre_prog (requiere contraseña e ID de usuario codificados)
echo      bldsqljs nombre_prog IDusuario contraseña
echo      bldsqljs nombre_prog IDusuario contraseña nombre_servidor
echo      bldsqljs nombre_prog IDusuario contraseña nombre_servidor número_puerto
echo      bldsqljs nombre_prog IDusuario contraseña nombre_servidor número_puerto nombre_bd
echo.
echo Valores por omisión:
echo      IDusuario      = %USR%
echo      contraseña     = %PSWD%
echo      nombre_servidor = %SERVER%
echo      número_puerto  = %PORTNUM%
echo      nombre_bd      = %DB%
goto exit

:start
rem Traducir y compilar el archivo fuente SQLJ
rem y vincular el paquete con la base de datos.
if "%DB2PATH%" == "" goto nodb2cmd
if "%1" == "" goto usage
if "%2" == "" goto case1
if "%3" == "" goto usage
if "%4" == "" goto case3
if "%5" == "" goto case4
if "%6" == "" goto case5
if "%7" == "" goto case6
goto usage

:case1
if "%USR%" == "" goto usage
if "%PSWD%" == "" goto usage
if "%SERVER%" == "" goto nohostname
sqlj %1.sqlj
db2sqljcustomize -url jdbc:db2://%SERVER%:%PORTNUM%/%DB% -user %USR%
                -password %PSWD% %1_SJProfile0
goto continue

:case3
if "%SERVER%" == "" goto nohostname
sqlj %1.sqlj
db2sqljcustomize -url jdbc:db2://%SERVER%:%PORTNUM%/%DB% -user %2
                -password %3 %1_SJProfile0
goto continue

:case4
sqlj %1.sqlj
db2sqljcustomize -url jdbc:db2://%4:%PORTNUM%/%DB% -user %2
                -password %3 %1_SJProfile0
goto continue

:case5
sqlj %1.sqlj
db2sqljcustomize -url jdbc:db2://%4:%5/%DB% -user %2
                -password %3 %1_SJProfile0
goto continue

:case6
sqlj %1.sqlj
db2sqljcustomize -url jdbc:db2://%4:%5/%6 -user %2
                -password %3 %1_SJProfile0
goto continue

:continue
rem Copiar los archivos *.class y *.ser en el directorio 'function'.
copy %1*.class %DB2PATH%\function\
copy %1*.ser %DB2PATH%\function\
goto exit

:nodb2cmd
echo Entorno de línea de mandatos de DB2 no inicializado. Ejecute de nuevo
echo db2cmd e inténtelo de nuevo.
goto exit

```

```

:hostname
echo No se ha podido determinar el nombre del servidor (hostname).
echo.
goto usage

:exit
@echo on

```

Opciones de rutinas SQLJ para Windows

La tabla siguiente contiene las opciones de conversión y personalización de SQLJ utilizadas en el archivo de proceso por lotes `bldsqljs.bat` en los sistemas operativos Windows. Estas son las opciones que DB2 recomienda que utilice para crear rutinas SQLJ (procedimientos almacenados y funciones definidas por el usuario).

Opciones de conversión y personalización para <code>bldsqljs.bat</code>	
sqlj	El conversor SQLJ (también compila el programa).
%1.sqlj	El archivo fuente de SQLJ.
db2sqljcustomize	El personalizador de perfiles de DB2 para Java.
-url	Especifica un URL de JDBC para establecer una conexión de base de datos, tal como <code>jdbc:db2://nombreServidor:50000/sample</code> .
-user	Especifica un ID de usuario.
-password	Especifica una contraseña.
%1_SJProfile0	Especifica un perfil serializado para el programa.

Tareas relacionadas:

- “Creación de rutinas SQLJ” en la página 136

Información relacionada:

- “Opciones de aplicaciones y applets SQLJ para Windows” en la página 135

Ejemplos relacionados:

- “`bldsqljs.bat` -- Builds a Java embedded SQL (SQLJ) stored procedure on Windows”

Capítulo 5. El procesador de línea de mandatos

Ejecución de scripts del procesador de línea de mandatos (CLP)	143	Llamada a procedimientos desde el Procesador de línea de mandatos (CLP)	144
--	-----	---	-----

Este capítulo proporciona información detallada para utilizar el Procesador de línea de mandatos para acceder a una base de datos DB2, ejecutando scripts de CLP, y llamar a procedimientos almacenados de DB2 en la línea de mandatos con la sentencia CALL.

Los procedimientos de SQL también se codifican en scripts de CLP. Puede encontrar información sobre la creación de procedimientos SQL en el apartado Capítulo 6, "Procedimientos de SQL", en la página 147.

Para conocer la información más reciente sobre el desarrollo de aplicaciones DB2, consulte la página Web situada en:

<http://www.ibm.com/software/data/db2/udb/ad>

Ejecución de scripts del procesador de línea de mandatos (CLP)

Se puede acceder directamente al CLP desde la línea de mandatos de una instancia de DB2. Se puede ejecutar de forma interactiva entrando "db2" en la línea de mandatos. También se puede ejecutar en modalidad no interactiva entrando mandatos de DB2, o scripts de DB2, precedidos de la palabra clave "db2". Los ejemplos que se utilizan aquí corresponden a la modalidad no interactiva.

Los scripts de CLP codifican sentencias de SQL directamente, en lugar de incorporarlas en un lenguaje de sistema principal, de forma que el usuario necesite conocer el SQL para programar un script del CLP. Y con los scripts del CLP no hay necesidad de compilar ni enlazar archivos fuente antes de ejecutar el script.

DB2 proporciona scripts de CLP para crear y ejecutar sentencias de SQL para la base de datos sample. Estos scripts están ubicados en el directorio `sql11ib/samples/clp` en UNIX, y en el directorio `sql11ib\samples\clp` en Windows. El archivo README que se encuentra en el directorio describe los programas y el modo de ejecutarlos.

Procedimiento:

El script del CLP, `cte.db2`, define dos expresiones de tabla comunes, `PAYLEVEL` y `PAYBYED`, a las que se accede mediante una sentencia `SELECT`.

Para ejecutar este script:

1. en primer lugar, conecte con la base de datos "sample":
`db2 connect to sample`
2. a continuación, entre el mandato siguiente:
`db2 -vf cte.db2 -t`

La letra "v" corresponde al distintivo de detalle, que no es necesario pero proporciona una salida detallada, por lo cual es recomendable.

Por omisión, el resultado se visualiza en la pantalla. Proporciona la expresión de tabla común de SQL y la salida de la misma:

```
WITH PAYLEVEL AS (SELECT EMPNO, YEAR(HIREDATE) AS HIREYEAR, EDLEVEL,
SALARY+BONUS+COMM AS TOTAL_PAY FROM EMPLOYEE WHERE EDLEVEL > 16 ),
PAYBYED (EDUC_LEVEL, YEAR_OF_HIRE, AVG_TOTAL_PAY) AS (SELECT EDLEVEL,
HIREYEAR, AVG(TOTAL_PAY) FROM PAYLEVEL GROUP BY EDLEVEL, HIREYEAR )
SELECT EMPNO, EDLEVEL, YEAR_OF_HIRE, TOTAL_PAY, AVG_TOTAL_PAY FROM
PAYLEVEL, PAYBYED WHERE EDLEVEL=EDUC_LEVEL AND HIREYEAR = YEAR_OF_HIRE
AND TOTAL_PAY < AVG_TOTAL_PAY
```

EMPNO	EDLEVEL	YEAR_OF_HIRE	TOTAL_PAY	AVG_TOTAL_PAY
000210	17	1979	20132.00	25896.500000000000000000000000

1 record(s) selected.

Conceptos relacionados:

- “Procesador de línea de mandatos (CLP)” en el manual *Consulta de mandatos*

Tareas relacionadas:

- “Llamada a procedimientos desde el Procesador de línea de mandatos (CLP)” en la página 144

Información relacionada:

- “Ejemplos de procesador de línea de mandatos (CLP)” en la página 82
- “db2: Mandato de invocación del procesador de línea de mandatos” en el manual *Consulta de mandatos*

Ejemplos relacionados:

- “cte.db2 -- How to create a COMMON TABLE EXPRESSION ”

Llamada a procedimientos desde el Procesador de línea de mandatos (CLP)

Puede llamar a procedimientos almacenados utilizando la sentencia CALL desde la interfaz del procesador de línea de mandatos de DB2. El procedimiento almacenado al que se llame debe estar definido en las tablas de catálogo del sistema DB2.

Procedimiento:

Para llamar al procedimiento almacenado, primero conéctese a la base de datos:

```
db2 connect to sample user IDusuario using contraseña
```

donde *IDusuario* y *contraseña* son el ID de usuario y la contraseña de la instancia donde reside la base de datos sample.

Para utilizar la sentencia CALL, entre el nombre del procedimiento almacenado más cualquier valor de parámetro IN o INOUT, así como el signo '?' en calidad de contenedor de cada valor de parámetro OUT.

Los parámetros de un procedimiento almacenado se proporcionan en la sentencia CREATE PROCEDURE para dicho procedimiento en el archivo fuente del programa.

Ejemplo de procedimiento de SQL

Para obtener información sobre la creación de un procedimiento de SQL, consulte Creación de procedimientos de SQL.

En el archivo whiles.db2, la sentencia CREATE PROCEDURE para la signatura de procedimiento DEPT_MEDIAN es la siguiente:

```
CREATE PROCEDURE DEPT_MEDIAN
(IN deptNumber SMALLINT, OUT medianSalary DOUBLE)
```

Para invocar este procedimiento, utilice la sentencia CALL en la que debe especificar el nombre del procedimiento y los argumentos de parámetro adecuados, que en este caso son el valor para el parámetro IN y un signo de interrogación, '?', para el valor del parámetro OUT. La sentencia SELECT del procedimiento utiliza el valor deptNumber de la columna DEPT de la tabla STAFF, por ello, a fin de obtener una salida significativa, el parámetro IN tiene que ser un valor correcto de la columna DEPT; por ejemplo, el valor "51":

```
db2 call dept_median (51, ?)
```

Nota: En las plataformas UNIX, los paréntesis tienen un significado especial para el shell de mandatos, por lo que deben ir precedidos del carácter "\" o se deben encerrar entre comillas, del modo siguiente:

```
db2 "call dept_median (51, ?)"
```

No sirven las comillas si se ha de utilizar la modalidad interactiva del procesador de línea de mandatos.

Después de ejecutar el mandato anterior, tendría que recibir un resultado como el siguiente:

```
Value of output parameters
-----
Parameter Name : MEDIANSALARY
Parameter Value : +1.765450000000000E+004

Return Status = 0
```

Ejemplo de procedimiento almacenado de C

Con el Procesador de línea de mandatos, también puede llamar a procedimientos almacenados creados a partir de los lenguajes principales soportados. En el directorio samples/c de UNIX y en el directorio samples\c de Windows, DB2 proporciona archivos de creación de procedimientos almacenados. La biblioteca compartida spserver contiene varios procedimientos almacenados que se pueden crear a partir del archivo fuente, spserver.sqc. El archivo spcreate.db2 cataloga los procedimientos almacenados.

En el archivo spcreate.db2, la sentencia CREATE PROCEDURE para el procedimiento MAIN_EXAMPLE empieza así:

```
CREATE PROCEDURE MAIN_EXAMPLE (IN job CHAR(8),
                                OUT salary DOUBLE,
                                OUT errorcode INTEGER)
```

Para llamar a este procedimiento almacenado, tiene que colocar un valor CHAR para el parámetro IN, job, y un signo de interrogación, '?', para cada uno de los parámetros OUT. La sentencia SELECT del procedimiento utiliza el valor job de la columna JOB de la tabla EMPLOYEE, por ello, a fin de obtener una salida significativa, el parámetro IN tiene que ser un valor correcto de la columna JOB. El

programa de ejemplo en C, `spclient`, que llama al procedimiento almacenado, utiliza 'DESIGNER' para el valor de JOB. Se puede hacer lo mismo, del modo siguiente:

```
db2 "call MAIN_EXAMPLE ('DESIGNER', ?, ?)"
```

Después de ejecutar el mandato anterior, tendría que recibir un resultado como el siguiente:

```
Value of output parameters
-----
Parameter Name  : SALARY
Parameter Value : +2.37312500000000E+004

Parameter Name  : ERRORCODE
Parameter Value : 0

Return Status = 0
```

Un ERRORCODE con el valor de cero indica un resultado satisfactorio.

Si se compara con el programa `spclient`, se puede observar que `spclient` ha formateado el resultado en decimal para facilitar la visualización:

```
CALL stored procedure named MAIN_EXAMPLE
Stored procedure returned successfully
Average salary for job DESIGNER = 23731.25
```

Tareas relacionadas:

- “Creación de procedimientos SQL” en la página 147
- “Invocación de procedimientos SQL mediante aplicaciones cliente” en la página 148
- “Llamada a procedimientos desde activadores o rutinas SQL” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de servidor*
- “Llamada a procedimientos desde aplicaciones o rutinas externas” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de servidor*

Ejemplos relacionados:

- “`spclient.sqc` -- Call various stored procedures (C)”
- “`spcreate.db2` -- How to catalog the stored procedures contained in `spserver.sqc` (C)”
- “`spserver.sqc` -- Definition of various types of stored procedures (C)”
- “`whiles.db2` -- To create the DEPT_MEDIAN SQL procedure ”
- “`whiles.sqc` -- To call the DEPT_MEDIAN SQL procedure”

Capítulo 6. Procedimientos de SQL

Creación de procedimientos SQL	147		Copia de seguridad y restauración de
Invocación de procedimientos SQL mediante			procedimientos SQL creados antes de DB2 8.2
aplicaciones cliente	148		Revinculación de procedimientos SQL
Personalización de las opciones de precompilación			
y vinculación para procedimientos SQL	149		

Creación de procedimientos SQL

Los scripts del Procesador de Línea de Mandatos (CLP) de DB2 (que terminan con la extensión `.db2` contenidos en el directorio `sql1lib/samples/sqlproc` en UNIX y en el directorio `sql1lib\samples\sqlproc` en Windows, ejecutan las sentencias `CREATE PROCEDURE` para crear procedimientos almacenados en el servidor. Cada script del CLP tiene asociado un archivo de aplicación cliente del mismo nombre, cuya extensión denota el lenguaje de programación y la interfaz de la aplicación: `.sqc` (para SQL incorporado en lenguaje C), `.c` (para la CLI de DB2) o `.java` (para JDBC).

Notas:

1. A partir de DB2 versión 8.2, la creación de procedimientos de SQL no requiere un compilador C o C++ en el servidor, por lo que no se precisa ninguna configuración de compilador C o C++. Cuando se crea un procedimiento de SQL, las sentencias procesales del mismo se convierten en una representación nativa que se almacena en los catálogos de la base de datos, tal como se hace con otras sentencias de SQL. Cuando se llama a un procedimiento de SQL, se carga esta representación desde los catálogos y el motor de DB2 la ejecuta.
2. `CALL` es una sentencia SQL en DB2 Versión 8. Esto significa que ya no pueden crearse procedimientos en ningún orden. El compilador comprueba la existencia de procedimientos invocados en tiempo de compilación y activa un error con `SQLCODE -440` si no se encuentra el procedimiento.

Procedimiento:

Para ejecutar un script `CREATE PROCEDURE` del Procesador de Línea de Mandatos, conecte con la base de datos "sample" mediante este mandato:

```
db2 connect to sample user IDusuario using contraseña
```

donde *IDusuario* y *contraseña* son el ID de usuario y la contraseña de la instancia donde reside la base de datos `sample`.

Para ejecutar la sentencia `CREATE PROCEDURE` contenida en el archivo de script `resultset.db2`, entre el siguiente mandato:

```
db2 -td@ -vf resultset.db2
```

Ahora, el procedimiento SQL está preparado para ser invocado.

Tareas relacionadas:

- "Personalización de las opciones de precompilación y vinculación para procedimientos SQL" en la página 149
- "Revinculación de procedimientos SQL" en la página 152

Ejemplos relacionados:

- “resultset.db2 -- To register and create the MEDIAN_RESULT_SET SQL procedure”

Invocación de procedimientos SQL mediante aplicaciones cliente

Después de haber creado un procedimiento SQL tal como se explica en el apartado “Creación de procedimientos SQL” en la página 147, puede invocar el procedimiento SQL creando y ejecutando una aplicación cliente. DB2 proporciona programas cliente de ejemplo en `sql1lib/samples/sqlproc` (UNIX) y en `sql1lib\samples\sqlproc` (Windows). Existen archivos fuente de cliente para CLI de DB2, SQL incorporado para C y JDBC. DB2 proporciona scripts de procesador de línea de mandatos en el directorio de UNIX, y archivos de proceso por lotes en el directorio de Windows, para invocar los procedimientos SQL..

Procedimiento:

Dependiendo de la interfaz de aplicación que esté utilizando, siga los ejemplos siguientes para crear y ejecutar un programa cliente de ejemplo para invocar procedimientos SQL:

CLI de DB2

Para crear la aplicación cliente de la CLI de DB2, `resultset`, a partir del archivo fuente `resultset.c`, entre:

```
bldcli resultset
```

Este mandato crea el archivo ejecutable, `resultset` en UNIX y `resultset.exe` en Windows.

Para invocar el procedimiento SQL, ejecute la aplicación cliente de ejemplo entrando el nombre del archivo ejecutable, el nombre de la base de datos a la que se está conectando, y el ID de usuario y contraseña de la instancia de base de datos:

```
resultset basedatos IDusuario contraseña
```

SQL incorporado de C

Para crear la aplicación cliente de SQL incorporado, `basecase`, a partir del archivo fuente `basecase.sqc`, entre el nombre del archivo de script, el nombre del ejecutable, la base de datos a la que se está conectando, y el ID de usuario y contraseña de la instancia de base de datos:

```
bldapp basecase basedatos IDusuario contraseña
```

El resultado es un archivo ejecutable, `basecase` en UNIX y `basecase.exe` en Windows.

Para invocar el procedimiento SQL, ejecute la aplicación cliente de ejemplo entrando:

```
basecase basedatos IDusuario contraseña
```

JDBC Para crear la aplicación cliente de JDBC, `NestedSP`, a partir del archivo fuente `NestedSP.java`, compile el archivo fuente:

```
javac NestedSP.java
```

El resultado es el archivo de clase, `NestedSP.class`.

Para invocar el procedimiento SQL, ejecute el intérprete java para la aplicación:

```
java NestedSP basedatos IDusuario contraseña
```

Tareas relacionadas:

- “Creación de procedimientos SQL” en la página 147
- “Llamada a procedimientos desde el Procesador de línea de mandatos (CLP)” en la página 144
- “Revinculación de procedimientos SQL” en la página 152

Ejemplos relacionados:

- “basecase.sqc -- To call the UPDATE_SALARY SQL procedure”
- “NestedSP.java -- Client application for invoking nested stored procedures ”
- “rsultset.c -- To call the MEDIAN_RESULT_SET SQL procedure”

Personalización de las opciones de precompilación y vinculación para procedimientos SQL

Procedimiento:

Las opciones de precompilación y vinculación para procedimientos SQL se pueden personalizar definiendo la variable DB2_SQLROUTINE_PREPOPTS del registro de DB2 mediante el mandato siguiente:

```
db2set DB2_SQLROUTINE_PREPOPTS=<opciones>
```

Sólo están permitidas las opciones siguientes:

```
BLOCKING {UNAMBIG | ALL | NO}
DATETIME {DEF | USA | EUR | ISO | JIS | LOC}
DEGREE {1 | grado-de-paralelismo | ANY}
DYNAMICRULES {BIND | RUN}
EXPLAIN {NO | YES | ALL}
EXPLSNAP {NO | YES | ALL}
FEDERATED {NO | YES}
INSERT {DEF | BUF}
ISOLATION {CS |RR |UR |RS |NC}
QUERYOPT nivel-optimización
VALIDATE {RUN | BIND}
```

Estas opciones se pueden cambiar a nivel de procedimiento mediante el procedimiento almacenado SET_ROUTINE_OPTS. Los valores de las opciones establecidos para crear procedimientos SQL en la sesión actual se pueden obtener mediante la función GET_ROUTINE_OPTS.

Ejemplo.

Los procedimientos SQL utilizados en este ejemplo se definirán en scripts del CLP (que se proporcionan a continuación). Estos scripts no están en el directorio de ejemplos sqlproc, pero los puede crear fácilmente cortando y pegando las sentencias de procedimiento CREATE en sus propios archivos.

En los ejemplos se utiliza una tabla denominada “expenses”, que puede crear en la base de datos de ejemplo del modo siguiente:

```
db2 connect to sample
db2 CREATE TABLE expenses(amount DOUBLE, date DATE)
db2 connect reset
```

Para empezar, especificaremos el uso del formato ISO para las fechas como valor para toda la instancia:

```
db2set DB2_SQLROUTINE_PREPOPTS="DATETIME ISO"  
db2stop  
db2start
```

Es necesario detener y rearrancar DB2 para que el cambio sea efectivo.

A continuación, conectaremos con la base de datos:

```
db2 connect to sample
```

El primer procedimiento se define en el script del CLP maxamount.db2 del modo siguiente:

```
CREATE PROCEDURE maxamount(OUT maxamnt DOUBLE)  
BEGIN  
    SELECT max(amount) INTO maxamnt FROM expenses;  
END @
```

Se creará con las opciones DATETIME ISO e ISOLATION UR:

```
db2 "CALL SET_ROUTINE_OPTS(GET_ROUTINE_OPTS() || ' ISOLATION UR')"  
db2 -td@ -vf maxamount.db2
```

El siguiente procedimiento se define en el script del CLP fullamount.db2 del modo siguiente:

```
CREATE PROCEDURE fullamount(OUT fullamnt DOUBLE)  
BEGIN  
    SELECT sum(amount) INTO fullamnt FROM expenses;  
END @
```

Se creará con la opción ISOLATION CS (observe que en este caso no estamos utilizando el valor DATETIME ISO para toda la instancia):

```
CALL SET_ROUTINE_OPTS('ISOLATION CS')  
db2 -td@ -vf fullamount.db2
```

El último procedimiento de nuestro ejemplo se define en el script del CLP perday.db2 del modo siguiente:

```
CREATE PROCEDURE perday()  
BEGIN  
    DECLARE cur1 CURSOR WITH RETURN FOR  
        SELECT date, sum(amount)  
        FROM expenses  
        GROUP BY date;  
  
    OPEN cur1;  
END @
```

La última llamada a SET_ROUTINE_OPTS utiliza el valor NULL como argumento. Así se restaura el valor global especificado en el registro DB2_SQLROUTINE_PREPOPTS, de forma que el último procedimiento se creará con la opción DATETIME ISO:

```
CALL SET_ROUTINE_OPTS(NULL)  
db2 -td@ -vf perday.db2
```

Tareas relacionadas:

- “Copia de seguridad y restauración de procedimientos SQL creados antes de DB2 8.2” en la página 151
- “Creación de procedimientos SQL” en la página 147

- “Llamada a procedimientos desde el Procesador de línea de mandatos (CLP)” en la página 144
- “Invocación de procedimientos SQL mediante aplicaciones cliente” en la página 148
- “Revinculación de procedimientos SQL” en la página 152

Información relacionada:

- “Mandato PRECOMPILE” en el manual *Consulta de mandatos*

Copia de seguridad y restauración de procedimientos SQL creados antes de DB2 8.2

En DB2 versión 8.2, los procedimientos SQL son objetos de base de datos de primera clase, en el sentido en que se gestionan completamente dentro de la base de datos, de forma parecida a los activadores y las vistas. Por consiguiente, no necesitan ninguna consideración especial durante las operaciones de copia de seguridad y restauración. Los procedimientos creados antes de la versión 8.2 requiere cierta consideración especial, que se explica más adelante, porque implican la creación de DLL (bibliotecas enlazadas dinámicamente) en el sistema de archivos. Cuando se crea un procedimiento SQL en una versión de DB2 anterior a V8.2, la biblioteca enlazada dinámicamente (DLL) generada se conserva en el catálogo de la base de datos. Por tanto, toda esta información se salva cuando copia una base de datos.

Procedimiento:

Durante la recuperación de una base de datos, se eliminan todos los ejecutables de procedimiento SQL contenidos en el sistema de archivos que pertenecen a la base de datos que se está recuperando. Si el parámetro de configuración para la creación de índices, *indexrec*, tiene el valor *RESTART*, todos los ejecutables de procedimientos SQL se extraerán de la tabla de catálogo y se devolverán al sistema de archivos durante la próxima conexión. En otro caso, los ejecutables de SQL se extraerán cuando se ejecuten por primera vez los procedimientos SQL.

Los ejecutables volverán a colocarse en el directorio siguiente:

```
UNIX $HOME/sqllib/function/routine/sqlproc/<nombre_base_datos>
```

Windows

```
sqllib\function\rutina\sqlproc\<nombre_base_datos>
```

donde *<nombre_base_datos>* representa la base de datos con la que se crearon los procedimientos SQL.

Si al intentar conectar por primera vez con una base de datos después de una operación de restauración se recibe este mensaje:

```
SQL2048N Se produjo un error al acceder al objeto "ARCHIVOS DE PROCEDIMIENTO DE SQL".
Código de razón: "7".
```

Detenga DB2 mediante *db2stop* y arranque con *db2start*.

Tareas relacionadas:

- “Personalización de las opciones de precompilación y vinculación para procedimientos SQL” en la página 149
- “Creación de procedimientos SQL” en la página 147

- “Llamada a procedimientos desde el Procesador de línea de mandatos (CLP)” en la página 144
- “Invocación de procedimientos SQL mediante aplicaciones cliente” en la página 148
- “Revinculación de procedimientos SQL” en la página 152

Revinculación de procedimientos SQL

Procedimiento:

Para volver a vincular el paquete correspondiente a un procedimiento SQL, llame al procedimiento almacenado interno `SYSPROC.REBIND_ROUTINE_PACKAGE`.

Por ejemplo, si la base de datos contiene un procedimiento SQL llamado `MYSHEMA.MYPROC`, su paquete se puede vincular de nuevo emitiendo el mandato siguiente desde el procesador de línea de mandatos (CLP):

```
CALL SYSPROC.REBIND_ROUTINE_PACKAGE('P', 'MYSHEMA.MYPROC', 'CONSERVATIVE')
```

donde 'P' indica que 'MYSHEMA.MYPROC' es un nombre de procedimiento. Si el valor del primer parámetro fuera 'SP', esto indicaría que 'MYSHEMA.MYPROC' es un nombre de procedimiento específico. 'CONSERVATIVE' indica que se aplicarán semánticas conservadoras para la revinculación. Consulte los enlaces proporcionados más abajo sobre el mandato REBIND para conocer más detalles sobre la revinculación conservadora.

Tareas relacionadas:

- “Personalización de las opciones de precompilación y vinculación para procedimientos SQL” en la página 149
- “Copia de seguridad y restauración de procedimientos SQL creados antes de DB2 8.2” en la página 151
- “Creación de procedimientos SQL” en la página 147
- “Llamada a procedimientos desde el Procesador de línea de mandatos (CLP)” en la página 144
- “Invocación de procedimientos SQL mediante aplicaciones cliente” en la página 148

Información relacionada:

- “Mandato REBIND” en el manual *Consulta de mandatos*

Capítulo 7. Perl

Creación de aplicaciones Perl 153

Este capítulo proporciona información detallada para crear programas Perl para acceder a una base de datos DB2.

Para conocer la información más reciente sobre el desarrollo de aplicaciones DB2, consulte la página Web situada en:

<http://www.ibm.com/software/data/db2/udb/ad>

Creación de aplicaciones Perl

DB2 da soporte al acceso a bases de datos para aplicaciones cliente escritas en Perl. En el momento de imprimir el presente manual, el Release 0.76 del controlador de DB2 UDB (DBD::DB2) correspondiente a Perl Database Interface (Perl DBI), Versión 0.93 o posterior, está disponible para AIX, HP-UX, Linux, Solaris y Windows. Para tener información sobre cómo obtener el controlador más reciente, visite:

<http://www.ibm.com/software/data/db2/perl>

DB2 proporciona programas de ejemplo en Perl, ubicados en el directorio `sqllib/samples/perl` para UNIX y en el directorio `sqllib\samples\perl` para Windows.

Configuración para entornos de 64 bits en UNIX

En UNIX, si no se está utilizando una versión de 64 bits de Perl, se debe seguir el mismo proceso recomendado para ejecutar aplicaciones de 32 bits en entornos de 64 bits, descrito en el apartado “Migración de aplicaciones desde entornos de 32 bits a entornos de 64 bits” en la página 56, a excepción de que las aplicaciones no se tienen que revincular. Si se utiliza un programa reiniciador, deberá sustituir `perl $1` por `$1`.

Nota: En los ejemplos siguientes, si se utiliza un programa reiniciador para entornos UNIX de 64 bits, sustituya `perl <nombre_programa>`, por `<nombre_reiniciador> <nombre_programa>`.

Procedimiento:

Para ejecutar el intérprete de Perl sobre un programa Perl de DB2 desde la línea de mandatos, entre el nombre del intérprete y el nombre del programa (incluyendo la extensión):

- Si está conectando localmente en el servidor:
`perl dbauth.pl`
- Si está conectando desde un cliente remoto:
`perl dbauth.pl sample <IDusuario> <contraseña>`

Algunos programas requieren archivos de soporte para que se puedan ejecutar. El programa de ejemplo `tbse1` requiere varias tablas creadas por el script `tbse1create.db2` del CLP. El script `tbse1init` (UNIX), o el archivo de proceso por lotes `tbse1init.bat` (Windows), en primer lugar llama a `tbse1drop.db2` para

eliminar las tablas, en caso de que existan, y luego llama a `tbselcreate.db2` para crearlas. Por lo tanto, para ejecutar el programa deberá entrar los mandatos siguientes:

- Si está conectando localmente en el servidor:

```
tbselinit
perl tbsel.pl
```

- Si está conectando desde un cliente remoto:

```
tbselinit
perl tbsel.pl sample <IDusuario> <contraseña>
```

Nota: Para un cliente remoto es necesario modificar la sentencia de conexión en los archivos `tbselinit` o `tbselinit.bat` para codificar el ID de usuario y la contraseña: `db2 connect to sample user <IDusuario> using <contraseña>`

Llamada a rutinas

Las aplicaciones cliente DB2 pueden acceder a rutinas (procedimientos almacenados y funciones definidas por el usuario) creadas por lenguajes del sistema principal soportados o por procedimientos de SQL. Por ejemplo, el programa de ejemplo `spclient.pl` puede acceder a la biblioteca compartida de los procedimientos de SQL `spserver`, en caso de que exista en la base de datos.

Nota: Para crear una rutina de lenguaje del sistema principal debe tener configurado el compilador apropiado en el servidor. Los procedimientos de SQL no necesitan compilador. La biblioteca compartida sólo se puede crear en el servidor, y no desde un cliente remoto.

Para mostrar procedimientos de SQL de llamada, vaya al directorio `samples/sqlproc` (UNIX) o al directorio `samples\sqlproc` (Windows) del servidor, y ejecute los mandatos siguientes para crear y catalogar los procedimientos de SQL en la biblioteca `spserver`:

```
db2 connect to sample
db2 -td@ -vf spserver.db2
```

A continuación, regrese al directorio de ejemplos `perl` (esto se puede hacer desde una máquina cliente remota) y ejecute el intérprete de Perl en el programa cliente para acceder a la biblioteca compartida `spserver`:

- Si está conectando localmente en el servidor, entre lo siguiente:

```
perl spclient
```

- Si está conectando desde un cliente remoto, entre lo siguiente:

```
perl spclient sample <IDusuario> <contraseña>
```

Conceptos relacionados:

- “Consideraciones sobre la programación en Perl” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente*
- “DBI Perl” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente*

Tareas relacionadas:

- “Migración de aplicaciones desde entornos de 32 bits a entornos de 64 bits” en la página 56

Información relacionada:

- “Ejemplos de Perl” en la página 97

Capítulo 8. PHP

Creación de aplicaciones PHP 155

Este capítulo proporciona información detallada para crear programas PHP para acceder a una base de datos DB2.

Para conocer la información más reciente sobre el desarrollo de aplicaciones DB2, consulte la página Web situada en:

<http://www.ibm.com/software/data/db2/udb/ad>

Creación de aplicaciones PHP

DB2 da soporte al acceso a bases de datos para aplicaciones cliente escritas en PHP. PHP es un lenguaje multiplataforma de programación que se ejecuta en el servidor y contiene HTML. Permite acceder a DB2 utilizando el método de acceso ODBC Unificado, en el cual el PHP a nivel de usuario se comunica con DB2 mediante llamadas a ODBC. A diferencia del ODBC estándar, en el método ODBC Unificado, la comunicación se realiza directamente con la capa CLI de DB2, no a través de la capa ODBC. Para obtener más información sobre el uso de PHP con DB2, consulte el sitio Web de soporte de DB2:

<http://www.ibm.com/software/data/db2/udb/winos2unix/support>

En el momento de imprimir esta publicación, la última versión es PHP 4.3.4. Puede bajar la versión más reciente de PHP desde:

<http://www.php.net>

Después de bajar el archivo tar, compílelo con la opción `--with-ibm-db2=<DIR>`, donde `<DIR>` lo determina la vía de acceso de la plataforma:

- En UNIX, la opción sería: `--with-ibm-db2=$HOME/sql1lib`
- En Windows, la opción sería `--with-ibm-db2=%DB2PATH%`

Después de descomprimir el archivo tar `php-4.3.4.tar` (o el archivo de una versión posterior de PHP, si está disponible), el directorio `php-4.3.4` contiene el archivo `php.ini-dist` que tiene los valores por omisión para las nuevas instalaciones de PHP. Este archivo se tiene que copiar en la vía de acceso de instalación cambiando su nombre por `php.ini`. Los mandatos siguientes asumen una vía de acceso de instalación típica:

- En UNIX:

```
cd ../php-4.x.y
cp php.ini-dist /usr/local/lib/php.ini
```

- En Windows:

```
cd ..\php-4.x.y
copy php.ini-dist C:\Windows\php.ini
```

Nota: Se puede especificar una vía de acceso de instalación distinta durante la instalación utilizando la opción siguiente:

`--with-config-file-path=<vía>`

Si se realiza la instalación utilizando RPM en Linux, la vía de acceso por omisión para php.ini es /etc. Para RPM en Linux, cree un nuevo archivo, .odbc.ini, en el directorio inicial e incluya la información siguiente para configurar ODBC con DB2:

```
[Fuentes de datos ODBC]
Sample          = <descripción>

[Sample]
Driver          = $HOME/sql/lib/lib/libdb2.so
Description     = <descripción>
Host            = localhost
UserName        = <usuario>
Password        = <contraseña>
Database        = sample
```

Si realiza la instalación utilizando InstallShield en Windows, añada el controlador IBM DB2 ODBC y configúrelo para la base de datos de ejemplo utilizando Fuentes de datos (ODBC) en las herramientas de administración.

Para obtener más información, consulte el archivo INSTALL.

Puede editar el archivo php.ini para establecer las opciones de PHP. DB2 le recomienda establecer las opciones siguientes para ejecutar los programas de ejemplo de DB2:

```
track_errors = 0n
register_globals = 0n
register_argc_argv = 0n
max_execution_time = 60
odbc.defaultlrl = 100000
```

Notas:

1. track_errors = 0n: permite hacer un seguimiento de los errores.
2. register_globals = 0n: si register_globals = Off, se deben utilizar \$_SERVER['argc'] y \$_SERVER['argv'][0], respectivamente, en lugar de \$argc y \$argv.
3. register_argc_argv = 0n: admite argumentos de línea de mandatos.
4. max_execution_time = 60: algunos scripts pueden tardar más en ejecutarse por lo que es necesario cambiar el valor por omisión.
5. odbc.defaultlrl = 100000: es, en bytes, suficientemente grande para contener los datos BLOB de los ejemplos.

DB2 proporciona programas de ejemplo en PHP, ubicados en el directorio sql/lib/samples/php para UNIX y en la carpeta sql/lib/samples/php para Windows.

Configuración para entornos de 64 bits en UNIX

En UNIX, si no se está utilizando una versión de 64 bits de PHP, se debe seguir el mismo proceso recomendado para ejecutar aplicaciones de 32 bits en entornos de 64 bits, descrito en el apartado “Migración de aplicaciones desde entornos de 32 bits a entornos de 64 bits” en la página 56, a excepción de que las aplicaciones no se tienen que revincular. Si se utiliza un programa reiniciador, deberá sustituir php \$1 por \$1.

Nota: En los ejemplos siguientes, si se utiliza un programa reiniciador para entornos UNIX de 64 bits, sustituya php <nombre_programa>, por <nombre_reiniciador> <nombre_programa>.

Procedimiento:

Para ejecutar el intérprete de Php sobre un programa PHP de DB2 desde la línea de mandatos, entre el nombre del intérprete y el nombre del archivo fuente (incluyendo la extensión):

- Si está conectando localmente en el servidor:

```
php dbauth.php
```
- Si está conectando desde un cliente remoto:

```
php dbauth.php sample <IDusuario> <contraseña>
```

Algunos programa requieren archivos de soporte para que se puedan ejecutar. El programa de ejemplo `tbsel` requiere varias tablas creadas por el script `tbselcreate.db2` del CLP. El script `tbselinit`, en UNIX, o el archivo de proceso por lotes `tbselinit.bat`, en Windows, en primer lugar llama a `tbseldrop.db2` para eliminar las tablas, en caso de que existan, y luego llama a `tbselcreate.db2` para crearlas. Por lo tanto, para ejecutar el programa deberá entrar los mandatos siguientes:

- Si está conectando localmente en el servidor:

```
tbselinit  
php tbsel.php
```
- Si está conectando desde un cliente remoto:

```
tbselinit  
php tbsel.php sample <IDusuario> <contraseña>
```

Nota: Para un cliente remoto es necesario modificar la sentencia de conexión en los archivos `tbselinit` o `tbselinit.bat` para codificar el ID de usuario y la contraseña: `db2 connect to sample user <IDusuario> using <contraseña>`

Llamada a funciones definidas por el usuario

Las aplicaciones cliente DB2 pueden acceder a funciones definidas por el usuario creadas por lenguajes del sistema principal soportados. Por ejemplo, el programa de ejemplo `udfcli.php` puede acceder a la biblioteca compartida de funciones C definidas por el usuario `udfsrv`, en caso de que exista en la base de datos.

Nota: Para crear una biblioteca compartida de funciones definidas por el usuario del lenguaje del sistema principal, debe tener configurado el compilador apropiado en el servidor. La biblioteca compartida sólo se puede crear en el servidor, y no desde un cliente remoto. PHP no soporta que los programas cliente llamen a procedimientos almacenados.

Suponiendo que hay un compilador C configurado en el servidor, demuestre la llamada a funciones definidas por el usuario yendo al directorio `samples/c` (UNIX) o al directorio `samples\c` (Windows) del servidor, y ejecute el mandato siguiente para crear la biblioteca `udfsrv` en la base de datos:

```
bldrtn udfsrv
```

A continuación, regrese al directorio de ejemplos php (esto se puede hacer en una máquina cliente remota) y ejecute el intérprete de Php en el programa cliente para acceder a la biblioteca compartida `udfsrv`:

- Si está conectando localmente en el servidor, entre lo siguiente:

```
php udfcli.php
```
- Si está conectando desde un cliente remoto, entre lo siguiente:

```
php udfcli.php sample <IDusuario> <contraseña>
```

| **Tareas relacionadas:**

- | • “Migración de aplicaciones desde entornos de 32 bits a entornos de 64 bits” en
| la página 56

| **Información relacionada:**

- | • “Ejemplos de PHP” en la página 98

Parte 3. Creación y ejecución de aplicaciones específicas de una plataforma

Capítulo 9. UNIX

Creación de aplicaciones C de UNIX	161	Creación de rutinas C++ de UNIX	172
Creación de aplicaciones C de conexión múltiple de UNIX	163	Creación de aplicaciones Micro Focus COBOL de UNIX	176
Creación de rutinas C de UNIX	165	Creación de rutinas Micro Focus COBOL de UNIX	177
Creación de aplicaciones C++ de UNIX.	168		
Creación de aplicaciones C++ de conexión múltiple de UNIX	170		

En este capítulo se describen los pasos comunes para crear aplicaciones y rutinas para los sistemas operativos UNIX soportados. Los detalles específicos de la plataforma, como por ejemplo las opciones del compilador, se proporcionan en los capítulos sobre plataformas que siguen a éste.

Los capítulos sobre plataformas contienen información para crear aplicaciones multihebra C/C++, puesto que cada sistema operativo tiene requisitos de creación específicos para estos programas.

Creación de aplicaciones C de UNIX

DB2 proporciona scripts de creación para compilar y enlazar programas C de SQL incorporado y de la API de DB2. Estos scripts residen en el directorio `sqllib/samples/c`, junto con programas de ejemplo que se pueden crear a partir de esos archivos.

El archivo de creación, `bldapp` contiene los mandatos para crear un programa de aplicación DB2.

El primer parámetro, `$1`, especifica el nombre del archivo fuente. Este parámetro obligatorio es el único necesario para los programas DB2 que no contienen SQL incorporado. Para crear programas de SQL incorporado es necesaria una conexión con la base de datos, por lo que también se proporcionan otros tres parámetros opcionales: el segundo parámetro, `$2`, especifica el nombre de la base de datos con la que se desea conectar; el tercer parámetro, `$3`, especifica el ID de usuario correspondiente a la base de datos, y `$4` especifica la contraseña.

Para un programa de SQL incorporado, `bldapp` pasa los parámetros al script de precompilación y vinculación, `embprep`. Si no se proporciona un nombre de base de datos, se utiliza la base de datos por omisión `sample`. Los parámetros de ID de usuario y contraseña sólo son necesarios si la instancia donde se crea el programa es diferente de la instancia donde reside la base de datos.

Procedimiento:

Los ejemplos siguientes muestran cómo crear y ejecutar programas de la API de DB2 y aplicaciones de SQL incorporado.

Para crear el programa de ejemplo de SQL no incorporado de la API de DB2, `cli_info`, a partir del archivo fuente `cli_info.c`, especifique:

```
bldapp cli_info
```

El resultado es un archivo ejecutable, `cli_info`.

Para ejecutar el archivo ejecutable, escriba el nombre del ejecutable:

```
cli_info
```

Creación y ejecución de aplicaciones de SQL incorporado

A partir del archivo fuente `tbmod.sql`, puede crear la aplicación de SQL incorporado, `tbmod`, de tres maneras:

1. Si conecta con la base de datos "sample" en la misma instancia, especifique:

```
bldapp tbmod
```

2. Si conecta con otra base de datos en la misma instancia, especifique también el nombre de la base de datos:

```
bldapp tbmod basedatos
```

3. Si conecta con una base de datos de otra instancia, especifique también el ID de usuario y la contraseña de la instancia de base de datos:

```
bldapp tbmod basedatos IDusuario contraseña
```

El resultado es un archivo ejecutable, `tbmod`.

Existen tres maneras de ejecutar esta aplicación de SQL incorporado:

1. Si desea acceder a la base de datos `sample` en la misma instancia, especifique simplemente el nombre del ejecutable:

```
tbmod
```

2. Si desea acceder a otra base de datos en la misma instancia, especifique el nombre del ejecutable y el nombre de la base de datos:

```
tbmod basedatos
```

3. Si desea acceder a una base de datos de otra instancia, especifique el nombre del ejecutable, el nombre de la base de datos, y el ID de usuario y la contraseña de la instancia de base de datos:

```
tbmod basedatos IDusuario contraseña
```

Conceptos relacionados:

- "Archivos de creación" en la página 110

Tareas relacionadas:

- "Creación de rutinas C de UNIX" en la página 165

Información relacionada:

- "Opciones de compilación y enlace para aplicaciones C de AIX" en la página 185
- "Opciones de compilación y enlace para aplicaciones C de HP-UX" en la página 212
- "Opciones de compilación y enlace para aplicaciones C de Linux" en la página 228
- "Opciones de compilación y enlace para aplicaciones C de Solaris" en la página 244

Ejemplos relacionados:

- "bldapp -- Builds AIX C application programs (C)"
- "bldapp -- Builds HP-UX C applications (C)"
- "bldapp -- Builds Linux C applications (C)"
- "bldapp -- Builds Solaris C applications (C)"
- "cli_info.c -- Set and get information at the client level (C)"

- "embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs (C)"
- "tbmod.sqc -- How to modify table data (C)"

Creación de aplicaciones C de conexión múltiple de UNIX

DB2 proporciona scripts de creación para compilar y enlazar programas C de SQL incorporado y de la API de DB2. Estos scripts residen en el directorio `sql11ib/samples/c`, junto con programas de ejemplo que se pueden crear a partir de esos archivos.

El archivo de creación, `blmnc`, contiene los mandatos para crear un programa de conexión múltiple de DB2, que requiere dos bases de datos. Las opciones de compilación y enlace son las mismas que las utilizadas en `bldapp`.

El primer parámetro, `$1`, especifica el nombre del archivo fuente. El segundo parámetro, `$2`, especifica el nombre de la primera base de datos a la que desea conectarse. El tercer parámetro, `$3`, especifica la segunda base de datos a la que desea conectarse. Todos estos parámetros son obligatorios.

Nota: El makefile codifica los valores por omisión "sample" y "sample2" como nombres de la base de datos (`$2` y `$3`, respectivamente) por lo que, si está utilizando el makefile y acepta estos valores, sólo tendrá que especificar el nombre del programa (parámetro `$1`). Si utiliza el script `blmnc` debe especificar los tres parámetros.

No se requieren parámetros opcionales para una conexión local, pero sí para conectar con un servidor desde un cliente remoto. Estos parámetros son: `$4` y `$5` para especificar, respectivamente, el ID de usuario y la contraseña para la primera base de datos; y `$6` y `$7` para especificar, respectivamente, el ID de usuario y la contraseña para la segunda base de datos.

Procedimiento:

Para el programa de ejemplo de conexión múltiple, `dbmcon`, se requieren dos bases de datos. Si aún no se ha creado la base de datos `sample`, la puede crear entrando `db2sample` en la línea de mandatos. La segunda base de datos, aquí llamada `sample2`, se puede crear mediante uno de los mandatos siguientes:

Si crea la base de datos localmente:

```
db2 create db sample2
```

Si crea la base de datos remotamente:

```
db2 attach to <nombre_nodo>
db2 create db sample2
db2 detach
db2 catalog db sample2 as sample2 at node <nombre_nodo>
```

donde `<nombre_nodo>` es el nodo en que reside la base de datos.

Una conexión múltiple también requiere que se esté ejecutando el receptor TCP/IP. Para asegurarse que es así, haga lo siguiente:

1. Establezca la variable de entorno `DB2COMM` en TCP/IP de la forma siguiente:

```
db2set DB2COMM=TCPIP
```

- |
- | 2. Actualice el archivo de configuración del gestor de bases de datos con el
- | nombre del servicio de TCP/IP que se haya especificado en el archivo de
- | servicios:

| `db2 update dbm cfg using SVCENAME <nombre del servicio de TCP/IP>`

| Cada instancia tiene un nombre de servicio TCP/IP listado en el archivo de

| servicios. Si no puede localizarlo o no tiene el permiso de archivo para cambiar

| el archivo de servicios, pida ayuda al administrador.

- | 3. Detenga y reinicie el gestor de bases de datos para que estos cambios entren en
- | vigor:

| `db2stop`

| `db2start`

| El programa `dbmcon` consta de cinco archivos:

| **dbmcon.sqc**

| Archivo fuente principal para conectar con ambas bases de datos.

| **dbmcon1.sqc**

| Archivo fuente para crear un paquete vinculado a la primera base de

| datos.

| **dbmcon1.h**

| Archivo de cabecera para `dbmcon1.sqc`, incluido en `dbmcon.sqc`, para

| acceder a las sentencias de SQL para crear y eliminar una tabla que se

| debe vincular a la primera base de datos.

| **dbmcon2.sqc**

| Archivo fuente para crear un paquete vinculado a la segunda base de

| datos.

| **dbmcon2.h**

| Archivo de cabecera para `dbmcon2.sqc`, incluido en `dbmcon.sqc`, para

| acceder a las sentencias de SQL para crear y eliminar una tabla que se

| debe vincular a la segunda base de datos.

| Para crear el programa de ejemplo de conexión múltiple, `dbmcon`, se entre lo

| siguiente:

| `blc dbmcon sample sample2`

| El resultado es un archivo ejecutable, `dbmcon`.

| Para ejecutar el archivo ejecutable, escriba el nombre del ejecutable:

| `dbmcon`

| El programa muestra una confirmación en una fase para dos bases de datos.

| **Conceptos relacionados:**

- | • “Archivos de creación” en la página 110

| **Tareas relacionadas:**

- | • “Creación de aplicaciones C de UNIX” en la página 161

| **Información relacionada:**

- | • “svcename - TCP/IP service name configuration parameter” en el manual
- | *Administration Guide: Performance*
- | • “Opciones de compilación y enlace para aplicaciones C de AIX” en la página 185

- “Opciones de compilación y enlace para aplicaciones C de HP-UX” en la página 212
- “Opciones de compilación y enlace para aplicaciones C de Linux” en la página 228
- “Opciones de compilación y enlace para aplicaciones C de Solaris” en la página 244

Ejemplos relacionados:

- “bldmc -- Builds AIX C multi-connection applications (C)”
- “bldmc -- Builds HP-UX C multi-connection applications (C)”
- “bldmc -- Builds Linux C multi-connection applications (C)”
- “bldmc -- Builds Solaris C multi-connection applications (C)”
- “dbmcon.sqc -- How to use multiple databases (C)”
- “dbmcon1.h -- Function declarations for the source file, dbmcon1.sqc (C)”
- “dbmcon1.sqc -- Functions used in the multiple databases program dbmcon.sqc (C)”
- “dbmcon2.h -- Function declarations for the source file, dbmcon2.sqc (C)”
- “dbmcon2.sqc -- Functions used in the multiple databases program dbmcon.sqc (C)”

Creación de rutinas C de UNIX

DB2 proporciona scripts de creación para compilar y enlazar programas C. Estos scripts residen en el directorio `sqllib/samples/c`, junto con programas de ejemplo que se pueden crear a partir de esos archivos.

El script `bldrtn` contiene los mandatos para crear rutinas (procedimientos almacenados y funciones definidas por el usuario). El script compila la rutina y crea una biblioteca compartida que puede ser cargada por el gestor de bases de datos y llamada por una aplicación cliente.

El primer parámetro, `$1`, especifica el nombre del archivo fuente. El segundo parámetro, `$2`, especifica el nombre de la base de datos a la que desea conectarse.

El parámetro correspondiente a la base de datos es opcional. Si no se proporciona un nombre de base de datos, el programa utiliza la base de datos por omisión `sample`. Debido a que el procedimiento almacenado se debe crear en la misma instancia donde reside la base de datos, no hay parámetros para el ID de usuario ni la contraseña.

Procedimiento:

Los ejemplos siguientes muestran cómo crear bibliotecas compartidas de rutinas mediante:

- procedimientos almacenados
- funciones definidas por el usuario (UDF) de SQL no incorporado
- funciones definidas por el usuario (UDF) de SQL incorporado

Biblioteca compartida de procedimientos almacenados

Para crear el programa de ejemplo `spserver` a partir del archivo fuente `spserver.sqc`:

- |
- | 1. Si conecta con la base de datos `sample`, especifique el nombre del script de
- | creación y el nombre del programa:
- |

| `bldrtn spserver`

| Si conecta con otra base de datos, especifique también el nombre de la base de

| datos:

| `bldrtn spserver basedatos`

| El script copia la biblioteca compartida en el directorio `sqllib/function` del

| servidor.

- | 2. A continuación, catalogue las rutinas ejecutando el script `spcat` en el servidor:
- |

| `spcat`

| Este script conecta con la base de datos "sample", descataloga mediante

| `spdrop.db2` las rutinas que se hubieran catalogado previamente, luego las

| cataloga llamando a `spcreate.db2`, y finalmente desconecta de la base de datos.

| Puede también ejecutar los scripts `spdrop.db2` y `spcreate.db2` por separado.

- | 3. A continuación, si no es la primera vez que se crea el procedimiento
- | almacenado, detenga y reinicie la base de datos para asegurarse que se
- | reconoce la nueva versión de la biblioteca compartida. Lo puede hacer entrando
- | en la línea de mandatos `db2stop` seguido de `db2start`.

| Una vez creada la biblioteca compartida, `spserver`, puede crear la aplicación

| cliente, `spclient`, la cual accede a la biblioteca compartida.

| Puede crear `spclient` utilizando el archivo de script, `bldapp`.

| Para invocar los procedimientos almacenados contenidos en la biblioteca

| compartida, ejecute la aplicación cliente de ejemplo, especificando lo siguiente:

| `spclient basedatos IDusuario contraseña`

| donde

| **basedatos**

| Es el nombre de la base de datos a la que desea conectarse. El nombre

| podría ser `sample`, o su alias, u otro nombre de base de datos.

| **IDusuario**

| Es un ID de usuario válido.

| **contraseña**

| Es una contraseña válida correspondiente al ID de usuario.

| La aplicación cliente accede a la biblioteca compartida, `spserver`, y ejecuta diversas

| funciones de procedimiento almacenado en la base de datos del servidor. Los datos

| resultantes se devuelven a la aplicación cliente.

| **Biblioteca compartida de UDF de SQL no incorporado**

| Para crear el programa de función definida por el usuario `udfsrv`, a partir del

| archivo fuente `udfsrv.c`, especifique el nombre del script de creación y el nombre

| del programa:

| `bldrtn udfsrv`

| El script copia la UDF en el directorio `sqllib/function`.

Una vez creado el programa `udfsrv`, puede crear la aplicación cliente, `udfcli`, que llama al programa. El programa se proporciona en versiones para CLI de DB2 y para SQL incorporado. Puede crear el programa cliente `udfcli` de la CLI de DB2 a partir del archivo fuente `udfcli.c`, contenido en `sqllib/samples/cli`, utilizando el script `bldapp`.

Puede crear el programa cliente `udfcli` de SQL incorporado a partir del archivo fuente `udfcli.sqc`, contenido en `sqllib/samples/c`, utilizando el script `bldapp`.

Para invocar las UDF contenidas en la biblioteca compartida, ejecute la aplicación cliente, especificando lo siguiente:

```
udfcli basedatos IDusuario contraseña
```

donde

basedatos

Es el nombre de la base de datos a la que desea conectarse. El nombre podría ser `sample`, o su alias, u otro nombre de base de datos.

IDusuario

Es un ID de usuario válido.

contraseña

Es una contraseña válida correspondiente al ID de usuario.

La aplicación cliente accede a la biblioteca compartida, `udfsrv`, y ejecuta las funciones definidas por el usuario contenidas en la base de datos del servidor. Los datos resultantes se devuelven a la aplicación cliente.

Biblioteca compartida de UDF de SQL incorporado

Puede crear el programa de función definida por el usuario de SQL incorporado, `udfemsrv`, a partir del archivo fuente `udfemsrv.sqc`. Si conecta con la base de datos `sample`, especifique el nombre del script de creación y el nombre del programa:

```
bldrtn udfemsrv
```

Si conecta con otra base de datos, especifique también el nombre de la base de datos:

```
bldrtn udfemsrv basedatos
```

El script copia la UDF en el directorio `sqllib/function`.

Una vez creado el programa `udfemsrv`, puede crear la aplicación cliente, `udfemcli`, que llama al programa. Puede crear el programa cliente `udfemcli` a partir del archivo fuente `udfemcli.sqc`, contenido en `sqllib/samples/c`, utilizando el script `bldapp`.

Para invocar las UDF contenidas en la biblioteca compartida, ejecute la aplicación cliente, especificando lo siguiente:

```
udfemcli basedatos IDusuario contraseña
```

donde

basedatos

Es el nombre de la base de datos a la que desea conectarse. El nombre podría ser `sample`, o su alias, u otro nombre de base de datos.

IDusuario

Es un ID de usuario válido.

contraseña

Es una contraseña válida correspondiente al ID de usuario.

La aplicación cliente accede a la biblioteca compartida, `udfemsrv`, y ejecuta las funciones definidas por el usuario contenidas en la base de datos del servidor. Los datos resultantes se devuelven a la aplicación cliente.

Conceptos relacionados:

- “Archivos de creación” en la página 110

Tareas relacionadas:

- “Creación de aplicaciones C de UNIX” en la página 161

Información relacionada:

- “Opciones de compilación y enlace para rutinas C de AIX” en la página 186
- “Opciones de compilación y enlace para rutinas C de HP-UX” en la página 215
- “Opciones de compilación y enlace para rutinas C de Linux” en la página 230
- “Opciones de compilación y enlace para rutinas C de Solaris” en la página 246

Ejemplos relacionados:

- “`bldrtn -- Builds AIX C routines (stored procedures and UDFs) (C)`”
- “`bldrtn -- Builds HP-UX C routines (stored procedures and UDFs) (C)`”
- “`bldrtn -- Builds Linux C routines (stored procedures or UDFs) (C)`”
- “`bldrtn -- Builds Solaris C routines (stored procedures or UDFs) (C)`”
- “`embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs (C)`”
- “`spclient.sqc -- Call various stored procedures (C)`”
- “`spserver.sqc -- Definition of various types of stored procedures (C)`”
- “`udfcli.sqc -- Call a variety of types of user-defined functions (C)`”
- “`udfemcli.sqc -- Call a variety of types of embedded SQL user-defined functions. (C)`”
- “`udfemsrv.sqc -- Call a variety of types of embedded SQL user-defined functions. (C)`”
- “`udfsrv.c -- Defines a variety of types of user-defined functions (C)`”

Creación de aplicaciones C++ de UNIX

DB2 proporciona scripts de creación para compilar y enlazar programas C++ de SQL incorporado y de la API de DB2. Estos scripts residen en el directorio `sqllib/samples/cpp`, junto con programas de ejemplo que se pueden crear a partir de esos archivos.

El archivo de creación `blapp` contiene los mandatos para crear aplicaciones de la API de DB2 y de SQL incorporado.

El primer parámetro, `$1`, especifica el nombre del archivo fuente. Éste es el único parámetro necesario para los programas DB2 que no contienen SQL incorporado. Para crear programas de SQL incorporado es necesaria una conexión con la base de datos, por lo que también se proporcionan otros tres parámetros opcionales: el

segundo parámetro, \$2, especifica el nombre de la base de datos con la que se desea conectar; el tercer parámetro, \$3, especifica el ID de usuario correspondiente a la base de datos, y \$4 especifica la contraseña.

Para un programa de SQL incorporado, `bldapp` pasa los parámetros al script de precompilación y vinculación, `embprep`. Si no se proporciona un nombre de base de datos, se utiliza la base de datos por omisión `sample`. Los parámetros de ID de usuario y contraseña sólo son necesarios si la instancia donde se crea el programa es diferente de la instancia donde reside la base de datos.

Procedimiento:

Los ejemplos siguientes muestran cómo crear y ejecutar programas de la API de DB2 y aplicaciones de SQL incorporado.

Para crear el programa de ejemplo de SQL no incorporado, `cli_info`, a partir del archivo fuente `cli_info.C`, especifique:

```
bldapp cli_info
```

El resultado es un archivo ejecutable, `cli_info`. Para ejecutar el archivo ejecutable sobre la base de datos `sample`, especifique:

```
cli_info
```

Creación y ejecución de aplicaciones de SQL incorporado

A partir del archivo fuente `tbmod.sqC`, puede crear la aplicación de SQL incorporado, `tbmod`, de tres maneras:

1. Si conecta con la base de datos "sample" en la misma instancia, especifique:

```
bldapp tbmod
```

2. Si conecta con otra base de datos en la misma instancia, especifique también el nombre de la base de datos:

```
bldapp tbmod basedatos
```

3. Si conecta con una base de datos de otra instancia, especifique también el ID de usuario y la contraseña de la instancia de base de datos:

```
bldapp tbmod basedatos IDusuario contraseña
```

El resultado es un archivo ejecutable, `tbmod`.

Existen tres maneras de ejecutar esta aplicación de SQL incorporado:

1. Si desea acceder a la base de datos `sample` en la misma instancia, especifique simplemente el nombre del ejecutable:

```
tbmod
```

2. Si desea acceder a otra base de datos en la misma instancia, especifique el nombre del ejecutable y el nombre de la base de datos:

```
tbmod basedatos
```

3. Si desea acceder a una base de datos de otra instancia, especifique el nombre del ejecutable, el nombre de la base de datos, y el ID de usuario y la contraseña de la instancia de base de datos:

```
tbmod basedatos IDusuario contraseña
```

Conceptos relacionados:

- "Archivos de creación" en la página 110

Tareas relacionadas:

- “Creación de rutinas C++ de UNIX” en la página 172

Información relacionada:

- “Opciones de compilación y enlace para aplicaciones C++ de AIX” en la página 189
- “Opciones de compilación y enlace para aplicaciones C++ de HP-UX” en la página 218
- “Opciones de compilación y enlace para aplicaciones C++ de Linux” en la página 233
- “Opciones de compilación y enlace para aplicaciones C++ de Solaris” en la página 249

Ejemplos relacionados:

- “bldapp -- Builds AIX C++ applications (C++)”
- “bldapp -- Builds HP-UX C++ applications (C++)”
- “bldapp -- Builds Linux C++ applications (C++)”
- “bldapp -- Builds Solaris C++ applications (C++)”
- “cli_info.C -- Set and get information at the client level (C++)”
- “tbmod.sqlC -- How to modify table data (C++)”
- “embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs (C)”

Creación de aplicaciones C++ de conexión múltiple de UNIX

DB2 proporciona scripts de creación para compilar y enlazar programas C++ de SQL incorporado y de la API de DB2. Estos scripts residen en el directorio `sql1lib/samples/cpp`, junto con programas de ejemplo que se pueden crear a partir de esos archivos.

El archivo de creación, `bldmc`, contiene los mandatos para crear un programa de conexión múltiple de DB2, que requiere dos bases de datos. Las opciones de compilación y enlace son las mismas que las utilizadas en `bldapp`.

El primer parámetro, `$1`, especifica el nombre del archivo fuente. El segundo parámetro, `$2`, especifica el nombre de la primera base de datos a la que desea conectarse. El tercer parámetro, `$3`, especifica la segunda base de datos a la que desea conectarse. Todos estos parámetros son obligatorios.

Nota: El makefile codifica los valores por omisión “sample” y “sample2” como nombres de la base de datos (`$2` y `$3`, respectivamente) por lo que, si está utilizando el makefile y acepta estos valores, sólo tendrá que especificar el nombre del programa (parámetro `$1`). Si utiliza el script `bldmc` debe especificar los tres parámetros.

No se requieren parámetros opcionales para una conexión local, pero sí para conectar con un servidor desde un cliente remoto. Estos parámetros son: `$4` y `$5` para especificar, respectivamente, el ID de usuario y la contraseña para la primera base de datos; y `$6` y `$7` para especificar, respectivamente, el ID de usuario y la contraseña para la segunda base de datos.

Procedimiento:

Para el programa de ejemplo de conexión múltiple, dbmcon, se requieren dos bases de datos. Si aún no se ha creado la base de datos sample, la puede crear entrando db2sample en la línea de mandatos. La segunda base de datos, aquí llamada sample2, se puede crear mediante uno de los mandatos siguientes:

Si crea la base de datos localmente:

```
db2 create db sample2
```

Si crea la base de datos remotamente:

```
db2 attach to <nombre_nodo>
db2 create db sample2
db2 detach
db2 catalog db sample2 as sample2 at node <nombre_nodo>
```

donde <nombre_nodo> es el nodo en que reside la base de datos.

Una conexión múltiple también requiere que se esté ejecutando el receptor TCP/IP. Para asegurarse que es así, haga lo siguiente:

1. Establezca la variable de entorno DB2COMM en TCP/IP de la forma siguiente:

```
db2set DB2COMM=TCPIP
```

2. Actualice el archivo de configuración del gestor de bases de datos con el nombre del servicio de TCP/IP que se haya especificado en el archivo de servicios:

```
db2 update dbm cfg using SVCENAME <nombre del servicio de TCP/IP>
```

Cada instancia tiene un nombre de servicio TCP/IP listado en el archivo de servicios. Si no puede localizarlo o no tiene el permiso de archivo para cambiar el archivo de servicios, pida ayuda al administrador.

3. Detenga y reinicie el gestor de bases de datos para que estos cambios entren en vigor:

```
db2stop
db2start
```

El programa dbmcon consta de cinco archivos:

dbmcon.sqC

Archivo fuente principal para conectar con ambas bases de datos.

dbmcon1.sqC

Archivo fuente para crear un paquete vinculado a la primera base de datos.

dbmcon1.h

Archivo de cabecera para dbmcon1.sqC, incluido en dbmcon.sqC, para acceder a las sentencias de SQL para crear y eliminar una tabla que se debe vincular a la primera base de datos.

dbmcon2.sqC

Archivo fuente para crear un paquete vinculado a la segunda base de datos.

dbmcon2.h

Archivo de cabecera para dbmcon2.sqC, incluido en dbmcon.sqC, para acceder a las sentencias de SQL para crear y eliminar una tabla que se debe vincular a la segunda base de datos.

Para crear el programa de ejemplo de conexión múltiple, dbmcon, se entre lo siguiente:

```
bldmc dbmcon sample sample2
```

El resultado es un archivo ejecutable, dbmcon.

Para ejecutar el archivo ejecutable, escriba el nombre del ejecutable:

```
dbmcon
```

El programa muestra una confirmación en una fase para dos bases de datos.

Conceptos relacionados:

- “Archivos de creación” en la página 110

Tareas relacionadas:

- “Creación de aplicaciones C++ de UNIX” en la página 168

Información relacionada:

- “svcename - TCP/IP service name configuration parameter” en el manual *Administration Guide: Performance*
- “Opciones de compilación y enlace para aplicaciones C++ de AIX” en la página 189
- “Opciones de compilación y enlace para aplicaciones C++ de HP-UX” en la página 218
- “Opciones de compilación y enlace para aplicaciones C++ de Linux” en la página 233
- “Opciones de compilación y enlace para aplicaciones C++ de Solaris” en la página 249

Ejemplos relacionados:

- “bldmc -- Builds AIX C++ multi-connection applications (C++)”
- “bldmc -- Builds HP-UX C++ multi-connection applications (C++)”
- “bldmc -- Builds Linux C++ multi-connection applications (C++)”
- “bldmc -- Builds Solaris C++ multi-connection applications (C++)”
- “dbmcon.sqC -- How to use multiple databases (C++)”
- “dbmcon1.h -- Class declaration for the source file, dbmcon1.sqC (C++)”
- “dbmcon1.sqC -- Functions used in the multiple databases program dbmcon.sqC (C++)”
- “dbmcon2.h -- Class declaration for the source file, dbmcon2.sqC (C++)”
- “dbmcon2.sqC -- Functions used in the multiple databases program dbmcon.sqC (C++)”

Creación de rutinas C++ de UNIX

DB2 proporciona scripts de creación para compilar y enlazar programas C++. Estos scripts residen en el directorio `sql1ib/samples/cpp`, junto con programas de ejemplo que se pueden crear a partir de esos archivos.

El archivo de script `bldrtn` contiene los mandatos para crear rutinas. El archivo de script compila las rutinas y crea una biblioteca compartida que puede ser cargada por el gestor de bases de datos y llamada por una aplicación cliente.

El primer parámetro, \$1, especifica el nombre del archivo fuente. El segundo parámetro, \$2, especifica el nombre de la base de datos a la que desea conectarse.

El parámetro correspondiente a la base de datos es opcional. Si no se proporciona un nombre de base de datos, el programa utiliza la base de datos por omisión `sample`. Debido a que el procedimiento almacenado se debe crear en la misma instancia donde reside la base de datos, no hay parámetros para el ID de usuario ni la contraseña.

Procedimiento:

Los ejemplos siguientes muestran cómo crear bibliotecas compartidas de rutinas mediante:

- procedimientos almacenados
- funciones definidas por el usuario (UDF) de SQL no incorporado
- funciones definidas por el usuario (UDF) de SQL incorporado

Biblioteca compartida de procedimientos almacenados

Puede crear el programa de ejemplo `spserver` a partir del archivo fuente `spserver.sql`.

1. Si conecta con la base de datos `sample`, especifique el nombre del script de creación y el nombre del programa:

```
bldrtn spserver
```

Si conecta con otra base de datos, especifique también el nombre de la base de datos:

```
bldrtn spserver basedatos
```

El archivo de script copia la biblioteca compartida en el directorio `sqllib/function` del servidor.

2. A continuación, catalogue las rutinas ejecutando el script `spcat` en el servidor:

```
spcat
```

Este script conecta con la base de datos "sample", descataloga mediante `spdrop.db2` las rutinas que se hubieran catalogado previamente, luego las cataloga llamando a `spcreate.db2`, y finalmente desconecta de la base de datos. Puede también ejecutar los scripts `spdrop.db2` y `spcreate.db2` por separado.

3. A continuación, si no es la primera vez que se crea el procedimiento almacenado, detenga y reinicie la base de datos para asegurarse que se reconoce la nueva versión de la biblioteca compartida. Lo puede hacer entrando en la línea de mandatos `db2stop` seguido de `db2start`.

Una vez creada la biblioteca compartida, `spserver`, puede crear la aplicación cliente, `spclient`, la cual accede a la biblioteca compartida. Puede crear `spclient` utilizando el archivo de script, `bldapp`.

Para invocar los procedimientos almacenados contenidos en la biblioteca compartida, ejecute la aplicación cliente de ejemplo, especificando lo siguiente:

```
spclient basedatos IDusuario contraseña
```

donde

| **basedatos**

| Es el nombre de la base de datos a la que desea conectarse. El nombre
| podría ser `sample`, o su alias, u otro nombre de base de datos.

| **IDusuario**

| Es un ID de usuario válido.

| **contraseña**

| Es una contraseña válida correspondiente al ID de usuario.

| La aplicación cliente accede a la biblioteca compartida, `spserver`, y ejecuta diversas
| funciones de procedimiento almacenado contenidas en la base de datos del
| servidor. Los datos resultantes se devuelven a la aplicación cliente.

| **Biblioteca compartida de UDF de SQL no incorporado**

| Para crear el programa de función definida por el usuario `udfsrv`, a partir del
| archivo fuente `udfsrv.C`, especifique el nombre del script de creación y el nombre
| del programa:

| `bldrtn udfsrv`

| El archivo de script copia la UDF en el directorio `sql11b/function`.

| Si es necesario, establezca la modalidad de archivo para la UDF a fin de que el
| gestor de bases de datos pueda acceder a ella.

| Una vez creado el programa `udfsrv`, puede crear la aplicación cliente, `udfcli`, que
| llama al programa. Puede crear `udfcli` a partir del archivo fuente `udfcli.sqC`
| utilizando el archivo de script `bldapp`.

| Para invocar las UDF contenidas en la biblioteca compartida, ejecute la aplicación
| cliente, especificando lo siguiente:

| `udfcli basedatos IDusuario contraseña`

| donde

| **basedatos**

| Es el nombre de la base de datos a la que desea conectarse. El nombre
| podría ser `sample`, o su alias, u otro nombre de base de datos.

| **IDusuario**

| Es un ID de usuario válido.

| **contraseña**

| Es una contraseña válida correspondiente al ID de usuario.

| La aplicación cliente accede a la biblioteca compartida, `udfsrv`, y ejecuta las
| funciones definidas por el usuario contenidas en la base de datos del servidor. Los
| datos resultantes se devuelven a la aplicación cliente.

| **Biblioteca compartida de UDF de SQL incorporado**

| Puede crear el programa de función definida por el usuario de SQL incorporado,
| `udfemsrv`, a partir del archivo fuente `udfemsrv.sqC`. Si conecta con la base de datos
| `sample`, especifique el nombre del script de creación y el nombre del programa:

| `bldrtn udfemsrv`

Si conecta con otra base de datos, especifique también el nombre de la base de datos:

```
bldrtn udfemsrv basedatos
```

El archivo de script copia la UDF en el directorio `sqllib/function`.

Una vez creado el programa `udfemsrv`, puede crear la aplicación cliente, `udfemcli`, que llama al programa. Puede crear `udfemcli` a partir del archivo fuente `udfemcli.sqC` utilizando el archivo de script `bldapp`.

Para invocar las UDF contenidas en la biblioteca compartida, ejecute la aplicación cliente, especificando lo siguiente:

```
udfemcli basedatos IDusuario contraseña
```

donde

basedatos

Es el nombre de la base de datos a la que desea conectarse. El nombre podría ser `sample`, o su alias, u otro nombre de base de datos.

IDusuario

Es un ID de usuario válido.

contraseña

Es una contraseña válida correspondiente al ID de usuario.

La aplicación cliente accede a la biblioteca compartida, `udfemsrv`, y ejecuta las funciones definidas por el usuario contenidas en la base de datos del servidor. Los datos resultantes se devuelven a la aplicación cliente.

Conceptos relacionados:

- “Archivos de creación” en la página 110

Tareas relacionadas:

- “Creación de aplicaciones C++ de UNIX” en la página 168

Información relacionada:

- “Opciones de compilación y enlace para rutinas C++ de AIX” en la página 191
- “Opciones de compilación y enlace para rutinas C++ de HP-UX” en la página 220
- “Opciones de compilación y enlace para rutinas C++ de Linux” en la página 235
- “Opciones de compilación y enlace para rutinas C++ de Solaris” en la página 251

Ejemplos relacionados:

- “`bldrtn -- Builds AIX C++ routines (stored procedures and UDFs) (C++)`”
- “`bldrtn -- Builds HP-UX C++ routines (stored procedures and UDFs) (C++)`”
- “`bldrtn -- Builds Linux C++ routines (stored procedures and UDFs) (C++)`”
- “`bldrtn -- Builds Solaris C++ routines (stored procedures or UDFs) (C++)`”
- “`spclient.sqC -- Call various stored procedures (C++)`”
- “`spserver.sqC -- Definition of various types of stored procedures (C++)`”
- “`udfcli.sqC -- Call a variety of types of user-defined functions (C++)`”

- “udfemcli.sqlC -- Call a variety of types of embedded SQL user-defined functions. (C++)”
- “udfemsrv.sqlC -- Call a variety of types of embedded SQL user-defined functions. (C++)”
- “udfsrv.C -- Defines a variety of types of user-defined functions (C++)”
- “embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs (C)”

Creación de aplicaciones Micro Focus COBOL de UNIX

DB2 proporciona scripts de creación para compilar y enlazar programas Micro Focus COBOL de SQL incorporado y de la API de DB2. Estos scripts residen en el directorio `sql1lib/samples/cobol_mf`, junto con programas de ejemplo que se pueden crear a partir de esos archivos.

El archivo de creación `bldapp` contiene los mandatos para crear un programa de aplicación DB2.

El primer parámetro, `$1`, especifica el nombre del archivo fuente. Éste es el único parámetro necesario para los programas que no contienen SQL incorporado. Para crear programas de SQL incorporado es necesaria una conexión con la base de datos, por lo que también se proporcionan otros tres parámetros opcionales: el segundo parámetro, `$2`, especifica el nombre de la base de datos con la que se desea conectar; el tercer parámetro, `$3`, especifica el ID de usuario correspondiente a la base de datos, y `$4` especifica la contraseña.

Para un programa de SQL incorporado, `bldapp` pasa los parámetros al script de precompilación y vinculación, `embprep`. Si no se proporciona un nombre de base de datos, se utiliza la base de datos por omisión `sample`. Los parámetros de ID de usuario y contraseña sólo son necesarios si la instancia donde se crea el programa es diferente de la instancia donde reside la base de datos.

Procedimiento:

Para crear el programa de ejemplo de SQL no incorporado, `client`, a partir del archivo fuente `client.cbl`, especifique:

```
bldapp client
```

El resultado es un archivo ejecutable, `client`. Para ejecutar el archivo ejecutable sobre la base de datos `sample`, especifique:

```
client
```

Creación y ejecución de aplicaciones de SQL incorporado

A partir del archivo fuente `updat.sql`, puede crear la aplicación de SQL incorporado, `updat`, de tres maneras:

1. Si conecta con la base de datos “sample” en la misma instancia, especifique:

```
bldapp updat
```

2. Si conecta con otra base de datos en la misma instancia, especifique también el nombre de la base de datos:

```
bldapp updat basedatos
```

3. Si conecta con una base de datos de otra instancia, especifique también el ID de usuario y la contraseña de la instancia de base de datos:


```
bldapp updat basedatos IDusuario contraseña
```

El resultado es un archivo ejecutable, updat.

Existen tres maneras de ejecutar esta aplicación de SQL incorporado:

1. Si desea acceder a la base de datos sample en la misma instancia, especifique simplemente el nombre del ejecutable:

```
updat
```

2. Si desea acceder a otra base de datos en la misma instancia, especifique el nombre del ejecutable y el nombre de la base de datos:

```
updat basedatos
```

3. Si desea acceder a una base de datos de otra instancia, especifique el nombre del ejecutable, el nombre de la base de datos, y el ID de usuario y la contraseña de la instancia de base de datos:

```
bldapp updat basedatos IDusuario contraseña
```

Tareas relacionadas:

- “Creación de rutinas Micro Focus COBOL de UNIX” en la página 177

Información relacionada:

- “Opciones de compilación y enlace para aplicaciones Micro Focus COBOL de AIX” en la página 207
- “Opciones de compilación y enlace para aplicaciones Micro Focus COBOL de HP-UX” en la página 224
- “Opciones de compilación y enlace para aplicaciones Micro Focus COBOL de Solaris” en la página 254
- “Opciones de compilación y enlace para aplicaciones Micro Focus COBOL de Linux” en la página 239

Ejemplos relacionados:

- “bldapp -- Builds AIX Micro Focus COBOL applications”
- “bldapp -- Builds HP-UX Micro Focus COBOL applications”
- “bldapp -- Builds Linux Micro Focus COBOL applications”
- “bldapp -- Builds Solaris Micro Focus COBOL applications”
- “client.cbl -- How to set and query a client (MF COBOL)”
- “updat.sqb -- How to update, delete and insert table data (MF COBOL)”
- “embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs (C)”

Creación de rutinas Micro Focus COBOL de UNIX

DB2 proporciona scripts de creación para compilar y enlazar programas Micro Focus COBOL de SQL incorporado y de la API de DB2. Estos scripts residen en el directorio `sqllib/samples/cobol_mf`, junto con programas de ejemplo que se pueden crear a partir de esos archivos.

El script `bldrtn` contiene los mandatos para crear rutinas (procedimientos almacenados). El script compila el archivo fuente de la rutina y crea una biblioteca compartida que puede ser llamada por una aplicación cliente.

El primer parámetro, \$1, especifica el nombre del archivo fuente. El script utiliza el nombre del archivo fuente como nombre de la biblioteca compartida. El segundo parámetro, \$2, especifica el nombre de la base de datos a la que desea conectarse. Debido a que la biblioteca compartida se debe crear en la misma instancia donde reside la base de datos, no hay parámetros para el ID de usuario ni la contraseña.

Sólo es obligatorio el primer parámetro, el nombre del archivo fuente. El nombre de la base de datos es opcional. Si no se proporciona un nombre de base de datos, el programa utiliza la base de datos por omisión `sample`.

Valores específicos de Solaris

Antes de crear las rutinas de Micro Focus en Solaris, ejecute estos mandatos:

```
db2stop
db2set DB2LIBPATH=$LD_LIBRARY_PATH
db2set DB2ENVLIST="COBDIR LD_LIBRARY_PATH"
db2set
db2start
```

Asegúrese de que `db2stop` detiene la base de datos. El último mandato `db2set` se emite para comprobar los valores: compruebe que `DB2LIBPATH` y `DB2ENVLIST` estén definidos correctamente.

Procedimiento:

Si conecta con la base de datos "sample", especifique lo siguiente para crear el programa de ejemplo `outsrv` a partir del archivo fuente `outsrv.sqb`:

```
bldrtn outsrv
```

Si conecta con otra base de datos, especifique también el nombre de la base de datos:

```
bldrtn outsrv basedatos
```

El archivo de script copia la biblioteca compartida en el directorio `sqllib/function` del servidor.

Una vez creado el procedimiento almacenado `outsrv`, puede crear la aplicación cliente `outcli` que llama al procedimiento almacenado. Puede crear `outcli` utilizando el archivo de script `blapp`.

Para invocar el procedimiento almacenado, ejecute la aplicación cliente de ejemplo, especificando lo siguiente:

```
outcli basedatos IDusuario contraseña
```

donde

basedatos

Es el nombre de la base de datos a la que desea conectarse. El nombre podría ser `sample`, o su alias, u otro nombre.

IDusuario

Es un ID de usuario válido.

contraseña

Es una contraseña válida correspondiente al ID de usuario.

| La aplicación cliente accede a la biblioteca compartida, `outsrv`, la cual ejecuta la
| función de procedimiento almacenado del mismo nombre contenida la base de
| datos el servidor. A continuación los datos resultantes se devuelven a la aplicación
| cliente.

| **Tareas relacionadas:**

- “Creación de aplicaciones Micro Focus COBOL de UNIX” en la página 176

| **Información relacionada:**

- “Opciones de compilación y enlace para rutinas Micro Focus COBOL de AIX” en la página 208
- “Opciones de compilación y enlace para rutinas Micro Focus COBOL de HP-UX” en la página 225
- “Opciones de compilación y enlace para rutinas Micro Focus COBOL de Solaris” en la página 256
- “Opciones de compilación y enlace para rutinas Micro Focus COBOL de Linux” en la página 240

| **Ejemplos relacionados:**

- “`bldrtn -- Builds AIX Micro Focus COBOL routines (stored procedures)`”
- “`bldrtn -- Builds HP-UX Micro Focus COBOL routines (stored procedures)`”
- “`bldrtn -- Builds Linux Micro Focus COBOL routines (stored procedures)`”
- “`bldrtn -- Builds Solaris Micro Focus COBOL routines (stored procedures)`”
- “`outcli.sqb -- Call stored procedures using the SQLDA structure (MF COBOL)`”
- “`outsrv.sqb -- Demonstrates stored procedures using the SQLDA structure (MF COBOL)`”
- “`embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs (C)`”

Capítulo 10. AIX

Consideraciones importantes	181	Creación de procedimientos almacenados C++ con archivos de configuración	196
Archivos de exportación de AIX para rutinas	181	Creación de funciones definidas por el usuario C++ con archivos de configuración	197
Las rutinas de AIX y la sentencia CREATE . . .	182	IBM COBOL Set para AIX	198
Sustitución de una biblioteca compartida de AIX	183	Configuración del compilador IBM COBOL en AIX	198
Consideraciones para la instalación de COBOL en AIX	183	Creación de aplicaciones IBM COBOL en AIX	199
IBM C.	184	Script de creación para aplicaciones COBOL DE IBM	201
Script de creación para aplicaciones C	184	Opciones de compilación y enlace para aplicaciones IBM COBOL de AIX.	201
Opciones de compilación y enlace para aplicaciones C de AIX	185	Creación de rutinas IBM COBOL en AIX . . .	202
Script de creación para rutinas C.	186	Script de creación para rutinas COBOL de IBM	203
Opciones de compilación y enlace para rutinas C de AIX.	186	Opciones de compilación y enlace para rutinas IBM COBOL de AIX	204
Creación de aplicaciones multihebra C en AIX	187	COBOL para Micro Focus	205
VisualAge C++.	188	Configuración del compilador Micro Focus COBOL en AIX.	205
Script de creación para aplicaciones C++ . . .	188	Script de creación para aplicaciones COBOL para Micro Focus	206
Opciones de compilación y enlace para aplicaciones C++ de AIX	189	Opciones de compilación y enlace para aplicaciones Micro Focus COBOL de AIX . . .	207
Script de creación para rutinas C++	190	Script de creación para rutinas COBOL para Micro Focus	207
Opciones de compilación y enlace para rutinas C++ de AIX	191	Opciones de compilación y enlace para rutinas Micro Focus COBOL de AIX	208
Creación de aplicaciones multihebra C++ en AIX	192	REXX	209
Archivos de configuración VisualAge C++. . . .	193	Creación de aplicaciones REXX en AIX	209
Creación de programas de VisualAge C++ con archivos de configuración	193		
Creación de aplicaciones C++ de la API de DB2 con archivos de configuración	194		
Creación de aplicaciones C++ de SQL incorporado con archivos de configuración . . .	195		

Este capítulo proporciona información detallada para crear aplicaciones en AIX. Para conocer la información más reciente sobre el desarrollo de aplicaciones DB2, consulte la página Web situada en:

<http://www.ibm.com/software/data/db2/udb/ad>

Consideraciones importantes

Esta sección proporciona información específica de AIX para crear aplicaciones DB2 en diversos compiladores soportados.

Archivos de exportación de AIX para rutinas

Las rutinas externas se compilan en el servidor, y se almacenan y ejecutan en bibliotecas compartidas del servidor. Estas bibliotecas compartidas se crean al compilar las rutinas.

En AIX®, debe proporcionar un archivo de exportación que especifique qué funciones globales de la biblioteca se pueden invocar desde fuera de ella, o puede crear bibliotecas y objetos compartidos de AIX con el comportamiento 'export all', utilizando el compilador tal como en el ejemplo siguiente:

```
x1C -qmkshrobj -q32 -g dlsources.C -o libtestit.so
```

El ejemplo de DB2[®] utiliza un archivo de exportación. Este archivo debe incluir los nombres de todas las rutinas contenidas en la biblioteca. Otras plataformas UNIX[®] simplemente exportan todas las funciones globales de la biblioteca. A continuación se muestra un ejemplo de archivo de exportación de AIX:

```
#! spserver export file
outlanguage
```

El archivo de exportación `spserver.exp` lista el procedimiento almacenado `outlanguage`. El editor de enlaces utiliza `spserver.exp` para crear la biblioteca compartida `spserver` donde reside el procedimiento almacenado `outlanguage`.

La documentación del editor de enlaces de AIX contiene más información sobre los archivos de exportación.

Conceptos relacionados:

- “Las rutinas de AIX y la sentencia CREATE” en la página 182

Tareas relacionadas:

- “Sustitución de una biblioteca compartida de AIX” en la página 183

Las rutinas de AIX y la sentencia CREATE

Esta sección describe la relación existente entre la compilación y el enlace de la rutina y la información que el usuario proporciona en la cláusula `EXTERNAL NAME` de la sentencia `CREATE`.

Cuando compila y enlaza el programa, puede identificar funciones externas utilizando un archivo de exportación especificado mediante la opción `-bE:`.

Suponga que la biblioteca `myrtns` contiene tres rutinas: `modify`, `remove` y `add`. Para identificar `modify` como punto de entrada por omisión, colóquelo como primera entrada en el archivo de exportación que se enlaza durante la edición de enlaces. Las funciones `remove` y `add` se identifican como funciones exportables adicionales incluyéndolas también en un archivo de exportación.

En el paso de edición de enlaces, especifique:

```
-bE:myrtns.exp
```

esto identifica el archivo de exportación `myrtns.exp`.

El archivo de exportación tiene este aspecto:

```
modify
remove
add
```

Finalmente, las cláusulas `EXTERNAL NAME` de las rutinas, que se implementan con las funciones `modify`, `remove` y `add`, se codifican así:

```
EXTERNAL NAME '/u/mydir/rutinas/myrtns!modify'
```

y

```
EXTERNAL NAME '/u/mydir/rutinas/myrtns!remove'
```

y

```
EXTERNAL NAME '/u/mydir/rutinas/myrtns!add'
```

Nota: La vía de acceso por omisión será `sqllib/function`. Esto significa que si la cláusula `EXTERNAL NAME` se especifica así:

```
EXTERNAL NAME 'myrtns!modify'
```

DB2® intentará cargar `myrtns` desde `sqllib/function`.

Conceptos relacionados:

- “Archivos de exportación de AIX para rutinas” en la página 181

Tareas relacionadas:

- “Sustitución de una biblioteca compartida de AIX” en la página 183

Sustitución de una biblioteca compartida de AIX

Procedimiento:

Una vez creada la biblioteca compartida, generalmente se copia en un directorio desde el que DB2 accederá a ella. Si desea sustituir una biblioteca compartida de rutina, debe ejecutar `/usr/sbin/slibclean` para vaciar la antememoria de la biblioteca compartida AIX, o bien eliminar la biblioteca en el directorio de destino y luego copiar la nueva biblioteca desde el directorio de origen al directorio de destino. De lo contrario, la operación de copia puede fallar porque AIX guarda en la antememoria las bibliotecas referenciadas y no permite que se sobrescriba la biblioteca.

Conceptos relacionados:

- “Archivos de exportación de AIX para rutinas” en la página 181
- “Las rutinas de AIX y la sentencia CREATE” en la página 182

Consideraciones para la instalación de COBOL en AIX

Debido a la forma en que AIX® carga las rutinas y resuelve las referencias a bibliotecas contenidas en ellas, existen ciertos requisitos respecto a la manera en que se debe instalar COBOL. Estos requisitos son importantes cuando un programa COBOL carga una biblioteca compartida (rutina) durante la ejecución.

Cuando se carga una rutina, también se debe cargar la cadena de bibliotecas referenciadas en la rutina. Cuando AIX busca una biblioteca que sólo está referenciada indirectamente por el programa del usuario, AIX debe utilizar la vía de acceso compilada en la biblioteca a la que hacía referencia cuando fue creada por el proveedor de lenguaje (IBM COBOL o Micro Focus COBOL). Esta vía de acceso puede no ser la misma que la vía de acceso donde se instaló el compilador. Si la biblioteca de la cadena no se puede encontrar, la rutina no se podrá cargar y el usuario recibirá el código de SQL -444.

Para asegurarse de que no ocurra eso, instale el compilador donde desee, luego cree enlaces simbólicos para todas las bibliotecas de lenguaje desde el directorio de instalación a `/usr/lib` (directorio donde casi siempre se busca cuando es necesario cargar una biblioteca). Puede enlazar las bibliotecas en `sqllib/function` (el directorio de rutinas), pero esto sólo es efectivo para una sola instancia de base de datos; `/usr/lib` puede utilizarse para todos los usuarios de la máquina.

Tareas relacionadas:

- “Configuración del entorno de desarrollo de aplicaciones UNIX” en la página 34

- “Configuración del compilador IBM COBOL en AIX” en la página 198
- “Configuración del compilador Micro Focus COBOL en AIX” en la página 205

IBM C

La creación de información para las aplicaciones y rutinas de la CLI de DB2 se encuentra en el manual *CLI Guide and Reference*.

Para obtener información sobre cómo crear aplicaciones C en sistemas operativos UNIX soportados, consulte el apartado “Creación de aplicaciones C de UNIX” en la página 161. Para obtener información sobre cómo crear rutinas C en sistemas operativos UNIX soportados, consulte el apartado “Creación de rutinas C de UNIX” en la página 165.

Script de creación para aplicaciones C

```

#!/bin/sh
# SCRIPT: bldapp
# Crea programas de aplicaciones C para AIX
# Uso: bldapp <nombre_prog> [ <nombre_bd> [ <IDusuario> <contraseña> ]]

# Definir DB2PATH de modo que apunte al lugar desde donde se accederá a DB2.
# El valor por omisión es la vía de acceso de instancia estándar.
DB2PATH=$HOME/sqllib

# Definir lib32 para programas de 32 bits, lib para 64 bits,
# y definir distintivo adicional de compilación para programas de 64 bits.
bitwidth=`LANG=C db2level | awk '/bits/{print $5}'`
if [ $bitwidth = "\"32\"" ]; then
    LIB=lib32
    EXTRA_CFLAG=
else
    LIB=lib
    EXTRA_CFLAG=-q64
fi

# Si es un programa de SQL incorporado, precompilar y vincular el programa.
# Nota: algunos archivos .sqc no contienen SQL sino que enlazan en
# utilemb.sqc, por lo que debe ignorar este aviso si lo recibe:
# SQL0053W No se ha encontrado ninguna sentencia SQL en el programa.
if [ -f $1".sqc" ]
then
./embprep $1 $2 $3 $4
# Compilar el programa de utilidad de comprobación de errores utilemb.c.
xlc $EXTRA_CFLAG -I$DB2PATH/include -c utilemb.c
else
# Compilar el programa de utilidad de comprobación de errores utilapi.c.
xlc $EXTRA_CFLAG -I$DB2PATH/include -c utilapi.c
fi

# Compilar el programa.
xlc $EXTRA_CFLAG -I$DB2PATH/include -c $1.c

if [ -f $1".sqc" ]
then
# Enlazar el programa con utilemb.o
xlc $EXTRA_CFLAG -o $1 $1.o utilemb.o -ldb2 -L$DB2PATH/$LIB
else
# Enlazar el programa con utilapi.o
xlc $EXTRA_CFLAG -o $1 $1.o utilapi.o -ldb2 -L$DB2PATH/$LIB
fi

```


Opciones de compilación y enlace para aplicaciones C de AIX

La tabla siguiente muestra las opciones de compilación y enlace que DB2 recomienda para crear aplicaciones C de SQL incorporado y de la API de DB2 con el compilador C de IBM para AIX, tal como muestra el script de creación bldapp.

Opciones de compilación y enlace para bldapp	
Opciones de compilación:	
x1c	Denota el compilador IBM C.
\$EXTRA_CFLAG	Contiene "-q64" para una instancia en que está habilitado el soporte de 64 bits; de lo contrario, no contiene ningún valor.
-I\$DB2PATH/include	Especifica la ubicación de los archivos de inclusión de DB2. Por ejemplo: \$HOME/sql1lib/include.
-c	Hace que se efectúe solamente la compilación, no la edición de enlaces. La compilación y la edición de enlaces son pasos separados.
Opciones de enlace:	
x1c	Hace que el compilador se utilice como interfaz del editor de enlaces.
\$EXTRA_CFLAG	Contiene "-q64" para una instancia en que está habilitado el soporte de 64 bits; de lo contrario, no contiene ningún valor.
-o \$1	Especifica el programa ejecutable.
\$1.o	Especifica el archivo objeto del programa.
utilemb.o	Si es un programa de SQL incorporado, incluir el archivo objeto del programa de utilidad de SQL incorporado para la comprobación de errores.
utilapi.o	Si crea un programa que no es de SQL incorporado, incluir el archivo objeto del programa de utilidad de la API de DB2 para la comprobación de errores.
-ldb2	Enlazar con la biblioteca DB2.
-L\$DB2PATH/\$LIB	Especifica la ubicación de las bibliotecas compartidas de tiempo de ejecución de DB2. Por ejemplo: \$HOME/sql1lib/\$LIB. Si el usuario no especifica la opción -L, el compilador utiliza esta vía de acceso: /usr/lib:/lib.
Consulte la documentación del compilador para conocer otras opciones de compilador.	

Tareas relacionadas:

- "Creación de aplicaciones C de UNIX" en la página 161

Información relacionada:

- "Opciones de compilación y enlace para rutinas C de AIX" en la página 186

Ejemplos relacionados:

- "bldapp -- Builds AIX C application programs (C)"

Script de creación para rutinas C

```
#!/bin/sh
# SCRIPT: bldrtn
# Crea rutinas C para AIX (procedimientos almacenados y UDF)
# Uso: bldrtn <nombre_prog> [ <nombre_bd> ]

# Definir DB2PATH de modo que apunte al lugar desde donde se accederá a DB2.
# El valor por omisión es la vía de acceso de instancia estándar.
DB2PATH=$HOME/sqlllib

# Definir lib32 para programas de 32 bits, lib para 64 bits,
# y definir distintivo adicional de compilación para programas de 64 bits.
bitwidth=`LANG=C db2level | awk '/bits/{print $5}'`
if [ $bitwidth = "\"32\"" ]; then
    LIB=lib32
    EXTRA_CFLAG=
else
    LIB=lib
    EXTRA_CFLAG=-q64
fi

# Si es un programa de SQL incorporado, precompilar y vincular el programa.
if [ -f $1".sqc" ]
then
./embprep $1 $2
fi

# Compilar el programa.
xlc_r $EXTRA_CFLAG -I$DB2PATH/include -c $1.c

# Enlazar el programa utilizando el archivo de exportación $1.exp,
xlc_r $EXTRA_CFLAG -qmksrobj -o $1 $1.o -ldb2 -L$DB2PATH/$LIB -bE:$1.exp

# Copiar la biblioteca compartida en el subdirectorío sqlllib/function.
# Nota: el usuario debe tener permiso de escritura sobre este directorio.
rm -f $DB2PATH/function/$1
cp $1 $DB2PATH/function
```

Opciones de compilación y enlace para rutinas C de AIX

La tabla siguiente muestra las opciones de compilación y enlace que DB2 recomienda para crear rutinas en C (procedimientos almacenados y funciones definidas por el usuario) con el compilador C de IBM para AIX, tal como muestra el script de creación bldrtn.

Opciones de compilación y enlace para bldrtn	
Opciones de compilación:	
xlc_r	Utilizar la versión multihebra del compilador C de IBM, que es necesaria porque las rutinas se pueden ejecutar en el mismo proceso que otras rutinas (THREADSAFE) o en el propio motor (NOT FENCED).
\$EXTRA_CFLAG	Contiene "-q64" para una instancia en que está habilitado el soporte de 64 bits; de lo contrario, no contiene ningún valor.
-I\$DB2PATH/include	Especifica la ubicación de los archivos de inclusión de DB2. Por ejemplo: \$HOME/sqlllib/include.
-c	Hace que se efectúe solamente la compilación, no la edición de enlaces. La compilación y la edición de enlaces son pasos separados.

Opciones de compilación y enlace para bldrtn

Opciones de enlace:

xlc_r Denota la utilización de la versión multihebra del compilador como interfaz del editor de enlaces.

\$EXTRA_CFLAG

Contiene "-q64" para una instancia en que está habilitado el soporte de 64 bits; de lo contrario, no contiene ningún valor.

-qmkshrobj

Hace que se cree la biblioteca compartida.

-o \$1 Especifica el nombre del archivo de salida.

\$1.o Especifica el archivo objeto.

-ldb2 Enlazar con la biblioteca DB2.

-L\$DB2PATH/\$LIB

Especifica la ubicación de las bibliotecas compartidas de tiempo de ejecución de DB2. Por ejemplo: \$HOME/sql1lib/\$LIB. Si el usuario no especifica la opción -L, el compilador utiliza esta vía de acceso: /usr/lib:/lib.

-bE:\$1.exp

Especifica un archivo de exportación. El archivo de exportación contiene una lista de las rutinas.

Consulte la documentación del compilador para conocer otras opciones de compilador.

Tareas relacionadas:

- "Creación de rutinas C de UNIX" en la página 165

Información relacionada:

- "Opciones de compilación y enlace para aplicaciones C de AIX" en la página 185

Ejemplos relacionados:

- "bldrtn -- Builds AIX C routines (stored procedures and UDFs) (C)"

Creación de aplicaciones multihebra C en AIX

Las aplicaciones C multihebra de AIX se deben compilar y enlazar con el compilador `xlc_r`, en lugar de utilizar el compilador `xlc o`, para C++, con el compilador `xlc_r` en lugar de `xlc`. Las versiones `_r` establecen las definiciones de preprocesador apropiadas para la compilación multihebra, y proporcionan al enlazador los nombres apropiados de las bibliotecas encadenadas.

Consulte la documentación del compilador para obtener más información sobre los valores de compilación y enlace utilizados en el compilador multihebra.

El archivo de script `blfmt`, situado en `sql1lib/samples/c`, contiene los mandatos para crear un programa multihebra de SQL incorporado. Excepto por el compilador `xlc_r` y la ausencia de un archivo de programa de utilidad enlazado, las opciones de compilación y enlace son las mismas que las utilizadas para el archivo de script de SQL incorporado, `bldapp`.

Procedimiento:

Para crear el programa multihebra de ejemplo, dbthrs, a partir del archivo fuente dbthrs.sqc, especifique:

```
bldmt dbthrs
```

El resultado es un archivo ejecutable, dbthrs. Para ejecutar el archivo ejecutable sobre la base de datos sample, escriba el nombre del ejecutable:

```
dbthrs
```

Conceptos relacionados:

- “Archivos de creación” en la página 110

Tareas relacionadas:

- “Creación de aplicaciones C de UNIX” en la página 161

Información relacionada:

- “Opciones de compilación y enlace para aplicaciones C de AIX” en la página 185
- “Ejemplos de C” en la página 71

Ejemplos relacionados:

- “bldmt -- Builds AIX C multi-threaded applications (C)”
- “dbthrs.sqc -- How to use multiple context APIs on UNIX (C)”
- “embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs (C)”

VisualAge C++

Para obtener información sobre cómo crear aplicaciones C++ en sistemas operativos UNIX soportados, consulte el apartado “Creación de aplicaciones C++ de UNIX” en la página 168. Para obtener información sobre cómo crear rutinas C++ en sistemas operativos UNIX soportados, consulte el apartado “Creación de rutinas C++ de UNIX” en la página 172.

Script de creación para aplicaciones C++

```
#!/bin/sh
# SCRIPT: bldapp
# Crea aplicaciones C++ para AIX
# Uso: bldapp <nombre_prog> [ <nombre_bd> [ <IDusuario> <contraseña> ]]

# Definir DB2PATH de modo que apunte al lugar desde donde se accederá a DB2.
# El valor por omisión es la vía de acceso de instancia estándar.
DB2PATH=$HOME/sql1lib

# Definir lib32 para programas de 32 bits, lib para 64 bits,
# y definir distintivo adicional de compilación para programas de 64 bits.
bitwidth=`LANG=C db2level | awk '/bits/{print $5}'`
if [ $bitwidth = "\"32\"" ]; then
    LIB=lib32
    EXTRA_CFLAG=
else
    LIB=lib
    EXTRA_CFLAG=-q64
fi

# Si es un programa de SQL incorporado, precompilar y vincular el programa.
# Nota: algunos archivos .sqc no contienen SQL sino que enlazan en
# utilemb.sqc, por lo que debe ignorar este aviso si lo recibe:
# SQL0053W No se ha encontrado ninguna sentencia SQL en el programa.
```

```

if [ -f $1".sqC" ]
then
./embprep $1 $2 $3 $4
# Compilar el programa de utilidad de comprobación de errores utilemb.C.
x1C $EXTRA_CFLAG -qstaticinline -I$DB2PATH/include -c utilemb.C
else
# Compilar el programa de utilidad de comprobación de errores utilapi.C.
x1C $EXTRA_CFLAG -qstaticinline -I$DB2PATH/include -c utilapi.C
fi

# Compilar el programa.
x1C $EXTRA_CFLAG -qstaticinline -I$DB2PATH/include -c $1.C

if [ -f $1".sqC" ]
then
# Enlazar el programa con utilemb.o
x1C $EXTRA_CFLAG -o $1 $1.o utilemb.o -ldb2 -L$DB2PATH/$LIB
else
# Enlazar el programa con utilapi.o
x1C $EXTRA_CFLAG -o $1 $1.o utilapi.o -ldb2 -L$DB2PATH/$LIB
fi

```

Opciones de compilación y enlace para aplicaciones C++ de AIX

La tabla siguiente muestra las opciones de compilación y enlace que DB2 recomienda para crear aplicaciones en C++ de SQL incorporado y de la API de DB2 con el compilador VisualAge C++ de IBM para AIX, tal como muestra el script de creación bldapp.

Opciones de compilación y enlace para bldapp	
Opciones de compilación:	
x1C	Denota el compilador VisualAge C++.
EXTRA_CFLAG	Contiene "-q64" para una instancia en que está habilitado el soporte de 64 bits; de lo contrario, no contiene ningún valor.
-qstaticinline	Brinda un enlace interno de opciones en línea sin línea, de forma que no se produce ningún aviso durante el enlace si la función existe en varios archivos de objetos.
-I\$DB2PATH/include	Especifica la ubicación de los archivos de inclusión de DB2. Por ejemplo: \$HOME/sqllib/include.
-c	Hace que se efectúe solamente la compilación, no la edición de enlaces. La compilación y la edición de enlaces son pasos separados.

Opciones de compilación y enlace para bldapp

Opciones de enlace:

x1C Hace que el compilador se utilice como interfaz del editor de enlaces.

EXTRA_CFLAG

Contiene "-q64" para una instancia en que está habilitado el soporte de 64 bits; de lo contrario, no contiene ningún valor.

-o \$1 Especifica el programa ejecutable.

\$1.o Especifica el archivo objeto del programa.

utilapi.o

Hace que se incluya el archivo objeto del programa de utilidad de la API para programas que no contienen SQL incorporado.

utilemb.o

Incluye el archivo objeto del programa de utilidad de SQL incorporado para los programas de SQL incorporado.

-ldb2 Enlazar con la biblioteca DB2.

-L\$DB2PATH/\$LIB

Especifica la ubicación de las bibliotecas compartidas de tiempo de ejecución de DB2. Por ejemplo: \$HOME/sqllib/\$LIB. Si el usuario no especifica la opción -L, el compilador utiliza esta vía de acceso: /usr/lib:/lib.

Consulte la documentación del compilador para conocer otras opciones de compilador.

Tareas relacionadas:

- "Creación de aplicaciones C++ de UNIX" en la página 168
- "Creación de aplicaciones C++ de SQL incorporado con archivos de configuración" en la página 195
- "Creación de aplicaciones C++ de la API de DB2 con archivos de configuración" en la página 194

Información relacionada:

- "Opciones de compilación y enlace para rutinas C++ de AIX" en la página 191

Ejemplos relacionados:

- "bldapp -- Builds AIX C++ applications (C++)"

Script de creación para rutinas C++

```
#!/bin/sh
# SCRIPT: bldrtn
# Crea rutinas C++ para AIX (procedimientos almacenados y UDF)
# Uso: bldrtn <nombre_prog> [ <nombre_bd> ]

# Definir DB2PATH de modo que apunte al lugar desde donde se accederá a DB2.
# El valor por omisión es la vía de acceso de instancia estándar.
DB2PATH=$HOME/sqllib

# Definir lib32 para programas de 32 bits, lib para 64 bits,
# y definir distintivo adicional de compilación para programas de 64 bits.
bitwidth=`LANG=C db2level | awk '/bits/{print $5}'`
if [ $bitwidth = "\"32\"" ]; then
    LIB=lib32
    EXTRA_CFLAG=
else
    LIB=lib
```

```

EXTRA_CFLAG=-q64
fi

# Si es un programa de SQL incorporado, precompilar y vincular el programa.
if [ -f $1".sqC" ]
then
./embprep $1 $2
fi

# Compilar el programa.
x1C_r $EXTRA_CFLAG -qstaticinline -I$DB2PATH/include -c $1.C

# Enlazar utilizando el archivo de exportación $1.exp, con lo que se
creará la biblioteca compartida $1
x1C_r $EXTRA_CFLAG -qmkshobj -o $1 $1.o -L$DB2PATH/$LIB -ldb2 -bE:$1.exp

# Copiar la biblioteca compartida en el subdirectorío sqllib/function.
# Nota: el usuario debe tener permiso de escritura sobre este directorio.
rm -f $DB2PATH/function/$1
cp $1 $DB2PATH/function

```

Opciones de compilación y enlace para rutinas C++ de AIX

La tabla siguiente muestra las opciones de compilación y enlace que DB2 recomienda para crear rutinas en C++ (procedimientos almacenados y funciones definidas por el usuario) con el compilador C++ de VisualAge para AIX, tal como muestra el script de creación bldrtn.

Opciones de compilación y enlace para bldrtn	
Opciones de compilación:	
x1C_r	Utilizar la versión multihebra del compilador IBM VisualAge C++, que es necesaria porque las rutinas se pueden ejecutar en el mismo proceso que otras rutinas (THREADSAFE) o en el propio motor (NOT FENCED).
\$EXTRA_CFLAG	Contiene "-q64" para una instancia en que está habilitado el soporte de 64 bits; de lo contrario, no contiene ningún valor.
-qstaticinline	Brinda un enlace interno de opciones en línea sin línea, de forma que no se produce ningún aviso durante el enlace si la función existe en varios archivos de objetos.
-I\$DB2PATH/include	Especifica la ubicación de los archivos de inclusión de DB2. Por ejemplo: \$HOME/sqllib/include.
-c	Hace que se efectúe solamente la compilación, no la edición de enlaces. La compilación y la edición de enlaces son pasos separados.

Opciones de compilación y enlace para bldrtn

Opciones de enlace:

x1C_r Denota la utilización de la versión multihebra del compilador como interfaz del editor de enlaces.

\$EXTRA_CFLAG

Contiene "-q64" para una instancia en que está habilitado el soporte de 64 bits; de lo contrario, no contiene ningún valor.

-qmkshrobj

Hace que se cree una biblioteca compartida.

-o \$1 Especifica la salida como archivo de biblioteca compartida.

\$1.o Especifica el archivo objeto del programa.

-L\$DB2PATH/\$LIB

Especifica la ubicación de las bibliotecas compartidas de tiempo de ejecución de DB2. Por ejemplo: \$HOME/sql1lib/\$LIB. Si el usuario no especifica la opción -L, el compilador utiliza esta vía de acceso: /usr/lib:/lib.

-ldb2 Enlazar con la biblioteca DB2.

-bE:\$1.exp

Especifica un archivo de exportación. El archivo de exportación contiene una lista de las rutinas.

Consulte la documentación del compilador para conocer otras opciones de compilador.

Tareas relacionadas:

- "Creación de rutinas C de UNIX" en la página 165
- "Creación de procedimientos almacenados C++ con archivos de configuración" en la página 196
- "Creación de funciones definidas por el usuario C++ con archivos de configuración" en la página 197

Información relacionada:

- "Opciones de compilación y enlace para aplicaciones C++ de AIX" en la página 189

Ejemplos relacionados:

- "bldrtn -- Builds AIX C++ routines (stored procedures and UDFs) (C++)"

Creación de aplicaciones multihebra C++ en AIX

Las aplicaciones C++ multihebra de AIX se deben compilar y enlazar con el compilador x1C_r, en lugar de utilizar el compilador x1C o, para C, con el compilador x1c_r en lugar de x1c. Las versiones _r establecen las definiciones de preprocesador apropiadas para la compilación multihebra, y proporcionan al enlazador los nombres apropiados de las bibliotecas encadenadas.

Consulte la documentación del compilador para obtener más información sobre los valores de compilación y enlace utilizados en el compilador multihebra.

El script bldmt contiene los mandatos para crear aplicaciones multihebra. Excepto por el compilador x1C_r, mencionado anteriormente, y la ausencia de un archivo de programa de utilidad enlazado, las opciones de compilación y enlace son las mismas que las utilizadas para el archivo de script de SQL incorporado, bldapp.

Procedimiento:

Para crear el programa multihebra de ejemplo, dbthrds, a partir del archivo fuente dbthrds.sqC, especifique:

```
bldmt dbthrds
```

El resultado es un archivo ejecutable, dbthrds. Para ejecutar el archivo ejecutable sobre la base de datos sample, escriba el nombre del ejecutable:

```
dbthrds
```

Conceptos relacionados:

- “Archivos de creación” en la página 110

Tareas relacionadas:

- “Creación de aplicaciones C++ de UNIX” en la página 168

Información relacionada:

- “Opciones de compilación y enlace para aplicaciones C++ de AIX” en la página 189
- “Ejemplos de C” en la página 71

Ejemplos relacionados:

- “bldmt -- Builds AIX C++ multi-threaded applications (C++)”
- “dbthrds.sqC -- How to use multiple context APIs on UNIX (C++)”
- “embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs (C)”

Archivos de configuración VisualAge C++

Nota: La creación de información para las aplicaciones y rutinas de la CLI se encuentra en el manual *CLI Guide and Reference*.

Creación de programas de VisualAge C++ con archivos de configuración

VisualAge C++ Versión 5.0 tiene un compilador incremental y un compilador de proceso por lotes. Mientras que el compilador de proceso por lotes utiliza makefiles y archivos de creación, el compilador incremental utiliza archivos de configuración en su lugar. Consulte la documentación de VisualAge C++ Versión 5.0 para conocer más sobre este tema.

DB2 proporciona archivos de configuración para los diferentes tipos de programas DB2 que el usuario puede crear con el compilador VisualAge C++.

Procedimiento:

Para utilizar un archivo de configuración de DB2, primero debe asignar una variable de entorno al nombre del programa que desea compilar. A continuación, compile el programa con un mandato proporcionado por VisualAge C++. Estos son los temas que describen cómo utilizar los archivos de configuración proporcionados por DB2 para compilar los diversos tipos de programas:

- Creación de aplicaciones C++ de SQL incorporado con archivos de configuración

- Creación de aplicaciones C++ de la API de DB2 con archivos de configuración
- Creación de procedimientos almacenados C++ con archivos de configuración
- Creación de funciones definidas por el usuario en C++ con archivos de configuración

Tareas relacionadas:

- “Creación de aplicaciones C++ de SQL incorporado con archivos de configuración” en la página 195
- “Creación de aplicaciones C++ de la API de DB2 con archivos de configuración” en la página 194
- “Creación de procedimientos almacenados C++ con archivos de configuración” en la página 196
- “Creación de funciones definidas por el usuario C++ con archivos de configuración” en la página 197
- “Creación de aplicaciones C de UNIX” en la página 161
- “Creación de rutinas C de UNIX” en la página 165
- “Creación de aplicaciones C++ de UNIX” en la página 168
- “Creación de rutinas C++ de UNIX” en la página 172

Creación de aplicaciones C++ de la API de DB2 con archivos de configuración

El archivo de configuración `api.icc`, contenido en `sqllib/samples/c` y `sqllib/samples/cpp`, le permite crear aplicaciones de la API de la DB2 utilizando C o C++ en AIX.

Procedimiento:

Para utilizar el archivo de configuración a fin de crear el programa de ejemplo de la API de DB2 `cli_info`, a partir del archivo fuente `cli_info.c`, siga estos pasos:

1. Asigne a la variable de entorno API el nombre del programa, de este modo:
 - En el shell bash o Korn:


```
export API=cli_info
```
 - En el shell C:


```
setenv API cli_info
```
2. Si su directorio de trabajo contiene un archivo `api.ics`, resultante de la creación de un programa diferente con el archivo `api.icc`, suprima el archivo `api.ics` mediante este mandato:


```
rm api.ics
```

Si para el mismo programa que ahora creará de nuevo ya existe un archivo `api.ics`, no es necesario suprimir el archivo.

3. Compile el programa de ejemplo, especificando:


```
vacbld api.icc
```

Nota: El mandato `vacbld` está incluido en VisualAge C++.

El resultado es un archivo ejecutable, `cli_info`. Puede ejecutar el programa entrando el nombre del ejecutable:

```
cli_info
```

Tareas relacionadas:

- “Creación de aplicaciones C++ de SQL incorporado con archivos de configuración” en la página 195
- “Creación de procedimientos almacenados C++ con archivos de configuración” en la página 196
- “Creación de funciones definidas por el usuario C++ con archivos de configuración” en la página 197

Creación de aplicaciones C++ de SQL incorporado con archivos de configuración

El archivo de configuración `emb.icc`, contenido en `sqllib/samples/c` y `sqllib/samples/cpp`, le permite crear aplicaciones de SQL incorporado para DB2 utilizando C o C++ en AIX.

Procedimiento:

Para utilizar el archivo de configuración a fin de crear la aplicación de SQL incorporado `tbmod`, a partir del archivo fuente `tbmod.sqc`, siga estos pasos:

1. Asigne a la variable de entorno EMB el nombre del programa, de este modo:
 - En el shell bash o Korn:

```
export EMB=tbmod
```
 - En el shell C:

```
setenv EMB tbmod
```
2. Si su directorio de trabajo contiene un archivo `emb.ics`, resultante de la creación de un programa diferente con el archivo `emb.icc`, suprima el archivo `emb.ics` mediante este mandato:

```
rm emb.ics
```

Si para el mismo programa que ahora creará de nuevo ya existe un archivo `emb.ics`, no es necesario suprimir el archivo.

3. Compile el programa de ejemplo, especificando:

```
vacbld emb.icc
```

Nota: El mandato `vacbld` está incluido en VisualAge C++.

El resultado es un archivo ejecutable, `tbmod`. Puede ejecutar el programa entrando el nombre del ejecutable:

```
tbmod
```

Tareas relacionadas:

- “Creación de aplicaciones C++ de la API de DB2 con archivos de configuración” en la página 194
- “Creación de procedimientos almacenados C++ con archivos de configuración” en la página 196
- “Creación de funciones definidas por el usuario C++ con archivos de configuración” en la página 197

Creación de procedimientos almacenados C++ con archivos de configuración

El archivo de configuración `stp.icc`, contenido en `sqllib/samples/c` y `sqllib/samples/cpp`, le permite crear procedimientos almacenados de SQL incorporado para DB2 utilizando C o C++ en AIX.

Procedimiento:

Para utilizar el archivo de configuración a fin de crear la biblioteca compartida de procedimientos almacenados de SQL incorporado, `spserver`, a partir del archivo fuente `spserver.sqc`, siga estos pasos:

1. Asigne a la variable de entorno STP el nombre del programa, de este modo:
 - En el shell bash o Korn:

```
export STP=spserver
```
 - En el shell C:

```
setenv STP spserver
```
2. Si su directorio de trabajo contiene un archivo `stp.ics`, resultante de la creación de un programa diferente con el archivo `stp.icc`, suprima el archivo `stp.ics` mediante este mandato:

```
rm stp.ics
```

Si para el mismo programa que ahora creará de nuevo ya existe un archivo `stp.ics`, no es necesario suprimir el archivo.

3. Compile el programa de ejemplo, especificando:

```
vacbld stp.icc
```

Nota: El mandato `vacbld` está incluido en VisualAge C++.

La biblioteca compartida de procedimientos almacenados se copia en el directorio `sqllib/function` del servidor.

A continuación, catalogue los procedimientos almacenados de la biblioteca compartida ejecutando el script `spcat` en el servidor:

```
spcat
```

Este script conecta con la base de datos "sample", descataloga mediante `spdrop.db2` los procedimientos almacenados que se hubieran catalogado previamente, luego los cataloga llamando a `spcreate.db2`, y finalmente desconecta de la base de datos. Puede también ejecutar los scripts `spdrop.db2` y `spcreate.db2` por separado.

A continuación, detenga y reorganice la base de datos para que se reconozca la nueva biblioteca compartida. Si es necesario, establezca la modalidad de archivo para la biblioteca compartida a fin de que la instancia de DB2 pueda acceder a la biblioteca.

Una vez creada la biblioteca compartida de procedimientos almacenados, `spserver`, puede crear la aplicación cliente, `spclient`, la cual llama a los procedimientos almacenados contenidos en la biblioteca. Puede crear `spclient` utilizando el archivo de configuración, `emb.icc`.

Para invocar los procedimientos almacenados, ejecute la aplicación cliente de ejemplo, especificando lo siguiente:

`spclient basedatos IDusuario contraseña`

donde

basedatos

Es el nombre de la base de datos a la que desea conectarse. El nombre podría ser `sample`, o su alias, u otro nombre.

IDusuario

Es un ID de usuario válido.

contraseña

Es una contraseña válida.

La aplicación cliente accede a la biblioteca compartida, `spserver`, la cual ejecuta diversas funciones de procedimiento almacenado contenidas en la base de datos del servidor. Los datos resultantes se devuelven a la aplicación cliente.

Tareas relacionadas:

- “Creación de aplicaciones C++ de SQL incorporado con archivos de configuración” en la página 195
- “Creación de aplicaciones C++ de la API de DB2 con archivos de configuración” en la página 194
- “Creación de funciones definidas por el usuario C++ con archivos de configuración” en la página 197

Creación de funciones definidas por el usuario C++ con archivos de configuración

El archivo de configuración `udf.icc`, contenido en `sqllib/samples/c` y `sqllib/samples/cpp`, le permite crear funciones definidas por el usuario utilizando C o C++ en AIX.

Procedimiento:

Para utilizar el archivo de configuración a fin de crear el programa de función definida por el usuario `udfsrv`, a partir del archivo fuente `udfsrv.c`, siga estos pasos:

1. Asigne a la variable de entorno UDF el nombre del programa, de este modo:
 - En el shell bash o Korn:

```
export UDF=udfsrv
```
 - En el shell C:

```
setenv UDF udfsrv
```
2. Si su directorio de trabajo contiene un archivo `udf.ics`, resultante de la creación de un programa diferente con el archivo `udf.icc`, suprima el archivo `udf.ics` mediante este mandato:

```
rm udf.ics
```

Si para el mismo programa que ahora creará de nuevo ya existe un archivo `udf.ics`, no es necesario suprimir el archivo.
3. Compile el programa de ejemplo, especificando:

```
vacbld udf.icc
```

Nota: El mandato `vacbld` está incluido en VisualAge C++.

La biblioteca de UDF se copia en el directorio `sqllib/function` del servidor.

Si es necesario, establezca la modalidad de archivo para la función definida por el usuario a fin de que la instancia de DB2 pueda ejecutar la función.

Una vez creado el programa `udfsrv`, puede crear la aplicación cliente, `udfcli`, que llama al programa. El programa se proporciona en versiones para CLI de DB2 y para SQL incorporado.

Puede crear el programa `udfcli` de la CLI de DB2, a partir del archivo fuente `udfcli.c`, contenido en `sqllib/samples/cli`, utilizando el archivo de configuración `cli.icc`.

Puede crear el programa `udfcli` de SQL incorporado, a partir del archivo fuente `udfcli.sqc`, contenido en `sqllib/samples/c`, utilizando el archivo de configuración `emb.icc`.

Para invocar la UDF, ejecute la aplicación solicitante de ejemplo entrando el nombre del ejecutable:

```
udfcli
```

La aplicación solicitante llama a la función `ScalarUDF` de la biblioteca `udfsrv`.

Tareas relacionadas:

- “Creación de aplicaciones C++ de SQL incorporado con archivos de configuración” en la página 195
- “Creación de aplicaciones C++ de la API de DB2 con archivos de configuración” en la página 194
- “Creación de procedimientos almacenados C++ con archivos de configuración” en la página 196

IBM COBOL Set para AIX

Configuración del compilador IBM COBOL en AIX

Debe seguir los pasos siguientes si desarrolla aplicaciones que contienen SQL incorporado y llamadas a la API de DB2, y está utilizando el compilador IBM COBOL Set para AIX.

Procedimiento:

- Cuando precompile la aplicación utilizando el mandato de procesador de línea de mandatos `db2 prep`, utilice la opción `target ibmcob`.
- No utilice caracteres de tabulación en los archivos fuente.
- Puede utilizar las palabras clave `PROCESS` y `CBL` en la primera línea de los archivos fuente para definir opciones de compilación.
- Si la aplicación sólo contiene SQL incorporado y ninguna llamada a la API de DB2, no es necesario que utilice la opción de compilación `pgmname(mixed)`. Si especifica llamadas a la API de DB2, debe utilizar la opción de compilación `pgmname(mixed)`.
- Si utiliza la opción “System/390 host data type support” del compilador IBM COBOL Set para AIX, los archivos de inclusión de DB2 correspondientes a las aplicaciones están contenidos en este directorio:

```
$HOME/sqllib/include/cobol_i
```

Si crea programas DB2 de ejemplo utilizando los archivos de script proporcionados, la vía de acceso de los archivos de inclusión especificada en los archivos de script se debe cambiar para que apunte al directorio `cobol_i` y no al directorio `cobol_a`.

Si NO utiliza la opción "System/390 host data type support" del compilador IBM COBOL Set para AIX, o está utilizando una versión anterior de este compilador, los archivos de inclusión de DB2 correspondientes a las aplicaciones están contenidos en este directorio:

```
$HOME/sql1lib/include/cobol_a
```

Especifique nombres de archivo de COPY de modo que incluyan la extensión `.cbl`, de la manera siguiente:

```
COPY "sql.cbl".
```

Conceptos relacionados:

- "Consideraciones para la instalación de COBOL en AIX" en la página 183

Tareas relacionadas:

- "Configuración del entorno de desarrollo de aplicaciones UNIX" en la página 34
- "Creación de aplicaciones IBM COBOL en AIX" en la página 199
- "Creación de rutinas IBM COBOL en AIX" en la página 202

Información relacionada:

- "Opciones de compilación y enlace para aplicaciones IBM COBOL de AIX" en la página 201
- "Opciones de compilación y enlace para rutinas IBM COBOL de AIX" en la página 204

Creación de aplicaciones IBM COBOL en AIX

DB2 proporciona scripts de creación para compilar y enlazar programas IBM COBOL de SQL incorporado y de la API de DB2. Estos scripts residen en el directorio `sql1lib/samples/cobol`, junto con programas de ejemplo que se pueden crear a partir de esos archivos.

El archivo de creación `bldapp` contiene los mandatos para crear un programa de aplicación DB2.

El primer parámetro, `$1`, especifica el nombre del archivo fuente. Éste es el único parámetro necesario para los programas que no contienen SQL incorporado. Para crear programas de SQL incorporado es necesaria una conexión con la base de datos, por lo que también se proporcionan otros tres parámetros opcionales: el segundo parámetro, `$2`, especifica el nombre de la base de datos con la que se desea conectar; el tercer parámetro, `$3`, especifica el ID de usuario correspondiente a la base de datos, y `$4` especifica la contraseña.

Para un programa de SQL incorporado, `bldapp` pasa los parámetros al script de precompilación y vinculación, `embprep`. Si no se proporciona un nombre de base de datos, se utiliza la base de datos por omisión `sample`. Los parámetros de ID de usuario y contraseña sólo son necesarios si la instancia donde se crea el programa es diferente de la instancia donde reside la base de datos.

Procedimiento:

Para crear el programa de ejemplo de SQL no incorporado, `client`, a partir del archivo fuente `client.cbl`, especifique:

```
bldapp client
```

El resultado es un archivo ejecutable, `client`. Para ejecutar el archivo ejecutable sobre la base de datos `sample`, especifique:

```
client
```

Creación y ejecución de aplicaciones de SQL incorporado

A partir del archivo fuente `updat.sqb`, puede crear la aplicación de SQL incorporado, `updat`, de tres maneras:

1. Si conecta con la base de datos "sample" en la misma instancia, especifique:

```
bldapp updat
```

2. Si conecta con otra base de datos en la misma instancia, especifique también el nombre de la base de datos:

```
bldapp updat basedatos
```

3. Si conecta con una base de datos de otra instancia, especifique también el ID de usuario y la contraseña de la instancia de base de datos:

```
bldapp updat basedatos IDusuario contraseña
```

El resultado es un archivo ejecutable, `updat`.

Existen tres maneras de ejecutar esta aplicación de SQL incorporado:

1. Si desea acceder a la base de datos `sample` en la misma instancia, especifique simplemente el nombre del ejecutable:

```
updat
```

2. Si desea acceder a otra base de datos en la misma instancia, especifique el nombre del ejecutable y el nombre de la base de datos:

```
updat basedatos
```

3. Si desea acceder a una base de datos de otra instancia, especifique el nombre del ejecutable, el nombre de la base de datos, y el ID de usuario y la contraseña de la instancia de base de datos:

```
bldapp updat basedatos IDusuario contraseña
```

Conceptos relacionados:

- "Archivos de creación" en la página 110

Tareas relacionadas:

- "Creación de rutinas IBM COBOL en AIX" en la página 202

Información relacionada:

- "Opciones de compilación y enlace para aplicaciones IBM COBOL de AIX" en la página 201
- "Ejemplos de COBOL" en la página 82

Ejemplos relacionados:

- "bldapp -- Builds AIX COBOL applications"
- "client.cbl -- How to set and query a client (IBM COBOL)"
- "embprep -- To prep and bind a COBOL embedded SQL sample on AIX"
- "updat.sqb -- How to update, delete and insert table data (IBM COBOL)"

Script de creación para aplicaciones COBOL DE IBM

```
#!/bin/sh
# SCRIPT: bldapp
# Crea aplicaciones COBOL para AIX
# Uso: bldapp <nombre_prog> [ <nombre_bd> [ <IDusuario> <contraseña> ]]

# Definir DB2PATH de modo que apunte al lugar desde donde se accederá a DB2.
# El valor por omisión es la vía de acceso de instancia estándar.
DB2PATH=$HOME/sql1lib

# Si es un programa de SQL incorporado, precompilar y vincular el programa.
if [ -f $1".sqb" ]
then
./embprep $1 $2 $3 $4
fi

# Compilar el programa de utilidad de comprobación de errores checkerr.cbl.
cob2 -qpgmname\mixed\ -qlib -I$DB2PATH/include/cobol_a \
-c checkerr.cbl

# Compilar el programa.
cob2 -qpgmname\mixed\ -qlib -I$DB2PATH/include/cobol_a \
-c $1.cbl

# Enlazar el programa.
cob2 -o $1 $1.o checkerr.o -L$DB2PATH/lib -ldb2
```

Opciones de compilación y enlace para aplicaciones IBM COBOL de AIX

A continuación se muestran las opciones de compilación y enlace que DB2 recomienda para crear aplicaciones COBOL de SQL incorporado y de la API de DB2 con el compilador COBOL Set de IBM AIX, tal como muestra el script de creación bldapp.

Opciones de compilación y enlace para bldapp	
Opciones de compilación:	
cob2	Denota el compilador IBM COBOL Set.
-qpgmname\mixed\	Indica al compilador que permita llamadas a los puntos de entrada de biblioteca utilizando nombres con mayúsculas y minúsculas mezcladas.
-qlib	Indica al compilador que procese las sentencias COPY.
-I\$DB2PATH/include/cobol_a	Especifica la ubicación de los archivos de inclusión de DB2. Por ejemplo: \$HOME/sql1lib/include/cobol_a.
-c	Hace que se efectúe solamente la compilación, no la edición de enlaces. La compilación y la edición de enlaces son pasos separados.

Opciones de compilación y enlace para bldapp	
Opciones de enlace:	
cob2	Hace que el compilador se utilice como interfaz del editor de enlaces.
-o \$1	Especifica el programa ejecutable.
\$1.o	Especifica el archivo objeto del programa.
checkerr.o	Hace que se incluya el archivo objeto de programa de utilidad para la comprobación de errores.
-L\$DB2PATH/lib	Especifica la ubicación de las bibliotecas compartidas de tiempo de ejecución de DB2. Por ejemplo: \$HOME/sqllib/lib. Si el usuario no especifica la opción -L, el compilador utiliza esta vía de acceso: /usr/lib:/lib.
-ldb2	Enlazar con la biblioteca del gestor de bases de datos.
Consulte la documentación del compilador para conocer otras opciones de compilador.	

Tareas relacionadas:

- “Creación de aplicaciones IBM COBOL en AIX” en la página 199

Información relacionada:

- “Opciones de compilación y enlace para rutinas IBM COBOL de AIX” en la página 204

Ejemplos relacionados:

- “bldapp -- Builds AIX COBOL applications”

Creación de rutinas IBM COBOL en AIX

DB2 proporciona scripts de creación para compilar y enlazar programas COBOL de SQL incorporado y de la API de DB2. Estos scripts residen en el directorio `sqllib/samples/cobol`, junto con programas de ejemplo que se pueden crear a partir de esos archivos.

El `bldrtn`, que reside en `sqllib/samples/cobol`, contiene los mandatos para crear rutinas (procedimientos almacenados). El script compila las rutinas y crea una biblioteca compartida que puede ser llamada por una aplicación cliente.

El primer parámetro, `$1`, especifica el nombre del archivo fuente. El segundo parámetro, `$2`, especifica el nombre de la base de datos a la que desea conectarse. Debido a que la biblioteca compartida se debe crear en la misma instancia donde reside la base de datos, no hay parámetros para el ID de usuario ni la contraseña.

Sólo es obligatorio el primer parámetro, el nombre del archivo fuente. El script utiliza el nombre del archivo fuente, `$1`, como nombre de la biblioteca compartida. El nombre de la base de datos es opcional. Si no se proporciona un nombre de base de datos, el programa utiliza la base de datos por omisión `sample`.

Procedimiento:

Si conecta con la base de datos “sample”, especifique lo siguiente para crear el programa de ejemplo `outsrv` a partir del archivo fuente `outsrv.sqb`:

```
bldrtn outsrv
```

Si conecta con otra base de datos, especifique también el nombre de la base de datos:

```
bldrtn outsrv basedatos
```

El archivo de script copia la biblioteca compartida en el directorio sqllib/function del servidor.

Una vez creada la biblioteca compartida de la rutina, outsrv, puede crear la aplicación cliente, outcli, la cual llama a la rutina contenida en la biblioteca. Puede crear outcli utilizando el archivo de script bldapp.

Para invocar la rutina, ejecute la aplicación cliente de ejemplo, especificando lo siguiente:

```
outcli basedatos IDusuario contraseña
```

donde

basedatos

Es el nombre de la base de datos a la que desea conectarse. El nombre podría ser sample, o su alias remoto, u otro nombre.

IDusuario

Es un ID de usuario válido.

contraseña

Es una contraseña válida correspondiente al ID de usuario.

La aplicación cliente accede a la biblioteca compartida, outsrv, la cual ejecuta la rutina del mismo nombre contenida en la base de datos del servidor, y devuelve los datos resultantes a la aplicación cliente.

Conceptos relacionados:

- “Archivos de creación” en la página 110

Tareas relacionadas:

- “Creación de aplicaciones IBM COBOL en AIX” en la página 199

Información relacionada:

- “Opciones de compilación y enlace para rutinas IBM COBOL de AIX” en la página 204
- “Ejemplos de COBOL” en la página 82

Ejemplos relacionados:

- “bldrtn -- Builds AIX COBOL routines (stored procedures)”
- “embprep -- To prep and bind a COBOL embedded SQL sample on AIX”
- “outcli.sqb -- Call stored procedures using the SQLDA structure (IBM COBOL)”
- “outsrv.sqb -- Demonstrates stored procedures using the SQLDA structure (IBM COBOL)”

Script de creación para rutinas COBOL de IBM

```
#!/bin/sh
# SCRIPT: bldrtn
# Crea rutinas COBOL para AIX (procedimientos almacenados)
```

```

# Uso: bldrtn <nombre_prog> [ <nombre_bd> ]

# Definir DB2PATH de modo que apunte al lugar desde donde se accederá a DB2.
# El valor por omisión es la vía de acceso de instancia estándar.
DB2PATH=$HOME/sqllib

# Precompilar y vincular el programa.
./embprep $1 $2

# Compilar el programa de utilidad de comprobación de errores checkerr.cbl.
cob2 -qpgmname\mixed\ -qlib -I$DB2PATH/include/cobol_a \
-c checkerr.cbl

# Compilar el programa.
cob2 -qpgmname\mixed\ -qlib -c -I$DB2PATH/include/cobol_a $1.cbl

# Enlazar el programa con el archivo de exportación $1.exp,
# con lo que se creará la biblioteca compartida $1
cob2 -o $1 $1.o checkerr.o -bnoentry -bE:$1.exp \
-L$DB2PATH/lib -ldb2

# Copiar la biblioteca compartida en el directorio sqllib/function de la
# instancia de DB2.
# Esto supone que el usuario tiene permiso de escritura para ese directorio.
rm -f $DB2PATH/function/$1
cp $1 $DB2PATH/function

```

Opciones de compilación y enlace para rutinas IBM COBOL de AIX

La tabla siguiente muestra las opciones de compilación y enlace que DB2 recomienda para crear rutinas COBOL (procedimientos almacenados) con el compilador COBOL Set de IBM para AIX, tal como muestra el script de creación bldrtn.

Opciones de compilación y enlace para bldrtn	
Opciones de compilación:	
cob2	Denota el compilador IBM COBOL Set.
-qpgmname\mixed\	Indica al compilador que permita llamadas a los puntos de entrada de biblioteca utilizando nombres con mayúsculas y minúsculas mezcladas.
-qlib	Indica al compilador que procese las sentencias COPY.
-c	Hace que se efectúe solamente la compilación, no la edición de enlaces. La compilación y la edición de enlaces son pasos separados.
-I\$DB2PATH/include/cobol_a	Especifica la ubicación de los archivos de inclusión de DB2. Por ejemplo: \$HOME/sqllib/include/cobol_a.

Opciones de compilación y enlace para bldrtn

Opciones de enlace:

- cob2** Denota la utilización del compilador para realizar la edición de enlaces.
- o \$1** Especifica la salida como archivo de biblioteca compartida.
- \$1.o** Especifica el archivo objeto del procedimiento almacenado.
- checkerr.o**
Hace que se incluya el archivo objeto de programa de utilidad para la comprobación de errores.
- bnoentry**
Denota que no se especifique el punto de entrada por omisión de la biblioteca compartida.
- bE:\$1.exp**
Especifica un archivo de exportación. El archivo de exportación contiene una lista de los procedimientos almacenados.
- L\$DB2PATH/lib**
Especifica la ubicación de las bibliotecas compartidas de tiempo de ejecución de DB2. Por ejemplo: \$HOME/sql11ib/lib. Si el usuario no especifica la opción -L, el compilador utiliza esta vía de acceso: /usr/lib:/lib.
- ldb2** Enlazar con la biblioteca del gestor de bases de datos.

Consulte la documentación del compilador para conocer otras opciones de compilador.

Tareas relacionadas:

- “Creación de rutinas IBM COBOL en AIX” en la página 202

Información relacionada:

- “Opciones de compilación y enlace para aplicaciones IBM COBOL de AIX” en la página 201

Ejemplos relacionados:

- “bldrtn -- Builds AIX COBOL routines (stored procedures)”

COBOL para Micro Focus

Para obtener información sobre cómo crear aplicaciones COBOL para Micro Focus en sistemas operativos UNIX soportados, consulte el apartado “Creación de aplicaciones Micro Focus COBOL de UNIX” en la página 176. Para obtener información sobre cómo crear rutinas COBOL para Micro Focus en sistemas operativos UNIX soportados, consulte el apartado “Creación de rutinas Micro Focus COBOL de UNIX” en la página 177.

Configuración del compilador Micro Focus COBOL en AIX

Siga los pasos siguientes si desarrolla aplicaciones que contienen SQL incorporado y llamadas a la API de DB2 y utiliza el compilador Micro Focus COBOL.

Procedimiento:

- Cuando precompile la aplicación utilizando el mandato db2 prep del procesador de línea de mandatos, utilice la opción target mfcob.
- Debe incluir el directorio del archivo COPY de DB2 para COBOL en la variable de entorno COBCOPY de Micro Focus COBOL. La variable de entorno COBCPY

especifica la ubicación de los archivos COPY. Los archivos COPY de DB2 correspondientes a Micro Focus COBOL residen en sqllib/include/cobol_mf, dentro del directorio de la instancia de la base de datos.

Para incluir el directorio en la variable de entorno, especifique:

- En el shell bash o Korn:
export COBCPY=\$COBCPY:\$HOME/sqllib/include/cobol_mf
- En el shell C:
setenv COBCPY \$COBCPY:\$HOME/sqllib/include/cobol_mf

Nota: Puede definir COBCPY dentro del archivo .profile o .login.

Conceptos relacionados:

- “Consideraciones para la instalación de COBOL en AIX” en la página 183

Tareas relacionadas:

- “Configuración del entorno de desarrollo de aplicaciones UNIX” en la página 34
- “Creación de aplicaciones Micro Focus COBOL de UNIX” en la página 176
- “Creación de rutinas Micro Focus COBOL de UNIX” en la página 177

Información relacionada:

- “Opciones de compilación y enlace para aplicaciones Micro Focus COBOL de AIX” en la página 207
- “Opciones de compilación y enlace para rutinas Micro Focus COBOL de AIX” en la página 208

Script de creación para aplicaciones COBOL para Micro Focus

```
#!/bin/sh
# SCRIPT: bldapp
# Crea aplicaciones COBOL para Micro Focus de AIX
# Uso: bldapp <nombre_prog> [ <nombre_bd> [ <IDusuario> <contraseña> ]]

# Definir DB2PATH de modo que apunte al lugar desde donde se accederá a DB2.
# El valor por omisión es la vía de acceso de instancia estándar.
DB2PATH=$HOME/sqllib

# Definir COBCPY de forma que incluya el directorio de los archivos COPY de DB2.
COBCPY=$COBCPY:$DB2PATH/include/cobol_mf

# Si es un programa de SQL incorporado, precompilar y vincular el programa.
if [ -f $1".sqb" ]
then
./embprep $1 $2 $3 $4
fi

# Compilar el programa de utilidad de comprobación de errores checkerr.cb1.
cob -c -x checkerr.cb1

# Compilar el programa.
cob -c -x $1.cb1

# Enlazar el programa.
cob -x -o $1 $1.o checkerr.o -L$DB2PATH/lib -ldb2 -ldb2gmf
```

Opciones de compilación y enlace para aplicaciones Micro Focus COBOL de AIX

La tabla siguiente muestra las opciones de compilación y enlace que DB2 recomienda para crear aplicaciones COBOL de SQL incorporado y de la API de DB2 con el compilador MicroFocus COBOL para AIX, tal como muestra el script de creación bldapp.

Opciones de compilación y enlace para bldapp	
Opciones de compilación:	
cob	Denota el compilador COBOL.
-c	Hace que se efectúe solamente la compilación, no la edición de enlaces.
-x	Cuando se utiliza con -c , crea un archivo objeto.
Opciones de enlace:	
cob	Hace que el compilador se utilice como interfaz del editor de enlaces.
-x	Crea un programa ejecutable.
-o \$1	Especifica el programa ejecutable.
\$1.o	Especifica el archivo objeto del programa.
-L\$DB2PATH/lib	Especifica la ubicación de las bibliotecas compartidas de tiempo de ejecución de DB2. Por ejemplo: \$HOME/sql1lib/lib. Si el usuario no especifica la opción -L , el compilador utiliza esta vía de acceso: /usr/lib:/lib.
-ldb2	Enlazar con la biblioteca DB2.
-ldb2gmf	Establece un enlace con la biblioteca del gestor de excepciones de DB2 correspondiente a Micro Focus COBOL.
Consulte la documentación del compilador para conocer otras opciones de compilador.	

Tareas relacionadas:

- “Creación de aplicaciones Micro Focus COBOL de UNIX” en la página 176

Información relacionada:

- “Opciones de compilación y enlace para rutinas Micro Focus COBOL de AIX” en la página 208

Ejemplos relacionados:

- “bldapp -- Builds AIX Micro Focus COBOL applications”

Script de creación para rutinas COBOL para Micro Focus

```
#!/bin/sh
# SCRIPT: bldrtn
# Crea rutinas COBOL para Micro Focus de AIX (procedimientos almacenados)
# Uso: bldrtn <nombre_prog> [ <nombre_bd> ]

# Definir DB2PATH de modo que apunte al lugar desde donde se accederá a DB2.
# El valor por omisión es la vía de acceso de instancia estándar.
DB2PATH=$HOME/sql1lib

# Definir COBCPY de forma que incluya el directorio de los archivos COPY de DB2.
COBCPY=$COBCPY:$DB2PATH/include/cobol_mf
```

```

# Precompilar y vincular el programa.
./embprep $1 $2

# Compilar el programa.
cob -c -x $1.cbl

# Enlazar el programa.
cob -x -o $1 $1.o -Q -bnoentry \
-Q -bI:$DB2PATH/lib/db2g.imp -L$DB2PATH/lib -ldb2 -ldb2gmf

# Copiar la biblioteca compartida en el subdirectorío sqllib/function.
# Nota: el usuario debe tener permiso de escritura sobre este directorio.
rm -f $DB2PATH/function/$1
cp $1 $DB2PATH/function

```

Opciones de compilación y enlace para rutinas Micro Focus COBOL de AIX

La tabla siguiente muestra las opciones de compilación y enlace que DB2 recomienda para crear rutinas COBOL (procedimientos almacenados) con el compilador Micro Focus COBOL para AIX, tal como muestra el script de creación bldrtn.

Opciones de compilación y enlace para bldrtn	
Opciones de compilación:	
cob	Denota el compilador COBOL.
-c	Hace que se efectúe solamente la compilación, no la edición de enlaces. La compilación y la edición de enlaces son pasos separados.
-x	Compila y crea un módulo objeto cuando se utiliza con la opción -c.
Opciones de enlace:	
cob	Hace que el compilador se utilice como interfaz del editor de enlaces.
-x	Hace que se cree una biblioteca compartida.
-o \$1	Especifica el programa ejecutable.
\$1.o	Especifica el archivo objeto del programa.
-Q -bnoentry	Denota que no se especifique el punto de entrada por omisión de la biblioteca compartida.
-Q -bI:\$DB2PATH/lib/db2g.imp	Proporciona una lista de puntos de entrada correspondientes a la biblioteca de aplicaciones de DB2.
-L\$DB2PATH/lib	Especifica la ubicación de las bibliotecas compartidas de tiempo de ejecución de DB2. Por ejemplo: \$HOME/sql1lib/lib. Si el usuario no especifica la opción -L, el compilador utiliza esta vía de acceso: /usr/lib:/lib.
-ldb2	Enlazar con la biblioteca DB2.
-ldb2gmf	Establece un enlace con la biblioteca del gestor de excepciones de DB2 correspondiente a Micro Focus COBOL.
Consulte la documentación del compilador para conocer otras opciones de compilador.	

Tareas relacionadas:

- “Creación de rutinas Micro Focus COBOL de UNIX” en la página 177

Información relacionada:

- “Opciones de compilación y enlace para aplicaciones Micro Focus COBOL de AIX” en la página 207

Ejemplos relacionados:

- “bldrtn -- Builds AIX Micro Focus COBOL routines (stored procedures)”

REXX

Creación de aplicaciones REXX en AIX

Esta sección le muestra cómo crear aplicaciones REXX en AIX. DB2 para AIX da soporte a REXX estándar y a Object REXX. Object REXX es una versión del lenguaje REXX orientada a objetos. Se han añadido extensiones orientadas a objetos al REXX estándar, pero no se han modificado sus funciones e instrucciones existentes. El programa intérprete de Object REXX es una versión mejorada de su predecesor, e incluye soporte adicional para:

- Clases, objetos y métodos
- Gestión de mensajes y polimorfismo
- Herencia simple y múltiple

Object REXX es totalmente compatible con el REXX estándar. En esta sección, siempre que se utiliza el término REXX se hace referencia a todas las versiones de REXX, incluido Object REXX.

No es necesario precompilar ni enlazar los programas REXX.

Procedimiento:

Para ejecutar programas REXX/SQL de DB2 en AIX, debe definir la variable de entorno LIBPATH de forma que especifique lib dentro del directorio de instalación de DB2.

En el shell bash o Korn, especifique:

```
export LIBPATH=$LIBPATH:/lib:/usr/lib:/usr/opt/db2_08_01/lib
```

En el shell C, especifique:

```
setenv LIBPATH $LIBPATH:/lib:/usr/lib:/usr/opt/db2_08_01/lib
```

En AIX, el archivo de la aplicación puede tener una extensión de archivo cualquiera. Puede ejecutar la aplicación utilizando cualquiera de estos dos métodos:

1. En el indicador de mandatos del shell, escriba `rexx nombre`, donde *nombre* es el nombre del programa REXX (incluida la extensión, si existe).
2. Si la primera línea del programa REXX contiene un “número mágico”, (`#!`)e identifica el directorio donde reside el intérprete REXX/6000, puede ejecutar el programa REXX entrando su nombre en el indicador de mandatos del shell. Por ejemplo, si el archivo del intérprete REXX/6000 está en el directorio `/usr/bin`, incluya lo siguiente como primera línea del programa REXX:

```
#! /usr/bin/rexx
```

A continuación, haga que el programa sea ejecutable entrando el mandato siguiente en el indicador de mandatos del shell:

```
chmod +x nombre
```

Ejecute el programa REXX entrando su nombre de archivo en el indicador de mandatos del shell.

Los programas REXX de ejemplo están contenidos en el directorio `sql11b/samples/rexx`. Para ejecutar el programa REXX de ejemplo `updat.cmd`, especifique:

```
updat.cmd
```

Tareas relacionadas:

- “Configuración del entorno de desarrollo de aplicaciones UNIX” en la página 34

Información relacionada:

- “Ejemplos de REXX” en la página 100

Capítulo 11. HP-UX

HP-UX C	211	Creación de aplicaciones multihebra C++ en HP-UX	222
Script de creación para aplicaciones C	211	COBOL para Micro Focus	222
Opciones de compilación y enlace para aplicaciones C de HP-UX	212	Configuración del compilador Micro Focus COBOL en HP-UX.	223
Script de creación para rutinas C.	214	Script de creación para aplicaciones COBOL para Micro Focus	223
Opciones de compilación y enlace para rutinas C de HP-UX.	215	Opciones de compilación y enlace para aplicaciones Micro Focus COBOL de HP-UX	224
Creación de aplicaciones multihebra C en HP-UX	216	Script de creación para rutinas COBOL para Micro Focus	224
HP-UX C++.	216	Opciones de compilación y enlace para rutinas Micro Focus COBOL de HP-UX	225
Script de creación para aplicaciones C++	216		
Opciones de compilación y enlace para aplicaciones C++ de HP-UX	218		
Script de creación para rutinas C++	219		
Opciones de compilación y enlace para rutinas C++ de HP-UX.	220		

Este capítulo proporciona información detallada para crear aplicaciones DB2 en HP-UX. Para conocer la información más reciente sobre el desarrollo de aplicaciones DB2, consulte la página Web sobre desarrollo de aplicaciones DB2 situada en:

<http://www.ibm.com/software/data/db2/udb/ad>

HP-UX C

La creación de información para las aplicaciones y rutinas de la CLI de DB2 CLI se encuentra en el manual *CLI Guide and Reference*.

Para obtener información sobre cómo crear aplicaciones C en sistemas operativos UNIX soportados, consulte el apartado "Creación de aplicaciones C de UNIX" en la página 161. Para obtener información sobre cómo crear rutinas C en sistemas operativos UNIX soportados, consulte el apartado "Creación de rutinas C de UNIX" en la página 165.

Script de creación para aplicaciones C

```
#!/bin/sh
# SCRIPT: bldapp
# Crea una aplicación C de HP-UX
# Uso: bldapp <nombre_prog> [ <nombre_bd> [ <IDusuario> <contraseña> ] ]

# Definir DB2PATH de modo que apunte al lugar desde donde se accederá a DB2.
# El valor por omisión es la vía de acceso de instancia estándar.
DB2PATH=$HOME/sqllib

# Determinar la plataforma HP y definir opciones correctas de compilación/enlace
hpplat=`uname -m`
bitwidth=`LANG=C db2level | awk '/bits/{print $5}'`
if [ $hpplat = "ia64" ]; then
    if [ $bitwidth = "\"64\"" ]; then
        EXTRA_CFLAG="+DD64"
        LIB="lib"
    else
        EXTRA_CFLAG="+DD32"
        LIB="lib32"
    fi
fi
```

```

else
    if [ $bitwidth = "\"64\"" ]; then
        EXTRA_CFLAG="+DA2.0W"
        LIB="lib"
    else
        EXTRA_CFLAG=
        LIB="lib32"
    fi
fi

# Se recomienda la vía de tiempo de ejecución para todas las aplicaciones.
# Si tiene necesidad de usar SHLIB_PATH o LD_LIBRARY_PATH, desestablezca
# la variable RUNTIME comentando la línea siguiente.
RUNTIME=true

if [ "$RUNTIME" != "" ]
then
    EXTRA_LFLAG="-Wl,+b$DB2PATH/$LIB"
else
    EXTRA_LFLAG=""
fi

# Si es un programa de SQL incorporado, precompilar y vincular el programa.
# Nota: algunos archivos .sqc no contienen SQL sino que enlazan en
# utilemb.sqc, por lo que debe ignorar este aviso si lo recibe:
# SQL0053W No se ha encontrado ninguna sentencia SQL en el programa.
if [ -f $1".sqc" ]
then
    ./embprep $1 $2 $3 $4
# Compilar el programa de utilidad de comprobación de errores utilemb.c.
    cc $EXTRA_CFLAG -Ae -I$DB2PATH/include -c utilemb.c
else
# Compilar el programa de utilidad de comprobación de errores utilapi.c.
    cc $EXTRA_CFLAG -Ae -I$DB2PATH/include -c utilapi.c
fi

# Compilar el programa.
cc $EXTRA_CFLAG -Ae -I$DB2PATH/include -c $1.c

if [ -f $1".sqc" ]
then
# Enlazar el programa con utilemb.o.
    cc $EXTRA_CFLAG -o $1 $1.o utilemb.o $EXTRA_LFLAG -L$DB2PATH/$LIB -ldb2
else
# Enlazar el programa con utilapi.o.
    cc $EXTRA_CFLAG -o $1 $1.o utilapi.o $EXTRA_LFLAG -L$DB2PATH/$LIB -ldb2
fi

```

Opciones de compilación y enlace para aplicaciones C de HP-UX

La tabla siguiente muestra las opciones de compilación y enlace que DB2 recomienda para crear aplicaciones C de SQL incorporado y de la API de DB2 con el compilador C de HP-UX, tal como muestra el script de creación bldapp.

Opciones de compilación y enlace para bldapp

Opciones de compilación:

cc Denota el compilador C.

\$EXTRA_CFLAG

Si la plataforma HP-UX es IA64 y está habilitado el soporte de 64 bits, este distintivo contiene el valor **+DD64**; si está habilitado el soporte de 32 bits, contiene el valor **+DD32**. Si la plataforma HP-UX es PA-RISC y está habilitado el soporte de 64 bits, contiene el valor **+DA2.0W**. Para el soporte de 32 bits en una plataforma PA-RISC, este distintivo no contiene ningún valor.

+DD64 Se debe utilizar para generar código de 64 bits para HP-UX en IA64.

+DD32 Se debe utilizar para generar código de 32 bits para HP-UX en IA64.

+DA2.0W

Se debe utilizar para generar código de 64 bits para HP-UX en PA-RISC.

-Ae Habilita la modalidad ampliada ANSI de HP.

-I\$DB2PATH/include

Especifica la ubicación de los archivos de inclusión de DB2.

-c Hace que se efectúe solamente la compilación, no la edición de enlaces. La compilación y la edición de enlaces son pasos separados.

Opciones de enlace:

cc Hace que el compilador se utilice como interfaz del editor de enlaces.

\$EXTRA_CFLAG

Si la plataforma HP-UX es IA64 y está habilitado el soporte de 64 bits, este distintivo contiene el valor **+DD64**; si está habilitado el soporte de 32 bits, contiene el valor **+DD32**. Si la plataforma HP-UX es PA-RISC y está habilitado el soporte de 64 bits, contiene el valor **+DA2.0W**. Para el soporte de 32 bits en una plataforma PA-RISC, este distintivo no contiene ningún valor.

+DD64 Se debe utilizar para generar código de 64 bits para HP-UX en IA64.

+DD32 Se debe utilizar para generar código de 32 bits para HP-UX en IA64.

+DA2.0W

Se debe utilizar para generar código de 64 bits para HP-UX en PA-RISC.

-o \$1 Especifica el ejecutable.

\$1.o Especifica el archivo objeto del programa.

utilemb.o

Si es un programa de SQL incorporado, incluir el archivo objeto del programa de utilidad de SQL incorporado para la comprobación de errores.

utilapi.o

Si es un programa de SQL no incorporado, incluir el archivo objeto del programa de utilidad de la API de DB2 para la comprobación de errores.

\$EXTRA_LFLAG

Especificar la vía de acceso de tiempo de ejecución. Si se establece, para 32 bits contiene el valor **-Wl,+b\$HOME/sql1lib/lib32**, y para 64 bits contiene **-Wl,+b\$HOME/sql1lib/lib**. Si no se establece, no contiene ningún valor.

-L\$DB2PATH/\$LIB

Especifica la ubicación de las bibliotecas compartidas de tiempo de ejecución de DB2. Para 32 bits: **\$HOME/sql1lib/lib32**; para 64 bits: **\$HOME/sql1lib/lib**.

-ldb2 Enlazar con la biblioteca DB2.

Consulte la documentación del compilador para conocer otras opciones de compilador.

Tareas relacionadas:

- “Creación de aplicaciones C de UNIX” en la página 161

Ejemplos relacionados:

- “bldapp -- Builds HP-UX C applications (C)”

Script de creación para rutinas C

```
#!/bin/sh
# SCRIPT: bldrtn
# Crea rutinas C de HP-UX (procedimientos almacenados y UDF)
# Uso: bldrtn <nombre_prog> [ <nombre_bd> ]

# Definir DB2PATH de modo que apunte al lugar desde donde se accederá a DB2.
# El valor por omisión es la vía de acceso de instancia estándar.
DB2PATH=$HOME/sqllib

# Determinar la plataforma HP y definir opciones correctas de compilación/enlace
hpplat=`uname -m`
bitwidth=`LANG=C db2level | awk '/bits/{print $5}'`
if [ $hpplat = "ia64" ]; then
    if [ $bitwidth = "\"64\"" ]; then
        EXTRA_CFLAG="+DD64"
        LIB="lib"
    else
        EXTRA_CFLAG="+DD32"
        LIB="lib32"
    fi
else
    if [ $bitwidth = "\"64\"" ]; then
        EXTRA_CFLAG="+DA2.0W"
        LIB="lib"
    else
        EXTRA_CFLAG=
        LIB="lib32"
    fi
fi

# Se recomienda la vía de tiempo de ejecución para todas las aplicaciones.
# Si tiene necesidad de usar SHLIB_PATH o LD_LIBRARY_PATH, desestablezca
# la variable RUNTIME comentando la línea siguiente.
RUNTIME=true

if [ "$RUNTIME" != "" ]
then
    EXTRA_LFLAG="+b$DB2PATH/$LIB"
else
    EXTRA_LFLAG=""
fi

# Si es un programa de SQL incorporado, precompilar y vincular el programa.
if [ -f $1".sqc" ]
then
    ./embprep $1 $2
fi

# Compilar el programa.
cc $EXTRA_CFLAG +u1 +z -Ae -I$DB2PATH/include \
-D_POSIX_C_SOURCE=199506L -c $1.c

# Enlazar el programa para crear una biblioteca compartida
ld -b -o $1 $1.o $EXTRA_LFLAG -L$DB2PATH/$LIB -ldb2 -lpthread

# Copiar la biblioteca compartida en el subdirectorío sqllib/function.
# Nota: el usuario debe tener permiso de escritura sobre este directorio.
rm -f $DB2PATH/function/$1
cp $1 $DB2PATH/function
```

Opciones de compilación y enlace para rutinas C de HP-UX

La tabla siguiente muestra las opciones de compilación y enlace que DB2 recomienda para crear rutinas en C (procedimientos almacenados y funciones definidas por el usuario) con el compilador C de HP-UX, tal como muestra el script de creación bldrtn.

Opciones de compilación y enlace para bldrtn	
Opciones de compilación:	
cc	Denota el compilador C.
\$EXTRA_CFLAG	Si la plataforma HP-UX es IA64 y está habilitado el soporte de 64 bits, este distintivo contiene el valor +DD64 ; si está habilitado el soporte de 32 bits, contiene el valor +DD32 . Si la plataforma HP-UX es PA-RISC y está habilitado el soporte de 64 bits, contiene el valor +DA2.0W . Para el soporte de 32 bits en una plataforma PA-RISC, este distintivo no contiene ningún valor.
	+DD64 Se debe utilizar para generar código de 64 bits para HP-UX en IA64.
	+DD32 Se debe utilizar para generar código de 32 bits para HP-UX en IA64.
	+DA2.0W
	Se debe utilizar para generar código de 64 bits para HP-UX en PA-RISC.
+u1	Permitir el acceso a datos no alineados. Sólo si la aplicación utiliza datos no alineados.
+z	Generar código independiente de la posición.
-Ae	Habilita la modalidad ampliada ANSI de HP.
-I\$DB2PATH/include	Especifica la ubicación de los archivos de inclusión de DB2. Por ejemplo: -I\$DB2PATH/include.
-D_POSIX_C_SOURCE=199506L	Opción de la biblioteca de hebras de POSIX que asegura que esté definido <code>_REENTRANT</code> , opción necesaria porque las rutinas se pueden ejecutar en el mismo proceso que otras rutinas (<code>THREADSAFE</code>) o en el propio motor (<code>NOT FENCED</code>).
-c	Hace que se efectúe solamente la compilación, no la edición de enlaces. La compilación y la edición de enlaces son pasos separados.
Opciones de enlace:	
ld	Denota la utilización del enlazador para la edición de enlaces.
-b	Crear una biblioteca compartida en lugar de un ejecutable normal.
-o \$1	Especifica la salida como archivo de biblioteca compartida.
\$1.o	Especifica el archivo objeto del programa.
\$EXTRA_LFLAG	Especificar la vía de acceso de tiempo de ejecución. Si se establece, para 32 bits contiene el valor +b\$HOME/sqllib/lib32 , y para 64 bits contiene +b\$HOME/sqllib/lib . Si no se establece, no contiene ningún valor.
-L\$DB2PATH/\$LIB	Especifica la ubicación de las bibliotecas compartidas de tiempo de ejecución de DB2. Para 32 bits: <code>\$HOME/sqllib/lib32</code> ; para 64 bits: <code>\$HOME/sqllib/lib</code> .
-ldb2	Enlazar con la biblioteca DB2.
-lpthread	Establece un enlace con la biblioteca de hebras de POSIX.
Consulte la documentación del compilador para conocer otras opciones de compilador.	

Tareas relacionadas:

- “Creación de rutinas C de UNIX” en la página 165

Ejemplos relacionados:

- “bldrtn -- Builds HP-UX C routines (stored procedures and UDFs) (C)”

Creación de aplicaciones multihebra C en HP-UX

HP-UX proporciona una biblioteca de hebras de POSIX y una biblioteca de hebras de DCE. DB2 sólo da soporte a las aplicaciones multihebra que hacen uso de la biblioteca de hebras de POSIX.

En HP-UX, las aplicaciones multihebra deben estar definidas como reentrantes (`_REENTRANT`) para su compilación. La documentación de HP-UX recomienda compilar utilizando `-D_POSIX_C_SOURCE=199506L`. Esto asegurará también que `_REENTRANT` esté definido. También es necesario enlazar las aplicaciones utilizando la opción `-lpthread`.

El archivo de script `bldmt` contiene los mandatos para crear aplicaciones multihebra. Además de las opciones especificadas anteriormente, las opciones de compilación y enlace son las mismas que las utilizadas para el archivo de script de SQL incorporado, `bldapp`.

Procedimiento:

Para crear el programa de ejemplo `dbthrs`, a partir del archivo fuente `dbthrs.sqc`, especifique:

```
bldmt dbthrs
```

El resultado es un archivo ejecutable, `dbthrs`. Para ejecutar el archivo ejecutable sobre la base de datos `sample`, escriba el nombre del ejecutable:

```
dbthrs
```

Conceptos relacionados:

- “Archivos de creación” en la página 110

Información relacionada:

- “Ejemplos de C” en la página 71
- “Opciones de compilación y enlace para aplicaciones C de HP-UX” en la página 212

Ejemplos relacionados:

- “`bldmt -- Builds HP-UX C multi-threaded applications (C)`”
- “`dbthrs.sqc -- How to use multiple context APIs on UNIX (C)`”
- “`embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs (C)`”

HP-UX C++

Para obtener información sobre cómo crear aplicaciones C++ en sistemas operativos UNIX soportados, consulte el apartado “Creación de aplicaciones C++ de UNIX” en la página 168. Para obtener información sobre cómo crear rutinas C++ en sistemas operativos UNIX soportados, consulte el apartado “Creación de rutinas C++ de UNIX” en la página 172.

Script de creación para aplicaciones C++

```
#!/bin/sh
# SCRIPT: bldapp
# Crea aplicaciones C++ de HP-UX
# Uso: bldapp <nombre_prog> [ <nombre_bd> [ <IDusuario> <contraseña> ] ]
```



```

# Definir DB2PATH de modo que apunte al lugar desde donde se accederá a DB2.
# El valor por omisión es la vía de acceso de instancia estándar.
DB2PATH=$HOME/sqllib

# Determinar la plataforma HP y definir opciones correctas de
compilación/enlace
hpplat=`uname -m`
bitwidth=`LANG=C db2level | awk '/bits/{print $5}'`
if [ $hpplat = "ia64" ]; then
    if [ $bitwidth = "\"64\"" ]; then
        EXTRA_CFLAG="+DD64 -AA"
        LIB="lib"
    else
        EXTRA_CFLAG="+DD32 -AA"
        LIB="lib32"
    fi
else
    if [ $bitwidth = "\"64\"" ]; then
        EXTRA_CFLAG="+DA2.0W"
        LIB="lib"
    else
        EXTRA_CFLAG=
        LIB="lib32"
    fi
fi

# Se recomienda la vía de tiempo de ejecución para todas las aplicaciones.
# Si tiene necesidad de usar SHLIB_PATH o LD_LIBRARY_PATH, desestablezca
# la variable RUNTIME comentando la línea siguiente.
RUNTIME=true

if [ "$RUNTIME" != "" ]
then
    EXTRA_LFLAG="-Wl,+b$DB2PATH/$LIB"
else
    EXTRA_LFLAG=""
fi

# Si es un programa de SQL incorporado, precompilar y vincular el programa.
# Nota: algunos archivos .sqc no contienen SQL sino que enlazan en
# utilemb.sqc, por lo que debe ignorar este aviso si lo recibe:
# SQL0053W No se ha encontrado ninguna sentencia SQL en el programa.
if [ -f $1".sqc" ]
then
    ./embprep $1 $2 $3 $4
# Compilar el programa de utilidad de comprobación de errores utilemb.C.
aCC $EXTRA_CFLAG -ext -I$DB2PATH/include -c utilemb.C
else
# Compilar el programa de utilidad de comprobación de errores utilapi.C.
aCC $EXTRA_CFLAG -ext -I$DB2PATH/include -c utilapi.C
fi

# Compilar el programa.
aCC $EXTRA_CFLAG -ext -I$DB2PATH/include -c $1.C

if [ -f $1".sqc" ]
then
# Enlazar el programa con utilemb.o.
aCC $EXTRA_CFLAG -o $1 $1.o utilemb.o $EXTRA_LFLAG -L$DB2PATH/$LIB -ldb2
else
# Enlazar el programa con utilapi.o.
aCC $EXTRA_CFLAG -o $1 $1.o utilapi.o $EXTRA_LFLAG -L$DB2PATH/$LIB -ldb2
fi

```

Opciones de compilación y enlace para aplicaciones C++ de HP-UX

La tabla siguiente muestra las opciones de compilación y enlace que DB2 recomienda para crear aplicaciones en C++ de SQL incorporado y de la API de DB2 con el compilador C++ de HP-UX, tal como muestra el script de creación bldapp.

Opciones de compilación y enlace para bldapp	
Opciones de compilación:	
aCC	Denota el compilador aC++ de HP.
\$EXTRA_CFLAG	Si la plataforma HP-UX es IA64 y está habilitado el soporte de 64 bits, este distintivo contiene el valor +DD64 -AA ; si está habilitado el soporte de 32 bits, contiene el valor +DD32 -AA . Si la plataforma HP-UX es PA-RISC y está habilitado el soporte de 64 bits, contiene el valor +DA2.0W . Para el soporte de 32 bits en una plataforma PA-RISC, este distintivo no contiene ningún valor.
+DD64	Se debe utilizar para generar código de 64 bits para HP-UX en IA64.
+DD32	Se debe utilizar para generar código de 32 bits para HP-UX en IA64.
-AA	Permite características estándar ANSI de C++, como por ejemplo el espacio de nombres std y la biblioteca estándar de C++ en IA64.
+DA2.0W	Se debe utilizar para generar código de 64 bits para HP-UX en PA-RISC.
-ext	Permite el uso de diversas extensiones de C++, que incluyen el soporte para "long long".
-\$DB2PATH/include	Especifica la ubicación de los archivos de inclusión de DB2. Por ejemplo: \$HOME/sql1lib/include
-c	Hace que se efectúe solamente la compilación, no la edición de enlaces. La compilación y la edición de enlaces son pasos separados.

Opciones de compilación y enlace para bldapp

Opciones de enlace:

aCC Hace que el compilador aC++ de HP se utilice como interfaz del editor de enlaces.

\$EXTRA_CFLAG

Si la plataforma HP-UX es IA64 y está habilitado el soporte de 64 bits, este distintivo contiene el valor **+DD64 -AA**; si está habilitado el soporte de 32 bits, contiene el valor **+DD32 -AA**. Si la plataforma HP-UX es PA-RISC y está habilitado el soporte de 64 bits, contiene el valor **+DA2.0W**. Para el soporte de 32 bits en una plataforma PA-RISC, este distintivo no contiene ningún valor.

+DD64 Se debe utilizar para generar código de 64 bits para HP-UX en IA64.

+DD32 Se debe utilizar para generar código de 32 bits para HP-UX en IA64.

-AA Permite características estándar ANSI de C++, como por ejemplo el espacio de nombres std y la biblioteca estándar de C++ en IA64.

+DA2.0W

Se debe utilizar para generar código de 64 bits para HP-UX en PA-RISC.

-o \$1 Especifica el ejecutable.

\$1.o Especifica el archivo objeto del programa.

utilemb.o

Si es un programa de SQL incorporado, incluir el archivo objeto del programa de utilidad de SQL incorporado para la comprobación de errores.

utilapi.o

Si es un programa de SQL no incorporado, incluir el archivo objeto del programa de utilidad de la API de DB2 para la comprobación de errores.

\$EXTRA_LFLAG

Especificar la vía de acceso de tiempo de ejecución. Si se establece, para 32 bits contiene el valor **-Wl,+b\$HOME/sql1lib/lib32**, y para 64 bits contiene **-Wl,+b\$HOME/sql1lib/lib**. Si no se establece, no contiene ningún valor.

-L\$DB2PATH/\$LIB

Especifica la ubicación de las bibliotecas compartidas de tiempo de ejecución de DB2. Para 32 bits: **\$HOME/sql1lib/lib32**; para 64 bits: **\$HOME/sql1lib/lib**.

-ldb2 Enlazar con la biblioteca DB2.

Consulte la documentación del compilador para conocer otras opciones de compilador.

Tareas relacionadas:

- “Creación de aplicaciones C++ de UNIX” en la página 168

Ejemplos relacionados:

- “bldapp -- Builds HP-UX C++ applications (C++)”

Script de creación para rutinas C++

```
#!/bin/sh
# SCRIPT: bldrtn
# Crea rutinas C++ de HP-UX (procedimientos almacenados y UDF)
# Uso: bldrtn <nombre_prog> [ <nombre_bd> ]

# Definir DB2PATH de modo que apunte al lugar desde donde se accederá a DB2.
# El valor por omisión es la vía de acceso de instancia estándar.
DB2PATH=$HOME/sql1lib

# Determinar la plataforma HP y definir opciones correctas de compilación/enlace
hpplat=`uname -m`
bitwidth=`LANG=C db2level | awk '/bits/{print $5}'`
if [ $hpplat = "ia64" ]; then
    if [ $bitwidth = "\"64\"" ]; then
        EXTRA_CFLAG="+DD64 -AA"
        LIB="lib"
    else

```

```

        EXTRA_CFLAG="+DD32 -AA"
        LIB="lib32"
    fi
else
    if [ $bitwidth = "\64\" ]; then
        EXTRA_CFLAG="+DA2.0W"
        LIB="lib"
    else
        EXTRA_CFLAG=
        LIB="lib32"
    fi
fi

# Se recomienda la vía de tiempo de ejecución para todas las aplicaciones.
# Si tiene necesidad de usar SHLIB_PATH o LD_LIBRARY_PATH, desestablezca
# la variable RUNTIME comentando la línea siguiente.
RUNTIME=true

if [ "$RUNTIME" != "" ]
then
    EXTRA_LFLAG="-Wl,+b$DB2PATH/$LIB"
else
    EXTRA_LFLAG=""
fi

# Si es un programa de SQL incorporado, precompilar y vincular el programa.
if [ -f $1".sqC" ]
then
    ./embprep $1 $2
fi

# Compilar el programa. Primero comprobar que está codificado con extern "C".
aCC $EXTRA_CFLAG +u1 +z -ext -mt -I$DB2PATH/include -c $1.C

# Enlazar el programa para crear una biblioteca compartida.
aCC $EXTRA_CFLAG -mt -b -o $1 $1.o $EXTRA_LFLAG -L$DB2PATH/$LIB -ldb2

# Copiar la biblioteca compartida en el subdirectorío sqllib/function.
# Nota: el usuario debe tener permiso de escritura sobre este directorio.
rm -f $DB2PATH/function/$1
cp $1 $DB2PATH/function

```

Opciones de compilación y enlace para rutinas C++ de HP-UX

La tabla siguiente muestra las opciones de compilación y enlace que DB2 recomienda para crear rutinas en C++ (procedimientos almacenados y funciones definidas por el usuario) con el compilador C++ de HP-UX, tal como muestra el script de creación bldrtn.

Opciones de compilación y enlace para bldrtn

Opciones de compilación:

aCC Denota el compilador aC++ de HP.

\$EXTRA_CFLAG

Si la plataforma HP-UX es IA64 y está habilitado el soporte de 64 bits, este distintivo contiene el valor **+DD64 -AA**; si está habilitado el soporte de 32 bits, contiene el valor **+DD32 -AA**. Si la plataforma HP-UX es PA-RISC y está habilitado el soporte de 64 bits, contiene el valor **+DA2.0W**. Para el soporte de 32 bits en una plataforma PA-RISC, este distintivo no contiene ningún valor.

+DD64 Se debe utilizar para generar código de 64 bits para HP-UX en IA64.

+DD32 Se debe utilizar para generar código de 32 bits para HP-UX en IA64.

-AA Permite características estándar ANSI de C++, como por ejemplo el espacio de nombres std y la biblioteca estándar de C++ en IA64.

+DA2.0W

Se debe utilizar para generar código de 64 bits para HP-UX en PA-RISC.

+u1 Permite el acceso a datos no alineados.

+z Generar código independiente de la posición.

-ext Permite el uso de diversas extensiones de C++, tales como el soporte para "long long".

-mt Permite el soporte de hebras para el compilador aC++ de HP, que es necesario porque las rutinas se pueden ejecutaren el mismo proceso que otras rutinas (THREADSAFE) o en el propio motor (NOT FENCED).

-I\$DB2PATH/include

Especifica la ubicación de los archivos de inclusión de DB2. Por ejemplo:
\$DB2PATH/include

-c Hace que se efectúe solamente la compilación, no la edición de enlaces. La compilación y la edición de enlaces son pasos separados.

Opciones de enlace:

aCC Hace que el compilador aC++ de HP se utilice como interfaz del editor de enlaces.

\$EXTRA_CFLAG

Si la plataforma HP-UX es IA64 y está habilitado el soporte de 64 bits, este distintivo contiene el valor **+DD64 -AA**; si está habilitado el soporte de 32 bits, contiene el valor **+DD32 -AA**. Si la plataforma HP-UX es PA-RISC y está habilitado el soporte de 64 bits, contiene el valor **+DA2.0W**. Para el soporte de 32 bits en una plataforma PA-RISC, este distintivo no contiene ningún valor.

+DD64 Se debe utilizar para generar código de 64 bits para HP-UX en IA64.

+DD32 Se debe utilizar para generar código de 32 bits para HP-UX en IA64.

-AA Permite características estándar ANSI de C++, como por ejemplo el espacio de nombres std y la biblioteca estándar de C++ en IA64.

+DA2.0W

Se debe utilizar para generar código de 64 bits para HP-UX en PA-RISC.

-mt Permite el soporte de hebras para el compilador aC++ de HP, que es necesario porque las rutinas se pueden ejecutaren el mismo proceso que otras rutinas (THREADSAFE) o en el propio motor (NOT FENCED).

-b Crear una biblioteca compartida en lugar de un ejecutable normal.

-o \$1 Especifica el ejecutable.

\$1.o Especifica el archivo objeto del programa.

\$EXTRA_LFLAG

Especificar la vía de acceso de tiempo de ejecución. Si se establece, para 32 bits contiene el valor **-Wl,+b\$HOME/sql1lib/lib32**, y para 64 bits contiene **-Wl,+b\$HOME/sql1lib/lib**. Si no se establece, no contiene ningún valor.

-L\$DB2PATH/\$LIB

Especifica la ubicación de las bibliotecas compartidas de tiempo de ejecución de DB2. Para 32 bits: **\$HOME/sql1lib/lib32**; para 64 bits: **\$HOME/sql1lib/lib**.

-ldb2 Enlazar con la biblioteca DB2.

Consulte la documentación del compilador para conocer otras opciones de compilador.

Tareas relacionadas:

- “Creación de rutinas C++ de UNIX” en la página 172

Ejemplos relacionados:

- “bldrtn -- Builds HP-UX C++ routines (stored procedures and UDFs) (C++)”

Creación de aplicaciones multihebra C++ en HP-UX

HP-UX proporciona una biblioteca de hebras de POSIX y una biblioteca de hebras de DCE. En HP-UX, DB2 sólo da soporte a las aplicaciones multihebra que hacen uso de la biblioteca de hebras de POSIX.

Cuando se utiliza el compilador C++ de HP-UX, se debe especificar `-mt` para las aplicaciones multihebra en los pasos de compilación y enlace.

El script `bldmt` contiene los mandatos para crear aplicaciones multihebra. Además de las opciones especificadas anteriormente, las opciones de compilación y enlace son las mismas que las utilizadas para el archivo de script de SQL incorporado, `bldapp`.

Procedimiento:

Para crear el programa de ejemplo `dbthrds`, a partir del archivo fuente `dbthrds.sqc`, especifique:

```
bldmt dbthrds
```

El resultado es un archivo ejecutable, `dbthrds`. Para ejecutar el archivo ejecutable sobre la base de datos `sample`, escriba el nombre del ejecutable:

```
dbthrds
```

Conceptos relacionados:

- “Archivos de creación” en la página 110

Información relacionada:

- “Ejemplos de C” en la página 71
- “Opciones de compilación y enlace para aplicaciones C++ de HP-UX” en la página 218

Ejemplos relacionados:

- “bldmt -- Builds HP-UX C++ multi-threaded applications (C++)”
- “dbthrds.sqc -- How to use multiple context APIs on UNIX (C++)”
- “embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs (C)”

COBOL para Micro Focus

Para obtener información sobre cómo crear aplicaciones COBOL para Micro Focus en sistemas operativos UNIX soportados, consulte el apartado “Creación de aplicaciones Micro Focus COBOL de UNIX” en la página 176. Para obtener información sobre cómo crear rutinas COBOL para Micro Focus en sistemas operativos UNIX soportados, consulte el apartado “Creación de rutinas Micro Focus COBOL de UNIX” en la página 177.

Configuración del compilador Micro Focus COBOL en HP-UX

Si desarrolla aplicaciones que contienen SQL incorporado y llamadas a la API de DB2, y está utilizando el compilador Micro Focus COBOL, debe tener en cuenta varias consideraciones.

Procedimiento:

- Cuando precompile la aplicación utilizando el mandato db2 prep del procesador de línea de mandatos, utilice la opción target mfcob.
- Debe incluir el directorio del archivo COPY de DB2 para COBOL en la variable de entorno COBCOPY de Micro Focus COBOL. La variable de entorno COBCPY especifica la ubicación de los archivos COPY. Los archivos COPY de DB2 correspondientes a Micro Focus COBOL residen en sqllib/include/cobol_mf, dentro del directorio de la instancia de la base de datos.

Para incluir el directorio en la variable de entorno,

- en el shell bash o Korn, especifique:

```
export COBCPY=${COBCPY}:${HOME}/sqllib/include/cobol_mf
```

- en el shell C, especifique:

```
setenv COBCPY ${COBCPY}:${HOME}/sqllib/include/cobol_mf
```

Nota: Puede definir COBCPY dentro del archivo .profile o .login.

Tareas relacionadas:

- “Creación de aplicaciones Micro Focus COBOL de UNIX” en la página 176
- “Creación de rutinas Micro Focus COBOL de UNIX” en la página 177

Información relacionada:

- “Opciones de compilación y enlace para aplicaciones Micro Focus COBOL de HP-UX” en la página 224
- “Opciones de compilación y enlace para rutinas Micro Focus COBOL de HP-UX” en la página 225

Script de creación para aplicaciones COBOL para Micro Focus

```
#!/bin/sh
# SCRIPT: bldapp
# Crea aplicaciones COBOL para Micro Focus de HP-UX
# Uso: bldapp <nombre_prog> [ <nombre_bd> [ <IDusuario> <contraseña> ]]

# Definir DB2PATH de modo que apunte al lugar desde donde se accederá a DB2.
# El valor por omisión es la vía de acceso de instancia estándar.
DB2PATH=${HOME}/sqllib

# Definir COBCPY de forma que incluya el directorio de los archivos COPY de DB2.
COBCPY=${COBCPY}:${DB2PATH}/include/cobol_mf

# Si es un programa de SQL incorporado, precompilar y vincular el programa.
if [ -f $1".sqb" ]
then
./embprep $1 $2 $3 $4
fi

# Compilar el programa de utilidad de comprobación de errores checkerr.cbl.
cob -cx checkerr.cbl

# Compilar el programa.
cob -cx $1.cbl
```

```
# Enlazar el programa.
cob -x $1.o checkerr.o -L$DB2PATH/lib -ldb2 -ldb2gmf
```

Opciones de compilación y enlace para aplicaciones Micro Focus COBOL de HP-UX

La tabla siguiente muestra las opciones de compilación y enlace que DB2 recomienda para crear aplicaciones COBOL de SQL incorporado y de la API de DB2 con el compilador MicroFocus COBOL de HP-UX, tal como muestra el script de creación bldapp.

Opciones de compilación y enlace para bldapp	
Opciones de compilación:	
cob	Denota el compilador Micro Focus COBOL.
-cx	Compila para crear el módulo objeto.
Opciones de enlace:	
cob	Hace que el compilador se utilice como interfaz del editor de enlaces.
-x	Especifica un programa ejecutable.
\$1.o	Incluir el archivo objeto del programa.
checkerr.o	Incluir el archivo objeto del programa de utilidad para la comprobación de errores.
-L\$DB2PATH/lib	Especifica la ubicación de las bibliotecas compartidas de tiempo de ejecución de DB2.
-ldb2	Enlazar con la biblioteca DB2.
-ldb2gmf	Establece un enlace con la biblioteca del gestor de excepciones de DB2 correspondiente a Micro Focus COBOL.
Consulte la documentación del compilador para conocer otras opciones de compilador.	

Tareas relacionadas:

- “Creación de aplicaciones Micro Focus COBOL de UNIX” en la página 176

Ejemplos relacionados:

- “bldapp -- Builds HP-UX Micro Focus COBOL applications”

Script de creación para rutinas COBOL para Micro Focus

```
#!/bin/sh
# SCRIPT: bldrtn
# Crea rutinas COBOL para Micro Focus de HP-UX (procedimientos almacenados)
# Uso: bldrtn <nombre_prog> [ <nombre_bd> ]

# Definir DB2PATH de modo que apunte al lugar desde donde se accederá a DB2.
# El valor por omisión es la vía de acceso de instancia estándar.
DB2PATH=$HOME/sqllib

# Definir COBCPY de forma que incluya el directorio de los archivos COPY de DB2.
COBCPY=$COBCPY:$DB2PATH/include/cobol_mf

# Si es un programa de SQL incorporado, precompilar y vincular el programa.
if [ -f $1".sqb" ]
then
embprep $1 $2
fi
```



```

# Compilar el programa.
cob +z -cx $1.cbl

# Enlazar el programa.
ld -b -o $1 $1.o -L$DB2PATH/lib -ldb2 -ldb2gmf \
-L$COBDIR/coblib -lcobol -lcrtn

# Copiar la biblioteca compartida en el subdirectorio sqllib/function.
# Nota: el usuario debe tener permiso de escritura sobre este directorio.
rm -f $DB2PATH/function/$1
cp $1 $DB2PATH/function

```

Opciones de compilación y enlace para rutinas Micro Focus COBOL de HP-UX

La tabla siguiente muestra las opciones de compilación y enlace que DB2 recomienda para crear rutinas COBOL (procedimientos almacenados) con el compilador Micro Focus COBOL de HP-UX, tal como muestra el script de creación bldrtn.

Opciones de compilación y enlace para bldrtn	
Opciones de compilación:	
cob	Denota el compilador COBOL.
+z	Generar código independiente de la posición.
-cx	Compila y crea el módulo objeto.
Opciones de enlace:	
ld	Denota la utilización del enlazador para la edición de enlaces.
-b	Crear una biblioteca compartida en lugar de un archivo ejecutable normal.
-o \$1	Especifica el ejecutable.
\$1.o	Incluir el archivo objeto del programa.
-L\$DB2PATH/lib	Especifica la ubicación de las bibliotecas compartidas de tiempo de ejecución de DB2.
-ldb2	Enlazar con la biblioteca compartida de DB2.
-ldb2gmf	Establece un enlace con la biblioteca del gestor de excepciones de DB2 correspondiente a Micro Focus COBOL.
-L\$COBDIR/coblib	Especifica la ubicación de las bibliotecas de tiempo de ejecución de COBOL.
-lcobol	Enlazar con la biblioteca de COBOL.
-lcrtn	Enlazar con la biblioteca crtn.
Consulte la documentación del compilador para conocer otras opciones de compilador.	

Tareas relacionadas:

- “Creación de rutinas Micro Focus COBOL de UNIX” en la página 177

Ejemplos relacionados:

- “bldrtn -- Builds HP-UX Micro Focus COBOL routines (stored procedures)”

Capítulo 12. Linux

Linux C	227	Creación de aplicaciones multihebra C++ en Linux	236
Script de creación para aplicaciones C	227	COBOL para Micro Focus	237
Opciones de compilación y enlace para aplicaciones C de Linux	228	Configuración del compilador Micro Focus COBOL en Linux	237
Script de creación para rutinas C	229	Script de creación para aplicaciones COBOL para Micro Focus	238
Opciones de compilación y enlace para rutinas C de Linux	230	Opciones de compilación y enlace para aplicaciones Micro Focus COBOL de Linux	239
Creación de aplicaciones multihebra C en Linux C++ para Linux	231	Script de creación para rutinas COBOL para Micro Focus	240
Script de creación para aplicaciones C++	232	Opciones de compilación y enlace para rutinas Micro Focus COBOL de Linux	240
Opciones de compilación y enlace para aplicaciones C++ de Linux	233		
Script de creación para rutinas C++	234		
Opciones de compilación y enlace para rutinas C++ de Linux	235		

Este capítulo proporciona información detallada para crear aplicaciones en Linux. Para obtener las actualizaciones más recientes sobre el desarrollo de aplicaciones de DB2 para Linux, visite la página Web en:

<http://www.ibm.com/software/data/db2/udb/ad>

Linux C

La creación de información para las aplicaciones y rutinas de la CLI de DB2 CLI se encuentra en el manual *CLI Guide and Reference*.

Para obtener información sobre cómo crear aplicaciones C en sistemas operativos UNIX soportados, consulte el apartado "Creación de aplicaciones C de UNIX" en la página 161. Para obtener información sobre cómo crear rutinas C en sistemas operativos UNIX soportados, consulte el apartado "Creación de rutinas C de UNIX" en la página 165.

Script de creación para aplicaciones C

```
#!/bin/sh
# SCRIPT: bldapp
# Crea aplicaciones C de Linux
# Uso: bldapp <nombre_prog> [ <nombre_bd> [ <IDusuario> <contraseña> ] ]

# Definir DB2PATH de modo que apunte al lugar desde donde se accederá a DB2.
# El valor por omisión es la vía de acceso de instancia estándar.
DB2PATH=$HOME/sql1ib

# Determinar si se estaba ejecutando con 32 bits y si
# se estaba ejecutando con 32 bits en Linux AMD64
LIB="lib"
EXTRA_C_FLAGS=""
HARDWAREPLAT=`uname -m`
bitwidth=`LANG=C db2level | awk '/bits/{print $5}'`
if [ $bitwidth = "\32\" ]; then
    LIB="lib32"
    if [ "$HARDWAREPLAT" = "x86_64" ]; then
        EXTRA_C_FLAGS="-m32"
    fi
fi

# Se recomienda la vía de tiempo de ejecución para todas las aplicaciones.
```

```

# Si tiene necesidad de usar LD_LIBRARY_PATH, desestablezca la variable
# RUNTIME comentando la línea siguiente.
RUNTIME=true

if [ "$RUNTIME" != "" ]
then
    EXTRA_LFLAG="-Wl,-rpath,$DB2PATH/$LIB"
else
    EXTRA_LFLAG=""
fi

# Si es un programa de SQL incorporado, precompilar y vincular el programa.
# Nota: algunos archivos .sqc no contienen SQL sino que enlazan en
# utilemb.sqc, por lo que debe ignorar este aviso si lo recibe:
# SQL0053W No se ha encontrado ninguna sentencia SQL en el programa.
if [ -f $1".sqc" ]
then
    ./embprep $1 $2 $3 $4
# Compilar el programa de utilidad de comprobación de errores utilemb.c.
    gcc $EXTRA_C_FLAGS -I$DB2PATH/include -c utilemb.c
else
# Compilar el programa de utilidad de comprobación de errores utilapi.c.
    gcc $EXTRA_C_FLAGS -I$DB2PATH/include -c utilapi.c
fi

# Compilar el programa.
gcc $EXTRA_C_FLAGS -I$DB2PATH/include -c $1.c

if [ -f $1".sqc" ]
then
# Enlazar el programa con utilemb.o.
    gcc $EXTRA_C_FLAGS -o $1 $1.o utilemb.o $EXTRA_LFLAG \
        -L$DB2PATH/$LIB -ldb2
else
# Enlazar el programa con utilapi.o.
    gcc $EXTRA_C_FLAGS -o $1 $1.o utilapi.o $EXTRA_LFLAG \
        -L$DB2PATH/$LIB -ldb2
fi

```

Opciones de compilación y enlace para aplicaciones C de Linux

La tabla siguiente muestra las opciones de compilación y enlace que DB2 recomienda para crear aplicaciones C de SQL incorporado y de la API de DB2 con el compilador C de Linux, tal como muestra el script de creación bldapp.

Opciones de compilación y enlace para bldapp	
Opciones de compilación:	
gcc	Denota el compilador C de GNU/Linux.
\$EXTRA_C_FLAGS	Contiene "-m32" para el soporte de 32 bits en AMD64; de lo contrario, no contiene ningún valor.
-I\$DB2PATH/include	Especifica la ubicación de los archivos de inclusión de DB2.
-c	Hace que se efectúe solamente la compilación, no la edición de enlaces. Este archivo de script tiene pasos separados para la compilación y la edición de enlaces.

Opciones de compilación y enlace para bldapp

Opciones de enlace:

gcc Hace que el compilador se utilice como interfaz del editor de enlaces.

\$EXTRA_C_FLAGS

Contiene "-m32" para el soporte de 32 bits en AMD64; de lo contrario, no contiene ningún valor.

-o \$1 Especifica el ejecutable.

\$1.o Especifica el archivo objeto.

utilemb.o

Si es un programa de SQL incorporado, incluir el archivo objeto del programa de utilidad de SQL incorporado para la comprobación de errores.

utilapi.o

Si es un programa de SQL no incorporado, incluir el archivo objeto del programa de utilidad de la API de DB2 para la comprobación de errores.

\$EXTRA_LFLAG

Si 'RUNTIME=true' no está comentado, para 32 bits contiene el valor "-Wl,-rpath,\$DB2PATH/lib32", y para 64 bits contiene el valor "-Wl,-rpath,\$DB2PATH/lib". De lo contrario, no contiene ningún valor.

-L\$DB2PATH/\$LIB

Especifica la ubicación de las bibliotecas estáticas y compartidas de DB2 durante el enlace. Por ejemplo, para 32 bits: \$HOME/sql1lib/lib32, y para 64 bits: \$HOME/sql1lib/lib. Si el usuario no especifica la opción -L, se utiliza la vía de acceso /usr/lib:/lib.

-ldb2 Enlazar con la biblioteca DB2.

Consulte la documentación del compilador para conocer otras opciones de compilador.

Tareas relacionadas:

- "Creación de aplicaciones C de UNIX" en la página 161

Ejemplos relacionados:

- "bldapp -- Builds Linux C applications (C)"

Script de creación para rutinas C

```
#!/bin/sh
# SCRIPT: bldrtn
# Crea rutinas C para Linux (procedimientos almacenados o UDF)
# Uso: bldrtn <nombre_prog> [ <nombre_bd> ]

# Definir DB2PATH de modo que apunte al lugar desde donde se accederá a DB2.
# El valor por omisión es la vía de acceso de instancia estándar.
DB2PATH=$HOME/sql1lib

# Determinar si se estaba ejecutando con 32 bits y si
# se estaba ejecutando con 32 bits en Linux AMD64
LIB="lib"
EXTRA_C_FLAGS=""
HARDWAREPLAT=`uname -m`
bitwidth=`LANG=C db2level | awk '/bits/{print $5}'`
if [ $bitwidth = "\"32\"" ]; then
    LIB="lib32"
    if [ "$HARDWAREPLAT" = "x86_64" ]; then
        EXTRA_C_FLAGS="-m32"
    fi
fi
```

```

fi

# Definir la vía de acceso de tiempo de ejecución.
EXTRA_LFLAG="-Wl,-rpath,$DB2PATH/$LIB"

# Si es un programa de SQL incorporado, precompilar y vincular el programa.
if [ -f $1".sqc" ]
then
./embprep $1 $2
fi

# Compilar el programa.
gcc $EXTRA_C_FLAGS -fpic -I$DB2PATH/include -c $1.c -D_REENTRANT

# Enlazar el programa y cree una biblioteca compartida.
gcc $EXTRA_C_FLAGS -shared -o $1 $1.o $EXTRA_LFLAG -L$DB2PATH/$LIB -ldb2 -lpthread

# Copiar la biblioteca compartida en el subdirectorio function.
# El usuario debe tener permiso de escritura sobre este directorio.
rm -f $DB2PATH/function/$1
cp $1 $DB2PATH/function

```

Opciones de compilación y enlace para rutinas C de Linux

La tabla siguiente muestra las opciones de compilación y enlace que DB2 recomienda para crear rutinas en C (procedimientos almacenados y funciones definidas por el usuario) con el compilador C de Linux, tal como muestra el script de creación bldrtn.

Opciones de compilación y enlace para bldrtn	
Opciones de compilación:	
gcc	Denota el compilador C de GNU/Linux.
\$EXTRA_C_FLAGS	Contiene "-m32" para el soporte de 32 bits en AMD64; de lo contrario, no contiene ningún valor.
-fpic	Genera código independiente de la posición.
-I\$DB2PATH/include	Especifica la ubicación de los archivos de inclusión de DB2.
-c	Hace que se efectúe solamente la compilación, no la edición de enlaces. Este archivo de script tiene pasos separados para la compilación y la edición de enlaces.
-D_REENTRANT	Define la opción <code>_REENTRANT</code> , que es necesaria pues las rutinas se pueden ejecutar en el mismo proceso que otras rutinas (THREADSAFE) o en el propio motor (NOT FENCED).

Opciones de compilación y enlace para bldrtn

Opciones de enlace:

gcc Hace que el compilador se utilice como interfaz del editor de enlaces.

\$EXTRA_CFLAG

Contiene "-m32" para el soporte de 32 bits en AMD64; de lo contrario, no contiene ningún valor.

-shared

Genera una biblioteca compartida.

-o \$1 Especifica el ejecutable.

\$1.o Incluir el archivo objeto del programa.

\$EXTRA_LFLAG

Especifica la ubicación de las bibliotecas compartidas de DB2 durante la ejecución. Para 32 bits contiene el valor "-Wl,-rpath,\$DB2PATH/lib32". Para 64 bits contiene el valor "-Wl,-rpath,\$DB2PATH/lib".

-L\$DB2PATH/\$LIB

Especifica la ubicación de las bibliotecas estáticas y compartidas de DB2 durante el enlace. Por ejemplo, para 32 bits: \$HOME/sql1lib/lib32, y para 64 bits: \$HOME/sql1lib/lib. Si el usuario no especifica la opción -L, se utiliza la vía de acceso: /usr/lib:/lib.

-ldb2 Enlazar con la biblioteca DB2.

-lpthread

Establece un enlace con la biblioteca de hebras de POSIX.

Consulte la documentación del compilador para conocer otras opciones de compilador.

Tareas relacionadas:

- "Creación de rutinas C de UNIX" en la página 165

Ejemplos relacionados:

- "bldrtn -- Builds Linux C routines (stored procedures or UDFs) (C)"

Creación de aplicaciones multihebra C en Linux

Las aplicaciones multihebra que hacen uso de Linux C se deben compilar con `-D_REENTRANT` y enlazar con `-lpthread`.

El script `blfmt` contiene los mandatos para crear aplicaciones multihebra. Además de las opciones especificadas anteriormente, las opciones de compilación y enlace son las mismas que las utilizadas para el archivo de script de SQL incorporado, `bldapp`.

Procedimiento:

Para crear el programa de ejemplo `dbthrs`, a partir del archivo fuente `dbthrs.sqc`, especifique:

```
blfmt dbthrs
```

El resultado es un archivo ejecutable, `dbthrs`. Para ejecutar el archivo ejecutable sobre la base de datos `sample`, especifique:

```
dbthrs
```

Conceptos relacionados:

- “Archivos de creación” en la página 110

Información relacionada:

- “Ejemplos de C” en la página 71
- “Opciones de compilación y enlace para aplicaciones C de Linux” en la página 228

Ejemplos relacionados:

- “bldmt -- Builds Linux C multi-threaded applications (C)”
- “dbthrds.sqc -- How to use multiple context APIs on UNIX (C)”
- “embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs (C)”

C++ para Linux

Para obtener información sobre cómo crear aplicaciones C++ en sistemas operativos UNIX soportados, consulte el apartado “Creación de aplicaciones C++ de UNIX” en la página 168. Para obtener información sobre cómo crear rutinas C++ en sistemas operativos UNIX soportados, consulte el apartado “Creación de rutinas C++ de UNIX” en la página 172.

Script de creación para aplicaciones C++

```

#!/bin/sh
# SCRIPT: bldapp
# Crea aplicaciones C++ para Linux
# Uso: bldapp <nombre_prog> [ <nombre_bd> [ <IDusuario> <contraseña> ]]

# Definir DB2PATH de modo que apunte al lugar desde donde se accederá a DB2.
# El valor por omisión es la vía de acceso de instancia estándar.
DB2PATH=$HOME/sql1lib

# Determinar si se estaba ejecutando con 32 bits y si
# se estaba ejecutando con 32 bits en Linux AMD64
LIB="lib"
EXTRA_C_FLAGS=""
HARDWAREPLAT=`uname -m`
bitwidth=`LANG=C db2level | awk '/bits/{print $5}'`
if [ $bitwidth = "\32\" ]; then
    LIB="lib32"
    if [ "$HARDWAREPLAT" = "x86_64" ]; then
        EXTRA_C_FLAGS="-m32"
    fi
fi

# Se recomienda la vía de tiempo de ejecución para todas las aplicaciones.
# Si tiene necesidad de usar LD_LIBRARY_PATH, desestablezca la variable
# RUNTIME comentando la línea siguiente.
RUNTIME=true

if [ "$RUNTIME" != "" ]
then
    EXTRA_LFLAG="-Wl,-rpath,$DB2PATH/$LIB"
else
    EXTRA_LFLAG=""
fi

# Si es un programa de SQL incorporado, precompilar y vincular el programa.
# Nota: algunos archivos .sqc no contienen SQL sino que enlazan en
# utilemb.sqc, por lo que debe ignorar este aviso si lo recibe:

```



```

# SQL0053W No se ha encontrado ninguna sentencia SQL en el programa.
if [ -f $1".sqC" ]
then
  ./embprep $1 $2 $3 $4
# Compilar el programa de utilidad de comprobación de errores utilemb.C.
  g++ $EXTRA_C_FLAGS -I$DB2PATH/include -c utilemb.C
else
# Compilar el programa de utilidad de comprobación de errores utilapi.C.
  g++ $EXTRA_C_FLAGS -I$DB2PATH/include -c utilapi.C
fi

# Compilar el programa.
g++ $EXTRA_C_FLAGS -I$DB2PATH/include -c $1.C

if [ -f $1".sqC" ]
then
# Enlazar el programa con utilemb.o
  g++ $EXTRA_C_FLAGS -o $1 $1.o utilemb.o $EXTRA_LFLAG -L$DB2PATH/$LIB -ldb2
else
# Enlazar el programa con utilapi.o
  g++ $EXTRA_C_FLAGS -o $1 $1.o utilapi.o $EXTRA_LFLAG -L$DB2PATH/$LIB -ldb2
fi

```

Opciones de compilación y enlace para aplicaciones C++ de Linux

La tabla siguiente muestra las opciones de compilación y enlace que DB2 recomienda para crear aplicaciones en C++ de SQL incorporado y de la API de DB2 con el compilador C++ de Linux, tal como muestra el script de creación bldapp.

Opciones de compilación y enlace para bldapp	
Opciones de compilación:	
g++	Denota el compilador C++ de GNU/Linux.
\$EXTRA_C_FLAGS	Contiene "-m32" para el soporte de 32 bits en AMD64; de lo contrario, no contiene ningún valor.
-I\$DB2PATH/include	Especifica la ubicación de los archivos de inclusión de DB2.
-c	Hace que se efectúe solamente la compilación, no la edición de enlaces. Este archivo de script tiene pasos separados para la compilación y la edición de enlaces.

Opciones de compilación y enlace para bldapp

Opciones de enlace:

g++ Hace que el compilador se utilice como interfaz del editor de enlaces.

\$EXTRA_C_FLAGS

Contiene "-m32" para el soporte de 32 bits en AMD64; de lo contrario, no contiene ningún valor.

-o \$1 Especifica el ejecutable.

\$1.o Incluir el archivo objeto del programa.

utilemb.o

Si es un programa de SQL incorporado, incluir el archivo objeto del programa de utilidad de SQL incorporado para la comprobación de errores.

utilapi.o

Si es un programa de SQL no incorporado, incluir el archivo objeto del programa de utilidad de la API de DB2 para la comprobación de errores.

\$EXTRA_LFLAG

Si 'RUNTIME=true' no está comentado, para 32 bits contiene el valor "-Wl,-rpath,\$DB2PATH/lib32", y para 64 bits contiene el valor "-Wl,-rpath,\$DB2PATH/lib". De lo contrario, no contiene ningún valor.

-L\$DB2PATH/\$LIB

Especifica la ubicación de las bibliotecas estáticas y compartidas de DB2 durante el enlace. Por ejemplo, para 32 bits: \$HOME/sql1lib/lib32, y para 64 bits: \$HOME/sql1lib/lib. Si el usuario no especifica la opción -L, se utiliza la vía de acceso /usr/lib:/lib.

-ldb2 Enlazar con la biblioteca DB2.

Consulte la documentación del compilador para conocer otras opciones de compilador.

Tareas relacionadas:

- "Creación de aplicaciones C++ de UNIX" en la página 168

Ejemplos relacionados:

- "bldapp -- Builds Linux C++ applications (C++)"

Script de creación para rutinas C++

```
#!/bin/sh
# SCRIPT: bldrtn
# Crea rutinas C++ para Linux (procedimientos almacenados y UDF)
# Uso: bldrtn <nombre_prog> [ <nombre_bd> ]

# Definir DB2PATH de modo que apunte al lugar desde donde se accederá a DB2.
# El valor por omisión es la vía de acceso de instancia estándar.
DB2PATH=$HOME/sql1lib

# Determinar si se estaba ejecutando con 32 bits y si
# se estaba ejecutando con 32 bits en Linux AMD64
LIB="lib"
EXTRA_C_FLAGS=""
HARDWAREPLAT=`uname -m`
bitwidth=`LANG=C db2level | awk '/bits/{print $5}'`
if [ $bitwidth = "\"32\"" ]; then
    LIB="lib32"
    if [ "$HARDWAREPLAT" = "x86_64" ]; then
        EXTRA_C_FLAGS="-m32"
    fi
fi
```

```

fi

# Definir la vía de acceso de tiempo de ejecución.
EXTRA_LFLAG="-Wl,-rpath,$DB2PATH/$LIB"

# Si es un programa de SQL incorporado, precompilar y vincular el programa.
if [ -f $1".sqC" ]
then
./embprep $1 $2
fi

# Compilar el programa.
g++ $EXTRA_C_FLAGS -fpic -I$DB2PATH/include -c $1.C -D_REENTRANT

# Enlazar el programa para crear una biblioteca compartida.
g++ $EXTRA_C_FLAGS -shared -o $1 $1.o $EXTRA_LFLAG -L$DB2PATH/$LIB -ldb2 -lpthread

# Copiar la biblioteca compartida en el subdirectorio function.
# El usuario debe tener permiso de escritura sobre este directorio.
rm -f $DB2PATH/function/$1
cp $1 $DB2PATH/function

```

Opciones de compilación y enlace para rutinas C++ de Linux

La tabla siguiente muestra las opciones de compilación y enlace que DB2 recomienda para crear rutinas en C++ (procedimientos almacenados y funciones definidas por el usuario) con el compilador C++ de Linux, tal como muestra el script de creación bldrtn.

Opciones de compilación y enlace para bldrtn	
Opciones de compilación:	
g++	Denota el compilador C++ de GNU/Linux.
\$EXTRA_C_FLAGS	Contiene "-m32" para el soporte de 32 bits en AMD64; de lo contrario, no contiene ningún valor.
-fpic	Genera código independiente de la posición.
-I\$DB2PATH/include	Especifica la ubicación de los archivos de inclusión de DB2.
-c	Hace que se efectúe solamente la compilación, no la edición de enlaces. Este archivo de script tiene pasos separados para la compilación y la edición de enlaces.
-D_REENTRANT	Define la opción <code>_REENTRANT</code> , que es necesaria pues las rutinas se pueden ejecutar en el mismo proceso que otras rutinas (THREADSAFE) o en el propio motor (NOT FENCED).

Opciones de compilación y enlace para bldrtn

Opciones de enlace:

g++ Hace que el compilador se utilice como interfaz del editor de enlaces.

\$EXTRA_C_FLAGS

Contiene "-m32" para el soporte de 32 bits en AMD64; de lo contrario, no contiene ningún valor.

-shared

Genera una biblioteca compartida.

-o \$1 Especifica el ejecutable.

\$1.o Incluir el archivo objeto del programa.

\$EXTRA_LFLAG

Especifica la ubicación de las bibliotecas compartidas de DB2 durante la ejecución. Para 32 bits contiene el valor "-Wl,-rpath,\$DB2PATH/lib32". Para 64 bits contiene el valor "-Wl,-rpath,\$DB2PATH/lib".

-L\$DB2PATH/\$LIB

Especifica la ubicación de las bibliotecas estáticas y compartidas de DB2 durante el enlace. Por ejemplo, para 32 bits: \$HOME/sql1lib/lib32, y para 64 bits: \$HOME/sql1lib/lib. Si el usuario no especifica la opción -L, se utiliza la vía de acceso: /usr/lib:/lib.

-ldb2 Enlazar con la biblioteca DB2.

-lpthread

Establece un enlace con la biblioteca de hebras de POSIX.

Consulte la documentación del compilador para conocer otras opciones de compilador.

Tareas relacionadas:

- "Creación de rutinas C++ de UNIX" en la página 172

Ejemplos relacionados:

- "bldrtn -- Builds Linux C++ routines (stored procedures and UDFs) (C++)"

Creación de aplicaciones multihebra C++ en Linux

Las aplicaciones multihebra que hacen uso de Linux C++ se deben compilar con `-D_REENTRANT` y enlazar con `-lpthread`.

El archivo de script `blfmt` contiene los mandatos para crear un programa multihebra de SQL incorporado. Además de las opciones especificadas anteriormente, las opciones de compilación y enlace son las mismas que las utilizadas para el archivo de script de SQL incorporado, `blapp`.

Procedimiento:

Para crear el programa de ejemplo `dbthrds`, a partir del archivo fuente `dbthrds.sqlc`, especifique:

```
blfmt dbthrds
```

El resultado es un archivo ejecutable, `dbthrds`. Para ejecutar el archivo ejecutable sobre la base de datos `sample`, especifique:

```
dbthrds
```

Conceptos relacionados:

- “Archivos de creación” en la página 110

Información relacionada:

- “Ejemplos de C” en la página 71
- “Opciones de compilación y enlace para aplicaciones C++ de Linux” en la página 233

Ejemplos relacionados:

- “bldmt -- Builds Linux C++ multi-threaded applications (C++)”
- “dbthrds.sqC -- How to use multiple context APIs on UNIX (C++)”
- “embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs (C)”

COBOL para Micro Focus

En Linux sólo se soporta COBOL para Micro Focus en las arquitecturas siguientes:

- **Linux en Intel x86 (32 bits)**
- **Linux en s/390**

Para obtener información sobre cómo crear aplicaciones COBOL para Micro Focus en sistemas operativos UNIX soportados, consulte el apartado “Creación de aplicaciones Micro Focus COBOL de UNIX” en la página 176. Para obtener información sobre cómo crear rutinas COBOL para Micro Focus en sistemas operativos UNIX soportados, consulte el apartado “Creación de rutinas Micro Focus COBOL de UNIX” en la página 177.

Configuración del compilador Micro Focus COBOL en Linux

Procedimiento:

Para ejecutar rutinas de Micro Focus COBOL, el editor de enlaces de Linux debe poder acceder a determinadas bibliotecas compartidas de Java, y DB2 debe poder cargar estas bibliotecas. Debido a que el programa que realiza esta carga se ejecuta con privilegios setuid, sólo buscará las bibliotecas dependientes en /usr/lib.

Cree enlaces simbólicos con /usr/lib para las bibliotecas compartidas de COBOL. Esto se debe hacer como usuario root. La manera más sencilla de hacerlo es enlazando todos los archivos de bibliotecas de COBOL de \$COBDIR/lib con /usr/lib:

```
ln -s $COBDIR/lib/libcob* /usr/lib
```

donde \$COBDIR es el lugar en que está instalado Micro Focus COBOL, normalmente /opt/lib/mfcobol.

Éstos son los mandatos para enlazar cada uno de los archivos individuales (suponiendo que Micro Focus COBOL está instalado en /opt/lib/mfcobol):

```
ln -s /opt/lib/mfcobol/lib/libcobrts.so /usr/lib
ln -s /opt/lib/mfcobol/lib/libcobrts_t.so /usr/lib
ln -s /opt/lib/mfcobol/lib/libcobrts.so.2 /usr/lib
ln -s /opt/lib/mfcobol/lib/libcobrts_t.so.2 /usr/lib
ln -s /opt/lib/mfcobol/lib/libcobcrtn.so /usr/lib
ln -s /opt/lib/mfcobol/lib/libcobcrtn.so.2 /usr/lib
ln -s /opt/lib/mfcobol/lib/libcobmisc.so /usr/lib
ln -s /opt/lib/mfcobol/lib/libcobmisc_t.so /usr/lib
```

```

ln -s /opt/lib/mfcobol/lib/libcobmisc.so.2 /usr/lib
ln -s /opt/lib/mfcobol/lib/libcobmisc_t.so.2 /usr/lib
ln -s /opt/lib/mfcobol/lib/libcobscreen.so /usr/lib
ln -s /opt/lib/mfcobol/lib/libcobscreen.so.2 /usr/lib
ln -s /opt/lib/mfcobol/lib/libcobtrace.so /usr/lib
ln -s /opt/lib/mfcobol/lib/libcobtrace_t.so /usr/lib
ln -s /opt/lib/mfcobol/lib/libcobtrace.so.2 /usr/lib
ln -s /opt/lib/mfcobol/lib/libcobtrace_t.so.2 /usr/lib

```

Es necesario hacer lo siguiente en cada instancia de DB2.

- Cuando precompile la aplicación utilizando el mandato db2 prep del procesador de línea de mandatos, utilice la opción target mfcob.
- Debe incluir el directorio del archivo COPY de DB2 para COBOL en la variable de entorno COBCOPY de Micro Focus COBOL. La variable de entorno COBCPY especifica la ubicación de los archivos COPY. Los archivos COPY de DB2 correspondientes a Micro Focus COBOL residen en sqllib/include/cobol_mf, dentro del directorio de la instancia de la base de datos.

Para incluir el directorio en la variable de entorno, especifique:

– En el shell bash o Korn:

```
export COBCPY=$HOME/sqllib/include/cobol_mf:$COBDIR/cpylib
```

– En el shell C:

```
setenv COBCPY $HOME/sqllib/include/cobol_mf:$COBDIR/cpylib
```

- Actualizar la variable de entorno:

– En el shell bash o Korn:

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$HOME/sqllib/lib:$COBDIR/lib
```

– En el shell C:

```
setenv LD_LIBRARY_PATH $LD_LIBRARY_PATH:$HOME/sqllib/lib:$COBDIR/lib
```

- Establecer la Lista de entornos de DB2:

```
db2set DB2ENVLIST="COBDIR LD_LIBRARY_PATH"
```

Nota: Es posible que desee establecer COBCPY, COBDIR y LD_LIBRARY_PATH en .bashrc, .kshrc (según el shell que se utilice), .bash_profile, .profile (según el shell que se utilice) o en .login. .

Tareas relacionadas:

- “Creación de aplicaciones Micro Focus COBOL de UNIX” en la página 176
- “Creación de rutinas Micro Focus COBOL de UNIX” en la página 177

Información relacionada:

- “Opciones de compilación y enlace para aplicaciones Micro Focus COBOL de Linux” en la página 239
- “Opciones de compilación y enlace para rutinas Micro Focus COBOL de Linux” en la página 240

Script de creación para aplicaciones COBOL para Micro Focus

```

#!/bin/sh
# SCRIPT: bldapp
# Crea aplicaciones COBOL para Micro Focus de Linux
# Uso: bldapp <nombre_prog> [ <nombre_bd> [ <IDusuario> <contraseña> ]]

# Definir DB2PATH de modo que apunte al lugar desde donde se accederá a DB2.
# El valor por omisión es la vía de acceso de instancia estándar.
DB2PATH=$HOME/sqllib

```

```

# Defina COBCPY de forma que incluya el directorio de los archivos COPY de DB2.
COBCPY=$COBCPY:$DB2PATH/include/cobol_mf

# Si es un programa de SQL incorporado, precompilar y vincular el programa.
if [ -f $1".sqb" ]
then
  ./embprep $1 $2 $3 $4
fi

# Compilar el programa de utilidad de comprobación de errores checkerr.cbl.
cob -cx checkerr.cbl

# Compilar el programa.
cob -cx $1.cbl

# Enlazar el programa.
cob -x -o $1 $1.o checkerr.o -L$DB2PATH/lib -ldb2 -ldb2gmf

```

Opciones de compilación y enlace para aplicaciones Micro Focus COBOL de Linux

La tabla siguiente muestra las opciones de compilación y enlace que DB2 recomienda para crear aplicaciones COBOL de SQL incorporado y de la API de DB2 con el compilador Micro Focus COBOL en Linux, tal como muestra el script de creación bldapp.

Opciones de compilación y enlace para bldapp	
Opciones de compilación:	
cob	Denota el compilador Micro Focus COBOL.
-cx	Compila y crea el módulo objeto.
Opciones de enlace:	
cob	Hace que el compilador se utilice como interfaz del editor de enlaces.
-x	Especifica un programa ejecutable.
-o \$1	Incluir el ejecutable.
\$1.o	Incluir el archivo objeto del programa.
checkerr.o	Incluir el archivo objeto del programa de utilidad para la comprobación de errores.
-L\$DB2PATH/lib	Especifica la ubicación de las bibliotecas compartidas de tiempo de ejecución de DB2.
-ldb2	Enlazar con la biblioteca DB2.
-ldb2gmf	Establece un enlace con la biblioteca del gestor de excepciones de DB2 correspondiente a Micro Focus COBOL.
Consulte la documentación del compilador para conocer otras opciones de compilador.	

Tareas relacionadas:

- “Creación de aplicaciones Micro Focus COBOL de UNIX” en la página 176
- “Configuración del compilador Micro Focus COBOL en Linux” en la página 237

Ejemplos relacionados:

- “bldapp -- Builds Linux Micro Focus COBOL applications”
- “embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs (C)”

Script de creación para rutinas COBOL para Micro Focus

```
#!/bin/sh
# SCRIPT: bldrtn
# Crea rutinas (procedimientos almacenados) COBOL para Micro Focus de Linux
# Uso: bldrtn <nombre_prog> [ <nombre_bd> ]

# Definir DB2PATH de modo que apunte al lugar desde donde se accederá a DB2.
# El valor por omisión es la vía de acceso de instancia estándar.
DB2PATH=$HOME/sqllib

# Defina COBCPY de forma que incluya el directorio de los archivos COPY de DB2.
COBCPY=$COBCPY:$DB2PATH/include/cobol_mf

# Precompilar y vincular el programa.
./embprep $1 $2

# Compilar el programa.
cob -cx $1.cbl

# Enlazar el programa.
cob -x -o $1 $1.o -Q -G -L$DB2PATH/lib -ldb2 -ldb2gmf

# Copiar la biblioteca compartida en el subdirectorio sqllib/function.
# Nota: el usuario debe tener permiso de escritura sobre este directorio.
rm -f $DB2PATH/function/$1
cp $1 $DB2PATH/function
```

Opciones de compilación y enlace para rutinas Micro Focus COBOL de Linux

La tabla siguiente muestra las opciones de compilación y enlace que DB2 recomienda para crear rutinas COBOL (procedimientos almacenados) con el compilador Micro Focus COBOL en Linux, tal como muestra el script de creación bldrtn.

Opciones de compilación y enlace para bldrtn	
Opciones de compilación:	
cob	Denota el compilador COBOL.
-cx	Compila y crea el módulo objeto.
Opciones de enlace:	
cob	Hace que el compilador se utilice como interfaz del editor de enlaces.
-x	Especifica un programa ejecutable.
-o \$1	Especifica el ejecutable.
\$1.o	Incluir el archivo objeto del programa.
-Q -G	Genera una biblioteca compartida.
-L\$DB2PATH/lib	Especifica la ubicación de las bibliotecas compartidas de tiempo de ejecución de DB2.
-ldb2	Enlazar con la biblioteca DB2.
-ldb2gmf	Establece un enlace con la biblioteca del gestor de excepciones de DB2 correspondiente a Micro Focus COBOL.
Consulte la documentación del compilador para conocer otras opciones de compilador.	

Tareas relacionadas:

- “Creación de rutinas Micro Focus COBOL de UNIX” en la página 177
- “Configuración del compilador Micro Focus COBOL en Linux” en la página 237

|
|
|
|

Ejemplos relacionados:

- "bldrtn -- Builds Linux Micro Focus COBOL routines (stored procedures)"
- "embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs (C)"

Capítulo 13. Solaris

C para Solaris	243	Creación de aplicaciones multihebra C++ en Solaris.	252
Script de creación para aplicaciones C	243	COBOL para Micro Focus	253
Opciones de compilación y enlace para aplicaciones C de Solaris	244	Configuración del compilador Micro Focus COBOL en Solaris	253
Script de creación para rutinas C.	245	Script de creación para aplicaciones COBOL para Micro Focus	254
Opciones de compilación y enlace para rutinas C de Solaris	246	Opciones de compilación y enlace para aplicaciones Micro Focus COBOL de Solaris	254
Creación de aplicaciones multihebra C en Solaris.	247	Script de creación para rutinas COBOL para Micro Focus	255
C++ para Solaris	248	Opciones de compilación y enlace para rutinas Micro Focus COBOL de Solaris	256
Script de creación para aplicaciones C++	248		
Opciones de compilación y enlace para aplicaciones C++ de Solaris.	249		
Script de creación para rutinas C++	250		
Opciones de compilación y enlace para rutinas C++ de Solaris	251		

Este capítulo proporciona información detallada para crear aplicaciones en el entorno operativo Solaris. Para obtener las actualizaciones más recientes sobre el desarrollo de aplicaciones de DB2 para Solaris, visite la página Web en:

<http://www.ibm.com/software/data/db2/udb/ad>

C para Solaris

La creación de información para las aplicaciones y rutinas de la CLI de DB2 CLI se encuentra en el manual *CLI Guide and Reference*.

Para obtener información sobre cómo crear aplicaciones C en sistemas operativos UNIX soportados, consulte el apartado "Creación de aplicaciones C de UNIX" en la página 161. Para obtener información sobre cómo crear rutinas C en sistemas operativos UNIX soportados, consulte el apartado "Creación de rutinas C de UNIX" en la página 165.

Script de creación para aplicaciones C

```
#!/bin/sh
# SCRIPT: bldapp
# Crear aplicaciones C de Solaris
# Uso: bldapp <nombre_prog> [ <nombre_bd> [ <IDusuario> <contraseña> ] ]

# Defina DB2PATH de modo que apunte al lugar desde donde se accederá a DB2.
# El valor por omisión es la vía de acceso de instancia estándar.
DB2PATH=$HOME/sqllib

# Definir distintivos de compilación y enlace para programas de 32 y 64 bits
bitwidth=`LANG=C db2level | awk '/bits/{print $5}'`
if [ $bitwidth = "\"64\"" ];
then
    CFLAG_ARCH=v9
    LIB=lib
else
    CFLAG_ARCH=v8plusa
    LIB=lib32
fi

# Definir la vía de acceso de tiempo de ejecución.
# Se seguirá LD_LIBRARY_PATH en vez de la vía de acceso de tiempo de ejecución
```

```

# a menos que antes se desestablezca LD_LIBRARY_PATH para permitir que se
# utilice la vía de acceso de tiempo de ejecución.
EXTRA_LFLAG="-R$DB2PATH/$LIB"

# Si es un programa de SQL incorporado, precompilar y vincular el programa.
# Nota: algunos archivos .sqc no contienen SQL sino que enlazan en
# utilemb.sqc, por lo que debe ignorar este aviso si lo recibe:
# SQL0053W No se ha encontrado ninguna sentencia SQL en el programa.
if [ -f $1".sqc" ]
then
./embprep $1 $2 $3 $4
# Compilar el programa de utilidad de comprobación de errores utilemb.c.
cc -xarch=$CFLAG_ARCH -I$DB2PATH/include -c utilemb.c
else
# Compilar el programa de utilidad de comprobación de errores utilapi.c.
cc -xarch=$CFLAG_ARCH -I$DB2PATH/include -c utilapi.c
fi

# Compilar el programa.
cc -xarch=$CFLAG_ARCH -I$DB2PATH/include -c $1.c

if [ -f $1".sqc" ]
then
# Enlazar el programa con utilemb.o
cc -xarch=$CFLAG_ARCH -mt -o $1 $1.o utilemb.o \
-L$DB2PATH/$LIB $EXTRA_LFLAG -ldb2
else
# Enlazar el programa con utilapi.o
cc -xarch=$CFLAG_ARCH -mt -o $1 $1.o utilapi.o \
-L$DB2PATH/$LIB $EXTRA_LFLAG -ldb2
fi

```

Opciones de compilación y enlace para aplicaciones C de Solaris

La tabla siguiente muestra las opciones de compilación y enlace que DB2 recomienda para crear aplicaciones C de SQL incorporado y de la API de DB2 con el compilador C de Forte, tal como muestra el script de creación bldapp.

Opciones de compilación y enlace para bldapp	
Opciones de compilación:	
cc	Denota el compilador C.
-xarch=\$CFLAG_ARCH	Esta opción asegura que el compilador creará ejecutables válidos cuando establezca un enlace con libdb2.so. El valor de \$CFLAG_ARCH es "v8plusa" para la modalidad de 32 bits, o "v9" para la modalidad de 64 bits.
-I\$DB2PATH/include	Especifica la ubicación de los archivos de inclusión de DB2. Por ejemplo: \$HOME/sql1lib/include
-c	Hace que se efectúe solamente la compilación, no la edición de enlaces. Este script tiene pasos separados para la compilación y la edición de enlaces.

Opciones de compilación y enlace para bldapp

Opciones de enlace:

cc Hace que el compilador se utilice como interfaz del editor de enlaces.

-xarch=\$CFLAG_ARCH

Esta opción asegura que el compilador creará ejecutables válidos cuando establezca un enlace con libdb2.so. El valor de \$CFLAG_ARCH es "v8plusa" para la modalidad de 32 bits, o "v9" para la modalidad de 64 bits.

-mt Crea enlaces cuando se utiliza el soporte multihebra. Esta opción es necesaria para enlazar con libdb2.

Nota: Si se utilizan hebras de POSIX, las aplicaciones DB2 se deben también enlazar con -lpthread, tengan o no varias hebras.

-o \$1 Especifica el ejecutable.

\$1.o Incluir el archivo objeto del programa.

utilemb.o

Si es un programa de SQL incorporado, incluir el archivo objeto del programa de utilidad de SQL incorporado para la comprobación de errores.

utilapi.o

Si crea un programa que no es de SQL incorporado, incluir el archivo objeto del programa de utilidad de la API de DB2 para la comprobación de errores.

-L\$DB2PATH/\$LIB

Especifica la ubicación de las bibliotecas estáticas y compartidas de DB2 durante el enlace. Por ejemplo, para 32 bits: \$HOME/sql1lib/lib32, y para 64 bits: \$HOME/sql1lib/lib.

\$EXTRA_LFLAG

Especifica la ubicación de las bibliotecas compartidas de DB2 durante la ejecución. Para 32 bits contiene el valor "-R\$DB2PATH/lib32", y para 64 bits contiene el valor "-R\$DB2PATH/lib".

-ldb2 Enlazar con la biblioteca DB2.

Consulte la documentación del compilador para conocer otras opciones de compilador.

Tareas relacionadas:

- "Creación de aplicaciones C de UNIX" en la página 161

Ejemplos relacionados:

- "bldapp -- Builds Solaris C applications (C)"

Script de creación para rutinas C

```
#!/bin/sh
# SCRIPT: bldrtn
# Crea rutinas C para Solaris (procedimientos almacenados o UDF)
# Uso: bldrtn <nombre_prog> [ <nombre_bd> ]

# Definir DB2PATH de modo que apunte al lugar desde donde se accederá a DB2.
# El valor por omisión es la vía de acceso de instancia estándar.
DB2PATH=$HOME/sql1lib

# Definir distintivos de compilación y enlace para programas de 32 y 64 bits
bitwidth=`LANG=C db2level | awk '/bits/{print $5}'`
if [ $bitwidth = "\"64\"" ];
then
    CFLAG_ARCH=v9
    LIB=lib
```

```

else
    CFLAG_ARCH=v8plusa
    LIB=lib32
fi

# Definir la vía de acceso de tiempo de ejecución.
# Se seguirá LD_LIBRARY_PATH en vez de la vía de acceso de tiempo de ejecución
# a menos que antes se desestablezca LD_LIBRARY_PATH para permitir que se
# utilice la vía de acceso de tiempo de ejecución.
EXTRA_LFLAG="-R$DB2PATH/$LIB"

# Si es un programa de SQL incorporado, precompilar y vincular el programa.
if [ -f $1".sqc" ]
then
./embprep $1 $2
fi

# Compilar el programa.
cc -xarch=$CFLAG_ARCH -mt -DUSE_UI_THREADS -Kpic \
-I$DB2PATH/include -c $1.c

# Enlazar el programa y crear una biblioteca compartida.
cc -xarch=$CFLAG_ARCH -mt -G -o $1 $1.o -L$DB2PATH/$LIB \
$EXTRA_LFLAG -ldb2

# Copiar la biblioteca compartida en el subdirectorío sqllib/function.
# Nota: el usuario debe tener permiso de escritura sobre este directorio.
rm -f $DB2PATH/function/$1
cp $1 $DB2PATH/function

```

Opciones de compilación y enlace para rutinas C de Solaris

La tabla siguiente muestra las opciones de compilación y enlace que DB2 recomienda para crear rutinas en C (procedimientos almacenados y funciones definidas por el usuario) con el compilador C de Forte, tal como muestra el script de creación bldrtn.

Opciones de compilación y enlace para bldrtn	
Opciones de compilación:	
cc	Denota el compilador C.
-xarch=\$CFLAG_ARCH	Esta opción asegura que el compilador creará ejecutables válidos cuando establezca un enlace con libdb2.so. El valor de \$CFLAG_ARCH es "v8plusa" para la modalidad de 32 bits, o "v9" para la modalidad de 64 bits.
-mt	Permite el soporte multihebra, que es necesario pues las rutinas se pueden ejecutar en el mismo proceso que otras rutinas (THREADSAFE) o en el propio motor (NOT FENCED).
-DUSE_UI_THREADS	Permite el uso de las API de hebras "UNIX International" de Sun.
-Kpic	Genera código independiente de la posición para bibliotecas compartidas.
-I\$DB2PATH/include	Especifica la ubicación de los archivos de inclusión de DB2.
-c	Hace que se efectúe solamente la compilación, no la edición de enlaces. Este script tiene pasos separados para la compilación y la edición de enlaces.

Opciones de compilación y enlace para bldrtn

Opciones de enlace:

cc Hace que el compilador se utilice como interfaz del editor de enlaces.

-xarch=\$CFLAG_ARCH

Esta opción asegura que el compilador creará ejecutables válidos cuando establezca un enlace con libdb2.so. El valor de \$CFLAG_ARCH es "v8plusa" para la modalidad de 32 bits, o "v9" para la modalidad de 64 bits.

-mt Esta opción es necesaria porque la biblioteca de DB2 se enlaza con -mt.

-G Genera una biblioteca compartida.

-o \$1 Especifica el ejecutable.

\$1.o Incluir el archivo objeto del programa.

-L\$DB2PATH/\$LIB

Especifica la ubicación de las bibliotecas estáticas y compartidas de DB2 durante el enlace. Por ejemplo, para 32 bits: \$HOME/sqllib/lib32, y para 64 bits: \$HOME/sqllib/lib.

\$EXTRA_LFLAG

Especifica la ubicación de las bibliotecas compartidas de DB2 durante la ejecución. Para 32 bits contiene el valor "-R\$DB2PATH/lib32", y para 64 bits contiene el valor "-R\$DB2PATH/lib".

-ldb2 Enlazar con la biblioteca DB2.

Consulte la documentación del compilador para conocer otras opciones de compilador.

Tareas relacionadas:

- "Creación de rutinas C de UNIX" en la página 165

Ejemplos relacionados:

- "bldrtn -- Builds Solaris C routines (stored procedures or UDFs) (C)"

Creación de aplicaciones multihebra C en Solaris

DB2 da soporte a aplicaciones multihebra que hacen uso de bibliotecas de hebras de SUN y POSIX. Por omisión se utilizan hebras de Sun. Las aplicaciones multihebra que hacen uso de Forte C en Solaris se deben compilar y enlazar con -mt. Esto hace que se pase la opción -D_REENTRANT al preprocesador y la opción -lthread al enlazador. También es necesario que especifique -DUSE_UI_THREADS para la compilación, a fin de utilizar las API de las hebras "Unix International" de Sun.

Nota: Si desea utilizar hebras de POSIX, debe añadir la opción de compilador -D_POSIX_PTHREAD_SEMANTICS, que permite variantes para POSIX de funciones tales como getpwnam_r(), y también añade la opción de enlace -lpthread. Si está utilizando el script bldmt proporcionado, debe también suprimir la opción -DUSE_UI_THREADS.

El script bldmt contiene los mandatos para crear aplicaciones multihebra. Además de las opciones especificadas anteriormente, las opciones de compilación y enlace son las mismas que las utilizadas para el archivo de script de SQL incorporado, bldapp.

Procedimiento:

Para crear el programa de ejemplo dbthrds, a partir del archivo fuente dbthrds.sqc, especifique:

```
bldmt dbthrds
```

El resultado es un archivo ejecutable, dbthrds. Para ejecutar el archivo ejecutable sobre la base de datos sample, especifique:

```
dbthrds
```

Nota: Para programas multihebra con un número elevado de conexiones, puede ser necesario definir valores para los parámetros de kernel semsys:seminfo_semume y shmsys:shminfo_shmseg que sean mayores que sus valores por omisión. Consulte el enlace mostrado más abajo para el programa de utilidad db2osconf para conocer los valores recomendados para esos parámetros.

Conceptos relacionados:

- “Archivos de creación” en la página 110

Información relacionada:

- “Ejemplos de C” en la página 71
- “Opciones de compilación y enlace para aplicaciones C de Solaris” en la página 244
- “db2osconf - Mandato Programa de utilidad para los valores de parámetro de kernel” en el manual *Consulta de mandatos*

Ejemplos relacionados:

- “bldmt -- Builds Solaris C multi-threaded applications (C)”
- “dbthrds.sqc -- How to use multiple context APIs on UNIX (C)”
- “embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs (C)”

C++ para Solaris

Para obtener información sobre cómo crear aplicaciones C++ en sistemas operativos UNIX soportados, consulte el apartado “Creación de aplicaciones C++ de UNIX” en la página 168. Para obtener información sobre cómo crear rutinas C++ en sistemas operativos UNIX soportados, consulte el apartado “Creación de rutinas C++ de UNIX” en la página 172.

Script de creación para aplicaciones C++

```
#!/bin/sh
# SCRIPT: bldapp
# Crea aplicaciones C++ para Solaris
# Uso: bldapp <nombre_prog> [ <nombre_bd> [ <IDusuario> <contraseña> ]]

# Definir DB2PATH de modo que apunte al lugar desde donde se accederá a DB2.
# El valor por omisión es la vía de acceso de instancia estándar.
DB2PATH=$HOME/sql1lib

# Definir distintivos de compilación y enlace para programas de 32 y 64 bits
bitwidth=`LANG=C db2level | awk '/bits/{print $5}'`
if [ $bitwidth = "\"64\"" ];
then
    CFLAG_ARCH=v9
    LIB=lib
else
```



```

        CFLAG_ARCH=v8plusa
        LIB=lib32
    fi

    # Definir la vía de acceso de tiempo de ejecución.
    # Se seguirá LD_LIBRARY_PATH en vez de la vía de acceso de tiempo de ejecución
    # a menos que antes se desestablezca LD_LIBRARY_PATH para permitir que se
    # utilice la vía de acceso de tiempo de ejecución.
    EXTRA_LFLAG="-R$DB2PATH/$LIB"

    # Si es un programa de SQL incorporado, precompilar y vincular el programa.
    # Nota: algunos archivos .sqC no contienen SQL sino que enlazan en
    # utilemb.sqC, por lo que debe ignorar este aviso si lo recibe:
    # SQL0053W No se ha encontrado ninguna sentencia SQL en el programa.
    if [ -f $1".sqC" ]
    then
        ./embprep $1 $2 $3 $4
        # Compilar el programa de utilidad de comprobación de errores utilemb.C.
        CC -xarch=$CFLAG_ARCH -I$DB2PATH/include -c utilemb.C
    else
        # Compilar el programa de utilidad de comprobación de errores utilapi.C.
        CC -xarch=$CFLAG_ARCH -I$DB2PATH/include -c utilapi.C
    fi

    # Compilar el programa.
    CC -xarch=$CFLAG_ARCH -I$DB2PATH/include -c $1.C

    if [ -f $1".sqC" ]
    then
        # Enlazar el programa con utilemb.o
        CC -xarch=$CFLAG_ARCH -mt -o $1 $1.o utilemb.o \
            -L$DB2PATH/$LIB $EXTRA_LFLAG -ldb2
    else
        # Enlazar el programa con utilapi.o
        CC -xarch=$CFLAG_ARCH -mt -o $1 $1.o utilapi.o \
            -L$DB2PATH/$LIB $EXTRA_LFLAG -ldb2
    fi

```

Opciones de compilación y enlace para aplicaciones C++ de Solaris

La tabla siguiente muestra las opciones de compilación y enlace que DB2 recomienda para crear aplicaciones en C++ de SQL incorporado y de la API de DB2 con el compilador C++ de Forte, tal como muestra el script de creación bldapp.

Opciones de compilación y enlace para bldapp	
Opciones de compilación:	
CC	Denota el compilador C++.
-xarch=\$CFLAG_ARCH	Esta opción asegura que el compilador creará ejecutables válidos cuando establezca un enlace con libdb2.so. El valor de \$CFLAG_ARCH es "v8plusa" para la modalidad de 32 bits, o "v9" para la modalidad de 64 bits.
-I\$DB2PATH/include	Especifica la ubicación de los archivos de inclusión de DB2. Por ejemplo: \$HOME/sql1lib/include
-c	Hace que se efectúe solamente la compilación, no la edición de enlaces. Este script tiene pasos separados para la compilación y la edición de enlaces.

Opciones de compilación y enlace para bldapp

Opciones de enlace:

- CC** Hace que el compilador se utilice como interfaz del editor de enlaces.
- xarch=\$CFLAG_ARCH**
Esta opción asegura que el compilador creará ejecutables válidos cuando establezca un enlace con libdb2.so. El valor de \$CFLAG_ARCH es "v8plusa" para la modalidad de 32 bits, o "v9" para la modalidad de 64 bits.
- mt** Crea enlaces cuando se utiliza el soporte multihebra. Esta opción es necesaria para enlazar con libdb2.
Nota: Si se utilizan hebras de POSIX, las aplicaciones DB2 se deben también enlazar con -lpthread, tengan o no varias hebras.
- o \$1** Especifica el ejecutable.
- \$1.o** Incluir el archivo objeto del programa.
- utilemb.o**
Si es un programa de SQL incorporado, incluir el archivo objeto del programa de utilidad de SQL incorporado para la comprobación de errores.
- utilapi.o**
Si es un programa de SQL no incorporado, incluir el archivo objeto del programa de utilidad de la API de DB2 para la comprobación de errores.
- L\$DB2PATH/\$LIB**
Especifica la ubicación de las bibliotecas estáticas y compartidas de DB2 durante el enlace. Por ejemplo, para 32 bits: \$HOME/sql1lib/lib32, y para 64 bits: \$HOME/sql1lib/lib.
- \$EXTRA_LFLAG**
Especifica la ubicación de las bibliotecas compartidas de DB2 durante la ejecución. Para 32 bits contiene el valor "-R\$DB2PATH/lib32", y para 64 bits contiene el valor "-R\$DB2PATH/lib".
- ldb2** Enlazar con la biblioteca DB2.

Consulte la documentación del compilador para conocer otras opciones de compilador.

Tareas relacionadas:

- "Creación de aplicaciones C++ de UNIX" en la página 168

Ejemplos relacionados:

- "bldapp -- Builds Solaris C++ applications (C++)"

Script de creación para rutinas C++

```
#!/bin/sh
# SCRIPT: bldrtn
# Crea rutinas C++ para Solaris C++ (procedimientos almacenados o UDF)
# Uso: bldrtn <nombre_prog> [ <nombre_bd> ]

# Definir DB2PATH de modo que apunte al lugar desde donde se accederá a DB2.
# El valor por omisión es la vía de acceso de instancia estándar.
DB2PATH=$HOME/sql1lib

# Definir distintivos de compilación y enlace para programas de 32 y 64 bits
bitwidth=`LANG=C db2level | awk '/bits/{print $5}'`
if [ $bitwidth = "\"64\"" ];
then
    CFLAG_ARCH=v9
    LIB=lib
```

```

else
  CFLAG_ARCH=v8plusa
  LIB=lib32
fi

# Definir la vía de acceso de tiempo de ejecución.
# Se seguirá LD_LIBRARY_PATH en vez de la vía de acceso de tiempo de ejecución
# a menos que antes se desestablezca LD_LIBRARY_PATH para permitir que se
# utilice la vía de acceso de tiempo de ejecución.
EXTRA_LFLAG="-R$DB2PATH/$LIB"

# Si es un programa de SQL incorporado, precompilar y vincular el programa.
if [ -f $1".sqc" ]
then
./embprep $1 $2
fi

# Compilar el programa.
CC -xarch=$CFLAG_ARCH -mt -DUSE_UI_THREADS -Kpic \
  -I$DB2PATH/include -c $1.C

# Enlazar el programa y crear una biblioteca compartida.
CC -xarch=$CFLAG_ARCH -mt -G -o $1 $1.o -L$DB2PATH/$LIB \
  $EXTRA_LFLAG -ldb2

# Copiar la biblioteca compartida en el subdirectorío sqllib/function.
# Nota: el usuario debe tener permiso de escritura sobre este directorio.
rm -f $DB2PATH/function/$1
cp $1 $DB2PATH/function

```

Opciones de compilación y enlace para rutinas C++ de Solaris

La tabla siguiente muestra las opciones de compilación y enlace que DB2 recomienda para crear rutinas en C++ (procedimientos almacenados y funciones definidas por el usuario) con el compilador C++ de Forte, tal como muestra el script de creación bldrtn.

Opciones de compilación y enlace para bldrtn	
Opciones de compilación:	
CC	Denota el compilador C++.
-xarch=\$CFLAG_ARCH	Esta opción asegura que el compilador creará ejecutables válidos cuando establezca un enlace con libdb2.so. El valor de \$CFLAG_ARCH es "v8plusa" para la modalidad de 32 bits, o "v9" para la modalidad de 64 bits.
-mt	Permite el soporte multihebra, que es necesario pues las rutinas se pueden ejecutar en el mismo proceso que otras rutinas (THREADSAFE) o en el propio motor (NOT FENCED).
-DUSE_UI_THREADS	Permite el uso de las API de hebras "UNIX International" de Sun.
-Kpic	Genera código independiente de la posición para bibliotecas compartidas.
-I\$DB2PATH/include	Especifica la ubicación de los archivos de inclusión de DB2.
-c	Hace que se efectúe solamente la compilación, no la edición de enlaces. Este script tiene pasos separados para la compilación y la edición de enlaces.

Opciones de compilación y enlace para bldrtn

Opciones de enlace:

- CC** Hace que el compilador se utilice como interfaz del editor de enlaces.
- xarch=\$CFLAG_ARCH**
Esta opción asegura que el compilador creará ejecutables válidos cuando establezca un enlace con libdb2.so. El valor de \$CFLAG_ARCH es "v8plusa" para la modalidad de 32 bits, o "v9" para la modalidad de 64 bits.
- mt** Esta opción es necesaria porque la biblioteca de DB2 se enlaza con -mt.
- G** Genera una biblioteca compartida.
- o \$1** Especifica el ejecutable.
- \$1.o** Incluir el archivo objeto del programa.
- L\$DB2PATH/\$LIB**
Especifica la ubicación de las bibliotecas estáticas y compartidas de DB2 durante el enlace. Por ejemplo, para 32 bits: \$HOME/sqllib/lib32, y para 64 bits: \$HOME/sqllib/lib.
- \$EXTRA_LFLAG**
Especifica la ubicación de las bibliotecas compartidas de DB2 durante la ejecución. Para 32 bits contiene el valor "-R\$DB2PATH/lib32", y para 64 bits contiene el valor "-R\$DB2PATH/lib".
- ldb2** Enlazar con la biblioteca DB2.

Consulte la documentación del compilador para conocer otras opciones de compilador.

Tareas relacionadas:

- "Creación de rutinas C++ de UNIX" en la página 172

Ejemplos relacionados:

- "bldrtn -- Builds Solaris C++ routines (stored procedures or UDFs) (C++)"

Creación de aplicaciones multihebra C++ en Solaris

DB2 da soporte a aplicaciones multihebra que hacen uso de bibliotecas de hebras de SUN y POSIX. Por omisión se utilizan hebras de Sun. Las aplicaciones multihebra que hacen uso de Forte C++ en Solaris se deben compilar y enlazar con -mt. Esto hace que se pase la opción -D_REENTRANT al preprocesador y la opción -lthread al enlazador. También es necesario que especifique -DUSE_UI_THREADS para la compilación, a fin de utilizar las API de las hebras "Unix International" de Sun.

Nota: Si desea utilizar hebras de POSIX, debe añadir la opción de compilador -D_POSIX_PTHREAD_SEMANTICS, que permite variantes para POSIX de funciones tales como getpwnam_r(), y también añade la opción de enlace -lpthread. Si está utilizando el script bldmt proporcionado, debe también suprimir la opción -DUSE_UI_THREADS.

El script bldmt contiene los mandatos para crear aplicaciones multihebra. Además de las opciones especificadas anteriormente, las opciones de compilación y enlace son las mismas que las utilizadas para el archivo de script de SQL incorporado, bldapp.

Procedimiento:

Para crear el programa de ejemplo dbthrds, a partir del archivo fuente dbthrds.sqC, especifique:

```
bldmt dbthrds
```

El resultado es un archivo ejecutable, dbthrds. Para ejecutar el archivo ejecutable sobre la base de datos sample, especifique:

```
dbthrds
```

Nota: Para programas multihebra con un número elevado de conexiones, puede ser necesario definir valores para los parámetros de kernel semsys:seminfo_semume y shmsys:shminfo_shmseg que sean mayores que sus valores por omisión. Consulte el enlace mostrado más abajo para el programa de utilidad db2osconf para conocer los valores recomendados para esos parámetros.

Conceptos relacionados:

- “Archivos de creación” en la página 110

Información relacionada:

- “Ejemplos de C” en la página 71
- “Opciones de compilación y enlace para aplicaciones C++ de Solaris” en la página 249
- “db2osconf - Mandato Programa de utilidad para los valores de parámetro de kernel” en el manual *Consulta de mandatos*

Ejemplos relacionados:

- “bldmt -- Builds Solaris C++ multi-threaded applications (C++)”
- “dbthrds.sqC -- How to use multiple context APIs on UNIX (C++)”
- “embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs (C)”

COBOL para Micro Focus

Para obtener información sobre cómo crear aplicaciones COBOL para Micro Focus en sistemas operativos UNIX soportados, consulte el apartado “Creación de aplicaciones Micro Focus COBOL de UNIX” en la página 176. Para obtener información sobre cómo crear rutinas COBOL para Micro Focus en sistemas operativos UNIX soportados, consulte el apartado “Creación de rutinas Micro Focus COBOL de UNIX” en la página 177.

Configuración del compilador Micro Focus COBOL en Solaris

Si desarrolla aplicaciones que contienen SQL incorporado y llamadas a la API de DB2, y está utilizando el compilador Micro Focus COBOL, debe tener en cuenta varias consideraciones.

Procedimiento:

- Cuando precompile la aplicación utilizando el mandato db2 prep del procesador de línea de mandatos, utilice la opción target mfcob.
- Debe incluir el directorio del archivo COPY de DB2 para COBOL en la variable de entorno COBCOPY de Micro Focus COBOL. La variable de entorno COBCPY especifica la ubicación de los archivos COPY. Los archivos COPY de DB2

correspondientes a Micro Focus COBOL residen en `sql1lib/include/cobol_mf`, dentro del directorio de la instancia de la base de datos.

Para incluir el directorio en la variable de entorno, especifique:

– En el shell bash o Korn:

```
export COBCPY=${COBCPY}:${HOME}/sql1lib/include/cobol_mf
```

– En el shell C:

```
setenv COBCPY ${COBCPY}:${HOME}/sql1lib/include/cobol_mf
```

Nota: Puede definir COBCPY dentro del archivo `.profile`.

Tareas relacionadas:

- “Creación de aplicaciones Micro Focus COBOL de UNIX” en la página 176
- “Creación de rutinas Micro Focus COBOL de UNIX” en la página 177

Información relacionada:

- “Opciones de compilación y enlace para aplicaciones Micro Focus COBOL de Solaris” en la página 254
- “Opciones de compilación y enlace para rutinas Micro Focus COBOL de Solaris” en la página 256

Script de creación para aplicaciones COBOL para Micro Focus

```
#!/bin/sh
# SCRIPT: bldapp
# Crea aplicaciones COBOL para Micro Focus de Solaris
# Uso: bldapp [ <nombre_bd> [ <IDusuario> <contraseña> ]

# Definir DB2PATH de modo que apunte al lugar desde donde se accederá a DB2.
# El valor por omisión es la vía de acceso de instancia estándar.
DB2PATH=${HOME}/sql1lib

# Defina COBCPY de forma que incluya el directorio de los archivos COPY de DB2.
COBCPY=${COBCPY}:${DB2PATH}/include/cobol_mf

# Si es un programa de SQL incorporado, precompilar y vincular el programa.
if [ -f $1".sqb" ]
then
  ./embprep $1 $2 $3 $4
fi

# Compilar el programa de utilidad de comprobación de errores checkerr.cbl.
cob -cx checkerr.cbl

# Compilar el programa.
cob -cx $1.cbl

# Enlazar el programa.
cob -x $1.o checkerr.o -L${DB2PATH}/lib -ldb2 -ldb2gmf
```

Opciones de compilación y enlace para aplicaciones Micro Focus COBOL de Solaris

La tabla siguiente muestra las opciones de compilación y enlace que DB2 recomienda para crear aplicaciones COBOL de SQL incorporado y de la API de DB2 con el compilador MicroFocus COBOL de Solaris, tal como muestra el script de creación `bldapp`.

Opciones de compilación y enlace para bldapp	
Opciones de compilación:	
cob	Denota el compilador Micro Focus COBOL.
-cx	Compila y crea el módulo objeto.
Opciones de enlace:	
cob	Hace que el compilador se utilice como interfaz del editor de enlaces.
-x	Especifica un programa ejecutable.
\$1.o	Incluir el archivo objeto del programa.
checkerr.o	Hace que se incluya el archivo objeto de programa de utilidad para la comprobación de errores.
-L\$DB2PATH/lib	Especifica la ubicación de las bibliotecas estáticas y compartidas de DB2 durante el enlace. Por ejemplo: \$HOME/sqllib/lib.
-ldb2	Enlazar con la biblioteca DB2.
-ldb2gmf	Denota enlazar con la biblioteca del gestor de excepciones de DB2 correspondiente a Micro Focus COBOL.
Consulte la documentación del compilador para conocer otras opciones de compilador.	

Tareas relacionadas:

- “Creación de aplicaciones Micro Focus COBOL de UNIX” en la página 176

Ejemplos relacionados:

- “bldapp -- Builds Solaris Micro Focus COBOL applications”

Script de creación para rutinas COBOL para Micro Focus

```

#!/bin/sh
# SCRIPT: bldrtn
# Crea rutinas COBOL para Micro Focus de Solaris (procedimientos almacenados)
# Uso: bldrtn <nombre_prog> [ <nombre_bd> ]

# Definir DB2PATH de modo que apunte al lugar desde donde se accederá a DB2.
# El valor por omisión es la vía de acceso de instancia estándar.
DB2PATH=$HOME/sqllib

# Defina COBCPY de forma que incluya el directorio de los archivos COPY de DB2.
COBCPY=$COBCPY:$DB2PATH/include/cobol_mf

# Si es un programa de SQL incorporado, precompilar y vincular el programa.
if [ -f $1".sqb" ]
then
./embprep $1 $2
fi

# Compilar el programa.
cob -cx $1.cbl

# Enlazar el programa.
cob -yo $1 $1.o -L$DB2PATH/lib -ldb2 -ldb2gmf

```

```

| # Copiar la biblioteca compartida en el subdirectorío sqllib/function.
| # El usuario debe tener permiso de escritura sobre este directorio.
| rm -f $DB2PATH/function/$1
| cp $1 $DB2PATH/function

```

Opciones de compilación y enlace para rutinas Micro Focus COBOL de Solaris

La tabla siguiente muestra las opciones de compilación y enlace que DB2 recomienda para crear rutinas COBOL (procedimientos almacenados) con el compilador Micro Focus COBOL de Solaris, tal como muestra el script de creación bldrtn.

Opciones de compilación y enlace para bldrtn	
Opciones de compilación:	
cob	Denota el compilador COBOL.
-cx	Compila y crea el módulo objeto.
Opciones de enlace:	
cob	Hace que el compilador se utilice como interfaz del editor de enlaces.
-y	Crea una biblioteca compartida autónoma auto-contenida.
-o \$1	Especifica el programa ejecutable.
\$1.o	Especifica el archivo objeto del programa.
-L\$DB2PATH/lib	Especifica la ubicación de las bibliotecas compartidas de tiempo de ejecución de DB2. Por ejemplo: \$HOME/sql1ib/lib. Si el usuario no especifica la opción -L, el compilador utiliza esta vía de acceso: /usr/lib:/lib.
-ldb2	Enlazar con la biblioteca DB2.
-ldb2gmf	Establece un enlace con la biblioteca del gestor de excepciones de DB2 correspondiente a Micro Focus COBOL.
Consulte la documentación del compilador para conocer otras opciones de compilador.	

Tareas relacionadas:

- “Creación de rutinas Micro Focus COBOL de UNIX” en la página 177

Ejemplos relacionados:

- “bldrtn -- Builds Solaris Micro Focus COBOL routines (stored procedures)”

Capítulo 14. Windows

La opción de precompilación WCHARTYPE CONVERT	257	Opciones de compilación y enlace para aplicaciones C/C++ de Windows	286
Funciones de tabla de base de datos OLE DB (base de datos de enlace e integración de objetos)	258	Creación de rutinas C/C++ en Windows	287
Windows Management Instrumentation (WMI)	259	Archivo de proceso por lotes para rutinas C/C++	290
Microsoft Visual Basic	260	Opciones de compilación y enlace para rutinas C/C++ de Windows	291
Creación de aplicaciones ADO con Visual Basic	260	Creación de aplicaciones C/C++ de conexión múltiple en Windows.	292
Creación de transacciones emparejadas débilmente con Visual Basic	262	IBM VisualAge COBOL	295
Resolución de problemas de un proyecto de transacción emparejada débilmente de Visual Basic	264	Configuración del compilador IBM COBOL en Windows.	295
Creación de aplicaciones RDO con Visual Basic	266	Creación de aplicaciones IBM COBOL en Windows.	296
Automatización del enlace e integración de objetos (OLE) con Visual Basic.	267	Archivo de proceso por lotes para aplicaciones COBOL de IBM	298
.NET	268	Opciones de compilación y enlace para aplicaciones IBM COBOL de Windows	299
Creación de aplicaciones C# .NET	268	Creación de rutinas IBM COBOL en Windows	300
Archivo de proceso por lotes para aplicaciones C# .NET	269	Archivo de proceso por lotes para rutinas COBOL de IBM	301
Opciones de compilación y enlace para aplicaciones C# .NET.	270	Opciones de compilación y enlace para rutinas IBM COBOL de Windows	302
Creación de aplicaciones Visual Basic .NET	272	Micro Focus COBOL	303
Archivo de proceso por lotes para aplicaciones Visual Basic .NET	273	Configuración del compilador Micro Focus COBOL en Windows	303
Opciones de compilación y enlace para aplicaciones Visual Basic .NET.	274	Creación de aplicaciones Micro Focus COBOL en Windows.	304
Creación de rutinas CLR (Common Language Runtime) .NET.	276	Archivo de proceso por lotes para aplicaciones COBOL para Micro Focus	306
Archivo de proceso por lotes para rutinas C# .NET	279	Opciones de compilación y enlace para aplicaciones Micro Focus COBOL de Windows	306
Archivo de proceso por lotes para rutinas Visual Basic .NET	279	Creación de rutinas Micro Focus COBOL en Windows.	307
Opciones de compilación y enlace para rutinas de CLR .NET	279	Archivo de proceso por lotes para rutinas COBOL para Micro Focus	308
Microsoft Visual C++.	280	Opciones de compilación y enlace para rutinas Micro Focus COBOL de Windows	309
Creación de aplicaciones ADO con Visual C++	281	Object REXX	309
Automatización del enlace e integración de objetos (OLE) con Visual C++	282	Creación de aplicaciones Object REXX en Windows.	309
Creación de aplicaciones C/C++ en Windows	283		
Archivo de proceso por lotes para aplicaciones C/C++	285		

Este capítulo proporciona información detallada para crear aplicaciones en sistemas operativos Windows. Para obtener las actualizaciones más recientes sobre el desarrollo de aplicaciones de DB2 para Windows, visite la página Web en:

<http://www.ibm.com/software/data/db2/udb/ad>

La opción de precompilación WCHARTYPE CONVERT

La opción de precompilación WCHARTYPE determina si los datos gráficos se deben manejar en formato de multibyte o en formato de carácter ancho utilizando el tipo de datos wchar_t.

En DB2[®] para sistemas operativos Windows[®], la opción WCHARTYPE CONVERT se puede utilizar para aplicaciones compiladas con el compilador Microsoft[®] Visual C++. Pero no utilice la opción CONVERT con este compilador si la aplicación inserta datos dentro de una base de datos DB2 en una página de códigos que es diferente de la página de códigos de la base de datos. En estos casos DB2 normalmente realiza una conversión de página de códigos; pero el entorno de ejecución de Microsoft C no maneja caracteres de sustitución para determinados caracteres de doble byte. Esto puede producir errores de conversión durante la ejecución.

La opción por omisión para WCHARTYPE es NOCONVERT. Cuando se utiliza la opción NOCONVERT, no tiene lugar ninguna conversión de caracteres implícita entre la aplicación y el gestor de bases de datos. Los datos de las variables gráficas de lenguaje principal se intercambian con el gestor de bases de datos en forma de caracteres inalterados del juego de caracteres de doble byte (juego DBCS).

Si necesita convertir datos gráficos al formato de múltiples bytes desde el formato de caracteres WIDE, utilice la función `wcstombs()`. Por ejemplo:

```
wchar_t widechar[200];
wchar_t mb[200];
wcstombs((char *)mb,widechar,200);

EXEC SQL INSERT INTO TABLENAME VALUES(:mb);
```

Similarmente, utilice la función `mbstowcs()` para convertir desde el formato de múltiples bytes al formato de caracteres WIDE.

No emita una llamada `setlocale()` desde su aplicación si ésta está vinculada estáticamente a las bibliotecas de ejecución de C, pues esto podría conducir a errores de conversión de C durante la ejecución. Puede utilizar `setlocale()` sin ningún problema si la aplicación está vinculada dinámicamente a la biblioteca de ejecución de C. Esto también es aplicable a rutinas (procedimientos almacenados y funciones definidas por el usuario).

Conceptos relacionados:

- “Tipos de datos `wchar_t` y `sqldbcchar` en C y C++” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente*
- “Opción del precompilador WCHARTYPE en C y C++” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente*
- “Variables gráficas del lenguaje principal en rutinas C/C++” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de servidor*

Información relacionada:

- “Mandato PRECOMPILE” en el manual *Consulta de mandatos*

Funciones de tabla de base de datos OLE DB (base de datos de enlace e integración de objetos)

DB2[®] da soporte a funciones de tabla de base de datos OLE. Para estas funciones, no es necesario crear aplicaciones, salvo la creación de la DLL CREATE FUNCTION. DB2 proporciona archivos de ejemplo para funciones de tabla de base de datos OLE en el directorio `sql1ib\samples\oledb`. Estos archivos son archivos del Procesador de Línea de Mandatos (CLP). Se pueden crear siguiendo estos pasos:

1. `db2 connect to nombre_basedatos`

2. `db2 -t -v -f nombre_archivo.db2`
3. `db2 terminate`

donde `nombre_basedatos` es la base de datos a la que se está conectando y `nombre_archivo` es el nombre del archivo de CLP, con la extensión `.db2`.

Estos mandatos se deben ejecutar en una ventana de mandatos de DB2.

Conceptos relacionados:

- “Funciones de tabla de OLE DB” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente*
- “Funciones de tabla de OLE DB definidas por el usuario” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de servidor*

Información relacionada:

- “Ejemplos de funciones de tabla de OLE DB (base de datos de enlace e integración de objetos)” en la página 97

Windows Management Instrumentation (WMI)

Windows[®] Management Instrumentation (WMI) es un componente clave de los servicios de gestión de Windows de Microsoft[®]. WMI proporciona un modelo coherente y descriptivo de la configuración, el estado y los aspectos funcionales de las aplicaciones y del sistema.

El componente WMI para DB2[®] permite a las aplicaciones WMI supervisar los servicios de servidor DB2, listar y crear bases de datos, configurar valores funcionales y realizar acciones de copia, restauración y actualización de bases de datos.

DB2 proporciona archivos de ejemplo de WMI para el lenguaje Visual Basic Scripting en el directorio `sql1lib\samples\wmi`. Antes de ejecutar los programas de ejemplo, asegúrese de que el proveedor de DB2 WMI esté registrado, ejecutando estos mandatos:

```
mofcomp %DB2PATH%\bin\db2wmi.mof
regsvr32 %DB2PATH%\bin\db2wmi.dll
```

donde `%DB2PATH%` es la vía donde está instalado DB2.

Utilice el mandato `cscript` para ejecutar los programas de ejemplo de Visual Basic Script. Por ejemplo, para ejecutar el script de ejemplo `listsrvr`, especifique:

```
cscript listsrvr.vbs
```

Conceptos relacionados:

- “Introduction to Windows Management Instrumentation (WMI)” en el manual *Administration Guide: Implementation*
- “DB2 Universal Database integration with Windows Management Instrumentation” en el manual *Administration Guide: Implementation*

Información relacionada:

- “Ejemplos de Windows Management Instrumentation” en la página 109

Creación de aplicaciones ADO con Visual Basic

ActiveX Data Objects (ADO) le permite escribir una aplicación para acceder y manejar datos de un servidor de bases de datos mediante un proveedor de bases de datos OLE. Las ventajas principales de ADO son una alta velocidad, la facilidad de manejo, una baja actividad de la memoria y una escasa ocupación de espacio de disco.

Los programas ADO de ejemplo creados con Visual Basic se encuentran en el directorio `sql1lib\samples\VB\ADO`.

Nota: Para ejecutar los programas de ejemplo ADO para DB2, son recomendables estas versiones (o versiones posteriores) de los componentes siguientes:

1. Visual Basic 6.0 Professional Edition
2. Microsoft Data Access 2.7 SDK (instalado opcionalmente con DB2 Versión 8)
3. Service Pack 5 de Visual Basic, que puede obtener en <http://msdn.microsoft.com/vstudio/sp/vs6sp5/vbfixes.asp>.
4. El Service Pack más reciente de Visual Studio, que se puede obtener en <http://msdn.microsoft.com/vstudio/>.

Procedimiento:

Puede utilizar cualquiera de estos dos proveedores compatibles con ODBC:

- Proveedor IBM OLE DB para DB2
- Proveedor Microsoft OLE DB para ODBC

Utilización del proveedor IBM OLE DB para DB2

Los clientes DB2 Versión 8.2 en sistemas operativos Windows instalarán opcionalmente IBM DADB2, que es el proveedor compatible IBM OLE DB 2.0 para DB2. El proveedor expone interfaces para los usuarios que desean acceder a datos de una base de datos DB2. El proveedor IBM OLE DB para DB2 da soporte a los tipos de aplicación ADO siguientes:

- Microsoft Active Server Pages (ASP)
- Aplicaciones Microsoft Visual Studio C++ y Visual Basic
- Microsoft Visual Interdev

Para conocer detalles sobre estos tipos de aplicaciones, consulte la documentación de ADO.

Para acceder a un servidor DB2 utilizando el proveedor IBM OLE DB para DB2, la aplicación Visual Basic debe especificar la palabra clave PROVIDER en la cadena de conexión de ADO, de esta manera:

```
Dim c1 As ADODB.Connection
Dim c1str As String
c1str = "Provider=ibmdadb2; DSN=aliasDB2; UID=IDusuario; PWD=contraseña"
c1.Open c1str
...
```

donde `aliasDB2` es el alias de la base de datos DB2 que está catalogada en el directorio de bases de datos DB2.

Nota: Si utiliza el proveedor IBM OLE DB para DB2, no es necesario que efectúe el paso de catalogación ODBC para la fuente de datos. Este paso es necesario cuando utiliza el proveedor OLE DB para ODBC.

Utilización del proveedor Microsoft OLE DB para ODBC

Para utilizar ADO con el proveedor Microsoft OLE DB y Visual Basic, necesita establecer una referencia que apunte a la biblioteca de tipos ADO. Siga estos pasos:

1. Seleccione "Referencias" en el menú Proyecto.
2. Seleccione la casilla correspondiente a "Microsoft ActiveX Data Objects <número_versión> Library"
3. Pulse "Bien".

donde <número_versión> es la versión actual de la biblioteca ADO.

Una vez efectuados los pasos anteriores, podrá acceder a los objetos, métodos y propiedades de ADO utilizando el Navegador de Objetos VBA y el Editor IDE.

Establezca una conexión:

```
Dim db As Connection  
Set db = New Connection
```

Defina los cursores del extremo cliente proporcionados por la biblioteca de cursores local:

```
db.CursorLocation = adUseClient
```

y defina el proveedor para que ADO utilice el controlador Microsoft ODBC.

Acceso a la base de datos "sample" con ADO

Un programa Visual Basic completo incluye formularios y otros elementos gráficos, y el usuario necesita visualizar el programa dentro del entorno Visual Basic. Después de conectarse a la base de datos sample de DB2 utilizando el proveedor IBM OLE DB o el proveedor Microsoft OLE DB, tal como se describió anteriormente, puede utilizar los siguientes mandatos de Visual Basic como parte de un programa para acceder a la base de datos.

Abra la base de datos sample sin especificar un ID de usuario ni una contraseña; es decir, utilice el usuario actual:

```
db.Open "SAMPLE"
```

Cree un conjunto de registros:

```
Set adoPrimaryRS = New Recordset
```

Utilice una sentencia SELECT para llenar el conjunto de registros:

```
adoPrimaryRS.Open "select EMPNO, LASTNAME, FIRSTNME, MIDINIT, EDLEVEL, JOB  
from EMPLOYEE Order by EMPNO", db
```

A partir de este punto, el programador puede utilizar los métodos ADO para acceder a los datos, por ejemplo, para pasar al conjunto de registros siguiente:

```
adoPrimaryRS.MoveNext
```

Suprimir el registro actual del conjunto de registros:

```
adoPrimaryRS.Delete
```

Además, el programador puede hacer lo siguiente para acceder a un campo individual:

```
Dim Text1 as String
Text1 = adoPrimaryRS!LASTNAME
```

Conceptos relacionados:

- “Objetivo de IBM OLE DB Provider para DB2” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente*
- “Tipos de aplicaciones soportados por IBM OLE DB Provider para DB2” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente*
- “Conexiones con fuentes de datos con aplicaciones ADO Visual Basic” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente*
- “Habilitación automática de servicios OLE DB por parte de IBM OLE DB Provider” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente*
- “Manipulación de objetos grandes con IBM OLE DB Provider” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente*
- “Soporte de transacciones distribuidas MTS y COM+ e IBM OLE DB Provider” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente*
- “Restricciones de IBM OLE DB Provider” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente*
- “Objetos de datos ActiveX y Objetos de datos remotos” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente*

Información relacionada:

- “Soporte de IBM OLE DB Provider de interfaces y componentes de OLE DB” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente*
- “Soporte de IBM OLE DB Provider de las propiedades de OLE DB” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente*
- “Soporte de IBM OLE DB Provider de propiedades y métodos ADO” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente*
- “Ejemplos de Visual Basic” en la página 105

Creación de transacciones emparejadas débilmente con Visual Basic

XA brinda dos maneras de que las hebras de aplicaciones de control puedan participar en una sola transacción global XA: emparejamiento fuerte y emparejamiento débil. El proyecto de ejemplo, `LCTransTest`, muestra transacciones XA emparejadas débilmente. Los archivos de ejemplo están ubicados en el directorio `sql11ib\samples\VB\MTS`.

Procedimiento:

Para crear y ejecutar el ejemplo de transacciones emparejadas débilmente, realice los pasos siguientes:

1. **Cree el proyecto `LCTransTest.vbp`**
 - a. Abra el proyecto “`LCTransTest.vbp`” realizando una doble pulsación sobre el mismo.
 - b. Si recibe el mensaje de error: “No se puede establecer el componente con versión compatible X:\...\LCTransTest.dll”, pulse “Bien” para continuar.

- c. Compile el proyecto. Vaya a "Archivo" -> "Crear LCTransTest.dll" y luego pulse "Bien".
 - d. Para resolver el problema de incompatibilidad de versión, pulse con el botón derecho del ratón el proyecto "LCTransTest (LCTransTest.vbp)" ubicado en el panel superior derecho. A continuación, seleccione "Propiedades de LCTransTest". En la ventana "LCTransTest - Propiedades del proyecto". Pulse la pestaña "Componente". En la sección "Compatibilidad de versión", seleccione "Compatibilidad binaria".
 - e. Guarde el proyecto. Vaya a "Archivo" -> "Guardar proyecto".
 - f. Cierre el proyecto.
- 2. Cree el proyecto Main.vbp**
- a. Abra el proyecto "Main.vbp" realizando una doble pulsación sobre el mismo.
 - b. Es probable que reciba el mensaje de aviso: "No se ha podido crear la referencia: X:\...\LCTransTest.dll", pulse "Bien" para continuar.
 - c. Vaya a "Proyecto" -> "Referencias" (en la barra de herramientas).
 - d. En la ventana "Referencias - main.vbp", asegúrese que el recuadro "Microsoft ActiveX Data Objects 2.7 Library" esté seleccionado. Vaya a "Examinar...". Busque el proyecto LCTransTest.dll que ha generado en el paso 1 y pulse "Abrir" para añadir esta referencia. Pulse "Bien" en la ventana "Referencias - main.vbp".
 - e. Compile el proyecto. Vaya a "Archivo" -> "Crear main.exe" y luego pulse "Bien".
 - f. Guarde el proyecto. Vaya a "Archivo" -> "Guardar proyecto".
 - g. Cierre el proyecto.
- 3. Otros valores**
- a. En Windows, vaya a "Inicio" -> "Configuración" -> "Panel de control" -> "Herramientas administrativas" -> "Servicios de componentes".
 - b. En la ventana "Servicios de componentes", expanda "Servicios de componentes" en el panel de la izquierda hasta que vea "Aplicaciones COM+".
 - c. Pulse con el botón derecho del ratón "Aplicaciones COM+", seleccione "Nuevo" -> "Aplicación".
 - d. En la ventana emergente, "Asistente para instalación de aplicación COM", pulse "Siguiente".
 - e. Seleccione "Crear una aplicación vacía".
 - f. Entre "LCTransTest" como nombre de la nueva aplicación. Mantenga "Tipo de activación" como "Aplicación de servidor". Pulse "Siguiente".
 - g. Pulse "Siguiente" y luego "Finalizar".
 - h. Expanda "LCTransTest", pulse con el botón derecho del ratón "Componentes". Vaya a "Nuevo" -> "Componentes" -> seleccione "Importar componentes que ya están registrados" -> pulse "LCTransTest.TestClass" -> "Siguiente" -> "Finalizar".
 - i. Expanda "Componentes". Pulse con el botón derecho del ratón "LCTransTest.TestClass". Vaya a "Propiedades". En la pestaña "Transacción", seleccione "Necesario" para "Soporte de transacción".
 - j. Reinicie el Microsoft Distributed Transaction Coordinator pulsando con el botón derecho del ratón "Servicios de componentes" -> "Sistemas" -> "Mi

PC". Seleccione "Detener MS DTC". Espere hasta que se detenga y, a continuación, pulse con el botón derecho del ratón "Mi PC" -> "Iniciar MS DTC" para reiniciar DTC.

4. Para ejecutar el ejemplo en modalidad de depuración

- a. Abra LCTransTest.vbp.
- b. En LCTransTest, asegúrese que en "propiedades del proyecto*", bajo la pestaña "Depuración", está seleccionado "Esperar a que se creen componentes". (Lo debe estar por omisión.)
- c. Coloque un punto de interrupción en la línea "con1.Open connString" (colocando el cursor sobre dicha línea y pulsando F9).
- d. Pulse F5 (o seleccione "Iniciar" en el menú desplegable "Ejecutar"). Cuando se cargue esta dll, y se ejecute este método, el depurador detendrá la ejecución en ese punto de interrupción.
- e. Abra main.vbp.
- f. En main.vbp, defina los argumentos de la línea de mandatos. En Propiedades del proyecto, bajo la pestaña "Crear", en el recuadro "Argumentos de la línea de mandatos", escriba lo siguiente:

```
provider=ibmdadb2;dsn=<nombrebd>;uid=<IDusuario>;pwd=<contraseña> <nombrearchivo>
```

donde <nombrebd> es el nombre de una base de datos que tenga y <nombrearchivo> es el nombre de un archivo (incluida la vía de acceso) que contendrá información de salida. Por ejemplo, C:\lctoutput.txt. A continuación, pulse Bien.
- g. En Visual Basic, puede utilizar "Debug ->Step Into" <F8> o "Debug->Step Over" <despl + F8> para ejecutar el ejecutable de línea de código en línea de código.
- h. Cuando llegue a la línea que llama a "transTest.RunTest" en el ejecutable principal, e intente evitarla, aflorará la otra ventana de Visual Basic (el proyecto LCTransTest que ha abierto) y se le detendrá en el punto de interrupción que ha colocado ahí. A continuación puede utilizar "Step Into" o "Step Over" para avanzar por el método RunTest de línea de código en línea de código.

Tareas relacionadas:

- "Creación de aplicaciones ADO con Visual Basic" en la página 260
- "Resolución de problemas de un proyecto de transacción emparejada débilmente de Visual Basic" en la página 264

Información relacionada:

- "Ejemplos de Visual Basic" en la página 105

Resolución de problemas de un proyecto de transacción emparejada débilmente de Visual Basic

Una característica del entorno integrado de Visual Basic es que cada vez que se compila una biblioteca enlazada por datos (dll), Visual Basic crea un nuevo identificador exclusivo globalmente (GUID) para la misma, y lo registra en el registro de Windows. Una vez que se crea la dll del proyecto, se registra con el coordinador de transacciones distribuidas (DTC) utilizando el GUID. Si se reconstruye la dll del proyecto, Visual Basic le asigna un nuevo GUID y también lo registra en el registro de Windows. Por consiguiente el GUID que utiliza el DTC

para hacer referencia a la dll del proyecto ahora está desfasado, y en el registro de Windows existen dos entradas distintas para el proyecto.

Procedimiento:

Para evitar este problema, después de crear el proyecto por primera vez debe modificar las propiedades del mismo:

1. En la pestaña Componente, seleccione "Compatibilidad binaria" y luego utilice el botón "... " para buscar la vía de acceso completa para la dll que acaba de crear.
2. A continuación, pulse el botón "Aceptar" y guarde el proyecto de inmediato.

Ahora, cada vez que recompila la dll del proyecto ésta conservará el mismo GUID. Sin embargo, si cambia la interfaz con la dll (por ejemplo, añadiendo o eliminando métodos, o cambiando los parámetros de los métodos existentes), tendrá que utilizar un nuevo GUID.

Si en el registro de Windows ya existen varias entradas de GUID, haga lo siguiente:

1. Busque el nombre del proyecto con el editor del registro.
2. Elimine todas las apariciones del nombre de proyecto que no sean la hallada en la lista de proyectos recientes de Visual Basic.

La información de conexión, incluidos el ID de usuario y la contraseña, debe ser idéntica en el DTC y en la aplicación Visual Basic para que se produzca la transacción emparejada débilmente. Normalmente, para acceder directamente a una dll se utiliza los métodos públicos de la misma. Sin embargo, cuando el DTC está implicado, éste encapsula la dll e intercepta todas las llamadas de entrada para los métodos, así como los resultados de salida de dichos métodos. De este modo, el DTC puede indicar cuándo la actividad de base de datos de uno de estos métodos se debe emparejar débilmente con la actividad de otra base de datos del mismo objeto, o con la actividad de base de datos de un objeto distinto.

Para evitar cualquier problema derivado de esta situación, haga lo siguiente:

1. En las Propiedades de aplicación del DTC, bajo la pestaña "Identidad", seleccione "Este usuario".
2. Utilice el botón Examinar para buscar el ID del usuario que va a ejecutar este proyecto.
3. Utilice el mismo ID de usuario y la misma contraseña en la serie de conexión del proyecto.

Si en la serie de conexión de la aplicación Visual Basic utiliza el mismo ID de usuario y la misma contraseña que ha utilizado para iniciar una sesión en el sistema, no es necesario que realice este paso adicional.

Para asegurarse que el proyecto de transacción emparejada débilmente está funcionando correctamente, puede hacer lo siguiente:

1. Observe el archivo de salida creado por el ejecutable y confirme que se están produciendo actualizaciones en la base de datos.
2. Examine un rastreo de la CLI para ENLIST_IN_DTC.

Si falla cualquiera de estas pruebas, la dll no está bien registrada con el DTC y no se están produciendo transacciones emparejadas débilmente,

Tareas relacionadas:

- "Creación de aplicaciones ADO con Visual Basic" en la página 260
- "Creación de transacciones emparejadas débilmente con Visual Basic" en la página 262

Información relacionada:

- "Ejemplos de Visual Basic" en la página 105

Creación de aplicaciones RDO con Visual Basic

Remote Data Objects (RDO) proporciona un modelo de información para acceder a fuentes de datos remotas a través de ODBC. RDO ofrece un conjunto de objetos que hacen más fácil el conectar con una base de datos, ejecutar consultas y procedimientos almacenados, manejar los resultados y confirmar los cambios en el servidor. RDO está pensado específicamente para acceder a fuentes remotas de datos relacionales ODBC, facilita el uso de ODBC sin ser necesario un código de aplicación complejo y es un medio básico para acceder a una base de datos relacional que se expone con un controlador ODBC. RDO implementa una capa de código sobre la API de ODBC (Open Database Connectivity) y el gestor de controladores; esta capa establece conexiones, crea conjuntos y cursores del resultado y ejecuta procedimientos complejos con un mínimo de recursos de la estación de trabajo.

DB2 proporciona programas de ejemplo de RDO para Visual Basic en el directorio `sql1lib\samples\VB`.

Procedimiento:

Para utilizar RDO con Microsoft Visual Basic, es necesario establecer una referencia con el proyecto de Visual Basic. Siga estos pasos:

1. Seleccione "Referencias" en el menú Proyecto.
2. Seleccione la casilla correspondiente a "Microsoft Remote Data Object <número_versión>"
3. Pulse "Bien".

donde <número_versión> es la versión actual de ADO.

Un programa Visual Basic completo incluye formularios y otros elementos gráficos, y el usuario necesita visualizar el programa dentro del entorno Visual Basic. A continuación siguen mandatos de Visual Basic que forman parte de un programa DB2 que conecta con la base de datos `sample`, abre un conjunto de registros que selecciona todas las columnas de la tabla `EMPLOYEE` y luego muestra los nombres de los empleados en una ventana de mensajes, uno a uno:

```
Dim rdoEn As rdoEngine
Dim rdoEv As rdoEnvironment
Dim rdoCn As rdoConnection
Dim Cnct$
Dim rdoRS As rdoResultset
Dim SQLQueryDB As String
```

Asignar la cadena de conexión:

```
Cnct$ = "DSN=SAMPLE;UID=;PWD=";
```

Definir el entorno RDO:

```
Set rdoEn = rdoEngine
Set rdoEv = rdoEn.rdoEnvironments(0)
```

Conectar con la base de datos:

```
Set rdoCn = rdoEv.OpenConnection("", , , Cnct$)
```

Asignar la sentencia SELECT para el conjunto de registros:

```
SQLQueryDB = "SELECT * FROM EMPLOYEE"
```

Abrir el conjunto de registros y ejecutar la consulta:

```
Set rdoRS = rdoCn.OpenResultset(SQLQueryDB)
```

Mientras no se alcance el final del conjunto de registros, abrir la ventana de mensajes y mostrar para cada empleado el apellido (LASTNAME) y el nombre (FIRSTNAME) a partir de la tabla:

```
While Not rdoRS.EOF  
MsgBox rdoRS!LASTNAME & ", " & rdoRS!FIRSTNAME
```

Pasar a la fila siguiente del conjunto de registros:

```
rdoRS.MoveNext  
Wend
```

Cerrar el programa:

```
rdoRS.Close  
rdoCn.Close  
rdoEv.Close
```

Conceptos relacionados:

- “Objetos de datos ActiveX y Objetos de datos remotos” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente*

Información relacionada:

- “Ejemplos de Visual Basic” en la página 105

Automatización del enlace e integración de objetos (OLE) con Visual Basic

Puede implementar funciones definidas por el usuario y procedimientos almacenados de OLE Automation en cualquier lenguaje, pues OLE es independiente del lenguaje. Para hacer esto, se exponen métodos de servidores de OLE Automation, y los métodos se registran en DB2[®] como UDF (funciones definidas por el usuario). Determinadas versiones de los entornos siguientes de desarrollo de aplicaciones dan soporte al desarrollo de servidores de OLE Automation: Microsoft[®] Visual Basic, Microsoft Visual C++, Microsoft Visual J++, Microsoft FoxPro, Borland Delphi, Powersoft PowerBuilder y Micro Focus COBOL. Además, los objetos de beans Java[™] que están envueltos debidamente para OLE, por ejemplo mediante Microsoft Visual J++, se pueden acceder utilizando OLE Automation.

Consulte la documentación del entorno apropiado de desarrollo de aplicaciones para obtener más información sobre el desarrollo de servidores de OLE Automation.

UDF (funciones definidas por el usuario) y procedimientos almacenados de OLE Automation

Microsoft Visual Basic da soporte a la creación de servidores de OLE Automation. En Visual Basic se crea una nueva clase de objeto añadiendo un módulo de clase al proyecto Visual Basic. Los métodos se crean añadiendo subprocedimientos públicos al módulo de clase. Estos procedimientos públicos se pueden registrar en DB2 como UDF (funciones definidas por el usuario) y procedimientos almacenados de OLE Automation. Para obtener más información sobre la creación de servidores OLE, consulte el manual de Microsoft Visual Basic, *Creating OLE Servers, Microsoft Corporation, 1995*, y los programas de ejemplo de OLE proporcionados por Microsoft Visual Basic.

DB2 proporciona ejemplos incorporados de funciones definidas por el usuario y procedimientos almacenados de OLE Automation en Microsoft Visual Basic, en el directorio `sqllib\samples\ole\msvb`. Para obtener información sobre la creación y ejecución de los ejemplos de funciones definidas por el usuario y procedimientos almacenados de OLE Automation, consulte el archivo README situado en `sqllib\samples\ole`.

Conceptos relacionados:

- “Diseño de rutinas de automatización de OLE” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de servidor*
- “Rutinas de automatización de OLE en BASIC y C++” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de servidor*

Información relacionada:

- “Ejemplos de OLE (Object Linking and Embedding)” en la página 96

.NET

Creación de aplicaciones C# .NET

DB2 proporciona un archivo de proceso por lotes, `bldapp.bat`, para compilar y vincular las aplicaciones C# .NET de DB2, ubicadas en el directorio `sqllib\samples\.NET\cs` junto a los programas de ejemplo que pueden crearse con este archivo. El archivo de proceso por lotes toma un parámetro, `%1`, para el nombre del archivo fuente que debe compilarse (sin la extensión `.cs`).

Procedimiento:

Para crear el programa `DbAuth` a partir del archivo fuente `DbAuth.cs`, entre:

```
bldapp DbAuth
```

Para asegurarse de que tenga los parámetros que necesitará cuando ejecute el archivo ejecutable, puede especificar distintas combinaciones de parámetros en función del número que haya entrado:

1. Ningún parámetro. Entre sólo el nombre del programa:
`DbAuth`
2. Un parámetro. Entre el nombre del programa más el alias de la base de datos:
`DbAuth <alias_base_datos>`
3. Dos parámetros. Entre el nombre del programa más el ID de usuario y la contraseña:
`DbAuth <ID_usuario> <contraseña>`
4. Tres parámetros. Entre el nombre del programa más el alias de la base de datos, el ID de usuario y la contraseña:

```
DbAuth <alias_base_datos> <ID_usuario> <contraseña>
```

5. Cuatro parámetros. Entre el nombre del programa más el nombre del servidor, el número de puerto, el ID de usuario y la contraseña:

```
DbAuth <servidor> <número_puerto> <ID_usuario> <contraseña>
```

6. Cinco parámetros. Entre el nombre del programa más el alias de la base de datos, el nombre del servidor, el número de puerto, el ID de usuario y la contraseña:

```
DbAuth <alias_base_datos> <servidor> <número_puerto> <ID_usuario> <contraseña>
```

Para crear y ejecutar el programa de ejemplo de LCTrans debe seguir instrucciones más detalladas que se proporcionan en el archivo fuente, LCTrans.cs.

Tareas relacionadas:

- “Creación de rutinas CLR (Common Language Runtime) .NET” en la página 276

Información relacionada:

- “Ejemplos de C#” en la página 78
- “Opciones de compilación y enlace para aplicaciones C# .NET” en la página 270

Ejemplos relacionados:

- “bldapp.bat -- Builds C# applications on Windows”
- “DbAuth.cs -- How to Grant, display and revoke privileges on database”
- “LCTrans.cs -- Demonstrates loosely coupled transactions (CSNET)”

Archivo de proceso por lotes para aplicaciones C# .NET

```
@echo off
rem BATCH FILE: bldapp.bat
rem Crea aplicaciones C# en Windows
rem Uso: bldapp nombre_prog

rem El compilador por omisión se establece en Microsoft C#
rem Para utilizar un compilador diferente, descomente "set BLDCOMP=csc"
rem y escriba 'set BLDCOMP=x', donde x es el compilador requerido
set BLDCOMP=csc

rem Cuando utilice .NET Framework Versión 1.0, apunte a netf10
rem set VERSION=netf10\
rem Cuando utilice .NET Framework Versión 1.1, apunte a netf11
set VERSION=netf11\

if exist %1.cs goto build

:build
if "%1"=="LCTrans" goto LCTransbuild
if "%1"=="SubCOM" goto SubCOMbuild
if "%1"=="RootCOM" goto RootCOMbuild
%BLDCOMP% %1.cs /r:"%DB2PATH%\bin\%VERSION%IBM.Data.DB2.d11
goto exit

:RootCOMbuild
%BLDCOMP% /out:RootCOM.d11 /target:library %1.cs /r:System.EnterpriseServices.d11
/r:"%DB2PATH%\bin\%VERSION%IBM.Data.DB2.d11 /r:System.Data.d11 /r:System.d11
/r:SubCOM.d11
goto exit

:SubCOMbuild
%BLDCOMP% /out:SubCOM.d11 /target:library %1.cs /r:System.EnterpriseServices.d11
/r:"%DB2PATH%\bin\%VERSION%IBM.Data.DB2.d11 /r:System.Data.d11 /r:System.Xml.d11
/r:System.d11
```

```

goto exit

:LCTransbuild
%BLDCOMP% %1.cs /r:System.EnterpriseServices.dll
  /r:"%DB2PATH%\bin\%VERSION%IBM.Data.DB2.dll /r:System.Data.dll /r:System.dll
  /r:SubCOM.dll /r:RootCOM.dll
goto exit

:exit
@echo on

```

Opciones de compilación y enlace para aplicaciones C# .NET

La tabla siguiente muestra las opciones de compilación y enlace que DB2 recomienda para crear aplicaciones de C# en Windows con el compilador Microsoft C#, tal como muestra el archivo de proceso por lotes bldapp.bat.

Opciones de compilación y enlace para bldapp
Opciones de compilación y enlace para aplicaciones C# autónomas:
%BLDCOMP% Variable del compilador. El valor por omisión es csc, que es el compilador C# de Microsoft.
/r:"%DB2PATH%\bin\%VERSION%IBM.Data.DB2.dll Referencia a la biblioteca de enlace de datos de DB2 para la versión de infraestructura de .NET que está utilizando.
%VERSION% Existen dos versiones de la infraestructura de .NET soportadas para las aplicaciones. DB2 tiene una biblioteca de enlace de datos para cada una de ellas en subdirectorios independientes. Para .NET Framework versión 1.0, %VERSION% apunta al subdirectorio netf10\; para .NET Framework versión 1.1, %VERSION% apunta al subdirectorio netf11\.

Opciones de compilación y enlace para bldapp

Opciones de compilación y enlace para el programa de ejemplo emparejado débilmente, LCTrans:

%BLDCOMP%

Variable del compilador. El valor por omisión es csc, que es el compilador C# de Microsoft.

/out:RootCOM.d11

Salida de la biblioteca de enlace de datos RootCOM, utilizada por la aplicación LCTrans, del archivo fuente RootCOM.cs,

/out:SubCOM.d11

Salida de la biblioteca de enlace de datos SubCOM, utilizada por la aplicación LCTrans, del archivo fuente SubCOM.cs,

/target:library %1.cs

Crear la biblioteca de enlace de datos a partir del archivo fuente de entrada (RootCOM.cs o SubCOM.cs).

/r:System.EnterpriseServices.d11

Referencia a la biblioteca de enlace de datos de Microsoft Windows System Enterprise Services.

/r:"%DB2PATH%\bin\%VERSION%IBM.Data.DB2.d11

Referencia a la biblioteca de enlace de datos de DB2 para la versión de infraestructura de .NET que está utilizando.

%VERSION%

Existen dos versiones de la infraestructura de .NET soportadas para las aplicaciones. DB2 tiene una biblioteca de enlace de datos para cada una de ellas en subdirectorios independientes. Para .NET Framework versión 1.0, %VERSION% apunta al subdirectorio netf10\; para .NET Framework versión 1.1, %VERSION% apunta al subdirectorio netf11\.

/r:System.Data.d11

Referencia a la biblioteca de enlace de datos de Microsoft Windows System Data.

/r:System.d11

Referencia a la biblioteca de enlace de datos de Microsoft Windows System.

/r:System.Xml.d11

Referencia a la biblioteca de enlace de datos de Microsoft Windows System XML (para SubCOM.cs).

/r:SubCOM.d11

Referencia a la biblioteca de enlace de datos de SubCOM (para RootCOM.cs y LCTrans.cs).

/r:RootCOM.d11

Referencia a la biblioteca de enlace de datos de RootCOM (para LCTrans.cs).

Consulte la documentación del compilador para conocer otras opciones de compilador.

Tareas relacionadas:

- “Creación de aplicaciones C# .NET” en la página 268

Ejemplos relacionados:

- “bldapp.bat -- Builds C# applications on Windows”

Creación de aplicaciones Visual Basic .NET

DB2 proporciona un archivo de proceso por lotes, bldapp.bat, para compilar y vincular las aplicaciones Visual Basic .NET de DB2, ubicadas en el directorio sqllib\samples\.NET\vb junto a los programas de ejemplo que pueden crearse con este archivo. El archivo de proceso por lotes toma un parámetro, %1, para el nombre del archivo fuente que debe compilarse (sin la extensión .vb).

Procedimiento:

Para crear el programa DbAuth a partir del archivo fuente DbAuth.vb, entre:

```
bldapp DbAuth
```

Para asegurarse de que tenga los parámetros que necesitará cuando ejecute el archivo ejecutable, puede especificar distintas combinaciones de parámetros en función del número que haya entrado:

1. Ningún parámetro. Entre sólo el nombre del programa:
DbAuth
2. Un parámetro. Entre el nombre del programa más el alias de la base de datos:
DbAuth <alias_base_datos>
3. Dos parámetros. Entre el nombre del programa más el ID de usuario y la contraseña:
DbAuth <ID_usuario> <contraseña>
4. Tres parámetros. Entre el nombre del programa más el alias de la base de datos, el ID de usuario y la contraseña:
DbAuth <alias_base_datos> <ID_usuario> <contraseña>
5. Cuatro parámetros. Entre el nombre del programa más el nombre del servidor, el número de puerto, el ID de usuario y la contraseña:
DbAuth <servidor> <número_puerto> <ID_usuario> <contraseña>
6. Cinco parámetros. Entre el nombre del programa más el alias de la base de datos, el nombre del servidor, el número de puerto, el ID de usuario y la contraseña:
DbAuth <alias_base_datos> <servidor> <número_puerto> <ID_usuario> <contraseña>

Para crear y ejecutar el programa de ejemplo de LCTrans debe seguir instrucciones más detalladas que se proporcionan en el archivo fuente, LCTrans.vb.

Tareas relacionadas:

- “Creación de rutinas CLR (Common Language Runtime) .NET” en la página 276

Información relacionada:

- “Ejemplos de Visual Basic .NET” en la página 107
- “Opciones de compilación y enlace para aplicaciones Visual Basic .NET” en la página 274

Ejemplos relacionados:

- “bldapp.bat -- Builds Visual Basic .Net applications on Windows”
- “DbAuth.vb -- How to Grant, display and revoke privileges on database”
- “LCTrans.vb -- Demonstrates loosely coupled transactions”

Archivo de proceso por lotes para aplicaciones Visual Basic .NET

```
@echo off
rem BATCH FILE: bldapp.bat
rem Crea aplicaciones Visual Basic .Net en Windows
rem Uso: bldapp nombre_prog [ nombre_bd [ IDusuario contraseña ]]

rem El compilador por omisión se establece en Microsoft Visual Basic .NET
rem Para utilizar un compilador diferente, descomente "set BLDCOMP=vbc"
rem y escriba 'set BLDCOMP=x', donde x es el compilador requerido
set BLDCOMP=vbc

rem Cuando utilice .NET Framework Versión 1.0, apunte a netf10
rem set VERSION=netf10\
rem Cuando utilice .NET Framework Versión 1.1, apunte a netf11
set VERSION=netf11\

if exist %1.vb goto build

:build
if "%1"=="LCTrans" goto LCTransbuild
if "%1"=="SubCOM" goto SubCOMbuild
if "%1"=="RootCOM" goto RootCOMbuild
%BLDCOMP% %1.vb /r:"%DB2PATH%\bin\%VERSION%IBM.Data.DB2.d11 /r:System.d11
/r:System.Data.d11 /r:System.Xml.d11
goto exit

:RootCOMbuild
%BLDCOMP% %1.vb /r:System.EnterpriseServices.d11
/r:"%DB2PATH%\bin\%VERSION%IBM.Data.DB2.d11 /target:library /r:System.Data.d11
/r:System.d11 /r:SubCOM.d11 /out:RootCOM.d11
goto exit

:SubCOMbuild
%BLDCOMP% %1.vb /r:System.EnterpriseServices.d11
/r:"%DB2PATH%\bin\%VERSION%IBM.Data.DB2.d11 /target:library /r:System.Data.d11
/r:System.Xml.d11 /r:System.d11 /out:SubCOM.d11
goto exit

:LCTransbuild
%BLDCOMP% %1.vb /r:System.EnterpriseServices.d11
/r:"%DB2PATH%\bin\%VERSION%IBM.Data.DB2.d11 /r:System.Data.d11 /r:System.d11
/r:SubCOM.d11 /r:RootCOM.d11
goto exit

:exit
@echo on
```

Opciones de compilación y enlace para aplicaciones Visual Basic .NET

La tabla siguiente muestra las opciones de compilación y enlace que DB2 recomienda para crear aplicaciones de Visual Basic .NET en Windows con el compilador Microsoft Visual Basic .NET, tal como muestra el archivo de proceso por lotes bldapp.bat.

Opciones de compilación y enlace para bldapp
Opciones de compilación y enlace para aplicaciones VB .NET autónomas:
%BLDCOMP% Variable del compilador. El valor por omisión es vbc, que es el compilador de Microsoft Visual Basic .NET.
/r:"%DB2PATH%\bin\%VERSION%IBM.Data.DB2.d11 Referencia a la biblioteca de enlace de datos de DB2 para la versión de infraestructura de .NET que está utilizando.
%VERSION% Existen dos versiones de la infraestructura de .NET soportadas para las aplicaciones. DB2 tiene una biblioteca de enlace de datos para cada una de ellas en subdirectorios independientes. Para .NET Framework versión 1.0, %VERSION% apunta al subdirectorio netf10\; para .NET Framework versión 1.1, %VERSION% apunta al subdirectorio netf11\.

Opciones de compilación y enlace para bldapp

Opciones de compilación y enlace para el programa de ejemplo emparejado débilmente, LCTrans:

%BLDCOMP%

Variable del compilador. El valor por omisión es vbc, que es el compilador de Microsoft Visual Basic .NET.

/out:RootCOM.d11

Salida de la biblioteca de enlace de datos RootCOM, utilizada por la aplicación LCTrans, del archivo fuente RootCOM.vb,

/out:SubCOM.d11

Salida de la biblioteca de enlace de datos SubCOM, utilizada por la aplicación LCTrans, del archivo fuente SubCOM.vb,

/target:library %1.cs

Crear la biblioteca de enlace de datos a partir del archivo fuente de entrada (RootCOM.vb o SubCOM.vb).

/r:System.EnterpriseServices.d11

Referencia a la biblioteca de enlace de datos de Microsoft Windows System Enterprise Services.

/r:"%DB2PATH%\bin\%VERSION%IBM.Data.DB2.d11

Referencia a la biblioteca de enlace de datos de DB2 para la versión de infraestructura de .NET que está utilizando.

%VERSION%

Existen dos versiones de la infraestructura de .NET soportadas para las aplicaciones. DB2 tiene una biblioteca de enlace de datos para cada una de ellas en subdirectorios independientes. Para .NET Framework versión 1.0, %VERSION% apunta al subdirectorio netf10\; para .NET Framework versión 1.1, %VERSION% apunta al subdirectorio netf11\.

/r:System.Data.d11

Referencia a la biblioteca de enlace de datos de Microsoft Windows System Data.

/r:System.d11

Referencia a la biblioteca de enlace de datos de Microsoft Windows System.

/r:System.Xml.d11

Referencia a la biblioteca de enlace de datos de Microsoft Windows System XML (para SubCOM.vb).

/r:SubCOM.d11

Referencia a la biblioteca de enlace de datos de SubCOM (para RootCOM.vb y LCTrans.vb).

/r:RootCOM.d11

Referencia a la biblioteca de enlace de datos de RootCOM (para LCTrans.vb).

Consulte la documentación del compilador para conocer otras opciones de compilador.

Tareas relacionadas:

- “Creación de aplicaciones Visual Basic .NET” en la página 272

Ejemplos relacionados:

- “bldapp.bat -- Builds Visual Basic .Net applications on Windows”

Creación de rutinas CLR (Common Language Runtime) .NET

DB2 proporciona archivos de proceso por lotes para compilar y enlazar programas .NET DB2. Estos archivos están ubicados en los directorios `sql11ib\samples\NET\cs` y `sql11ib\samples\NET\vb`, junto con programas de ejemplo que se pueden crear con esos archivos.

El archivo de proceso por lotes `bldrtn.bat` contiene los mandatos para crear rutinas (procedimientos almacenados y funciones definidas por el usuario) CLR. El archivo de proceso por lotes crea una DLL de conjunto .NET en el servidor. Utiliza dos parámetros como entrada, que están representados dentro del archivo de proceso por lotes por las variables `%1` y `%2`.

El primer parámetro, `%1`, especifica el nombre del archivo fuente. El archivo de proceso por lotes utiliza el nombre del archivo fuente para el nombre de la DLL de conjunto. El segundo parámetro, `%2`, especifica el nombre de la base de datos a la que desea conectarse. Puesto que la DLL de conjunto se debe crear en la misma instancia donde reside la base de datos, no hay parámetros para el ID de usuario ni la contraseña.

Sólo es obligatorio el primer parámetro, el nombre del archivo fuente. El nombre de la base de datos es opcional. Si no se proporciona un nombre de base de datos, el programa utiliza la base de datos por omisión `sample`.

Prerrequisitos:

El servidor de bases de datos debe ejecutar un sistema operativo Windows instalado con Microsoft .NET Framework versión 1.1 (para las aplicaciones cliente se soportan tanto .NET Framework 1.0 como .NET Framework 1.1). .NET Framework está disponible de forma independiente o formando parte del Kit de desarrollo de software Microsoft .NET Framework 1.1.

Se deben instalar las versiones siguientes de DB2:

Servidor:

DB2 8.2 o posteriores

Cliente:

DB2 7.2 o posteriores

Se debe otorgar autorización para ejecutar la sentencia `CREATE` para la rutina. Para conocer los privilegios requeridos para ejecutar la sentencia `CREATE`, vea la sentencia `CREATE` para el tipo de rutina: `CREATE PROCEDURE` o `CREATE FUNCTION`.

Procedimiento:

Los ejemplos siguientes muestran cómo crear las DLL de conjunto de rutinas mediante procedimientos almacenados y funciones definidas por el usuario.

DLL de conjunto de procedimiento almacenado

Para crear la DLL de conjunto `SpServer` a partir del archivo fuente VB .NET, `SpServer.vb`, o a partir del archivo fuente C#, `SpServer.cs`:

1. Escriba el nombre del archivo de proceso por lotes y el nombre del programa (sin la extensión):

bldrtn SpServer

Si conecta con una base de datos que no sea la base de datos por omisión, `sample`, especifique también el nombre de la base de datos:

bldrtn SpServer *basedatos*

El archivo de proceso por lotes copia la DLL de conjunto, `SpServer.dll`, en el directorio `sql\lib\function`.

2. A continuación, catalogue las rutinas ejecutando el script `spcat` en el servidor:

SpCat

Este script conecta con la base de datos "sample", descataloga mediante `SpDrop.db2` las rutinas que se hubieran catalogado previamente, luego las cataloga llamando a `SpCreate.db2`, y finalmente desconecta de la base de datos. También puede llamar a los scripts `SpDrop.db2` y `SpCreate.db2` por separado.

3. A continuación, a menos que sea la primera vez que se ha creado la DLL de conjunto, detenga y arranque la base de datos para que se reconozca la nueva versión de la DLL de conjunto. Si es necesario, establezca la modalidad de archivo para la DLL de conjunto de forma que la instancia de DB2 pueda acceder a ella.

Una vez creada la DLL de conjunto, `SpServer`, puede crear la aplicación cliente `SpClient`, que la llama.

Puede crear `SpClient` utilizando el archivo de proceso por lotes, `bldapp.bat`.

Para asegurarse de tener los parámetros que necesitará cuando ejecute el ejecutable, puede especificar distintas combinaciones de parámetros, en lugar de aceptar los valores por omisión, en función del número de parámetros que haya entrado:

1. Ningún parámetro. Entre sólo el nombre de programa (para llamar localmente en la instancia del servidor):

SpClient

2. Un parámetro. Entre el nombre del programa más el alias de la base de datos (para llamar a una base de datos distinta de la base de datos `sample` localmente en la instancia del servidor):

SpClient <alias_bd>

3. Tres parámetros. Entre el nombre del programa más el alias de la base de datos, el ID de usuario y la contraseña (para llamar desde un cliente remoto):

SpClient <alias_bd> <IDusuario> <contraseña>

4. Cinco parámetros. Entre el nombre del programa más el alias de la base de datos, el nombre del servidor, el número de puerto, el ID de usuario y la contraseña (para llamar desde un cliente remoto):

SpClient <alias_bd> <servidor> <númeropuerto> <IDusuario> <contraseña>

La aplicación cliente accede a la DLL de conjunto, `SpServer` y ejecuta varias rutinas contenidas en la base de datos del servidor. Los datos resultantes se devuelven a la aplicación cliente.

DLL de conjunto de función definida por el usuario

Para crear la DLL de conjunto de función definida por el usuario `UDFsrv` a partir del archivo fuente VB `.NET`, `UDFsrv.vb`, o a partir del archivo fuente C#, `UDFsrv.cs`:

```
bldrtn UDFsrv
```

Si conecta con una base de datos que no sea la base de datos por omisión, `sample`, especifique también el nombre de la base de datos:

```
bldrtn UDFsrv basedatos
```

El archivo de proceso por lotes copia la DLL de conjunto de función definida por el usuario, `UDFsrv.dll`, en el directorio `sql11ib\function`.

Una vez creada `UDFsrv`, puede crear la aplicación cliente, `udfcli`, que la llama.

Puede crear `UDFcli` utilizando el archivo de proceso por lotes, `bldapp.bat`.

Para asegurarse de tener los parámetros que necesitará cuando ejecute el ejecutable, puede especificar distintas combinaciones de parámetros, en lugar de aceptar los valores por omisión, en función del número de parámetros que haya entrado:

1. Ningún parámetro. Entre sólo el nombre de programa (para llamar localmente en la instancia del servidor):

```
UDFcli
```
2. Un parámetro. Entre el nombre del programa más el alias de la base de datos (para llamar a una base de datos distinta de la base de datos `sample` localmente en la instancia del servidor):

```
UDFcli <alias_bd>
```
3. Tres parámetros. Entre el nombre del programa más el alias de la base de datos, el ID de usuario y la contraseña (para llamar desde un cliente remoto):

```
UDFcli <alias_bd> <IDusuario> <contraseña>
```
4. Cinco parámetros. Entre el nombre del programa más el alias de la base de datos, el nombre del servidor, el número de puerto, el ID de usuario y la contraseña (para llamar desde un cliente remoto):

```
UDFcli <alias_bd> <servidor> <númpuerto> <IDusuario> <contraseña>
```

La aplicación llamante invoca la función `ScalarUDF` de la DLL de conjunto `udfsrv`.

Conceptos relacionados:

- “Rutinas CLR (Common Language Runtime)” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de servidor*

Ejemplos relacionados:

- “`bldrtn.bat` -- Builds C# routines (stored procedures and UDFs)”
- “`SpServer.cs` -- C# external code implementation of procedures created in `spcat.db2`”
- “`SpClient.cs` -- Call different types of stored procedures from `SpServer.java`”
- “`UDFcli.cs` -- Client application that calls the user-defined functions ”
- “`UDFsrv.cs` -- User-defined scalar functions called by `udfcli.cs`”
- “`bldrtn.bat` -- Builds Visual Basic .NET routines (stored procedures and UDFs)”
- “`SpServer.vb` -- VB.NET implementation of procedures created in `SpCat.db2`”
- “`SpClient.vb` -- Call different types of stored procedures from `SpServer.java`”
- “`UDFcli.vb` -- Client application that calls the user-defined functions ”
- “`UDFsrv.vb` -- User-defined scalar functions called by `udfcli.vb` ”

Archivo de proceso por lotes para rutinas C# .NET

```
@echo off
rem BATCH FILE: bldrtn.bat
rem Crea rutinas C# (procedimientos almacenados y UDF)
rem Uso: bldrtn nombre_prog

rem Cuando utilice .NET Framework Versión 1.1, apunte a netf11
set VERSION=netf11\

rem Compile el programa.
csc /out:%1.dll /target:library /debug /lib:"%DB2PATH%\bin\netf11\
/reference:"%DB2PATH%\bin\%VERSION%IBM.Data.DB2.dll %1.cs

if exist "%DB2PATH%\function\%1.dll" goto delete else goto copydll

:delete
del "%DB2PATH%\function\%1.dll"
goto copydll

:copydll
rem Copiar la DLL de ensamblaje de la rutina al directorio 'function'
copy "%1.dll" "%DB2PATH%\function"

@echo on
```

Archivo de proceso por lotes para rutinas Visual Basic .NET

```
@echo off
rem BATCH FILE: bldrtn.bat
rem Crea rutinas Visual Basic .NET (procedimientos almacenados y UDF)
rem Uso: bldrtn nombre_prog

rem Compile el programa.
vbc %1.vb /out:%1.dll /target:library /debug /libpath:"%DB2PATH%\bin\netf11\
/reference:IBM.Data.DB2.dll /reference:System.dll /reference:System.Data.dll

if exist "%DB2PATH%\function\%1.dll" goto delete else goto copydll

:delete
del "%DB2PATH%\function\%1.dll"
goto copydll

:copydll
rem Copiar la DLL de ensamblaje de la rutina al directorio 'function'
copy "%1.dll" "%DB2PATH%\function"

@echo on
```

Opciones de compilación y enlace para rutinas de CLR .NET

A continuación se muestran las opciones de compilación y enlace que DB2 recomienda para crear rutinas de Common Language Runtime (CLR) .NET en Windows con el compilador Microsoft Visual Basic .NET o con el compilador Microsoft C#, tal como muestran los archivos de proceso `samples\.NET\cs\bldrtn.bat` y `samples\.NET\vb\bldrtn.bat`.

Opciones de compilación y enlace para bldrtn

Opciones de compilación y enlace utilizando el compilador Microsoft C#:

csc El compilador de Microsoft C#.

/out:%1.dll /target:library

Produce como salida la biblioteca de enlace de datos en forma de dll de conjunto de procedimientos almacenados.

/debug Utilizar el depurador.

/lib: "%DB2PATH%\bin\netf11

Utilizar la vía de acceso de bibliotecas para .NET Framework versión 1.1.

/reference:IBM.Data.DB2.dll

Utilizar la biblioteca de enlace de datos de DB2 para .NET Framework versión 1.1.

Consulte la documentación del compilador para conocer otras opciones de compilador.

Opciones de compilación y enlace utilizando el compilador Microsoft Visual Basic .NET:

vb El compilador de Microsoft Visual Basic .NET.

/out:%1.dll /target:library

Produce como salida la biblioteca de enlace de datos en forma de dll de conjunto de procedimientos almacenados.

/debug Utilizar el depurador.

/libpath:"%DB2PATH%\bin\netf11

Utilizar la vía de acceso de bibliotecas para .NET Framework versión 1.1.

/reference:IBM.Data.DB2.dll

Utilizar la biblioteca de enlace de datos de DB2 para .NET Framework versión 1.1.

/reference:System.dll

Referencia a la biblioteca de enlace de datos de Microsoft Windows System.

/reference:System.Data.dll

Referencia a la biblioteca de enlace de datos de Microsoft Windows System Data.

Consulte la documentación del compilador para conocer otras opciones de compilador.

Tareas relacionadas:

- “Creación de rutinas CLR (Common Language Runtime) .NET” en la página 276

Ejemplos relacionados:

- “bldrtn.bat -- Builds C# routines (stored procedures and UDFs)”
- “bldrtn.bat -- Builds Visual Basic .NET routines (stored procedures and UDFs)”

Microsoft Visual C++

En esta sección se describe la creación de aplicaciones con ActiveX Data Objects (ADO), Object Linking and Embedding (OLE), así como con SQL incorporado y las API de DB2.

La creación de información para las aplicaciones y rutinas de la CLI de DB2 se encuentra en el manual *CLI Guide and Reference*.

Creación de aplicaciones ADO con Visual C++

ActiveX Data Objects (ADO) le permite escribir una aplicación para acceder y manejar datos de un servidor de bases de datos mediante un proveedor de bases de datos OLE. Las ventajas principales de ADO son una alta velocidad, la facilidad de manejo, una baja actividad de la memoria y una escasa ocupación de espacio de disco.

DB2 proporciona programas de ejemplo ADO para Visual C++ en el directorio `sql1ib\samples\VC`.

Procedimiento:

Puede utilizar cualquiera de estos dos proveedores compatibles con ODBC:

- Proveedor IBM OLE DB para DB2
- Proveedor Microsoft OLE DB para ODBC

Utilización del proveedor IBM OLE DB para DB2

Los clientes DB2 Versión 8.2 en sistemas operativos Windows instalarán opcionalmente `IBMDADB2`, que es el proveedor compatible IBM OLE DB 2.0 para DB2. El proveedor expone interfaces para los usuarios que desean acceder a datos de una base de datos DB2. El proveedor IBM OLE DB para DB2 da soporte a los tipos de aplicación ADO siguientes:

- Microsoft Active Server Pages (ASP)
- Aplicaciones Microsoft Visual Studio C++ y Visual Basic
- Microsoft Visual Interdev

Para conocer detalles sobre estos tipos de aplicaciones, consulte la documentación de ADO.

Utilización del proveedor Microsoft OLE DB para ODBC

Los programas ADO para DB2 que hacen uso del proveedor Microsoft OLE DB y Visual C++ se pueden compilar del mismo modo que los programas C++ normales, una vez hecho el cambio siguiente.

Para que el programa fuente C++ se ejecute como un programa ADO, debe colocar la siguiente sentencia de importación al comienzo del archivo del programa fuente:

```
#import "C:\program files\common files\system\ado\msado<número_versión>.dll" \  
no_namespace \  
rename( "EOF", "adoEOF")
```

donde `<número_versión>` es el número de versión de la biblioteca ADO.

Cuando se compila el programa, el usuario debe verificar que `msado<número_versión>.dll` está en la vía especificada. Como alternativa, puede añadir `C:\program files\common files\system\ado` a la variable de entorno `LIBPATH` y luego colocar esta sentencia de importación más corta en el archivo fuente:

```
#import <msado<número_versión>.dll> \  
no_namespace \  
rename( "EOF", "adoEOF")
```

Este es el método utilizado en el programa de ejemplo de DB2, `BLOBAccess.dsp`.

Mediante esta sentencia de importación, el programa DB2 podrá acceder a la biblioteca ADO. Ahora puede compilar el programa Visual C++ tal como haría con cualquier otro programa. Si también está utilizando otra interfaz de programación, tal como las API de DB2 o la CLI de DB2, consulte el apartado apropiado para obtener más información sobre la creación del programa.

Conceptos relacionados:

- “Objetivo de IBM OLE DB Provider para DB2” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente*
- “Tipos de aplicaciones soportados por IBM OLE DB Provider para DB2” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente*
- “Compilación y enlace de aplicaciones C/C++ e IBM OLE DB Provider” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente*
- “Conexiones con fuentes de datos en aplicaciones C/C++ mediante IBM OLE DB Provider” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente*
- “Habilitación automática de servicios OLE DB por parte de IBM OLE DB Provider” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente*
- “Manipulación de objetos grandes con IBM OLE DB Provider” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente*
- “Restricciones de IBM OLE DB Provider” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente*

Información relacionada:

- “Correlaciones de tipos de datos entre DB2 y OLE DB” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente*
- “Conversión de datos para establecer datos de tipos OLE DB en tipos DB2” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente*
- “Conversión de datos para establecer datos de tipos DB2 en tipos OLE DB” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente*
- “Soporte de IBM OLE DB Provider de interfaces y componentes de OLE DB” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente*
- “Soporte de IBM OLE DB Provider de las propiedades de OLE DB” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente*
- “Soporte de IBM OLE DB Provider de propiedades y métodos ADO” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente*
- “Ejemplos de Visual C++” en la página 108

Automatización del enlace e integración de objetos (OLE) con Visual C++

Puede implementar funciones definidas por el usuario y procedimientos almacenados de OLE Automation en cualquier lenguaje, pues OLE es independiente del lenguaje. Para hacer esto, se exponen métodos de servidores de OLE Automation, y los métodos se registran en DB2® como UDF (funciones definidas por el usuario). Determinadas versiones de los entornos siguientes de desarrollo de aplicaciones dan soporte al desarrollo de servidores de OLE Automation: Microsoft® Visual Basic, Microsoft Visual C++, Microsoft Visual J++, Microsoft FoxPro, Borland Delphi, Powersoft PowerBuilder y Micro Focus COBOL.

Además, los objetos de beans Java™ que están envueltos debidamente para OLE, por ejemplo mediante Microsoft Visual J++, se pueden acceder utilizando OLE Automation.

Consulte la documentación del entorno apropiado de desarrollo de aplicaciones para obtener más información sobre el desarrollo de servidores de OLE Automation.

UDF (funciones definidas por el usuario) y procedimientos almacenados de OLE Automation

Microsoft Visual C++ da soporte a la creación de servidores de OLE Automation. Los servidores se pueden implementar utilizando Microsoft Foundation Classes y el asistente de la aplicación Microsoft Foundation Class, o bien se pueden implementar como aplicaciones Win32. Los servidores pueden ser una DLL o un EXE. Consulte la documentación de Microsoft Visual C++ y los ejemplos de OLE proporcionados por Microsoft Visual C++ para obtener más información.

DB2 proporciona ejemplos independientes de UDF (funciones definidas por el usuario) y procedimientos almacenados de OLE Automation en Microsoft Visual C++, en el directorio `sqllib\samples\ole\msvc`. Para obtener información sobre la creación y ejecución de los ejemplos de funciones definidas por el usuario y procedimientos almacenados de OLE Automation, consulte el archivo README situado en `sqllib\samples\ole`.

Conceptos relacionados:

- “Diseño de rutinas de automatización de OLE” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de servidor*
- “Rutinas de automatización de OLE en BASIC y C++” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de servidor*

Información relacionada:

- “Ejemplos de OLE (Object Linking and Embedding)” en la página 96

Creación de aplicaciones C/C++ en Windows

DB2 proporciona archivos de proceso por lotes para compilar y enlazar programas C/C++ de la API de DB2 y de SQL incorporado. Estos archivos residen en los directorios `sqllib/samples/c` y `sqllib/samples/cpp`, junto con programas de ejemplo que se pueden crear a partir de esos archivos.

El archivo de proceso por lotes `bldapp.bat` contiene los mandatos para crear programas de la API de DB2 y de SQL incorporado. Utiliza un máximo de cuatro parámetros como entrada, que están representados dentro del archivo de proceso por lotes por las variables `%1`, `%2`, `%3` y `%4`.

El primer parámetro, `%1`, especifica el nombre del archivo fuente. Éste es el único parámetro necesario para los programas que no contienen SQL incorporado. Para crear programas de SQL incorporado es necesaria una conexión con la base de datos, por lo que también se proporcionan otros tres parámetros: el segundo parámetro, `%2`, especifica el nombre de la base de datos con la que se desea conectar; el tercer parámetro, `%3`, especifica el ID de usuario correspondiente a la base de datos, y `%4` especifica la contraseña.

Para un programa de SQL incorporado, `bldapp` pasa los parámetros al archivo de precompilación y vinculación, `embprep.bat`. Si no se proporciona un nombre de base de datos, se utiliza la base de datos por omisión `sample`. Los parámetros de ID de usuario y contraseña sólo son necesarios si la instancia donde se crea el programa es diferente de la instancia donde reside la base de datos.

Procedimiento:

Los ejemplos siguientes muestran cómo crear y ejecutar programas de la API de DB2 y aplicaciones de SQL incorporado.

Para crear el programa de ejemplo de la API de DB2 de SQL no incorporado, `cli_info`, a partir del archivo fuente `cli_info.c`, situado en `sqllib\samples\c`, o a partir del archivo fuente `cli_info.cxx`, situado en `sqllib\samples\cpp`, especifique:

```
bldapp cli_info
```

El resultado es un archivo ejecutable, `cli_info.exe`. Puede ejecutar el archivo ejecutable escribiendo el nombre del ejecutable (sin la extensión) en la línea de mandatos:

```
cli_info
```

Creación y ejecución de aplicaciones de SQL incorporado

Existen tres maneras para crear la aplicación de SQL incorporado `tbmod`, a partir del archivo fuente C `tbmod.sqc` de `sqllib\samples\c`, o partir del archivo C++ `tbmod.sqx` de `sqllib\samples\cpp`:

1. Si conecta con la base de datos "sample" en la misma instancia, especifique:

```
bldapp tbmod
```

2. Si conecta con otra base de datos en la misma instancia, especifique también el nombre de la base de datos:

```
bldapp tbmod basedatos
```

3. Si conecta con una base de datos de otra instancia, especifique también el ID de usuario y la contraseña de la instancia de base de datos:

```
bldapp tbmod basedatos IDusuario contraseña
```

El resultado es un archivo ejecutable, `tbmod.exe`.

Existen tres maneras de ejecutar esta aplicación de SQL incorporado:

1. Si desea acceder a la base de datos `sample` en la misma instancia, especifique simplemente el nombre del ejecutable:

```
tbmod
```

2. Si desea acceder a otra base de datos en la misma instancia, especifique el nombre del ejecutable y el nombre de la base de datos:

```
tbmod basedatos
```

3. Si desea acceder a una base de datos de otra instancia, especifique el nombre del ejecutable, el nombre de la base de datos, y el ID de usuario y la contraseña de la instancia de base de datos:

```
tbmod basedatos IDusuario contraseña
```

Creación y ejecución de aplicaciones de varias hebras

Las aplicaciones C/C++ de varias hebras en Windows se tienen que compilar con las opciones `-MT` o `-MD`. La opción `-MT` se enlazará utilizando la biblioteca estática

LIBCMT.LIB, y -MD se enlazará utilizando la biblioteca dinámica MSVCRT.LIB. El binario enlazado con -MD será más pequeño pero dependiente de MSVCRT.DLL, mientras que el binario enlazado con -MT será más grande pero independiente respecto al momento de ejecución.

El archivo de proceso por lotes bldmt.bat utiliza la opción -MT para crear un programa de varias hebras. Todas las otras opciones de compilación y enlace son iguales a las utilizadas por el archivo de proceso por lotes bldapp.bat para crear aplicaciones autónomas normales.

Para crear el programa multihebra de ejemplo, dbthrds, a partir de los archivos fuente samples\c\dbthrds.sqc o samples\cpp\dbthrds.sqx, especifique:

```
bldmt dbthrds
```

El resultado es un archivo ejecutable, dbthrds.exe.

Existen tres maneras de ejecutar esta aplicación multihebra:

1. Si accede a la base de datos sample en la misma instancia, especifique simplemente el nombre del ejecutable (sin la extensión):
dbthrds
2. Si desea acceder a otra base de datos en la misma instancia, especifique el nombre del ejecutable y el nombre de la base de datos:
dbthrds basedatos
3. Si desea acceder a una base de datos de otra instancia, especifique el nombre del ejecutable, el nombre de la base de datos, y el ID de usuario y la contraseña de la instancia de base de datos:

```
dbthrds basedatos IDusuario contraseña
```

Información relacionada:

- “Opciones de compilación y enlace para aplicaciones C/C++ de Windows” en la página 286

Ejemplos relacionados:

- “bldapp.bat -- Builds C applications on Windows”
- “bldmt.bat -- Builds C multi-threaded applications on Windows”
- “cli_info.c -- Set and get information at the client level (C)”
- “dbthrds.sqc -- How to use multiple context APIs on Windows (C)”
- “embprep.bat -- Prep and binds a C/C++ or Micro Focus COBOL embedded SQL program on Windows”
- “tbmod.sqc -- How to modify table data (C)”
- “bldapp.bat -- Builds C++ applications on Windows”
- “bldmt.bat -- Builds C++ multi-threaded applications on Windows”
- “cli_info.C -- Set and get information at the client level (C++)”
- “dbthrds.sqC -- How to use multiple context APIs on Windows (C++)”
- “tbmod.sqC -- How to modify table data (C++)”

Archivo de proceso por lotes para aplicaciones C/C++

```
@echo off
rem BATCH FILE: bldapp.bat
rem Crea aplicaciones C/C++ en Windows
rem Uso: bldapp nombre_prog [ nombre_bd [ IDusuario contraseña ]]
```

```

rem El compilador por omisión se establece en Microsoft Visual C++
rem Para utilizar un compilador diferente, comente "set BLDCOMP=c1"
rem y descomente 'set BLDCOMP=icl' o 'set BLDCOMP=ec1'
rem Compilador Microsoft C/C++
set BLDCOMP=c1

rem Compilador Intel C++ para aplicaciones de 32 bits
rem set BLDCOMP=icl

rem Compilador Intel C++ para aplicaciones Itanium de 64 bits
rem set BLDCOMP=ec1

if exist "%1.sqx" goto embedded
if exist "%1.sqc" goto embedded
goto non_embedded

:embedded
rem Precompilar y vincular el programa.
rem Nota: algunos archivos .sqc/.sqx no contienen SQL sino que enlazan en
rem utilemb.sqc/.sqx, por lo que debe ignorar este aviso si lo recibe:
rem SQL0053W No se ha encontrado ninguna sentencia SQL en el programa.
call embprep %1 %2 %3 %4
rem Compile el programa.
if exist "%1.cxx" goto cpp_emb
%BLDCOMP% -Zi -Od -c -W2 -DWIN32 %1.c utilemb.c
goto link_embedded
:cpp_emb
%BLDCOMP% -Zi -Od -c -W2 -DWIN32 %1.cxx utilemb.cxx
rem Enlazar el programa.
:link_embedded
link -debug -out:%1.exe %1.obj utilemb.obj db2api.lib
goto exit

:non_embedded
rem Compile el programa.
if exist "%1.cxx" goto cpp_non
%BLDCOMP% -Zi -Od -c -W2 -DWIN32 %1.c utilapi.c
goto link_non_embedded
:cpp_non
%BLDCOMP% -Zi -Od -c -W2 -DWIN32 %1.cxx utilapi.cxx
rem Enlazar el programa.
:link_non_embedded
link -debug -out:%1.exe %1.obj utilapi.obj db2api.lib
:exit
@echo on

```

Opciones de compilación y enlace para aplicaciones C/C++ de Windows

La tabla siguiente muestra las opciones de compilación y enlace que DB2 recomienda para crear aplicaciones en C/C++ de SQL incorporado y de la API de DB2 cuando se utiliza Windows y el compilador Microsoft Visual C++, tal como muestra el archivo de proceso por lotes bldapp.bat.

Opciones de compilación y enlace para bldapp	
Opciones de compilación:	
%BLDCOMP%	Variable del compilador. El valor por omisión es <code>cl</code> , que denota el compilador Microsoft Visual C++. Puede también tener el valor <code>icl</code> , que denota el compilador Intel C++ para aplicaciones de 32 bits, o <code>ec1</code> , que denota el compilador Intel C++ para aplicaciones Itanium de 64 bits.
-Zi	Habilitar la información de depuración.
-Od	Inhabilitar las optimizaciones. Es más fácil utilizar un depurador si la optimización está inhabilitada.
-c	Hace que se efectúe solamente la compilación, no la edición de enlaces. El archivo de proceso por lotes tiene pasos separados para la compilación y la edición de enlaces.
-W2	Emitir mensajes de aviso, de error, de error grave y de error no recuperable.
-DWIN32	Opción de compilador necesaria para los sistemas operativos Windows.
Opciones de enlace:	
link	Denota la utilización del enlazador para realizar la edición de enlaces.
-debug	Hace que se incluya información de depuración.
-out:%1.exe	Especifique un nombre de archivo.
%1.obj	Incluir el archivo objeto.
utilemb.obj	Si es un programa de SQL incorporado, incluir el archivo objeto del programa de utilidad de SQL incorporado para la comprobación de errores.
utilapi.obj	Si crea un programa que no es de SQL incorporado, incluir el archivo objeto del programa de utilidad de la API de DB2 para la comprobación de errores.
db2api.lib	Enlazar con la biblioteca DB2.
Consulte la documentación del compilador para conocer otras opciones de compilador.	

Tareas relacionadas:

- “Creación de aplicaciones C/C++ en Windows” en la página 283

Ejemplos relacionados:

- “bldapp.bat -- Builds C applications on Windows”
- “bldapp.bat -- Builds C++ applications on Windows”

Creación de rutinas C/C++ en Windows

DB2 proporciona archivos de proceso por lotes para compilar y enlazar programas de la API de DB2 y de SQL incorporado en C y C++. Estos archivos residen en los directorios `sqllib\samples\c` y `sqllib\samples\cpp`, junto con programas de ejemplo que se pueden crear a partir de esos archivos.

El archivo de proceso por lotes `bldrtn.bat` contiene los mandatos para crear rutinas (procedimientos almacenados y funciones definidas por el usuario) de SQL

incorporado. El archivo de proceso por lotes crea una DLL en el servidor. Utiliza dos parámetros como entrada, que están representados dentro del archivo de proceso por lotes por las variables %1 y %2.

El primer parámetro, %1, especifica el nombre del archivo fuente. El archivo de proceso por lotes utiliza el nombre del archivo fuente para el nombre de la DLL. El segundo parámetro, %2, especifica el nombre de la base de datos a la que desea conectarse. Debido a que la DLL se debe crear en la misma instancia donde reside la base de datos, no hay parámetros para el ID de usuario ni la contraseña.

Sólo es obligatorio el primer parámetro, el nombre del archivo fuente. El nombre de la base de datos es opcional. Si no se proporciona un nombre de base de datos, el programa utiliza la base de datos por omisión `sample`.

Procedimiento:

Los ejemplos siguientes muestran cómo crear las DLL de rutinas mediante:

- procedimientos almacenados
- funciones definidas por el usuario (UDF) de SQL no incorporado
- funciones definidas por el usuario (UDF) de SQL incorporado

DLL de procedimiento almacenado

Para crear la DLL `spserver` a partir del archivo fuente C, `spserver.sqc`, o a partir del archivo fuente C++, `spserver.sqx`:

1. Escriba el nombre del archivo de proceso por lotes y el nombre del programa:

```
bldrtn spserver
```

Si conecta con otra base de datos, especifique también el nombre de la base de datos:

```
bldrtn spserver basedatos
```

El archivo de proceso por lotes utiliza el archivo de definición de módulos, `spserver.def`, que reside en el mismo directorio que los programas de ejemplo, para crear la DLL. El archivo de proceso por lotes copia la DLL, `spserver.dll`, en el directorio `sqllib\function` del servidor.

2. A continuación, catalogue las rutinas ejecutando el script `spcat` en el servidor:

```
spcat
```

Este script conecta con la base de datos "sample", descataloga mediante `spdrop.db2` las rutinas que se hubieran catalogado previamente, luego las cataloga llamando a `spcreate.db2`, y finalmente desconecta de la base de datos. Puede también ejecutar los scripts `spdrop.db2` y `spcreate.db2` por separado.

3. A continuación, detenga y reorganice la base de datos para que se reconozca la nueva DLL. Si es necesario, establezca la modalidad de archivo para la DLL a fin de que la instancia de DB2 pueda acceder a la DLL.

Una vez creada la DLL, `spserver`, puede crear la aplicación cliente `spclient`, la cual accede a la DLL.

Puede crear `spclient` utilizando el archivo de proceso por lotes, `bldapp.bat`.

Para invocar la DLL, ejecute la aplicación cliente de ejemplo, especificando lo siguiente:

`spclient basedatos IDusuario contraseña`

donde

basedatos

Es el nombre de la base de datos a la que desea conectarse. El nombre podría ser `sample`, o su alias, u otro nombre de base de datos.

IDusuario

Es un ID de usuario válido.

contraseña

Es una contraseña válida correspondiente al ID de usuario.

La aplicación cliente accede a la DLL, `spserver`, la cual ejecuta varias rutinas contenidas en la base de datos del servidor. Los datos resultantes se devuelven a la aplicación cliente.

DLL de funciones definidas por el usuario de SQL no incorporado

Para crear la función definida por el usuario `udfsrv` a partir del archivo fuente `udfsrv.c`, especifique:

```
bldrtn udfsrv
```

El archivo de proceso por lotes utiliza el archivo de definición de módulos, `udfsrv.def`, que reside en el mismo directorio que los programas de ejemplo, para crear la DLL de función definida por el usuario. El archivo de proceso por lotes copia la DLL de función definida por el usuario, `udfsrv.dll`, en el directorio `sqllib\function` del servidor.

Una vez creado el programa `udfsrv`, puede crear la aplicación cliente, `udfcli`, la cual llama al programa. Se proporcionan versiones en C y C++ de este programa para la CLI de DB2 y de SQL incorporado.

Puede crear el programa `udfcli` de la CLI de DB2 a partir del archivo fuente `udfcli.c`, situado en `sqllib\samples\cli`, utilizando el archivo de proceso por lotes `bldapp`.

Puede crear el programa en C `udfcli` de SQL incorporado a partir del archivo fuente `udfcli.sqc`, situado en `sqllib\samples\c`, utilizando el archivo de proceso por lotes `bldapp`.

Puede crear el programa en C++ `udfcli` de SQL incorporado a partir del archivo fuente `udfcli.sqx`, situado en `sqllib\samples\cpp`, utilizando el archivo de proceso por lotes `bldapp`.

Para ejecutar la UDF (función definida por el usuario), especifique:

```
udfcli
```

La aplicación solicitante llama a la función `ScalarUDF` de la DLL `udfsrv`.

DLL de UDF de SQL incorporado

Para crear la biblioteca de función definida por el usuario de SQL incorporado `udfmsrv`, a partir del archivo fuente C `udfmsrv.sqc` de `sqllib\samples\c`, o a partir del archivo C++ `udfmsrv.sqx` de `sqllib\samples\cpp`, especifique:

```
bldrtn udfmsrv
```

Si conecta con otra base de datos, especifique también el nombre de la base de datos:

```
bldrtn udfemsrv basedatos
```

El archivo de proceso por lotes utiliza el archivo de definición de módulos, `udfemsrv.def`, que reside en el mismo directorio que los programas de ejemplo, para crear la DLL de función definida por el usuario. El archivo de proceso por lotes copia la DLL de función definida por el usuario, `udfemsrv.dll`, en el directorio `sqllib\function` del servidor.

Una vez creado el programa `udfemsrv`, puede crear la aplicación cliente, `udfemcli`, que llama al programa. Puede crear `udfemcli` a partir del archivo fuente C `udfemcli.sqc` de `sqllib\samples\c`, o a partir del archivo fuente C++ `udfemcli.sqx` de `sqllib\samples\cpp`, utilizando el archivo de proceso por lotes `bldapp`.

Para ejecutar la UDF (función definida por el usuario), especifique:

```
udfemcli
```

La aplicación solicitante llama a las UDF contenidas en la DLL `udfemsrv`.

Información relacionada:

- “Opciones de compilación y enlace para rutinas C/C++ de Windows” en la página 291

Ejemplos relacionados:

- “`bldrtn.bat` -- Builds C routines (stored procedures and UDFs) on Windows”
- “`embprep.bat` -- Prep and binds a C/C++ or Micro Focus COBOL embedded SQL program on Windows”
- “`spclient.sqc` -- Call various stored procedures (C)”
- “`spserver.sqc` -- Definition of various types of stored procedures (C)”
- “`udfcli.sqc` -- Call a variety of types of user-defined functions (C)”
- “`udfemcli.sqc` -- Call a variety of types of embedded SQL user-defined functions. (C)”
- “`udfemsrv.sqc` -- Call a variety of types of embedded SQL user-defined functions. (C)”
- “`udfsrv.c` -- Defines a variety of types of user-defined functions (C)”
- “`bldrtn.bat` -- Builds C++ routines (stored procedures and UDFs) on Windows”
- “`spclient.sqC` -- Call various stored procedures (C++)”
- “`spserver.sqC` -- Definition of various types of stored procedures (C++)”
- “`udfcli.sqC` -- Call a variety of types of user-defined functions (C++)”
- “`udfemcli.sqC` -- Call a variety of types of embedded SQL user-defined functions. (C++)”
- “`udfemsrv.sqC` -- Call a variety of types of embedded SQL user-defined functions. (C++)”
- “`udfsrv.C` -- Defines a variety of types of user-defined functions (C++)”

Archivo de proceso por lotes para rutinas C/C++

```
@echo off
rem BATCH FILE: bldrtn.bat
rem Crea rutinas C/C++ para Linux (procedimientos almacenados y UDF) en Windows
rem Uso: bldrtn nombre_prog [ nombre_bd ]
```

```

rem El compilador por omisión se establece en Microsoft Visual C++
rem Para utilizar un compilador diferente, comente "set BLDCOMP=c1"
rem y descomente 'set BLDCOMP=icl' o 'set BLDCOMP=ec1'
rem Compilador Microsoft C/C++
set BLDCOMP=c1

rem Compilador Intel C++ para aplicaciones de 32 bits
rem set BLDCOMP=icl

rem Compilador Intel C++ para aplicaciones Itanium de 64 bits
rem set BLDCOMP=ec1

if exist "%1.sqc" goto embedded
if exist "%1.sqx" goto embedded
goto compile

:embedded
rem Precompilar y vincular el programa.
call embprep %1 %2

:compile
rem Compile el programa.
if exist "%1.cxx" goto cpp
%BLDCOMP% -Zi -Od -c -W2 -DWIN32 -MD %1.c
goto link_step
:cpp
%BLDCOMP% -Zi -Od -c -W2 -DWIN32 -MD %1.cxx

:link_step
rem Enlazar el programa.
link -debug -out:%1.dll -dll %1.obj db2api.lib -def:%1.def

rem Copiar la DLL de la rutina al directorio 'function'
copy %1.dll "%DB2PATH%\function"
@echo on

```

Opciones de compilación y enlace para rutinas C/C++ de Windows

La tabla siguiente muestra las opciones de compilación y enlace que DB2 recomienda para crear rutinas en C/C++ (procedimientos almacenados y funciones definidas por el usuario) cuando se utiliza Windows y el compilador Microsoft Visual C++, tal como muestra el archivo de proceso por lotes bldrtn.bat.

Opciones de compilación y enlace para bldrtn	
Opciones de compilación:	
%BLDCOMP%	Variable del compilador. El valor por omisión es <code>cl</code> , que denota el compilador Microsoft Visual C++. Puede también tener el valor <code>icl</code> , que denota el compilador Intel C++ para aplicaciones de 32 bits, o <code>ec1</code> , que denota el compilador Intel C++ para aplicaciones Itanium de 64 bits.
-Zi	Habilitar la información de depuración.
-Od	Inhabilitar la optimización.
-c	Hace que se efectúe solamente la compilación, no la edición de enlaces. La compilación y la edición de enlaces son pasos separados.
-W2	Emitir mensajes de aviso, de error, de error grave y de error no recuperable.
-DWIN32	Opción de compilador necesaria para los sistemas operativos Windows.
-MD	Crear un DLL de varias hebras, utilizando MSVCRT.LIB
Opciones de enlace:	
link	Denota la utilización del enlazador para realizar la edición de enlaces.
-debug	Hace que se incluya información de depuración.
-out:%1.dll	Crear un archivo .DLL.
%1.obj	Incluir el archivo objeto.
db2api.lib	Enlazar con la biblioteca DB2.
-def:%1.def	Archivo de definición de módulos.
Consulte la documentación del compilador para conocer otras opciones de compilador.	

Tareas relacionadas:

- “Creación de rutinas C/C++ en Windows” en la página 287

Ejemplos relacionados:

- “bldrtn.bat -- Builds C routines (stored procedures and UDFs) on Windows”
- “bldrtn.bat -- Builds C++ routines (stored procedures and UDFs) on Windows”

Creación de aplicaciones C/C++ de conexión múltiple en Windows

DB2 proporciona archivos de proceso por lotes para compilar y enlazar programas C y C++ de SQL incorporado y de la API de DB2. Estos archivos residen en los directorios `sqllib\samples\c` y `sqllib\samples\cpp`, junto con programas de ejemplo que se pueden crear a partir de esos archivos.

El archivo de proceso por lotes, `bldmc.bat`, contiene los mandatos para crear un programa de conexión múltiple de DB2, que requiere dos bases de datos. Las opciones de compilación y enlace son las mismas que las utilizadas en el archivo `bldapp.bat`.

El primer parámetro, %1, especifica el nombre del archivo fuente. El segundo parámetro, %2, especifica el nombre de la primera base de datos a la que desea conectarse. El tercer parámetro, %3, especifica la segunda base de datos a la que desea conectarse. Todos estos parámetros son obligatorios.

Nota: El makefile codifica los valores por omisión "sample" y "sample2" como nombres de la base de datos (%2 y %3, respectivamente) por lo que, si está utilizando el makefile y acepta estos valores, sólo tendrá que especificar el nombre del programa (parámetro %1). Si utiliza el script b1dmc.bat debe especificar los tres parámetros.

No se requieren parámetros opcionales para una conexión local, pero sí para conectar con un servidor desde un cliente remoto. Estos parámetros son: %4 y %5 para especificar, respectivamente, el ID de usuario y la contraseña para la primera base de datos; y %6 y %7 para especificar, respectivamente, el ID de usuario y la contraseña para la segunda base de datos.

Procedimiento:

Para el programa de ejemplo de conexión múltiple, dbmcon.exe, se requieren dos bases de datos. Si aún no se ha creado la base de datos sample, la puede crear entrando db2samp1 en la línea de mandatos de una ventana de mandatos de DB2. La segunda base de datos, aquí llamada sample2, se puede crear mediante uno de los mandatos siguientes:

Si crea la base de datos localmente:

```
db2 create db sample2
```

Si crea la base de datos remotamente:

```
db2 attach to <nombre_nodo>
db2 create db sample2
db2 detach
db2 catalog db sample2 as sample2 at node <nombre_nodo>
```

donde <nombre_nodo> es el nodo en que reside la base de datos.

Una conexión múltiple también requiere que se esté ejecutando el receptor TCP/IP. Para asegurarse que es así, haga lo siguiente:

1. Establezca la variable de entorno DB2COMM en TCP/IP de la forma siguiente:

```
db2set DB2COMM=TCPIP
```

2. Actualice el archivo de configuración del gestor de bases de datos con el nombre del servicio de TCP/IP que se haya especificado en el archivo de servicios:

```
db2 update dbm cfg using SVCENAME <nombre del servicio de TCP/IP>
```

Cada instancia tiene un nombre de servicio TCP/IP listado en el archivo de servicios. Si no puede localizarlo o no tiene el permiso de archivo para cambiar el archivo de servicios, pida ayuda al administrador.

3. Detenga y reinicie el gestor de bases de datos para que estos cambios entren en vigor:

```
db2stop
db2start
```

El programa dbmcon.exe se crea a partir de cinco archivos de los directorios samples\c o samples\cpp:

dbmcon.sqc o dbmcon.sqx

Archivo fuente principal para conectar con ambas bases de datos.

dbmcon1.sqc o dbmcon1.sqx

Archivo fuente para crear un paquete vinculado a la primera base de datos.

dbmcon1.h

Archivo de cabecera para dbmcon1.sqc o dbmcon1.sqx, incluido en el archivo fuente principal, dbmcon.sqc o dbmcon.sqx, para acceder a las sentencias de SQL para crear y eliminar una tabla que se debe vincular a la primera base de datos.

dbmcon2.sqc o dbmcon2.sqx

Archivo fuente para crear un paquete vinculado a la segunda base de datos.

dbmcon2.h

Archivo de cabecera para dbmcon2.sqc o dbmcon2.sqx, incluido en el archivo fuente principal, dbmcon.sqc o dbmcon.sqx, para acceder a las sentencias de SQL para crear y eliminar una tabla que se debe vincular a la segunda base de datos.

Para crear el programa de ejemplo de conexión múltiple, dbmcon.exe, se entre lo siguiente:

```
bldmc dbmcon sample sample2
```

El resultado es un archivo ejecutable, dbmcon.exe.

Para ejecutar el archivo ejecutable, entre el nombre del ejecutable sin la extensión:

```
dbmcon
```

El programa muestra una confirmación en una fase para dos bases de datos.

Conceptos relacionados:

- “Archivos de creación” en la página 110

Información relacionada:

- “svcname - TCP/IP service name configuration parameter” en el manual *Administration Guide: Performance*
- “Opciones de compilación y enlace para aplicaciones C/C++ de Windows” en la página 286

Ejemplos relacionados:

- “bldmc.bat -- Builds C multi-connection application on Windows”
- “dbmcon.sqc -- How to use multiple databases (C)”
- “dbmcon1.h -- Function declarations for the source file, dbmcon1.sqc (C)”
- “dbmcon1.sqc -- Functions used in the multiple databases program dbmcon.sqc (C)”
- “dbmcon2.h -- Function declarations for the source file, dbmcon2.sqc (C)”
- “dbmcon2.sqc -- Functions used in the multiple databases program dbmcon.sqc (C)”
- “bldmc.bat -- Builds C++ multi-connection application on Windows”
- “dbmcon.sqC -- How to use multiple databases (C++)”
- “dbmcon1.h -- Class declaration for the source file, dbmcon1.sqC (C++)”

- “dbmcon1.sqC -- Functions used in the multiple databases program dbmcon.sqC (C++)”
- “dbmcon2.h -- Class declaration for the source file, dbmcon2.sqC (C++)”
- “dbmcon2.sqC -- Functions used in the multiple databases program dbmcon.sqC (C++)”

IBM VisualAge COBOL

Configuración del compilador IBM COBOL en Windows

Si desarrolla aplicaciones que contienen SQL incorporado y llamadas a la API de DB2, y está utilizando el compilador IBM VisualAge COBOL, existen varias consideraciones para tener en cuenta.

Procedimiento:

- Cuando precompile la aplicación utilizando el precompilador de DB2 y use el mandato de procesador de línea de mandatos db2 prep, utilice la opción target ibmcob.
- No utilice caracteres de tabulación en los archivos fuente.
- Utilice las palabras clave PROCESS y CBL en los archivos fuente para definir opciones de compilación. Coloque las palabras clave en las columnas 8 a la 72 solamente.
- Si la aplicación sólo contiene SQL incorporado y ninguna llamada a la API de DB2, no es necesario que utilice la opción de compilación pgmname(mixed). Si especifica llamadas a la API de DB2, debe utilizar la opción de compilación pgmname(mixed).
- Si utiliza la opción “System/390 host data type support” del compilador IBM VisualAge COBOL, los archivos de inclusión de DB2 correspondientes a las aplicaciones están contenidos en este directorio:
%DB2PATH%\include\cobl_i

Si crea programas DB2 de ejemplo utilizando los archivos de proceso por lotes proporcionados, la vía de acceso de los archivos de inclusión especificada en los archivos de proceso por lotes se debe cambiar para que apunte al directorio cobl_i y no al directorio cobl_a.

Si NO utiliza la opción “System/390 host data type support” del compilador IBM VisualAge COBOL, o está utilizando una versión anterior de este compilador, los archivos de inclusión de DB2 correspondientes a las aplicaciones están contenidos en este directorio:

%DB2PATH%\include\cobl_a

El directorio cobl_a es el directorio por omisión.

- Especifique nombres de archivo de COPY de modo que incluyan la extensión .cbl, de la manera siguiente:
COPY "sql.cbl".

Tareas relacionadas:

- “Creación de aplicaciones IBM COBOL en Windows” en la página 296
- “Creación de rutinas IBM COBOL en Windows” en la página 300

Información relacionada:

- “Opciones de compilación y enlace para aplicaciones IBM COBOL de Windows” en la página 299
- “Opciones de compilación y enlace para rutinas IBM COBOL de Windows” en la página 302

Creación de aplicaciones IBM COBOL en Windows

DB2 proporciona archivos de proceso por lotes para compilar y enlazar programas de la API de DB2 y de SQL incorporado. Estos archivos residen en el directorio `sqllib\samples\cobol`, junto con programas de ejemplo que se pueden crear a partir de esos archivos.

DB2 da soporte a dos precompiladores para crear aplicaciones IBM COBOL en Windows, el precompilador de DB2 y el precompilador de IBM COBOL. El valor por omisión es el precompilador de DB2. El precompilador de IBM COBOL se puede seleccionar descomentando la línea apropiada del archivo de proceso por lotes que se esté utilizando. La precompilación con IBM COBOL la realiza el propio compilador, utilizando opciones de precompilación específicas.

El archivo de proceso por lotes `bldapp.bat` contiene los mandatos para crear programas de aplicación DB2. Utiliza un máximo de cuatro parámetros como entrada, que están representados dentro del archivo de proceso por lotes por las variables `%1`, `%2`, `%3` y `%4`.

El primer parámetro, `%1`, especifica el nombre del archivo fuente. Éste es el único parámetro necesario para los programas que no contienen SQL incorporado. Para crear programas de SQL incorporado es necesaria una conexión con la base de datos, por lo que también se proporcionan otros tres parámetros: el segundo parámetro, `%2`, especifica el nombre de la base de datos con la que se desea conectar; el tercer parámetro, `%3`, especifica el ID de usuario correspondiente a la base de datos, y `%4` especifica la contraseña.

Para un programa de SQL incorporado que utiliza el precompilador de DB2 por omisión, `bldrtn.bat` pasa los parámetros al archivo de precompilación y enlace, `embprep.bat`.

Para un programa de SQL incorporado que utiliza el precompilador de IBM COBOL, `bldrtn.bat` copia el archivo fuente `.sqb` en un archivo fuente `.cb1`. El compilador realiza la precompilación sobre el archivo fuente `.cb1` con opciones de precompilación específicas.

Para cualquiera de los precompiladores, si no se proporciona un nombre de base de datos, se utiliza la base de datos por omisión `sample`. Los parámetros de ID de usuario y contraseña sólo son necesarios si la instancia donde se crea el programa es diferente de la instancia donde reside la base de datos.

Procedimiento:

Los ejemplos siguientes muestran cómo crear y ejecutar programas de la API de DB2 y aplicaciones de SQL incorporado.

Para crear el programa de ejemplo de SQL no incorporado, `client`, a partir del archivo fuente `client.cb1`, especifique:

```
bldapp client
```


El resultado es un archivo ejecutable, `client.exe`. Para ejecutar el archivo ejecutable sobre la base de datos `sample`, especifique el nombre del ejecutable (sin la extensión):

```
client
```

Creación y ejecución de aplicaciones de SQL incorporado

A partir del archivo fuente `updat.sqb`, puede crear la aplicación de SQL incorporado, `updat`, de tres maneras:

1. Si conecta con la base de datos "sample" en la misma instancia, especifique:

```
bldapp updat
```

2. Si conecta con otra base de datos en la misma instancia, especifique también el nombre de la base de datos:

```
bldapp updat basedatos
```

3. Si conecta con una base de datos de otra instancia, especifique también el ID de usuario y la contraseña de la instancia de base de datos:

```
bldapp updat basedatos IDusuario contraseña
```

El resultado es un archivo ejecutable, `updat`.

Existen tres maneras de ejecutar esta aplicación de SQL incorporado:

1. Si desea acceder a la base de datos `sample` en la misma instancia, especifique simplemente el nombre del ejecutable:

```
updat
```

2. Si desea acceder a otra base de datos en la misma instancia, especifique el nombre del ejecutable y el nombre de la base de datos:

```
updat basedatos
```

3. Si desea acceder a una base de datos de otra instancia, especifique el nombre del ejecutable, el nombre de la base de datos, y el ID de usuario y la contraseña de la instancia de base de datos:

```
bldapp updat basedatos IDusuario contraseña
```

Conceptos relacionados:

- "Archivos de creación" en la página 110

Información relacionada:

- "Opciones de compilación y enlace para aplicaciones IBM COBOL de Windows" en la página 299
- "Ejemplos de COBOL" en la página 82

Ejemplos relacionados:

- "bldapp.bat -- Builds Windows VisualAge COBOL applications"
- "client.cbl -- How to set and query a client (IBM COBOL)"
- "embprep.bat -- To prep and bind a COBOL embedded SQL program on Windows"
- "updat.sqb -- How to update, delete and insert table data (IBM COBOL)"

Archivo de proceso por lotes para aplicaciones COBOL de IBM

```
@echo off
rem BATCH FILE: bldapp.bat
rem Crea aplicaciones VisualAge COBOL para Windows
rem Uso: bldapp nombre_prog [ nombre_bd [ IDusuario contraseña ]]

set IBMCOB_PRECOMP=
set EXTRA_COMPFLAG=

rem Para utilizar el precompilador COBOL de IBM, descomente la línea siguiente.
rem establezca IBMCOB_PRECOMP=true

rem Si se utiliza el precompilador COBOL de IBM
if "%IBMCOB_PRECOMP%" == "true" goto IBMCOB_precompile_step

rem Utilizando el precompilador DB2 por omisión,
rem Si es un programa de SQL incorporado, precompílelo y vincúlelo.
if not exist "%1.sqb" goto compile_step
call embprep %1 %2 %3 %4
goto compile_step

:IBMCOB_precompile_step
rem Utilizando el precompilador COBOL de IBM,
rem Copiar el archivo <nombre_prog>.sqb en <nombre_prog>.cbl.
if exist "%1.sqb" cp -f %1.sqb %1.cbl
rem Asignar parámetros de entrada a la variable EXTRA_COMPFLAG
if "%1" == "" goto error
if "%2" == "" goto case1
if "%3" == "" goto case2
if "%4" == "" goto error
goto case3

:case1
set EXTRA_COMPFLAG=-q"SQL('database sample CALL_RESOLUTION DEFERRED')"
goto compile_step
:case2
set EXTRA_COMPFLAG=-q"SQL('database %2 CALL_RESOLUTION DEFERRED')"
goto compile_step
:case3
set EXTRA_COMPFLAG=-q"SQL('database %2 user %3 using %4
CALL_RESOLUTION DEFERRED')"
goto compile_step

:compile_step
rem Compilar el programa de utilidad de comprobación de errores.
cob2 -qpgmname(mixed) -c -qlib -I"%DB2PATH%\include\cobol_a" checkerr.cbl

rem Compile el programa.
cob2 -qpgmname(mixed) -c -qlib -I"%DB2PATH%\include\cobol_a" %1.cbl
%EXTRA_COMPFLAG%

rem Enlazar el programa.
cob2 %1.obj checkerr.obj db2api.lib
goto exit

:error
echo Uso: bldapp nombre_prog [ nombre_bd [ IDusuario contraseña ]]

:exit
@echo on
```

Opciones de compilación y enlace para aplicaciones IBM COBOL de Windows

La tabla siguiente muestra las opciones de compilación y enlace que DB2 recomienda para crear aplicaciones en COBOL de SQL incorporado y de la API de DB2 cuando se utiliza Windows y el compilador IBM VisualAge COBOL, tal como muestra el archivo de proceso por lotes bldapp.bat.

Opciones de compilación y enlace para bldapp	
Opciones de compilación:	
cob2	Denota el compilador IBM VisualAge COBOL.
-qpgmname(mixed)	Indica al compilador que permita llamadas a los puntos de entrada de biblioteca utilizando nombres con mayúsculas y minúsculas mezcladas.
-c	Hace que se efectúe solamente la compilación, no la edición de enlaces. La compilación y la edición de enlaces son pasos separados.
-q1ib	Indica al compilador que procese las sentencias COPY.
-Ivía	Especifica la ubicación de los archivos de inclusión de DB2. Por ejemplo: -I"%DB2PATH%\include\cobol_a".
%EXTRA_COMPFLAG%	Si "set IBMCOB_PRECOMP=true" está descomentado, las opciones de precompilación de IBM COBOL se utilizan con una de las formulaciones siguientes, en función de los parámetros de entrada:
-q"SQL('database sample CALL_RESOLUTION DEFERRED')"	precompilar utilizando la base de datos de ejemplo por omisión, y diferir la resolución de la llamada.
-q"SQL('database %2 CALL_RESOLUTION DEFERRED')"	precompilar utilizando una base de datos especificada por el usuario, y diferir la resolución de la llamada.
-q"SQL('database %2 user %3 using %4 CALL_RESOLUTION DEFERRED')"	precompilar utilizando una base de datos, un ID de usuario y una contraseña especificados por el usuario, y diferir la resolución de la llamada. Éste es el formato para el acceso de clientes remotos.
Opciones de enlace:	
cob2	Utilizar el compilador como interfaz del editor de enlaces.
%1.obj	Incluir el archivo objeto del programa.
checkerr.obj	Incluir el archivo objeto del programa de utilidad de comprobación de errores.
db2api.lib	Enlazar con la biblioteca DB2.
Consulte la documentación del compilador para conocer otras opciones de compilador.	

Tareas relacionadas:

- "Creación de aplicaciones IBM COBOL en Windows" en la página 296

Ejemplos relacionados:

- "bldapp.bat -- Builds Windows VisualAge COBOL applications"

Creación de rutinas IBM COBOL en Windows

DB2 proporciona archivos de proceso por lotes para compilar y enlazar programas de la API de DB2 y de SQL incorporado en IBM COBOL. Estos archivos residen en el directorio `sqllib\samples\cobol`, junto con programas de ejemplo que se pueden crear a partir de esos archivos.

DB2 da soporte a dos precompiladores para crear aplicaciones IBM COBOL en Windows, el precompilador de DB2 y el precompilador de IBM COBOL. El valor por omisión es el precompilador de DB2. El precompilador de IBM COBOL se puede seleccionar descomentando la línea apropiada del archivo de proceso por lotes que se esté utilizando. La precompilación con IBM COBOL la realiza el propio compilador, utilizando opciones de precompilación específicas.

El archivo de proceso por lotes `bldrtn.bat` contiene los mandatos para crear rutinas (procedimientos almacenados) de SQL incorporado. El archivo de proceso por lotes compila las rutinas y crea una DLL en el servidor. Utiliza dos parámetros como entrada, que están representados dentro del archivo de proceso por lotes por las variables `%1` y `%2`.

El primer parámetro, `%1`, especifica el nombre del archivo fuente. El archivo de proceso por lotes utiliza el nombre del archivo fuente, `%1`, como nombre de la DLL. El segundo parámetro, `%2`, especifica el nombre de la base de datos a la que desea conectarse. Debido a que el procedimiento almacenado se debe crear en la misma instancia donde reside la base de datos, no hay parámetros para el ID de usuario ni la contraseña.

Sólo es obligatorio el primer parámetro, el nombre del archivo fuente. El nombre de la base de datos es opcional. Si no se proporciona un nombre de base de datos, el programa utiliza la base de datos por omisión `sample`.

Si se utiliza el precompilador de DB2 por omisión, `bldrtn.bat` pasa los parámetros al archivo de precompilación y enlace, `embprep.bat`.

Si se utiliza el precompilador de IBM COBOL, `bldrtn.bat` copia el archivo fuente `.sqb` en un archivo fuente `.cbl`. El compilador realiza la precompilación sobre el archivo fuente `.cbl` con opciones de precompilación específicas.

Procedimiento:

Si conecta con la base de datos "sample", especifique lo siguiente para crear el programa de ejemplo `outsrv` a partir del archivo fuente `outsrv.sqb`:

```
bldrtn outsrv
```

Si conecta con otra base de datos, especifique también el nombre de la base de datos:

```
bldrtn outsrv basedatos
```

El archivo de proceso por lotes copia la DLL en la vía de acceso `sqllib\function` del servidor.

Una vez creada la DLL `outsrv`, puede crear la aplicación cliente `outcli`, la cual llama a la rutina contenida en la DLL y que tiene el mismo que la DLL. Puede crear `outcli` utilizando el archivo de proceso por lotes `bldapp.bat`.

Para invocar la rutina `outsrv`, ejecute la aplicación cliente de ejemplo, especificando lo siguiente:

```
outcli basedatos IDusuario contraseña
```

donde

basedatos

Es el nombre de la base de datos a la que desea conectarse. El nombre podría ser `sample`, o su alias remoto, u otro nombre.

IDusuario

Es un ID de usuario válido.

contraseña

Es una contraseña válida correspondiente al ID de usuario.

La aplicación cliente accede a la DLL `outsrv`, la cual ejecuta la rutina del mismo nombre contenida en la base de datos del servidor, y devuelve los datos resultantes a la aplicación cliente.

Conceptos relacionados:

- “Archivos de creación” en la página 110

Información relacionada:

- “Opciones de compilación y enlace para rutinas IBM COBOL de Windows” en la página 302
- “Ejemplos de COBOL” en la página 82

Ejemplos relacionados:

- “`bldrtn.bat` -- Builds Windows VisualAge COBOL routines (stored procedures)”
- “`embprep.bat` -- To prep and bind a COBOL embedded SQL program on Windows”
- “`outcli.sqb` -- Call stored procedures using the SQLDA structure (IBM COBOL)”
- “`outsrv.sqb` -- Demonstrates stored procedures using the SQLDA structure (IBM COBOL)”

Archivo de proceso por lotes para rutinas COBOL de IBM

```
| @echo off
| rem BATCH FILE: bldrtn.bat
| rem Crear rutinas COBOL de VisualAge para Windows (procedimientos almacenados)
| rem Uso: bldrtn nombre_prog [ nombre_bd ]
|
| set IBMCOB_PRECOMP=
| set EXTRA_COMPFLAG=
|
| rem Para utilizar el precompilador COBOL de IBM, descomente la línea siguiente.
| rem establezca IBMCOB_PRECOMP=true
|
| rem Si se utiliza el precompilador COBOL de IBM
| if "%IBMCOB_PRECOMP%" == "true" goto IBMCOB_precompile_step
|
| rem Utilizando el precompilador DB2 por omisión,
| rem Precompilar y vincular el programa.
| call embprep %1 %2
| goto compile_step
|
| :IBMCOB_precompile_step
| rem Utilizando el precompilador COBOL de IBM,
| rem Copiar el archivo <nombre_prog>.sqb en <nombre_prog>.cbl.
```

```

if exist "%1.sqb" cp -f %1.sqb %1.cbl
rem Asignar parámetros de entrada a la variable EXTRA_COMPFLAG
if "%1" == "" goto error
if "%2" == "" goto case1

set EXTRA_COMPFLAG=-q"SQL('database %2 CALL_RESOLUTION DEFERRED')"
goto compile_step

:case1
set EXTRA_COMPFLAG=-q"SQL('database sample CALL_RESOLUTION DEFERRED')"
goto compile_step

:compile_step
rem Compile el procedimiento almacenado.
cob2 -qpgmname(mixed) -c -qlib -I"%DB2PATH%\include\cobol_a" %1.cbl %EXTRA_COMPFLAG%

rem Enlace el procedimiento almacenado y cree una biblioteca compartida.
ilib /no1 /gi:%1 %1.obj
ilink /free /no1 /dll db2api.lib %1.exp %1.obj iwzrwin3.obj

rem Copiar el procedimiento almacenado al directorio %DB2PATH%\function.
copy %1.dll "%DB2PATH%\function"
goto exit

:error
echo Uso: bldrtn nombre_prog [ nombre_bd ]

:exit
@echo on

```

Opciones de compilación y enlace para rutinas IBM COBOL de Windows

La tabla siguiente muestra las opciones de compilación y enlace que DB2 recomienda para crear rutinas en COBOL (procedimientos almacenados y funciones definidas por el usuario) cuando se utiliza Windows y el compilador IBM VisualAge COBOL, tal como muestra el archivo de proceso por lotes bldrtn.bat.

Opciones de compilación y enlace para bldrtn

Opciones de compilación:

cob2 Denota el compilador IBM VisualAge COBOL.

-qpgmname(mixed)

Indica al compilador que permita llamadas a los puntos de entrada de biblioteca utilizando nombres con mayúsculas y minúsculas mezcladas.

-c Hace que se efectúe solamente la compilación, no la edición de enlaces. El archivo de proceso por lotes tiene pasos separados para la compilación y la edición de enlaces.

-qlib Indica al compilador que procese las sentencias COPY.

-Ivía Especifica la ubicación de los archivos de inclusión de DB2. Por ejemplo:

`-I"%DB2PATH%\include\cobol_a".`

%EXTRA_COMPFLAG%

Si "set IBMCOB_PRECOMP=true" está descomentado, las opciones de precompilación de IBM COBOL se utilizan con una de las formulaciones siguientes, en función de los parámetros de entrada:

-q"SQL('database sample CALL_RESOLUTION DEFERRED')"

precompilar utilizando la base de datos de ejemplo por omisión, y diferir la resolución de la llamada.

-q"SQL('database %2 CALL_RESOLUTION DEFERRED')"

precompilar utilizando una base de datos especificada por el usuario, y diferir la resolución de la llamada.

Opciones de enlace:

ilink Utilizar el enlazador IBM VisualAge COBOL.

/free Formato libre.

/no1 Sin logotipo.

/d11 Crear la DLL con el nombre del programa fuente.

db2api.lib

Enlazar con la biblioteca DB2.

%1.exp Incluir el archivo de exportación.

%1.obj Incluir el archivo objeto del programa.

iwzrwin3.obj

Incluir el archivo objeto proporcionado por IBM VisualAge COBOL.

Consulte la documentación del compilador para conocer otras opciones de compilador.

Tareas relacionadas:

- "Creación de rutinas IBM COBOL en Windows" en la página 300

Ejemplos relacionados:

- "bldrtn.bat -- Builds Windows VisualAge COBOL routines (stored procedures)"

Micro Focus COBOL

Configuración del compilador Micro Focus COBOL en Windows

Si desarrolla aplicaciones que contienen SQL incorporado y llamadas a la API de DB2, y está utilizando el compilador Micro Focus, existen varias consideraciones para tener en cuenta.

Procedimiento:

- Cuando precompile la aplicación utilizando el mandato db2 prep del procesador de línea de mandatos, utilice la opción target mfcob.
- Asegúrese de que la variable de entorno LIB apunta a %DB2PATH%\lib, utilizando este mandato:

```
set LIB="%DB2PATH%\lib;%LIB%"
```
- Los archivos de COPY para DB2 correspondientes a Micro Focus COBOL residen en %DB2PATH%\include\cobol_mf. Defina la variable de entorno COBCPY para que incluya ese directorio, de esta manera:

```
set COBCPY="%DB2PATH%\include\cobol_mf;%COBCPY%"
```

Las llamadas a las API de DB2 se deben realizar utilizando el convenio de llamada 74. El precompilador DB2 COBOL inserta automáticamente una cláusula CALL-CONVENTION en un párrafo SPECIAL-NAMES. Si el párrafo SPECIAL-NAMES no existe, el precompilador DB2 COBOL lo crea, de esta manera:

```
Identification Division
Program-ID. "static".
special-names.
    call-convention 74 is DB2API.
```

Además, el precompilador inserta automáticamente el símbolo DB2API, que sirve para identificar el convenio de llamada, a continuación de la palabra clave "call" cada vez que se llama a una API de DB2. Esto ocurre, por ejemplo, cada vez que el precompilador emite una llamada a una API de DB2 a partir de una sentencia de SQL incorporado.

Si las llamadas a las API de DB2 se realizan en una aplicación que no está precompilada, es necesario crear manualmente un párrafo SPECIAL-NAMES en la aplicación, similar al proporcionado más arriba. Si llama a una API de DB2 directamente, necesitará añadir manualmente el símbolo DB2API a continuación de la palabra clave "call".

Tareas relacionadas:

- "Creación de aplicaciones Micro Focus COBOL en Windows" en la página 304
- "Creación de rutinas Micro Focus COBOL en Windows" en la página 307

Información relacionada:

- "Opciones de compilación y enlace para aplicaciones Micro Focus COBOL de Windows" en la página 306
- "Opciones de compilación y enlace para rutinas Micro Focus COBOL de Windows" en la página 309

Creación de aplicaciones Micro Focus COBOL en Windows

DB2 proporciona archivos de proceso por lotes para compilar y enlazar programas de la API de DB2 y de SQL incorporado. Estos archivos residen en el directorio sql1lib\samples\cobol_mf, junto con programas de ejemplo que se pueden crear a partir de esos archivos.

El archivo de proceso por lotes b1dapp.bat contiene los mandatos para crear programas de aplicación DB2. Utiliza un máximo de cuatro parámetros como entrada, que están representados dentro del archivo de proceso por lotes por las variables %1, %2, %3 y %4.

El primer parámetro, %1, especifica el nombre del archivo fuente. Éste es el único parámetro necesario para los programas que no contienen SQL incorporado. Para

crear programas de SQL incorporado es necesaria una conexión con la base de datos, por lo que también se proporcionan otros tres parámetros: el segundo parámetro, %2, especifica el nombre de la base de datos con la que se desea conectar; el tercer parámetro, %3, especifica el ID de usuario correspondiente a la base de datos, y %4 especifica la contraseña.

Para un programa de SQL incorporado, `bldapp` pasa los parámetros al archivo de precompilación y vinculación, `embprep.bat`. Si no se proporciona un nombre de base de datos, se utiliza la base de datos por omisión `sample`. Los parámetros de ID de usuario y contraseña sólo son necesarios si la instancia donde se crea el programa es diferente de la instancia donde reside la base de datos.

Procedimiento:

Los ejemplos siguientes muestran cómo crear y ejecutar programas de la API de DB2 y aplicaciones de SQL incorporado.

Para crear el programa de ejemplo de SQL no incorporado, `client`, a partir del archivo fuente `client.cbl`, especifique:

```
bldapp client
```

El resultado es un archivo ejecutable, `client.exe`. Para ejecutar el archivo ejecutable sobre la base de datos `sample`, especifique el nombre del ejecutable (sin la extensión):

```
client
```

Creación y ejecución de aplicaciones de SQL incorporado

A partir del archivo fuente `updat.sqb`, puede crear la aplicación de SQL incorporado, `updat`, de tres maneras:

1. Si conecta con la base de datos "sample" en la misma instancia, especifique:

```
bldapp updat
```

2. Si conecta con otra base de datos en la misma instancia, especifique también el nombre de la base de datos:

```
bldapp updat basedatos
```

3. Si conecta con una base de datos de otra instancia, especifique también el ID de usuario y la contraseña de la instancia de base de datos:

```
bldapp updat basedatos IDusuario contraseña
```

El resultado es un archivo ejecutable, `updat.exe`.

Existen tres maneras de ejecutar esta aplicación de SQL incorporado:

1. Si accede a la base de datos `sample` en la misma instancia, especifique simplemente el nombre del ejecutable (sin la extensión):

```
updat
```

2. Si desea acceder a otra base de datos en la misma instancia, especifique el nombre del ejecutable y el nombre de la base de datos:

```
updat basedatos
```

3. Si desea acceder a una base de datos de otra instancia, especifique el nombre del ejecutable, el nombre de la base de datos, y el ID de usuario y la contraseña de la instancia de base de datos:

```
bldapp updat basedatos IDusuario contraseña
```

Conceptos relacionados:

- “Archivos de creación” en la página 110

Información relacionada:

- “Opciones de compilación y enlace para aplicaciones Micro Focus COBOL de Windows” en la página 306
- “Ejemplos de COBOL” en la página 82

Ejemplos relacionados:

- “bldapp.bat -- Builds Windows Micro Focus Cobol applications”
- “client.cbl -- How to set and query a client (MF COBOL)”
- “updat.sqb -- How to update, delete and insert table data (MF COBOL)”
- “embprep.bat -- Prep and binds a C/C++ or Micro Focus COBOL embedded SQL program on Windows”

Archivo de proceso por lotes para aplicaciones COBOL para Micro Focus

```
@echo off
rem BATCH FILE: bldapp.bat
rem Crear aplicaciones Cobol de Micro Focus para Windows
rem Uso: bldapp <nombre_prog> [ <nombre_db> [ <IDusuario> <contraseña> ]]

rem Si es un programa de SQL incorporado, precompílelo y vincúlelo.
if not exist "%1.sqb" goto compile_step
call embprep %1 %2 %3 %4

:compile_step
rem Compilar el programa de utilidad de comprobación de errores.
cobol checkerr.cbl;

rem Compile el programa.
cobol %1.cbl;

rem Enlazar el programa.
cbl1link -l %1.obj checkerr.obj db2api.lib
@echo on
```

Opciones de compilación y enlace para aplicaciones Micro Focus COBOL de Windows

La tabla siguiente muestra las opciones de compilación y enlace que DB2 recomienda para crear aplicaciones en COBOL de SQL incorporado y de la API de DB2 cuando se utiliza Windows y el compilador Micro Focus COBOL, tal como muestra el archivo de proceso por lotes bldapp.bat.

Opciones de compilación y enlace para bldapp	
Opción de compilación:	
cobol	Denota el compilador Micro Focus COBOL.

Opciones de compilación y enlace para bldapp	
Opciones de enlace:	
cbllink	Utilizar el enlazador para la edición de enlaces.
-1	Enlazar con la biblioteca lcbol.
checkerr.obj	Enlazar con el archivo objeto del programa de utilidad de comprobación de errores.
db2api.lib	Enlazar con la biblioteca DB2 de la API
Consulte la documentación del compilador para conocer otras opciones de compilador.	

Tareas relacionadas:

- “Creación de aplicaciones Micro Focus COBOL en Windows” en la página 304

Ejemplos relacionados:

- “bldapp.bat -- Builds Windows Micro Focus Cobol applications”

Creación de rutinas Micro Focus COBOL en Windows

DB2 proporciona archivos de proceso por lotes para compilar y enlazar programas de la API de DB2 y de SQL incorporado en Micro Focus COBOL. Estos archivos residen en el directorio `sql11ib\samples\cobol_mf`, junto con programas de ejemplo que se pueden crear a partir de esos archivos.

El archivo de proceso por lotes `bldrtn.bat` contiene los mandatos para crear rutinas (procedimientos almacenados) de SQL incorporado. El archivo de proceso por lotes compila las rutinas y crea una DLL en el servidor. El archivo utiliza dos parámetros como entrada, que están representados dentro del archivo de proceso por lotes por las variables `%1` y `%2`.

El primer parámetro, `%1`, especifica el nombre del archivo fuente. El archivo de proceso por lotes utiliza el nombre del archivo fuente, `%1`, como nombre de la DLL. El segundo parámetro, `%2`, especifica el nombre de la base de datos a la que desea conectarse. Debido a que el procedimiento almacenado se debe crear en la misma instancia donde reside la base de datos, no hay parámetros para el ID de usuario ni la contraseña.

Sólo es obligatorio el primer parámetro, el nombre del archivo fuente. El nombre de la base de datos es opcional. Si no se proporciona un nombre de base de datos, el programa utiliza la base de datos por omisión `sample`.

Procedimiento:

Si conecta con la base de datos “sample”, especifique lo siguiente para crear el programa de ejemplo `outsrv` a partir del archivo fuente `outsrv.sqb`:

```
bldrtn outsrv
```

Si conecta con otra base de datos, especifique también el nombre de la base de datos:

```
bldrtn outsrv basedatos
```

El archivo de script copia la DLL en el directorio sqllib/function del servidor.

Una vez creada la DLL outsrv, puede crear la aplicación cliente outcli, la cual llama a la rutina contenida en la DLL y que tiene el mismo que la DLL. Puede crear outcli utilizando el archivo de proceso por lotes bldapp.bat.

Para invocar la rutina outsrv, ejecute la aplicación cliente de ejemplo, especificando lo siguiente:

```
outcli basedatos IDusuario contraseña
```

donde

basedatos

Es el nombre de la base de datos a la que desea conectarse. El nombre podría ser sample, o su alias, u otro nombre.

IDusuario

Es un ID de usuario válido.

contraseña

Es una contraseña válida correspondiente al ID de usuario.

La aplicación cliente accede a la DLL outsrv, la cual ejecuta la rutina del mismo nombre contenida en la base de datos del servidor. A continuación los datos resultantes se devuelven a la aplicación cliente.

Conceptos relacionados:

- “Archivos de creación” en la página 110

Información relacionada:

- “Opciones de compilación y enlace para rutinas Micro Focus COBOL de Windows” en la página 309
- “Ejemplos de COBOL” en la página 82

Ejemplos relacionados:

- “bldrtn.bat -- Builds Windows Micro Focus Cobol routines (stored procedures)”
- “outcli.sqb -- Call stored procedures using the SQLDA structure (MF COBOL)”
- “outsrv.sqb -- Demonstrates stored procedures using the SQLDA structure (MF COBOL)”
- “embprep.bat -- Prep and binds a C/C++ or Micro Focus COBOL embedded SQL program on Windows”

Archivo de proceso por lotes para rutinas COBOL para Micro Focus

```
@echo off
rem BATCH FILE: bldrtn.bat
rem Crear rutinas Cobol de Micro Focus para Windows (procedimientos almacenados)
rem Uso: bldsrv <nombre_prog> [ <nombre_bd> ]

rem Precompilar y vincular el programa.
call embprep %1 %2

rem Compile el procedimiento almacenado.
cobol %1.cbl /case;

rem Enlace el procedimiento almacenado y cree una biblioteca compartida.
cbllink /d %1.obj db2api.lib
```

```
rem Copiar el procedimiento almacenado al directorio %DB2PATH%\function.
copy %1.dll "%DB2PATH%\function"
@echo on
```

Opciones de compilación y enlace para rutinas Micro Focus COBOL de Windows

La tabla siguiente muestra las opciones de compilación y enlace que DB2 recomienda para crear rutinas en COBOL (procedimientos almacenados y funciones definidas por el usuario) cuando se utiliza Windows y el compilador Micro Focus COBOL, tal como muestra el archivo de proceso por lotes bldrtn.bat.

Opciones de compilación y enlace para bldrtn	
Opciones de compilación:	
cobo1	Denota el compilador Micro Focus COBOL.
/case	Impide que los símbolos externos se conviertan en mayúsculas.
Opciones de enlace:	
cb1link	Utilizar el enlazador Micro Focus COBOL para la edición de enlaces.
/d	Crear un archivo .dll.
db2api.lib	Enlazar con la biblioteca DB2 de la API.
Consulte la documentación del compilador para conocer otras opciones de compilador.	

Tareas relacionadas:

- “Creación de rutinas Micro Focus COBOL en Windows” en la página 307

Ejemplos relacionados:

- “bldrtn.bat -- Builds Windows Micro Focus Cobol routines (stored procedures)”

Object REXX

Creación de aplicaciones Object REXX en Windows

Object REXX es un versión del lenguaje REXX orientada a objetos. Se han añadido extensiones orientadas a objetos al REXX estándar, pero no se han modificado sus funciones e instrucciones existentes. El programa intérprete de Object REXX es una versión mejorada de su predecesor, e incluye soporte adicional para:

- Clases, objetos y métodos
- Gestión de mensajes y polimorfismo
- Herencia simple y múltiple

Object REXX es totalmente compatible con el REXX estándar. En esta sección, siempre que se utiliza el término REXX se hace referencia a todas las versiones de REXX, incluido Object REXX.

No es necesario precompilar ni enlazar los programas REXX.

En Windows, no es necesario que los programas REXX comiencen con un comentario. Sin embargo, por razones de portabilidad, es recomendable que en cada programa REXX coloque un comentario que comience en la primera columna de la primera línea. Esto permitirá diferenciar el programa respecto de un mandato de proceso por lotes cuando se ejecute en otras plataformas:

```
/* Coloque aquí un comentario cualquiera. */
```

Los programas REXX de ejemplo están contenidos en el directorio `sql11ib\samples\rexx`.

Procedimiento:

Para ejecutar el programa REXX de ejemplo `updat`, especifique:

```
rexx updat.cmd
```

Conceptos relacionados:

- “Consideraciones sobre la programación en REXX” en el manual *Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente*

Información relacionada:

- “Ejemplos de REXX” en la página 100

Parte 4. Apéndices

Apéndice A. Información técnica sobre DB2 Universal Database

Documentación y ayuda de DB2

Está disponible información técnica de DB2® a través de las herramientas y los métodos siguientes:

- Centro de información de DB2
 - Temas
 - Herramientas de ayuda para DB2
 - Programas de ejemplo
 - Guías de aprendizaje
- Archivos PDF descargables y en CD y manuales impresos
 - Guías
 - Manuales de consulta
- Ayuda de línea de mandatos
 - Ayuda de mandatos
 - Ayuda de mensajes
 - Ayuda para estados de SQL
- Código fuente instalado
 - Programas de ejemplo

Puede acceder a información técnica adicional de DB2 Universal Database™ como, por ejemplo, notas técnicas, white papers y Redbooks™ en línea en ibm.com®. Acceda al sitio de la biblioteca de software de gestión de información de DB2 en www.ibm.com/software/data/pubs/.

Actualizaciones de la documentación de DB2

De forma periódica, IBM® puede realizar FixPaks de la documentación y otras actualizaciones de la misma en el Centro de información de DB2 disponible. Si accede al Centro de información de DB2 en <http://publib.boulder.ibm.com/infocenter/db2help/>, siempre visualizará la información más actualizada. Si ha instalado el Centro de información de DB2 localmente, tendrá que instalar cualquier actualización de forma manual para poder visualizarla. Las actualizaciones de la documentación le permiten actualizar la información que ha instalado desde el *CD del Centro de información de DB2* cuando está disponible nueva información.

El Centro de información se actualiza con mayor frecuencia que los manuales PDF o en copia impresa. Para conseguir la información técnica de DB2 más actualizada, instale las actualizaciones de la documentación a medida que estén disponibles o diríjase al Centro de información de DB2 en el sitio www.ibm.com.

Conceptos relacionados:

- “CLI sample programs” en el manual *CLI Guide and Reference, Volume 1*
- “Programas Java de ejemplo” en la página 121
- “Centro de información de DB2” en la página 314

Tareas relacionadas:

- “Invocación de ayuda según contexto desde una herramienta de DB2” en la página 332
- “Actualización del Centro de información de DB2 instalado en el sistema o en un servidor de intranet” en la página 324
- “Invocación de la ayuda de mensajes desde el procesador de línea de mandatos” en la página 334
- “Invocación de la ayuda de mandatos desde el procesador de línea de mandatos” en la página 334
- “Invocación de la ayuda para estados de SQL desde el procesador de línea de mandatos” en la página 335

Información relacionada:

- “Documentación PDF e impresa de DB2” en la página 326

Centro de información de DB2

El Centro de información de DB2[®] le proporciona acceso a toda la información que necesita para obtener el máximo provecho de los productos de la familia de DB2, incluidos DB2 Universal Database[™], DB2 Connect[™], DB2 Information Integrator y DB2 Query Patroller[™]. El Centro de información de DB2 también contiene información relativa a las características y los componentes principales de DB2, como la duplicación, el depósito de datos y DB2 Extenders.

El Centro de información de DB2 presenta las características siguientes si se visualiza en Mozilla 1.0 o posterior o bien en Microsoft[®] Internet Explorer 5.5 o posterior. Algunas características requieren que se habilite el soporte de JavaScript[™]:

Opciones flexibles de instalación

Puede elegir visualizar la documentación de DB2 utilizando la opción que mejor se ajuste a sus necesidades:

- Para asegurarse fácilmente de que la documentación siempre esté actualizada, puede acceder a toda la documentación directamente desde el Centro de información de DB2 incluido en el sitio Web de IBM[®] de <http://publib.boulder.ibm.com/infocenter/db2help/>
- Para minimizar el esfuerzo de actualización y mantener el tráfico de red en su intranet, puede instalar la documentación de DB2 en un solo servidor de la intranet
- Para maximizar la flexibilidad y reducir la dependencia de las conexiones de red, puede instalar la documentación de DB2 en su propio sistema

Búsqueda

Es posible buscar en todos los temas del Centro de información de DB2 entrando un término de búsqueda en el campo de texto **Buscar**. Puede recuperar coincidencias exactas encerrando los términos entre comillas y puede afinar la búsqueda mediante operadores de comodín (*, ?) y operadores booleanos (AND, NOT, OR).

Tabla de contenido orientada a tareas

Puede localizar los temas en la documentación de DB2 a partir de una sola tabla de contenido. La tabla de contenido está organizada principalmente

según la clase de tareas que puede desear realizar, pero también incluye entradas para visiones generales de productos, objetivos, información de consulta, un índice y un glosario.

- Las visiones generales de los productos describen la relación entre los productos disponibles en la familia de DB2, las características que ofrece cada uno de estos productos y proporcionan información actualizada del release de cada uno de estos productos.
- Las categorías de objetivos, como la instalación, la administración y el desarrollo, incluyen temas que permiten realizar rápidamente tareas y desarrollar un conocimiento más profundo de la información de fondo para realizar dichas tareas.
- Los temas de consulta proporcionan información detallada sobre un tema, incluida la sintaxis de sentencias y mandatos, la ayuda de mensajes y los parámetros de configuración.

Mostrar el tema actual en la tabla de contenido

Puede mostrar dónde encaja el tema actual en la tabla de contenido pulsando el botón **Renovar / Mostrar tema actual** en el marco de la tabla de contenido o pulsando el botón **Mostrar en tabla de contenido** en el marco del contenido. Esta característica es útil si ha seguido varios enlaces con temas relacionados en varios archivos o ha llegado a un tema a partir de resultados de una búsqueda.

Índice Es posible acceder a toda la documentación desde el índice. El índice está organizado en orden alfabético por términos del índice.

Glosario

Puede utilizar el glosario a fin de buscar definiciones de términos utilizados en la documentación de DB2. El glosario está organizado en orden alfabético por términos del glosario.

Información adaptada integrada

El Centro de información de DB2 visualiza la información en el idioma preferido que se ha establecido en las preferencias de navegador. Si un tema no está disponible en el idioma preferido del usuario, el Centro de información de DB2 visualiza la versión inglesa de ese tema.

Si desea información técnica sobre iSeries™, consulte el centro de información de IBM eServer™ iSeries en www.ibm.com/eserver/iseriess/infocenter/.

Conceptos relacionados:

- “Escenarios de instalación del Centro de información de DB2” en la página 316

Tareas relacionadas:

- “Actualización del Centro de información de DB2 instalado en el sistema o en un servidor de intranet” en la página 324
- “Visualización de temas en el idioma preferido en el Centro de información de DB2” en la página 325
- “Invocación del Centro de información de DB2” en la página 323
- “Instalación del Centro de información de DB2 utilizando el asistente de instalación de DB2 (UNIX)” en la página 318
- “Instalación del Centro de información de DB2 utilizando el asistente de instalación de DB2 (Windows)” en la página 321

Escenarios de instalación del Centro de información de DB2

Los entornos de trabajo distintos pueden plantear requisitos distintos para el modo de acceder a la información de DB2[®]. Se puede acceder al Centro de información de DB2 en el sitio Web de IBM[®], en un servidor de la red de la organización o en una versión instalada en el sistema. En los tres casos, la documentación está incluida en el Centro de información de DB2, el cual consiste en una Web estructurada de información que se organiza en temas y que se visualiza mediante un navegador. Por omisión, los productos de DB2 acceden al Centro de información de DB2 en el sitio Web de IBM. No obstante, si desea acceder al Centro de información de DB2 en un servidor de intranet o en su propio sistema, es necesario que instale el Centro de información de DB2 utilizando el CD del Centro de información de DB2 que encontrará en el Paquete de soportes del producto. Consulte el siguiente resumen de opciones para acceder a la documentación de DB2, junto con los tres escenarios de instalación, como ayuda para determinar qué método de acceso al Centro de información de DB2 le funciona mejor en su entorno de trabajo y qué cuestiones relacionadas con la instalación se pueden tener en cuenta.

Resumen de opciones para acceder a la documentación de DB2:

La siguiente tabla proporciona recomendaciones sobre las opciones que son posibles en su entorno de trabajo a la hora de acceder a la documentación de productos de DB2 del Centro de información de DB2.

Acceso a Internet	Acceso a Intranet	Recomendación
Sí	Sí	Acceda al Centro de información de DB2 en el sitio Web de IBM o acceda al Centro de información de DB2 instalado en un servidor de intranet.
Sí	No	Acceda al Centro de información de DB2 en el sitio Web de IBM.
No	Sí	Acceda al Centro de información de DB2 instalado en un servidor de intranet.
No	No	Acceda al Centro de información de DB2 en un sistema local.

Escenario: Acceso al Centro de información de DB2 en su sistema:

Tsu-Chen es propietario de una fábrica en una pequeña ciudad que no dispone de ISP local para proporcionarle acceso a Internet. Ha adquirido DB2 Universal Database[™] para la gestión de su inventario, pedidos de productos, información de cuentas bancarias y gastos empresariales. Puesto que nunca había utilizado un producto de DB2 anteriormente, Tsu-Chen tendrá que aprender a partir de la documentación de productos de DB2.

Después de instalar DB2 Universal Database en el sistema utilizando la opción de instalación típica, Tsu-Chen intenta acceder a la documentación de DB2. Sin embargo, el navegador emite un mensaje de error que indica que la página que ha intentado abrir no se encuentra. Tsu-Chen comprueba el manual de instalación de su producto de DB2 y descubre que tiene que instalar el Centro de información de DB2 si desea acceder a la documentación de DB2 en su sistema. Encuentra el *CD del Centro de información de DB2* en el paquete de soportes y lo instala.

Desde el programa ejecutor de aplicaciones del sistema operativo, Tsu-Chen dispone ahora de acceso al Centro de información de DB2 y puede aprender a utilizar el producto de DB2 para incrementar el éxito de su empresa.

Escenario: Acceso al Centro de información de DB2 en el sitio Web de IBM:

Colin es un consultor de tecnologías de la información con una empresa de formación. Está especializado en tecnología de bases de datos y SQL y ofrece clases sobre estos temas a empresas por toda Norteamérica utilizando DB2 Universal Database. Parte de las clases de Colin incluye el uso de la documentación de DB2 como una herramienta didáctica. Por ejemplo, mientras imparte los cursos sobre SQL, Colin utiliza la documentación de DB2 relativa a SQL como un modo de enseñar sintaxis básica y avanzada para las consultas de base de datos.

La mayoría de las empresas en las que Colin imparte cursos tienen acceso a Internet. Esta situación ha influido en la decisión de Colin de configurar su sistema portátil para que acceda al Centro de información de DB2 en el sitio Web de IBM cuando ha instalado la versión más reciente de DB2 Universal Database. Dicha configuración permite a Colin disponer de acceso en línea a la documentación más reciente de DB2 durante sus clases.

Sin embargo, a veces, mientras viaja, Colin no tiene acceso a Internet. Esto le planteaba un problema, especialmente cuando necesitaba acceder a la documentación de DB2 para preparar las clases. A fin de evitar tales situaciones, Colin ha instalado una copia del Centro de información de DB2 en el sistema portátil.

Colin disfruta de la flexibilidad que supone tener siempre una copia de la documentación de DB2 a su disposición. Mediante el mandato **db2set**, puede configurar fácilmente las variables de registro en el sistema portátil para acceder al Centro de información de DB2 en el sitio Web de IBM o en el sistema portátil, según su situación.

Escenario: Acceso al Centro de información de DB2 en un servidor de intranet:

El trabajo de Eva es el de administrador sénior de bases de datos en una compañía de seguros de vida. Sus responsabilidades administrativas incluyen la instalación y configuración de la versión más reciente de DB2 Universal Database en los servidores de bases de datos UNIX[®] de la compañía. Recientemente, la compañía ha informado a sus empleados de que, por razones de seguridad, no se les proporcionará acceso a Internet en el trabajo. Dado que la compañía tiene un entorno de red, Eva decide instalar una copia del Centro de información de DB2 en un servidor de intranet a fin de que todos los empleados de la compañía que utilicen el depósito de datos de la misma de forma regular (representantes de ventas, gestores de ventas y analistas de empresa) tengan acceso a la documentación de DB2.

Eva indica a su equipo encargado de las bases de datos que instalen la versión más reciente de DB2 Universal Database en los sistemas de todos los empleados a través de un archivo de respuestas, para asegurarse de que cada sistema esté configurado de manera que acceda al Centro de información de DB2 utilizando el nombre de sistema principal y el número de puerto del servidor de intranet.

No obstante, debido a un malentendido, Miguel, un administrador de bases de datos auxiliar del equipo de Eva, instala una copia del Centro de información de DB2 en varios sistemas de los empleados en lugar de configurar DB2 Universal

Database para que acceda al Centro de información de DB2 en el servidor de intranet. Con el fin de corregir esta situación, Eva indica a Miguel que utilice el mandato **db2set** para cambiar las variables de registro del Centro de información de DB2 (DB2_DOCHOST para el nombre de sistema principal y DB2_DOCPORT para el número de puerto) en cada uno de esos sistemas. Ahora todos los sistemas correspondientes de la red tienen acceso al Centro de información de DB2, y los empleados pueden hallar las respuestas a sus preguntas sobre DB2 en la documentación de DB2.

Conceptos relacionados:

- “Centro de información de DB2” en la página 314

Tareas relacionadas:

- “Actualización del Centro de información de DB2 instalado en el sistema o en un servidor de intranet” en la página 324
- “Instalación del Centro de información de DB2 utilizando el asistente de instalación de DB2 (UNIX)” en la página 318
- “Instalación del Centro de información de DB2 utilizando el asistente de instalación de DB2 (Windows)” en la página 321
- “Establecimiento de la ubicación para acceder al Centro de información de DB2”

Información relacionada:

- “db2set - Mandato Registro de perfiles de DB2” en el manual *Consulta de mandatos*

Instalación del Centro de información de DB2 utilizando el asistente de instalación de DB2 (UNIX)

Se puede acceder a la documentación de los productos de DB2 de tres maneras: en el sitio Web de IBM, en un servidor de intranet o en una versión instalada en el sistema. Por omisión, el acceso de los productos de DB2 dentro de la documentación de DB2 se efectúa en el sitio Web de IBM. Si desea acceder a la documentación de DB2 en un servidor de intranet o en su propio sistema, deberá instalar la documentación desde el *CD del Centro de información de DB2*. Mediante el asistente de instalación de DB2, puede definir sus preferencias de instalación e instalar el Centro de información de DB2 en un sistema que utilice un sistema operativo UNIX.

Prerrequisitos:

Este apartado lista los requisitos de hardware, sistema operativo, software y comunicaciones para instalar el Centro de información de DB2 en los sistemas UNIX.

• **Requisitos de hardware**

Necesita uno de los procesadores siguientes:

- PowerPC (AIX)
- HP 9000 (HP-UX)
- Intel de 32 bits (Linux)
- Sistemas Solaris UltraSPARC (Entorno operativo Solaris)

• **Requisitos de sistema operativo**

Necesita uno de los sistemas operativos siguientes:

- IBM AIX 5.1 (en PowerPC)
- HP-UX 11i (en HP 9000)
- Red Hat Linux 8.0 (en Intel de 32 bits)
- SuSE Linux 8.1 (en Intel de 32 bits)
- Sun Solaris Versión 8 (en sistemas UltraSPARC del Entorno operativo Solaris)

Nota: El Centro de información de DB2 se ejecuta en un subconjunto de los sistemas operativos UNIX en los que están soportados los clientes DB2. Por consiguiente, es recomendable que acceda al Centro de información de DB2 desde el sitio Web de IBM o que instale el Centro de información de DB2 y acceda al mismo en un servidor de intranet.

- **Requisitos de software**

- Está soportado el navegador siguiente:
 - Mozilla Versión 1.0 o superior
- El asistente de instalación de DB2 es un instalador gráfico. Debe disponer de una implementación del software X Window System capaz de representar una interfaz gráfica de usuario para que el asistente de instalación de DB2 se ejecute en el sistema. A fin de ejecutar el asistente de instalación de DB2, debe asegurarse de que ha exportado debidamente la visualización. Por ejemplo, entre el mandato siguiente en el indicador de mandatos:


```
export DISPLAY=9.26.163.144:0.
```

- **Requisitos de comunicaciones**

- TCP/IP

Procedimiento:

Para instalar el Centro de información de DB2 utilizando el asistente de instalación de DB2:

1. Inicie una sesión en el sistema.
2. Inserte y monte el CD del producto Centro de información de DB2 en el sistema.
3. Vaya al directorio en el que está montado el CD entrando el mandato siguiente:


```
cd /cd
```

donde */cd* representa el punto de montaje del CD.

4. Entre el mandato **`./db2setup`** para iniciar el asistente de instalación de DB2.
5. Se abrirá el Área de ejecución para la instalación de IBM DB2. Para continuar directamente con la instalación del Centro de información de DB2, pulse en **Instalar producto**. Existe ayuda en línea disponible para guiarle durante los pasos restantes. Para invocar la ayuda en línea, pulse en **Ayuda**. Puede pulsar en **Cancelar** en cualquier momento para interrumpir la instalación.
6. En la página **Seleccione el producto que desee instalar**, pulse en **Siguiente**.
7. Pulse en **Siguiente** en la página **Bienvenido al asistente de instalación de DB2**. El asistente de instalación de DB2 le guiará durante el proceso de instalación del programa.
8. Para continuar con la instalación, debe aceptar el contrato de licencia. En la página **Contrato de licencia**, seleccione **Acepto los términos del contrato de licencia** y pulse en **Siguiente**.
9. Seleccione **Instalar el Centro de información de DB2 en este sistema** en la página **Seleccionar la acción de instalación**. Si desea utilizar un archivo de

respuestas para instalar el Centro de información de DB2 en éste o en otros sistemas más adelante, seleccione **Guardar los valores en un archivo de respuestas**. Pulse en **Siguiente**.

10. Seleccione los idiomas en los que se instalará el Centro de información de DB2 en la página **Seleccionar los idiomas a instalar**. Pulse en **Siguiente**.
11. Configure el Centro de información de DB2 para las comunicaciones entrantes en la página **Especificar el puerto del Centro de información de DB2**. Pulse en **Siguiente** para continuar la instalación.
12. Revise las opciones de instalación que ha elegido en la página **Comenzar a copiar archivos**. Para cambiar cualquier valor, pulse en **Anterior**. Pulse en **Instalar** para copiar los archivos del Centro de información de DB2 en el sistema.

También puede instalar el Centro de información de DB2 utilizando un archivo de respuestas.

Los archivos de anotaciones cronológicas de instalación db2setup.his, db2setup.log y db2setup.err están ubicados, por omisión, en el directorio /tmp.

El archivo db2setup.log capta toda la información de instalación del producto de DB2, incluidos los errores. El archivo db2setup.his registra todas las instalaciones de productos de DB2 en el sistema. DB2 añade el archivo db2setup.log al archivo db2setup.his. El archivo db2setup.err capta cualquier salida de errores devuelta por Java, como, por ejemplo, información de interrupciones y excepciones.

Cuando se haya completado la instalación, el Centro de información de DB2 estará instalado en uno de los directorios siguientes, según el sistema operativo UNIX:

- AIX: /usr/opt/db2_08_01
- HP-UX: /opt/IBM/db2/V8.1
- Linux: /opt/IBM/db2/V8.1
- Entorno operativo Solaris: /opt/IBM/db2/V8.1

Conceptos relacionados:

- “Centro de información de DB2” en la página 314
- “Escenarios de instalación del Centro de información de DB2” en la página 316

Tareas relacionadas:

- “Instalación de DB2 utilizando un archivo de respuestas (UNIX)” en el manual *Suplemento de instalación y configuración*
- “Actualización del Centro de información de DB2 instalado en el sistema o en un servidor de intranet” en la página 324
- “Visualización de temas en el idioma preferido en el Centro de información de DB2” en la página 325
- “Invocación del Centro de información de DB2” en la página 323
- “Instalación del Centro de información de DB2 utilizando el asistente de instalación de DB2 (Windows)” en la página 321

Instalación del Centro de información de DB2 utilizando el asistente de instalación de DB2 (Windows)

Se puede acceder a la documentación de los productos de DB2 de tres maneras: en el sitio Web de IBM, en un servidor de intranet o en una versión instalada en el sistema. Por omisión, el acceso de los productos de DB2 dentro de la documentación de DB2 se efectúa en el sitio Web de IBM. Si desea acceder a la documentación de DB2 en un servidor de intranet o en su propio sistema, deberá instalar la documentación de DB2 desde el *CD del Centro de información de DB2*. Mediante el asistente de instalación de DB2, puede definir sus preferencias de instalación e instalar el Centro de información de DB2 en un sistema que utilice un sistema operativo Windows.

Prerrequisitos:

Este apartado lista los requisitos de hardware, sistema operativo, software y comunicaciones para instalar el Centro de información de DB2 en Windows.

- **Requisitos de hardware**

Necesita uno de los procesadores siguientes:

- Sistemas de 32 bits: una CPU Pentium o compatible con Pentium

- **Requisitos de sistema operativo**

Necesita uno de los sistemas operativos siguientes:

- Windows 2000
- Windows XP

Nota: El Centro de información de DB2 se ejecuta en un subconjunto de los sistemas operativos Windows en los que están soportados los clientes DB2. Por consiguiente, es recomendable que acceda al Centro de información de DB2 en el sitio Web de IBM o que instale el Centro de información de DB2 y acceda al mismo en un servidor de intranet.

- **Requisitos de software**

– Están soportados los navegadores siguientes:

- Mozilla 1.0 o superior
- Internet Explorer Versión 5.5 ó 6.0 (Versión 6.0 para Windows XP)

- **Requisitos de comunicaciones**

- TCP/IP

Restricciones:

- Necesita una cuenta con privilegios administrativos para instalar el Centro de información de DB2.

Procedimiento:

Para instalar el Centro de información de DB2 utilizando el asistente de instalación de DB2:

1. Inicie una sesión en el sistema con la cuenta que ha definido para la instalación del Centro de información de DB2.
2. Inserte el CD en la unidad. Si está habilitada, la característica de ejecución automática inicia el Área de ejecución para la instalación de IBM DB2.
3. El asistente de instalación de DB2 determina el idioma del sistema y ejecuta el programa de instalación para ese idioma. Si desea ejecutar el programa de

instalación en un idioma distinto del inglés o bien el programa de instalación no se inicia de forma automática, puede iniciar el asistente de instalación de DB2 manualmente.

Para iniciar el asistente de instalación de DB2 manualmente:

- a. Pulse en **Inicio** y seleccione **Ejecutar**.
- b. En el campo **Abrir**, escriba el mandato siguiente:

```
x:\setup.exe /i identificador de idioma de 2 letras
```

donde *x*: representa la unidad de CD, e *identificador de idioma de 2 letras* representa el idioma en el que se ejecutará el programa de instalación.

- c. Pulse en **Aceptar**.
4. Se abrirá el Área de ejecución para la instalación de IBM DB2. Para continuar directamente con la instalación del Centro de información de DB2, pulse en **Instalar producto**. Existe ayuda en línea disponible para guiarle durante los pasos restantes. Para invocar la ayuda en línea, pulse en **Ayuda**. Puede pulsar en **Cancelar** en cualquier momento para interrumpir la instalación.
5. En la página **Seleccione el producto que desee instalar**, pulse en **Siguiente**.
6. Pulse en **Siguiente** en la página **Bienvenido al asistente de instalación de DB2**. El asistente de instalación de DB2 le guiará durante el proceso de instalación del programa.
7. Para continuar con la instalación, debe aceptar el contrato de licencia. En la página **Contrato de licencia**, seleccione **Acepto los términos del contrato de licencia** y pulse en **Siguiente**.
8. Seleccione **Instalar el Centro de información de DB2 en este sistema** en la página **Seleccionar la acción de instalación**. Si desea utilizar un archivo de respuestas para instalar el Centro de información de DB2 en éste o en otros sistemas más adelante, seleccione **Guardar los valores en un archivo de respuestas**. Pulse en **Siguiente**.
9. Seleccione los idiomas en los que se instalará el Centro de información de DB2 en la página **Seleccionar los idiomas a instalar**. Pulse en **Siguiente**.
10. Configure el Centro de información de DB2 para las comunicaciones entrantes en la página **Especificar el puerto del Centro de información de DB2**. Pulse en **Siguiente** para continuar la instalación.
11. Revise las opciones de instalación que ha elegido en la página **Comenzar a copiar archivos**. Para cambiar cualquier valor, pulse en **Anterior**. Pulse en **Instalar** para copiar los archivos del Centro de información de DB2 en el sistema.

Puede instalar el Centro de información de DB2 utilizando un archivo de respuestas. También es posible utilizar el mandato **db2rspgn** a fin de generar un archivo de respuestas basado en una instalación existente.

Para obtener información sobre los errores encontrados durante la instalación, consulte los archivos db2.log y db2wi.log ubicados en el directorio 'Mis documentos'\DB2LOG\. La ubicación del directorio 'Mis documentos' dependerá de la configuración de su sistema.

El archivo db2wi.log capta la información de la instalación de DB2 más reciente. El archivo db2.log capta el historial de instalaciones de productos de DB2.

Conceptos relacionados:

- "Centro de información de DB2" en la página 314

- “Escenarios de instalación del Centro de información de DB2” en la página 316

Tareas relacionadas:

- “Instalación de un producto DB2 utilizando un archivo de respuestas (Windows)” en el manual *Suplemento de instalación y configuración*
- “Actualización del Centro de información de DB2 instalado en el sistema o en un servidor de intranet” en la página 324
- “Visualización de temas en el idioma preferido en el Centro de información de DB2” en la página 325
- “Invocación del Centro de información de DB2” en la página 323
- “Instalación del Centro de información de DB2 utilizando el asistente de instalación de DB2 (UNIX)” en la página 318

Información relacionada:

- “db2rspgn - Mandato del Generador de archivos de respuestas (Windows)” en el manual *Consulta de mandatos*

Invocación del Centro de información de DB2

El Centro de información de DB2 proporciona acceso a toda la información que necesita para utilizar productos de DB2 para los sistemas operativos Linux, UNIX y Windows, tales como DB2 Universal Database, DB2 Connect, DB2 Information Integrator y DB2 Query Patroller.

Puede invocar el Centro de información de DB2 desde una de las ubicaciones siguientes:

- Sistemas en los que está instalado un cliente o servidor DB2 UDB
- Un servidor de intranet o sistema local en el que está instalado el Centro de información de DB2
- El sitio Web de IBM

Prerrequisitos:

Antes de invocar el Centro de información de DB2:

- *Opcional:* Configure el navegador para que visualice los temas en su idioma preferido
- *Opcional:* Configure el cliente DB2 para que utilice el Centro de información de DB2 instalado en el sistema o servidor de intranet

Procedimiento:

Para invocar el Centro de información de DB2 en un sistema en el que está instalado un cliente o servidor DB2 UDB:

- Desde el menú Inicio (sistema operativo Windows): Pulse en **Inicio** → **Programas** → **IBM DB2** → **Información** → **Centro de información**.
- Desde el indicador de línea de mandatos:
 - En los sistemas operativos Linux y UNIX, emita el mandato **db2icdocs**.
 - En el sistema operativo Windows, emita el mandato **db2icdocs.exe**.

Para abrir el Centro de información de DB2 instalado en un servidor de intranet o sistema local en un navegador Web:

- Abra la página Web en <http://<nombre-sistemaprincipal>:<número-puerto>/>, donde <nombre-sistemaprincipal> representa el nombre de sistema principal y <número-puerto> representa el número de puerto en el que está disponible el Centro de información de DB2.

Para abrir el Centro de información de DB2 en el sitio Web de IBM en un navegador Web:

- Abra la página Web en publib.boulder.ibm.com/infocenter/db2help/.

Conceptos relacionados:

- “Centro de información de DB2” en la página 314
- “Escenarios de instalación del Centro de información de DB2” en la página 316

Tareas relacionadas:

- “Visualización de temas en el idioma preferido en el Centro de información de DB2” en la página 325
- “Invocación de ayuda según contexto desde una herramienta de DB2” en la página 332
- “Actualización del Centro de información de DB2 instalado en el sistema o en un servidor de intranet” en la página 324
- “Invocación de la ayuda de mandatos desde el procesador de línea de mandatos” en la página 334
- “Establecimiento de la ubicación para acceder al Centro de información de DB2”

Información relacionada:

- “Mandato HELP” en el manual *Consulta de mandatos*

Actualización del Centro de información de DB2 instalado en el sistema o en un servidor de intranet

El Centro de información de DB2 que hay disponible en <http://publib.boulder.ibm.com/infocenter/db2help/> se actualizará periódicamente con documentación nueva o modificada. Asimismo, IBM puede efectuar actualizaciones del Centro de información de DB2 disponibles para descargar e instalar en el sistema o servidor de intranet. La actualización del Centro de información de DB2 no actualiza los productos de cliente o servidor DB2.

Prerrequisitos:

Es necesario tener acceso a un sistema que esté conectado a Internet.

Procedimiento:

Para actualizar el Centro de información de DB2 instalado en el sistema o servidor de intranet:

1. Abra el Centro de información de DB2 que se encuentra en el sitio Web de IBM de: <http://publib.boulder.ibm.com/infocenter/db2help/>
2. En la sección de descargas de la página de bienvenida, bajo la cabecera de servicio y soporte, pulse en el enlace de **documentación de DB2 Universal Database**.
3. Determine si la versión de su Centro de información de DB2 está anticuada comparando el nivel de la última imagen de documentación renovada con el

nivel de documentación que tenga instalado. El nivel de documentación que ha instalado aparece listado en la página de bienvenida del Centro de información de DB2.

4. Si se encuentra disponible una versión más reciente del Centro de información de DB2, descargue la última imagen renovada del *Centro de información de DB2* aplicable a su sistema operativo.
5. Para instalar la imagen renovada del *Centro de información de DB2*, siga las instrucciones proporcionadas en la página Web.

Conceptos relacionados:

- “Escenarios de instalación del Centro de información de DB2” en la página 316

Tareas relacionadas:

- “Invocación del Centro de información de DB2” en la página 323
- “Instalación del Centro de información de DB2 utilizando el asistente de instalación de DB2 (UNIX)” en la página 318
- “Instalación del Centro de información de DB2 utilizando el asistente de instalación de DB2 (Windows)” en la página 321

Visualización de temas en el idioma preferido en el Centro de información de DB2

El Centro de información de DB2 intenta visualizar los temas en el idioma especificado en las preferencias de navegador. Si un tema no se ha traducido al idioma preferido del usuario, el Centro de información de DB2 visualiza dicho tema en inglés.

Procedimiento:

Para visualizar temas en su idioma preferido en el navegador Internet Explorer:

1. En Internet Explorer, pulse el botón **Herramientas** —> **Opciones de Internet** —> **Idiomas...** Se abrirá la ventana Preferencias de idioma.
2. Asegúrese de que su idioma preferido esté especificado como la primera entrada de la lista de idiomas.
 - Para añadir un nuevo idioma a la lista, pulse el botón **Agregar...**

Nota: La adición de un idioma no garantiza que el sistema tenga los fonts necesarios para visualizar los temas en el idioma preferido.

- Para mover un idioma hacia el principio de la lista, seleccione el idioma y pulse el botón **Subir** hasta que el idioma esté en primer lugar en la lista de idiomas.
3. Renueve la página a fin de visualizar el Centro de información de DB2 en su idioma preferido.

Para visualizar temas en su idioma preferido en el navegador Mozilla:

1. En Mozilla, seleccione el botón **Edit** —> **Preferences** —> **Languages**. Se visualizará el panel Languages en la ventana Preferences.
2. Asegúrese de que su idioma preferido esté especificado como la primera entrada de la lista de idiomas.
 - Para añadir un nuevo idioma a la lista, pulse el botón **Add...** a fin de seleccionar un idioma en la ventana Add Languages.

- Para mover un idioma hacia el principio de la lista, seleccione el idioma y pulse el botón **Move Up** hasta que el idioma esté en primer lugar en la lista de idiomas.
3. Renueve la página a fin de visualizar el Centro de información de DB2 en su idioma preferido.

Conceptos relacionados:

- “Centro de información de DB2” en la página 314

Documentación PDF e impresión de DB2

Las tablas siguientes proporcionan los nombres oficiales de los manuales, los números de documento y los nombres de los archivos PDF. Para solicitar manuales en copia impresa, debe conocer el nombre oficial del manual. Para imprimir un archivo PDF, debe conocer el nombre del archivo PDF.

La documentación de DB2 está categorizada según las cabeceras siguientes:

- Información básica de DB2
- Información de administración
- Información para el desarrollo de aplicaciones
- Información de Business Intelligence
- Información de DB2 Connect
- Información de iniciación
- Información de aprendizaje
- Información sobre componentes opcionales
- Notas del release

Las tablas siguientes describen, para cada manual de la biblioteca de DB2, la información necesaria para solicitar la copia impresa o para imprimir o ver el PDF correspondiente al manual en cuestión. Se encuentra una descripción completa de cada uno de los manuales de la biblioteca de DB2 en el Centro de publicaciones de IBM de www.ibm.com/shop/publications/order

Información básica de DB2

La información de estos manuales es fundamental para todos los usuarios de DB2; encontrará útil esta información tanto si es programador o administrador de bases de datos como si trabaja con DB2 Connect, DB2 Warehouse Manager u otros productos de DB2.

Tabla 42. Información básica de DB2

Nombre	Número de documento	Nombre de archivo PDF
IBM DB2 Universal Database Consulta de mandatos	SC10-3725	db2n0x81
IBM DB2 Universal Database Glosario	Sin número de documento	db2t0x81
IBM DB2 Universal Database Consulta de mensajes, Volumen 1	GC10-3728, no disponible en copia impresa	db2m1x81
IBM DB2 Universal Database Consulta de mensajes, Volumen 2	GC10-3729, no disponible en copia impresa	db2m2x81
IBM DB2 Universal Database Novedades	SC10-3734	db2q0x81

Información de administración

La información de estos manuales incluye los temas necesarios para diseñar, implementar y mantener de forma efectiva bases de datos de DB2, depósitos de datos y sistemas federados.

Tabla 43. Información de administración

Nombre	Número de documento	Nombre de archivo PDF
<i>IBM DB2 Universal Database Administration Guide: Planning</i>	SC09-4822	db2d1x81
<i>IBM DB2 Universal Database Administration Guide: Implementation</i>	SC09-4820	db2d2x81
<i>IBM DB2 Universal Database Administration Guide: Performance</i>	SC09-4821	db2d3x81
<i>IBM DB2 Universal Database Administrative API Reference</i>	SC09-4824	db2b0x81
<i>IBM DB2 Universal Database Data Movement Utilities Guide and Reference</i>	SC09-4830	db2dmx81
<i>IBM DB2 Universal Database Data Recovery and High Availability Guide and Reference</i>	SC09-4831	db2hax81
<i>IBM DB2 Universal Database Data Warehouse Center Administration Guide</i>	SC27-1123	db2ddx81
<i>IBM DB2 Universal Database Consulta de SQL, Volumen 1</i>	SC10-3730	db2s1x81
<i>IBM DB2 Universal Database Consulta de SQL, Volumen 2</i>	SC10-3731	db2s2x81
<i>IBM DB2 Universal Database System Monitor Guide and Reference</i>	SC09-4847	db2f0x81

Información para el desarrollo de aplicaciones

La información de estos manuales es de especial interés para los programadores de aplicaciones o programadores que trabajan con DB2 Universal Database (DB2 UDB). Hallará información acerca de los lenguajes y compiladores soportados, así como la documentación necesaria para acceder a DB2 UDB utilizando las diversas interfaces de programación soportadas, como, por ejemplo, SQL incorporado, ODBC, JDBC, SQLJ y CLI. Si utiliza el Centro de información de DB2, también podrá acceder a versiones HTML del código fuente para los programas de ejemplo.

Tabla 44. Información para el desarrollo de aplicaciones

Nombre	Número de documento	Nombre de archivo PDF
<i>IBM DB2 Universal Database Guía de desarrollo de aplicaciones: Creación y ejecución de aplicaciones</i>	SC10-3733	db2axx81

Tabla 44. Información para el desarrollo de aplicaciones (continuación)

Nombre	Número de documento	Nombre de archivo PDF
IBM DB2 Universal Database Guía de desarrollo de aplicaciones: Programación de aplicaciones de cliente	SC10-3723	db2a1x81
IBM DB2 Universal Database Guía de desarrollo de aplicaciones: Programación de aplicaciones de servidor	SC10-3724	db2a2x81
IBM DB2 Universal Database Call Level Interface Guide and Reference, Volume 1	SC09-4849	db2l1x81
IBM DB2 Universal Database Call Level Interface Guide and Reference, Volume 2	SC09-4850	db2l2x81
IBM DB2 Universal Database Data Warehouse Center Application Integration Guide	SC27-1124	db2adx81
IBM DB2 XML Extender Administración y programación	SC10-3750	db2sxx81

Información de Business Intelligence

La información de estos manuales describe cómo utilizar los componentes que mejoran las posibilidades de análisis y de depósito de datos de DB2 Universal Database.

Tabla 45. Información de Business Intelligence

Nombre	Número de documento	Nombre de archivo PDF
IBM DB2 Warehouse Manager Standard Edition Information Catalog Center Administration Guide	SC27-1125	db2dix81
IBM DB2 Warehouse Manager Standard Edition Installation Guide	GC27-1122	db2idx81
IBM DB2 Warehouse Manager Standard Edition Managing ETI Solution Conversion Programs with DB2 Warehouse Manager	SC18-7727	iwhe1mstx80

Información de DB2 Connect

La información incluida en esta categoría describe cómo acceder a datos de servidores de sistema principal y de sistema medio utilizando DB2 Connect Enterprise Edition o DB2 Connect Personal Edition.

Tabla 46. Información de DB2 Connect

Nombre	Número de documento	Nombre de archivo PDF
IBM Connectivity Supplement	Sin número de documento	db2h1x81

Tabla 46. Información de DB2 Connect (continuación)

Nombre	Número de documento	Nombre de archivo PDF
IBM DB2 Connect Guía rápida de iniciación para DB2 Enterprise Edition	GC10-3774	db2c6x81
IBM DB2 Connect Quick Beginnings for DB2 Connect Personal Edition	GC09-4834	db2c1x81
IBM DB2 Connect User's Guide	SC09-4835	db2c0x81

Información de iniciación

La información de esta categoría es útil cuando se van a instalar y configurar servidores, clientes y otros productos de DB2.

Tabla 47. Información de iniciación

Nombre	Número de documento	Nombre de archivo PDF
IBM DB2 Universal Database Guía rápida de iniciación para clientes DB2	GC10-3775, no disponible en copia impresa	db2itx81
IBM DB2 Universal Database Guía rápida de iniciación para servidores DB2	GC10-3773	db2isx81
IBM DB2 Universal Database Guía rápida de iniciación para DB2 Personal Edition	GC10-3771	db2i1x81
IBM DB2 Universal Database Suplemento de instalación y configuración	GC10-3772, no disponible en copia impresa	db2iyx81
IBM DB2 Universal Database Guía rápida de iniciación para DB2 Data Links Manager	GC10-3726	db2z6x81

Información de aprendizaje

La información de aprendizaje presenta las características de DB2 y explica cómo realizar diversas tareas.

Tabla 48. Información de aprendizaje

Nombre	Número de documento	Nombre de archivo PDF
Guía de aprendizaje de Business Intelligence: Introducción al Centro de depósito de datos	Sin número de documento	db2tux81
Guía de aprendizaje de Business Intelligence: Lecciones ampliadas sobre depósito de datos	Sin número de documento	db2tax81
Information Catalog Center Tutorial	Sin número de documento	db2aix81
Guía de aprendizaje de Video Central para e-business	Sin número de documento	db2twx81
Guía de aprendizaje de Visual Explain	Sin número de documento	db2tvx81

Información sobre componentes opcionales

La información de esta categoría describe cómo trabajar con los componentes opcionales de DB2.

Tabla 49. Información sobre componentes opcionales

Nombre	Número de documento	Nombre de archivo PDF
IBM DB2 Cube Views Guía y consulta	SC10-3868	db2aax81
IBM DB2 Query Patroller Guide: Installation, Administration and Usage Guide	GC09-7658	db2dwx81
IBM DB2 Spatial Extender and Geodetic Extender Guía del usuario y de consulta	SC10-3755	db2sbx81
IBM DB2 Universal Database Data Links Manager Administration Guide and Reference	SC27-1221	db2z0x82
DB2 Net Search Extender Administración y guía del usuario	SH10-9305	N/D

Nota: El HTML para este documento *no* se instala desde el CD de documentación HTML.

Notas del release

Las notas del release proporcionan información adicional específica del release y nivel de FixPak del producto. Las notas del release también proporcionan resúmenes de las actualizaciones de la documentación que se han incorporado en cada release, actualización y FixPak.

Tabla 50. Notas del release

Nombre	Número de documento	Nombre de archivo PDF
Notas del release de DB2	Ver nota.	Ver nota.
Notas de instalación de DB2	Sólo disponible en el CD-ROM del producto.	No disponible.

Nota: Las Notas del release están disponibles en:

- XHTML y formato de texto, en los CD de los productos
- Formato PDF, en el CD de documentación PDF

Además, las partes de las Notas del release que tratan *Problemas conocidos y soluciones alternativas* e *Incompatibilidades entre releases* también aparecen en el Centro de información de DB2.

Para ver las Notas del release en formato de texto en las plataformas basadas en UNIX, consulte el archivo Release.Notes. Este archivo se encuentra en el directorio DB2DIR/Readme/%L, donde %L representa el nombre de entorno nacional y DB2DIR representa:

- En los sistemas operativos AIX: /usr/opt/db2_08_01
- En los otros sistemas operativos basados en UNIX: /opt/IBM/db2/V8.1

Conceptos relacionados:

- “Documentación y ayuda de DB2” en la página 313

Tareas relacionadas:

- “Impresión de manuales de DB2 desde archivos PDF” en la página 331
- “Solicitud de manuales de DB2 impresos” en la página 332
- “Invocación de ayuda según contexto desde una herramienta de DB2” en la página 332

Impresión de manuales de DB2 desde archivos PDF

Puede imprimir los manuales de DB2 desde los archivos PDF del *CD de documentación PDF de DB2*. Mediante la utilización de Adobe Acrobat Reader, puede imprimir el manual entero o un rango específico de páginas.

Prerrequisitos:

Asegúrese de que tiene instalado Adobe Acrobat Reader. Si ha de instalar Adobe Acrobat Reader, está disponible desde el sitio Web de Adobe en www.adobe.com

Procedimiento:

Para imprimir un manual de DB2 desde un archivo PDF:

1. Inserte el *CD de documentación PDF de DB2*. En sistemas operativos UNIX, monte el CD de documentación PDF de DB2. Consulte el manual *Iniciación rápida* para obtener detalles sobre cómo montar un CD en sistemas operativos UNIX.
2. Abra `index.htm`. El archivo se abre en una ventana de navegador.
3. Pulse el título del PDF que desee ver. El PDF se abrirá en Acrobat Reader.
4. Seleccione **Archivo** → **Imprimir** para imprimir cualquier parte que desee del manual.

Conceptos relacionados:

- “Centro de información de DB2” en la página 314

Tareas relacionadas:

- “Montaje del CD-ROM (AIX)” en el manual *Guía rápida de iniciación para servidores DB2*
- “Cómo montar el CD-ROM (HP-UX)” en el manual *Guía rápida de iniciación para servidores DB2*
- “Montaje del CD-ROM (Linux)” en el manual *Guía rápida de iniciación para servidores DB2*
- “Solicitud de manuales de DB2 impresos” en la página 332
- “Montaje del CD-ROM (Entorno operativo Solaris)” en el manual *Guía rápida de iniciación para servidores DB2*

Información relacionada:

- “Documentación PDF e impresa de DB2” en la página 326

Solicitud de manuales de DB2 impresos

Si prefiere utilizar manuales en copia impresa, puede solicitarlos de tres modos distintos.

Procedimiento:

Los manuales impresos se pueden solicitar en algunos países o regiones. Compruebe, en el sitio Web de publicaciones de IBM correspondiente a su país o región, si este servicio está disponible en su país o región. Cuando las publicaciones estén disponibles para su solicitud, puede realizar lo siguiente:

- Póngase en contacto con el distribuidor autorizado o representante de marketing de IBM. Para encontrar un representante local de IBM, consulte el directorio mundial de contactos de IBM en la página Web www.ibm.com/planetwide
- Llame al teléfono 1-800-879-2755, si está en los EE.UU. o al 1-800-IBM-4YOU, si está en Canadá.
- Visite el Centro de publicaciones de IBM en <http://www.ibm.com/shop/publications/order>. La capacidad de solicitar manuales desde el Centro de publicaciones de IBM puede no estar disponible en todos los países.

En el momento en que un producto de DB2 se encuentra disponible, los manuales impresos son los mismos que aparecen en formato PDF en el *CD de documentación PDF de DB2*. El contenido de los manuales impresos que se halla en el *CD del Centro de información de DB2* también es el mismo. No obstante, existe contenido adicional en el CD del Centro de información de DB2 que no aparece en ninguno de los manuales PDF (por ejemplo, rutinas de administración de SQL y ejemplos de HTML). No todos los manuales incluidos en el CD de documentación PDF de DB2 se pueden solicitar en copia impresa.

Nota: El Centro de información de DB2 se actualiza con mayor frecuencia que los manuales PDF o en copia impresa; instale las actualizaciones de la documentación a medida que estén disponibles o consulte el Centro de información de DB2 en <http://publib.boulder.ibm.com/infocenter/db2help/> para obtener la información más actualizada.

Tareas relacionadas:

- “Impresión de manuales de DB2 desde archivos PDF” en la página 331

Información relacionada:

- “Documentación PDF e impresa de DB2” en la página 326

Invocación de ayuda según contexto desde una herramienta de DB2

La ayuda según contexto proporciona información sobre las tareas o controles que están asociados con una ventana, cuaderno, asistente o asesor determinado. La ayuda según contexto está disponible desde las herramientas de administración y desarrollo de DB2 que tienen interfaces gráficas de usuario. Existen dos tipos de ayuda según contexto:

- Ayuda a la que se accede mediante el botón **Ayuda** ubicado en cada ventana o cuaderno.

- Ventanas emergentes de información, que son ventanas que se visualizan cuando el cursor del ratón se coloca sobre un campo o control o cuando se selecciona un campo o control en una ventana, cuaderno, asistente o asesor y se pulsa F1.

El botón **Ayuda** proporciona acceso a la información de visión general, de prerequisites y de tareas. Las ventanas emergentes de información describen los campos y controles individuales.

Procedimiento:

Para invocar la ayuda según contexto:

- Para la ayuda de ventana y de cuaderno, inicie una de las herramientas de DB2 y, luego, abra cualquier ventana o cuaderno. Pulse el botón **Ayuda** situado en la esquina inferior derecha de la ventana o del cuaderno a fin de invocar la ayuda según contexto.

También puede acceder a la ayuda según contexto desde el elemento de menú **Ayuda** situado en la parte superior de cada uno de los centros de herramientas de DB2.

Para los asistentes y asesores, pulse en el enlace Visión general de tareas, de la primera página, si desea ver ayuda según contexto.

- Para obtener ayuda sobre controles individuales de una ventana o un cuaderno en una ventana emergente de información, pulse el control y, a continuación, pulse F1. La información emergente que contiene detalles sobre el control se visualizará en una ventana amarilla.

Nota: Para visualizar ventanas emergentes de información simplemente manteniendo el cursor del ratón sobre un campo o control, seleccione el recuadro de selección **Visualizar automáticamente ventanas emergentes de información** en la página **Documentación** del cuaderno Valores de herramientas.

Similar a las ventanas emergentes de información, la información emergente de diagnóstico es otra forma de ayuda según contexto; en ella se incluyen reglas para la entrada de datos. La información emergente de diagnóstico se visualiza en una ventana de color morado que aparece cuando se entran datos que no son válidos o que son insuficientes. La información emergente de diagnóstico puede aparecer para:

- Campos obligatorios.
- Campos cuyos datos tengan un formato preciso como, por ejemplo, un campo de fecha.

Tareas relacionadas:

- “Invocación del Centro de información de DB2” en la página 323
- “Invocación de la ayuda de mensajes desde el procesador de línea de mandatos” en la página 334
- “Invocación de la ayuda de mandatos desde el procesador de línea de mandatos” en la página 334
- “Invocación de la ayuda para estados de SQL desde el procesador de línea de mandatos” en la página 335
- “Acceso al Centro de información de DB2”
- “Cómo utilizar la ayuda de DB2 UDB”
- “Establecimiento de la ubicación para acceder al Centro de información de DB2”
- “Configuración del acceso a documentación y ayuda contextual de DB2”

Invocación de la ayuda de mensajes desde el procesador de línea de mandatos

La ayuda de mensajes describe la causa de un mensaje y describe la acción que se debe realizar en respuesta al error.

Procedimiento:

Para invocar la ayuda de mensajes, abra el procesador de línea de mandatos y entre:

? XXXnnnnn

donde XXXnnnnn representa un identificador de mensaje válido.

Por ejemplo, ? SQL30081 muestra la ayuda acerca del mensaje SQL30081.

Conceptos relacionados:

- “Introducción a los mensajes” en el manual *Consulta de mensajes Volumen 1*

Información relacionada:

- “db2: Mandato de invocación del procesador de línea de mandatos” en el manual *Consulta de mandatos*

Invocación de la ayuda de mandatos desde el procesador de línea de mandatos

La ayuda de mandatos explica la sintaxis de los mandatos del procesador de línea de mandatos.

Procedimiento:

Para invocar la ayuda de mandatos, abra el procesador de línea de mandatos y entre:

? mandato

donde *mandato* representa una palabra clave o el mandato completo.

Por ejemplo, ? catalog visualiza ayuda para todos los mandatos CATALOG, mientras que ? catalog database visualiza ayuda solamente para el mandato CATALOG DATABASE.

Tareas relacionadas:

- “Invocación de ayuda según contexto desde una herramienta de DB2” en la página 332
- “Invocación del Centro de información de DB2” en la página 323
- “Invocación de la ayuda de mensajes desde el procesador de línea de mandatos” en la página 334
- “Invocación de la ayuda para estados de SQL desde el procesador de línea de mandatos” en la página 335

Información relacionada:

- “db2: Mandato de invocación del procesador de línea de mandatos” en el manual *Consulta de mandatos*

Invocación de la ayuda para estados de SQL desde el procesador de línea de mandatos

DB2 Universal Database devuelve un valor de SQLSTATE para las condiciones que pueden ser el resultado de una sentencia de SQL. La ayuda de SQLSTATE explica los significados de los estados de SQL y los códigos de las clases de estados de SQL.

Procedimiento:

Para invocar la ayuda para estados de SQL, abra el procesador de línea de mandatos y entre:

? sqlstate o *? código de clase*

donde *sqlstate* representa un estado de SQL válido de cinco dígitos y *código de clase* representa los dos primeros dígitos del estado de SQL.

Por ejemplo, *? 08003* visualiza la ayuda para el estado de SQL 08003, y *? 08* visualiza la ayuda para el código de clase 08.

Tareas relacionadas:

- “Invocación del Centro de información de DB2” en la página 323
- “Invocación de la ayuda de mensajes desde el procesador de línea de mandatos” en la página 334
- “Invocación de la ayuda de mandatos desde el procesador de línea de mandatos” en la página 334

Guías de aprendizaje de DB2

Las guías de aprendizaje de DB2 ayudan a conocer los diversos aspectos de DB2 Universal Database. Las guías de aprendizaje proporcionan ejercicios con instrucciones paso a paso en las áreas de desarrollo de aplicaciones, ajuste del rendimiento de las consultas de SQL, trabajo con depósitos de datos, gestión de metadatos y desarrollo de servicios Web utilizando DB2.

Antes de empezar:

Puede ver las versiones XHTML de las guías de aprendizaje desde el Centro de información en <http://publib.boulder.ibm.com/infocenter/db2help/>.

Algunos ejercicios de las guías de aprendizaje utilizan datos o código de ejemplo. Consulte cada guía de aprendizaje para obtener una descripción de los prerrequisitos para las tareas específicas.

Guías de aprendizaje de DB2 Universal Database:

Pulse en el título de una guía de aprendizaje de la lista siguiente para ver esa guía de aprendizaje.

Guía de aprendizaje de Business Intelligence: Introducción al Centro de depósito de datos
Realizar tareas de introducción de depósito de datos utilizando el Centro de depósito de datos.

Guía de aprendizaje de Business Intelligence: Lecciones ampliadas sobre depósito de datos
Realizar tareas avanzadas de depósito de datos utilizando el Centro de depósito de datos.

Information Catalog Center Tutorial
Crear y gestionar un catálogo de información para localizar y usar metadatos utilizando el Centro de catálogos de información.

Guía de aprendizaje de Visual Explain
Analizar, optimizar y ajustar sentencias de SQL para obtener un mejor rendimiento al utilizar Visual Explain.

Información de resolución de problemas de DB2

Existe una gran variedad de información para la resolución de problemas y la determinación de problemas para ayudarle a utilizar los productos DB2®.

Documentación de DB2

La información de resolución de problemas se puede encontrar en todo el Centro de información de DB2, así como en todos los manuales PDF que componen la biblioteca de DB2. Puede consultar la rama sobre soporte y resolución de problemas, del árbol de navegación del Centro de información de DB2 (en el panel izquierdo de la ventana del navegador), para obtener un listado completo de la documentación de resolución de problemas de DB2.

Sitio Web de soporte técnico de DB2

Consulte el sitio Web de soporte técnico de DB2 si tiene problemas y desea obtener ayuda para encontrar las causas y las soluciones posibles. El sitio de soporte técnico tiene enlaces con las últimas publicaciones de DB2, notas técnicas, Informes autorizados de análisis del programa (APAR), FixPaks y el listado más reciente de códigos de error internos de DB2, además de otros recursos. Puede buscar en esta base de conocimiento para encontrar posibles soluciones a los problemas.

Para acceder al sitio Web de soporte de DB2, vaya a
<http://www.ibm.com/software/data/db2/udb/winos2unix/support>

DB2 Problem Determination Tutorial Series (Serie de guías de aprendizaje para la determinación de problemas de DB2)

Consulte el sitio Web DB2 Problem Determination Tutorial Series para encontrar información sobre cómo identificar y resolver rápidamente los problemas que puedan surgir mientras trabaja con DB2. Una de las guías de aprendizaje ofrece una presentación de los recursos y las herramientas de determinación de problemas de DB2 disponibles y le ayuda a decidir cuándo utilizarlos. Otras de las guías de aprendizaje tratan temas relacionados como, por ejemplo, "Determinación de problemas del motor de base de datos", "Determinación de problemas de rendimiento" y "Determinación de problemas de aplicaciones".

Consulte el conjunto completo de guías de aprendizaje de determinación de problemas de DB2 en el sitio de soporte técnico de DB2 de
<http://www.ibm.com/software/data/support/pdm/db2tutorials.html>

Conceptos relacionados:

- “Centro de información de DB2” en la página 314
- “Introduction to problem determination - DB2 Technical Support tutorial” en la publicación *Troubleshooting Guide*

Accesibilidad

Las características de accesibilidad ayudan a los usuarios con discapacidades físicas, por ejemplo movilidad o visión limitada, a utilizar los productos de software satisfactoriamente. La lista siguiente especifica las características de accesibilidad principales de los productos de DB2® Versión 8:

- Toda la funcionalidad de DB2 está disponible utilizando el teclado para la navegación en lugar del ratón. Si desea más información, consulte el apartado “Entrada de teclado y navegación”.
- Puede personalizar el tamaño y color de los fonts en las interfaces de usuario de DB2. Si desea más información, consulte el apartado “Pantalla accesible”.
- Los productos de DB2 dan soporte a aplicaciones de accesibilidad que utilizan la API de accesibilidad de Java™. Si desea más información, consulte el apartado “Compatibilidad con tecnologías de asistencia” en la página 338.
- La documentación de DB2 se proporciona en un formato accesible. Si desea más información, consulte el apartado “Documentación accesible” en la página 338.

Entrada de teclado y navegación

Entrada de teclado

Puede trabajar con las herramientas de DB2 utilizando solamente el teclado. Puede utilizar teclas o combinaciones de teclas para llevar a cabo operaciones que también se pueden realizar con el ratón. Las pulsaciones estándares del sistema operativo se utilizan para operaciones estándares del sistema operativo.

Para obtener más información sobre el uso de teclas o combinaciones de teclas al realizar operaciones, consulte Accesos directos y aceleradores del teclado.

Navegación de teclado

Puede navegar por la interfaz de usuario de las herramientas de DB2 mediante teclas o combinaciones de teclas.

Para obtener más información sobre el uso de teclas o combinaciones de teclas al navegar por las herramientas de DB2, consulte Accesos directos y aceleradores del teclado.

Foco del teclado

En los sistemas operativos UNIX®, se resalta el área de la ventana activa en la que las pulsaciones tendrán efecto.

Pantalla accesible

Las herramientas de DB2 presentan características que mejoran la accesibilidad de los usuarios con poca visión u otras discapacidades visuales. Estas mejoras de la accesibilidad incluyen soporte para propiedades de font personalizables.

Valores de font

Puede seleccionar el color, tamaño y font del texto en menús y ventanas de diálogo utilizando el cuaderno Valores de herramientas.

Para obtener más información sobre cómo especificar valores de font, consulte [Modificación de fonts para menús y texto](#).

No dependencia del color

No es necesario distinguir los colores para utilizar cualquiera de las funciones de este producto.

Compatibilidad con tecnologías de asistencia

Las interfaces de las herramientas de DB2 dan soporte a la API de accesibilidad de Java, que le permite utilizar lectores de pantalla y otras tecnologías de asistencia con los productos de DB2.

Documentación accesible

La documentación de DB2 se proporciona en formato XHTML 1.0, que se puede visualizar en la mayoría de los navegadores Web. XHTML le permite visualizar la documentación de acuerdo con las preferencias de pantalla establecidas en el navegador. También permite utilizar lectores de pantalla y otras tecnologías de asistencia.

Los diagramas de sintaxis se proporcionan en formato decimal con puntos. Este formato sólo está disponible si se accede a la documentación en línea mediante un lector de pantalla.

Conceptos relacionados:

- “Diagramas de sintaxis en formato decimal con puntos” en la página 338

Tareas relacionadas:

- “Accesos directos y aceleradores del teclado”
- “Modificación de fonts para menús y texto”

Diagramas de sintaxis en formato decimal con puntos

Se proporcionan diagramas de sintaxis en formato decimal con puntos para los usuarios que acceden al Centro de información utilizando un lector de pantalla.

En formato decimal con puntos, cada elemento de sintaxis se escribe en una línea distinta. Si dos o más elementos de sintaxis siempre aparecen juntos (o siempre están ausentes los dos a la vez), pueden aparecer en la misma línea, puesto que se pueden considerar un elemento de sintaxis compuesto.

Cada línea empieza por un número decimal con puntos; por ejemplo, 3 ó 3.1 ó 3.1.1. Para oír estos números correctamente, asegúrese de que su lector de pantalla esté configurado para leer la puntuación. Todos los elementos de sintaxis que tienen el mismo número decimal con puntos (por ejemplo, todos los elementos de sintaxis que tienen el número 3.1) son alternativas mutuamente excluyentes. Si oye las líneas 3.1 USERID y 3.1 SYSTEMID, sabrá que la sintaxis puede incluir o USERID o SYSTEMID, pero no ambos.

El nivel de numeración decimal con puntos denota el nivel jerárquico. Por ejemplo, si un elemento de sintaxis con el número decimal con puntos 3 va seguido de una serie de elementos de sintaxis con el número decimal 3.1, todos los elementos de sintaxis con la numeración 3.1 son subordinados de los elementos de sintaxis identificados por el número 3.

Junto a los números decimales con puntos se utilizan determinados símbolos y palabras para añadir información sobre los elementos de sintaxis. A veces, estos símbolos y palabras pueden aparecer al principio del propio elemento. Para facilitar la identificación, si la palabra o el símbolo forman parte del elemento de sintaxis, van precedidos por una barra inclinada invertida (\). El símbolo * se puede utilizar junto a un número decimal con puntos para indicar que el elemento de sintaxis se repite. Por ejemplo, el elemento de sintaxis *FILE con el número decimal con puntos 3 adopta el formato 3 * FILE. El formato 3* FILE indica que el elemento de sintaxis FILE se repite. El formato 3* * FILE indica que el elemento de sintaxis * FILE se repite.

Los caracteres como las comas, que se utilizan para separar una serie de elementos de sintaxis, se muestran en la sintaxis justo antes de los elementos que separan. Estos caracteres pueden aparecer en la misma línea que cada elemento o en una línea distinta con el mismo número decimal con puntos que los elementos en cuestión. En la línea también puede aparecer otro símbolo que proporcione información sobre los elementos de sintaxis. Por ejemplo, las líneas 5.1*, 5.1 LASTRUN y 5.1 DELETE significan que si se utiliza más de uno de los elementos de sintaxis LASTRUN y DELETE, los elementos deben estar separados por comas. Si no hay ningún separador, suponga que utiliza un espacio en blanco para separar cada elemento de sintaxis.

Si un elemento de sintaxis va precedido del símbolo %, esto indica una referencia que está definida en cualquier otro lugar. La serie que aparece después del símbolo % es el nombre de un fragmento de sintaxis en lugar de un literal. Por ejemplo, la línea 2.1 %OP1 significa que se debe hacer referencia al fragmento de sintaxis separado OP1.

Junto a los números decimales con puntos se utilizan los símbolos y las palabras siguientes:

- ? indica un elemento de sintaxis opcional. Un número decimal con puntos seguido del símbolo ? indica que todos los elementos de sintaxis con un número decimal con puntos correspondiente y elementos de sintaxis subordinados son opcionales. Si sólo hay un elemento de sintaxis con un número decimal con puntos, el símbolo ? aparecerá en la misma línea que el elemento de sintaxis (por ejemplo, 5? NOTIFY). Si hay más de un elemento de sintaxis con un número decimal con puntos, el símbolo ? aparecerá en una línea propia, seguido de los elementos de sintaxis opcionales. Por ejemplo, si oye las líneas 5 ?, 5 NOTIFY y 5 UPDATE, sabrá que los elementos de sintaxis NOTIFY y UPDATE son opcionales; es decir, puede seleccionar uno o ninguno de dichos elementos. El símbolo ? es equivalente a una línea de desvío de un diagrama de vías.
- ! indica un elemento de sintaxis por omisión. Un número decimal con puntos seguido del símbolo ! y un elemento de sintaxis indica que el elemento de sintaxis es la opción por omisión para todos los elementos de sintaxis que comparten el mismo número decimal con puntos. Sólo uno de los elementos de sintaxis que comparten el mismo número decimal con puntos puede especificar un símbolo !. Por ejemplo, si oye las líneas 2? FILE, 2.1! (KEEP) y 2.1 (DELETE), sabrá que (KEEP) es la opción por omisión correspondiente a la palabra clave FILE. En este ejemplo, si incluye la palabra clave FILE pero no especifica ninguna opción, se aplicará la opción por omisión KEEP. También se aplicará una opción por omisión al siguiente número decimal con puntos más alto. En este ejemplo, si se omite la palabra clave FILE, se utiliza el valor por omisión FILE(KEEP). No obstante, si oye las líneas 2? FILE, 2.1, 2.1.1! (KEEP) y 2.1.1 (DELETE), la opción por omisión KEEP sólo se aplicará al siguiente número

decimal con puntos más alto, 2.1 (que no tiene una palabra clave asociada) y no se aplicará a 2? FILE. Si se omite la palabra clave FILE, no se utilizará nada.

- * indica un elemento de sintaxis que se puede repetir 0 o más veces. Un número decimal con puntos seguido del símbolo * indica que este elemento de sintaxis se puede utilizar cero o más veces; es decir, es opcional y se puede repetir. Por ejemplo, si oye la línea 5.1* data area, sabrá que puede incluir un área de datos, más de un área de datos o ningún área de datos. Si oye las líneas 3*, 3 HOST y 3 STATE, sabrá que puede incluir HOST, STATE, los dos juntos o ninguno de los dos.

Notas:

1. Si un número decimal con puntos tiene un asterisco (*) al lado y sólo hay un elemento con dicho número decimal con puntos, podrá repetir el mismo elemento más de una vez.
 2. Si un número decimal con puntos tiene un asterisco al lado y hay varios elementos que tienen dicho número decimal con puntos, podrá utilizar más de un elemento de la lista, pero no podrá utilizar los elementos más de una vez cada uno. En el ejemplo anterior, podría escribir HOST STATE pero no podría escribir HOST HOST.
 3. El símbolo * es equivalente a una línea de bucle de retorno de un diagrama de sintaxis de vías.
- + indica un elemento de sintaxis que se debe incluir una o más veces. Un número decimal con puntos seguido del símbolo + indica que este elemento de sintaxis se debe incluir una o más veces; es decir, se debe incluir como mínimo una vez y se puede repetir. Por ejemplo, si oye la línea 6.1+ data area, deberá incluir como mínimo un área de datos. Si oye las líneas 2+, 2 HOST y 2 STATE, sabrá que debe incluir HOST, STATE o ambos. De manera similar al símbolo *, el símbolo + sólo puede repetir un elemento determinado si éste es el único elemento que tiene el número decimal con puntos en cuestión. El símbolo +, al igual que el símbolo *, es equivalente a una línea de bucle de retorno de un diagrama de sintaxis de vías.

Conceptos relacionados:

- “Accesibilidad” en la página 337

Tareas relacionadas:

- “Accesos directos y aceleradores del teclado”

Información relacionada:

- “Cómo se leen los diagramas de sintaxis” en el manual *Consulta de SQL, Volumen 2*

Certificación Common Criteria de productos DB2 Universal Database

Se está evaluando DB2 Universal Database para obtener la certificación Common Criteria en el nivel de garantía de evaluación 4 (EAL4). Para más información acerca de Common Criteria, consulte el sitio Web de Common Criteria en: <http://niap.nist.gov/cc-scheme/>.

Apéndice B. Avisos

Es posible que IBM no comercialice en todos los países algunos productos, servicios o características descritos en este manual. Consulte al representante local de IBM para obtener información sobre los productos y servicios que actualmente pueden adquirirse en su zona. Cualquier referencia a un producto, programa o servicio de IBM no pretende afirmar ni implicar que sólo se pueda utilizar dicho producto, programa o servicio de IBM. En su lugar se puede utilizar cualquier producto, programa o servicio funcionalmente equivalente que no vulnere ninguno de los derechos de propiedad intelectual de IBM. Sin embargo, es responsabilidad del usuario evaluar y verificar el funcionamiento de cualquier producto, programa o servicio que no sea de IBM.

IBM puede tener patentes o solicitudes de patentes en tramitación que afecten al tema tratado en este documento. La posesión de este documento no confiere ninguna licencia sobre dichas patentes. Puede realizar consultas sobre licencias escribiendo a:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
EE.UU.

Para realizar consultas sobre licencias referentes a información de doble byte (DBCS), puede ponerse en contacto con el Departamento de Propiedad Intelectual de IBM de su país/región o escribir a:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokio 106, Japón

El párrafo siguiente no es aplicable al Reino Unido ni a ningún país/región en donde tales disposiciones sean incompatibles con la legislación local:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROPORCIONA ESTA PUBLICACIÓN "TAL CUAL", SIN GARANTÍA DE NINGUNA CLASE, NI EXPLÍCITA NI IMPLÍCITA, INCLUIDAS, PERO SIN LIMITARSE A ELLAS, LAS GARANTÍAS IMPLÍCITAS DE NO VULNERACIÓN DE DERECHOS, COMERCIALIZACIÓN O IDONEIDAD PARA UN FIN DETERMINADO. Algunos estados no permiten la exclusión de garantías expresas o implícitas en determinadas transacciones, por lo que es posible que esta declaración no sea aplicable en su caso.

Esta publicación puede contener inexactitudes técnicas o errores tipográficos. Periódicamente se efectúan cambios en la información aquí contenida; dichos cambios se incorporarán a las nuevas ediciones de la publicación. IBM puede efectuar, en cualquier momento y sin previo aviso, mejoras y cambios en los productos y programas descritos en esta publicación.

Las referencias hechas en esta publicación a sitios Web que no son de IBM se proporcionan sólo para la comodidad del usuario y no constituyen un aval de esos

sitios Web. La información contenida en esos sitios Web no forma parte de la información del presente producto IBM y el usuario es responsable de la utilización de dichos sitios Web.

IBM puede utilizar o distribuir cualquier información que se le facilite de la manera que considere adecuada, sin contraer por ello ninguna obligación con el remitente.

Los licenciarios de este programa que deseen obtener información sobre él con el fin de habilitar: (i) el intercambio de información entre programas creados de forma independiente y otros programas (incluido éste) y (ii) el uso mutuo de la información intercambiada, deben ponerse en contacto con:

IBM Canada Limited
Office of the Lab Director
8200 Warden Avenue
Markham, Ontario
L6G 1C7
CANADÁ

Dicha información puede estar disponible, sujeta a los términos y condiciones apropiados, incluido en algunos casos el pago de una tarifa.

El programa bajo licencia descrito en este documento y todo el material bajo licencia asociado a él, los proporciona IBM según los términos del Acuerdo de Cliente de IBM, el Acuerdo Internacional de Programas Bajo Licencia de IBM o cualquier acuerdo equivalente entre el usuario e IBM.

Los datos de rendimiento contenidos en este documento se obtuvieron en un entorno controlado. Por lo tanto, los resultados obtenidos en otros entornos operativos pueden variar significativamente. Algunas mediciones pueden haberse realizado en sistemas experimentales y no es seguro que estas mediciones sean las mismas en los sistemas disponibles comercialmente. Además, algunas mediciones pueden haberse calculado mediante extrapolación. Los resultados reales pueden variar. Los usuarios del presente manual deben verificar los datos aplicables para su entorno específico.

La información referente a productos que no son de IBM se ha obtenido de los proveedores de esos productos, de sus anuncios publicados o de otras fuentes disponibles públicamente. IBM no ha probado esos productos y no puede confirmar la exactitud del rendimiento, la compatibilidad ni ninguna otra afirmación referente a productos que no son de IBM. Las preguntas sobre las prestaciones de productos que no son de IBM deben dirigirse a los proveedores de esos productos.

Todas las declaraciones de intenciones de IBM están sujetas a cambio o cancelación sin previo aviso, y sólo representan objetivos.

Este manual puede contener ejemplos de datos e informes que se utilizan en operaciones comerciales diarias. 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 similitud con nombres y direcciones utilizados por una empresa real es totalmente fortuita.

LICENCIA DE COPYRIGHT:

Este manual puede contener programas de aplicaciones de ejemplo escritos en lenguaje fuente, que muestran técnicas de programación en diversas plataformas operativas. Puede copiar, modificar y distribuir estos programas de ejemplo como desee, sin pago alguno a IBM, con la intención de desarrollar, utilizar, comercializar o distribuir programas de aplicaciones de acuerdo con la interfaz de programación de aplicaciones correspondiente a la plataforma operativa para la que están escritos los programas de ejemplo. Estos ejemplos no se han probado exhaustivamente bajo todas las condiciones. Por lo tanto, IBM no puede asegurar ni implicar la fiabilidad, utilidad o función de estos programas.

Cada copia o parte de estos programas de ejemplo o cualquier trabajo derivado debe incluir una nota de copyright como la siguiente:

© (*nombre de la empresa*) (*año*). Partes de este código proceden de programas de ejemplo de IBM Corp. © Copyright IBM Corp. *_entre el o los años_*. Reservados todos los derechos.

Marcas registradas

Los términos siguientes son marcas registradas de International Business Machines Corporation en los EE.UU. y/o en otros países y se han utilizado como mínimo en uno de los documentos de la biblioteca de documentación de DB2 UDB.

ACF/VTAM	iSeriesLAN Distance
AISPO	MVS
AIX	MVS/ESA
AIXwindows	MVS/XA
AnyNet	Net.Data
APPN	NetView
AS/400	OS/390
BookManager	OS/400
C Set++	PowerPC
C/370	pSeries
CICS	QBIC
Database 2	QMF
DataHub	RACF
DataJoiner	RISC System/6000
DataPropagator	RS/6000
DataRefresher	S/370
DB2	SP
DB2 Connect	SQL/400
DB2 Extenders	SQL/DS
DB2 OLAP Server	System/370
DB2 Information Integrator	System/390
DB2 Query Patroller	SystemView
DB2 Universal Database	Tivoli
Distributed Relational Database Architecture	VisualAge
DRDA	VM/ESA
eServer	VSE/ESA
Extended Services	VTAM
FFST	WebExplorer
First Failure Support Technology	WebSphere
IBM	WIN-OS/2
IMS	z/OS
IMS/ESA	zSeries

Los términos siguientes son marcas registradas de otras empresas y se han utilizado como mínimo en uno de los documentos de la biblioteca de documentación de DB2 UDB:

Microsoft, Windows, Windows NT y el logotipo de Windows son marcas registradas de Microsoft Corporation en los EE.UU. y/o en otros países.

Intel y Pentium son marcas registradas de Intel Corporation en los EE.UU. y/o en otros países.

Java y todas las marcas registradas basadas en Java son marcas registradas de Sun Microsystems, Inc. en los EE.UU. y/o en otros países.

UNIX es marca registrada de The Open Group en los EE.UU. y/o en otros países.

Otros nombres de empresas, productos o servicios, pueden ser marcas registradas o marcas de servicio de otras empresas.

Índice

Caracteres Especiales

.NET

- aplicaciones C#
 - creación en Windows 268
 - opciones de compilación y enlace 270
- aplicaciones Visual Basic
 - creación en Windows 272
 - opciones de compilación y enlace 274
- archivos de proceso por lotes 110
- ejemplos de C# 78
- ejemplos de Visual Basic 107
- rutinas
 - creación en Windows 276
 - opciones de compilación y enlace 279

Números

- 32 bits, aplicaciones
 - migración a entornos de 64 bits 56

A

- acceso
 - características 337
 - diagramas de sintaxis decimales con puntos 338
- Actualización
 - documentación HTML 324
- AIX
 - aplicaciones C
 - opciones de compilación y enlace 185
 - aplicaciones C++
 - opciones de compilación y enlace 189
 - aplicaciones C multihebra
 - creación 187
 - aplicaciones COBOL de IBM
 - creación 199
 - opciones de compilación y enlace 201
 - aplicaciones COBOL de Micro Focus
 - opciones de compilación y enlace 207
 - aplicaciones de la API en C++
 - creación con archivos de configuración 194
 - aplicaciones REXX
 - creación 209
 - funciones definidas por el usuario en C++
 - creación con archivos de configuración 197
- Java
 - configurar entorno 37

AIX (continuación)

- procedimientos almacenados en C++
 - creación con archivos de configuración 196
- rutinas C
 - opciones de compilación y enlace 186
- rutinas C++
 - opciones de compilación y enlace 191
- rutinas COBOL de IBM
 - creación 202
 - opciones de compilación y enlace 204
- rutinas COBOL de Micro Focus
 - opciones de compilación y enlace 208
- SQL incorporado en C++
 - creación con archivos de configuración 195

API

- archivos de configuración de C++ de AIX 194

aplicaciones, desarrollo

- configuración de la base de datos de ejemplo 48
- configuración entorno UNIX 34
- configuración entorno Windows 42
- configurar entorno 25
- configurar entorno Java 28
- migración
 - ejecución en dos versiones de DB2 59
- Perl
 - creación de aplicaciones 153
- PHP
 - creación de aplicaciones 155

aplicaciones multihebra

- archivos de creación 110
- creación con C++ de AIX 192
- creación con C++ de HP-UX 222
- creación con C++ de Linux 236
- creación con C++ de Solaris 252
- creación con C/C++ de Windows 283
- creación con C de AIX 187
- creación con C de HP-UX 216
- creación con C de Linux 231
- creación con C de Solaris 247

applets

- creación de JDBC 124
- creación de SQLJ 129
- ejemplos de JDBC 87
- ejemplos de SQLJ 90
- temas para utilizar 122

archivos de configuración

- para VisualAge C++ en AIX 193
- archivos de creación 110
- archivos de proceso por lotes 110
- atajos del teclado
 - soporte de 337

ayuda

- para mandatos
 - invocación 334
- para mensajes
 - invocación 334
- para sentencias de SQL
 - invocación 335
 - visualización 323, 325
- ayuda para mandatos
 - invocación 334

B

base de datos de ejemplo

- catalogación 50
 - configuración 48
 - creación 48
 - creación en sistemas principales 50
 - vinculación 51
- ### bibliotecas compartidas
- rutina de revinculación 27
 - sustitución de AIX 183

C

C

- aplicaciones
 - creación en UNIX 161
 - creación en Windows 283
 - opciones de compilación en AIX 185
 - opciones de compilación en HP-UX 212
 - opciones de compilación en Linux 228
 - opciones de compilación en Solaris 244
 - opciones de compilación en Windows 286
- aplicaciones multihebra
 - AIX 187
 - HP-UX 216
 - Linux 231
 - Solaris 247
 - Windows 283
- archivos de creación 110
- archivos del programa de utilidad de comprobación de errores 116
- ejemplos 71
- makefiles 113
- multiconexión, aplicaciones
 - creación en UNIX 163
 - creación en Windows 292
- rutinas
 - creación en UNIX 165
 - creación en Windows 287
 - opciones de compilación en AIX 186
 - opciones de compilación en HP-UX 215

- C (continuación)
 - rutinas (continuación)
 - opciones de compilación en Linux 230
 - opciones de compilación en Solaris 246
 - opciones de compilación en Windows 291
 - versiones soportadas de AIX 9
 - versiones soportadas de HP-UX 11
 - versiones soportadas de Linux 13
 - versiones soportadas de Solaris 18
 - versiones soportadas de Windows 20
 - C++
 - aplicaciones
 - creación en UNIX 168
 - creación en Windows 283
 - opciones de compilación en AIX 189
 - opciones de compilación en HP-UX 218
 - opciones de compilación en Linux 233
 - opciones de compilación en Solaris 249
 - opciones de compilación en Windows 286
 - aplicaciones multihebra
 - AIX 192
 - HP-UX 222
 - Linux 236
 - Solaris Operating Environment 252
 - Windows 283
 - archivos de configuración VisualAge en AIX 193
 - archivos de creación 110
 - archivos del programa de utilidad de comprobación de errores 116
 - automatización OLE con Visual C++ 282
 - ejemplos 75
 - makefiles 113
 - multiconexión, aplicaciones
 - creación en UNIX 170
 - creación en Windows 292
 - rutinas
 - creación en UNIX 172
 - creación en Windows 287
 - opciones de compilación en AIX 191
 - opciones de compilación en HP-UX 220
 - opciones de compilación en Linux 235
 - opciones de compilación en Solaris 251
 - opciones de compilación en Windows 291
 - versiones soportadas de AIX 9
 - versiones soportadas de HP-UX 11
 - versiones soportadas de Linux 13
 - versiones soportadas de Solaris 18
 - versiones soportadas de Windows 20
 - C# .NET (continuación)
 - aplicaciones (continuación)
 - opciones de compilación y enlace 270
 - archivos de proceso por lotes 110
 - ejemplos 78
 - versiones soportadas de Windows 20
 - catalogación
 - base de datos de ejemplo 50
 - Centro de desarrollo
 - soporte de DB2 AD Client 3
 - Centro de información
 - instalación 316, 318, 321
 - Centro de información de DB2
 - 314
 - invocación 323
 - CLI (interfaz a nivel de llamada)
 - archivos de programas de ejemplo 79
 - CLI de DB2
 - archivos de programas de ejemplo 79
 - CLR (ejecución en lenguaje común)
 - rutinas
 - creación en Windows 276
 - opciones de compilación y enlace 279
 - COBOL, lenguaje
 - AIX
 - compilador de IBM 198
 - compilador de Micro Focus 205
 - instalar y ejecutar en 183
 - aplicaciones COBOL de IBM
 - creación en AIX 199
 - creación en Windows 296
 - opciones de compilación en AIX 201
 - opciones de compilación en Windows 299
 - aplicaciones de Micro Focus
 - creación en UNIX 176
 - creación en Windows 304
 - opciones de compilación en AIX 207
 - opciones de compilación en HP-UX 224
 - opciones de compilación en Linux 239
 - opciones de compilación en Solaris 254
 - opciones de compilación en Windows 306
 - archivos de creación 110
 - archivos del programa de utilidad de comprobación de errores 116
 - ejemplos 82
 - HP-UX
 - utilización del compilador de Micro Focus 223
 - Linux
 - compilador de Micro Focus 237
 - makefiles 113
 - rutinas COBOL de IBM
 - creación en AIX 202
 - creación en Windows 300
 - opciones de compilación en AIX 204
 - COBOL, lenguaje (continuación)
 - rutinas COBOL de IBM (continuación)
 - opciones de compilación en Windows 302
 - rutinas de Micro Focus
 - creación en UNIX 177
 - creación en Windows 307
 - opciones de compilación en AIX 208
 - opciones de compilación en HP-UX 225
 - opciones de compilación en Linux 240
 - opciones de compilación en Solaris 256
 - opciones de compilación en Windows 309
 - Solaris Operating Environment
 - compilador de Micro Focus 253
 - versiones soportadas de AIX 9
 - versiones soportadas de HP-UX 11
 - versiones soportadas de Linux 13
 - versiones soportadas de Solaris 18
 - versiones soportadas de Windows 20
 - Windows
 - compilador de IBM 295
 - compilador de Micro Focus 303
 - compartidas, bibliotecas
 - rutina de revinculación 27
 - sustitución de AIX 183
 - compiladores
 - crear archivos para 110
 - makefiles para 113
 - utilización de COBOL de IBM en AIX 198
 - utilización de COBOL de IBM en Windows 295
 - utilización de COBOL de Micro Focus en AIX 205
 - utilización de COBOL de Micro Focus en HP-UX 223
 - utilización de COBOL de Micro Focus en Solaris 253
 - utilización de COBOL de Micro Focus en Windows 303
 - versiones soportadas de AIX 9
 - versiones soportadas de HP-UX 11
 - versiones soportadas de Linux 13
 - versiones soportadas de Solaris 18
 - versiones soportadas de Windows 20
 - comprobación de errores
 - archivos de programas de utilidad 116
 - configuración dinámica
 - ejemplos 86
 - Conformidad con SQL 92 y MVS, indicador
 - soporte de DB2 AD Client 3
 - copia de seguridad
 - procedimientos de SQL 151
 - CREATE, sentencia
 - y rutinas de AIX 182
 - CREATE PROCEDURE, sentencia
 - con procedimientos de SQL 147
- aplicaciones
 - creación en Windows 268

D

DB2 Personal Developer's Edition ix
DB2 Universal Developer's Edition ix
DB2INSTANCE, variable de entorno 48
DB2INSTPROF
 y gestor de bases de datos 5
DB2PATH
 y gestor de bases de datos 5
diagramas de sintaxis decimales con
 puntos 338
discapacidad 337
documentación
 visualización 323

E

ejemplos
 archivos de programa 65
 C 71
 C++ 75
 COBOL 82
 configuración dinámica 86
 de salida de usuario de gestión de
 anotaciones 94
 Java WebSphere 93
 JDBC 87
 lenguajes soportados 65
 Object Linking and Embedding 96
 funciones de tabla de base de
 datos 97
 Perl 97
 PHP 98
 plugin de Java 93
 Procesador de línea de mandatos
 (CLP) 82
 programas
 directorios de ejemplo de Java
 para 121
 SQLJ 90
en línea
 ayuda, acceso 332
entorno
 desarrollo de aplicaciones
 configurar 25
extensiones de archivo
 ejemplos 65
EXTERNAL NAME, cláusula
 CREATE, sentencia 182

F

FORTTRAN, lenguaje
 soporte de DB2 8
funciones de tabla
 Object Linking and Embedding
 ejemplos 97
 OLE DB 258
funciones definidas por el usuario (UDF)
 archivos de configuración de C++ de
 AIX 197
 automatización OLE con Visual
 Basic 267
 automatización OLE con Visual
 C++ 282

G

gestión de archivos de anotaciones
 archivos de ejemplo de salida de
 usuario 94
gestor de bases de datos
 instancias 5
guías de aprendizaje 335
 resolución y determinación de
 problemas 336
guías de aprendizaje de DB2 335

H

HP-UX
 aplicaciones C
 opciones de compilación y
 enlace 212
 aplicaciones C++
 opciones de compilación y
 enlace 218
 aplicaciones C multihebra
 creación 216
 aplicaciones COBOL de Micro Focus
 opciones de compilación y
 enlace 224
Java
 configurar entorno 38
rutinas C
 opciones de compilación y
 enlace 215
rutinas C++
 opciones de compilación y
 enlace 220
rutinas COBOL de Micro Focus
 opciones de compilación y
 enlace 225
HTML, documentación
 actualización 324

I

impresión
 archivos PDF 331
instalación
 Centro de información 316, 318, 321
instancias
 gestor de bases de datos 5
interfaz a nivel de llamada (CLI)
 soporte de DB2 AD Client 3
invocación
 ayuda para mandatos 334
 mensajes, ayuda para 334
 SQL, ayuda para sentencias de 335

J

Java
 applets, temas para utilizar 122
 archivos de ejemplo de plugin 93
 archivos de ejemplo de
 WebSphere 93
 configuración entorno UNIX 35
 configurar entorno 28
 configurar entorno AIX 37
 configurar entorno HP-UX 38

Java (continuación)

 creación
 aplicaciones JDBC 126
 aplicaciones SQLJ 128, 131
 applets de JDBC 124
 applets de SQLJ 129
 creación de rutinas JDBC 126
 creación de rutinas SQLJ 136
 de ejemplo
 directorios 121
 ejemplos de JDBC 87
 ejemplos de SQLJ 90
 JDK de AIX soportados 9
 JDK de HP-UX soportados 11
 JDK de Linux soportados 13
 JDK de Solaris soportados 18
 Linux
 configurar entorno 40
 makefiles 113
 migración de aplicaciones 54
 Solaris Operating Environment
 configurar 41
 soporte de DB2 AD Client 3
 Windows
 configurar 45
 versiones JDK soportadas 20
JDBC (conectividad de base de datos
Java)
 applets, temas para utilizar 122
 creación de aplicaciones 126
 creación de applets 124
 creación de rutinas 126
 ejemplos 87
 soporte de DB2 AD Client 3
JDK_PATH, palabra clave de
 configuración del Gestor de bases de
 datos 27

K

KEEPFENCED, palabra clave de
 configuración del Gestor de bases de
 datos 27
Kerberos
 protocolos de seguridad
 ejemplos 102

L

lenguaje REXX
 creación de aplicaciones en AIX 209
 creación de aplicaciones para
 Windows 309
 ejemplos 100
 soporte de DB2 8
 versiones soportadas de AIX 9
 versiones soportadas de Windows 20
Linux
 aplicaciones C
 opciones de compilación y
 enlace 228
 aplicaciones C++
 opciones de compilación y
 enlace 233
 aplicaciones C multihebra
 creación 231

- Linux *(continuación)*
 - aplicaciones COBOL de Micro Focus
 - opciones de compilación y enlace 239
 - Java
 - configurar entorno 40
 - Micro Focus COBOL
 - configuración del compilador 237
 - rutinas C
 - opciones de compilación y enlace 230
 - rutinas C++
 - opciones de compilación y enlace 235
 - rutinas COBOL de Micro Focus
 - opciones de compilación y enlace 240

LL

- llamada a procedimientos de SQL
 - aplicaciones cliente 148

M

- makefiles 113
- Management Instrumentation de Windows 259
 - ejemplos 109
- manuales impresos, solicitud 332
- mensajes, ayuda para invocación 334
- Microsoft Transaction Server
 - ejemplos de Visual Basic 105
- migración
 - aplicaciones 53
 - ejecución en dos versiones de DB2 59
 - entornos de 32 bits a entornos de 64 bits 56
 - aplicaciones de Java 54
 - applets de Java 54
 - portabilidad de las aplicaciones 58
 - rutinas de Java 54
- MQ, funciones definidas por el usuario configurar 30
- multiconexión, aplicaciones
 - archivos de creación 110
 - creación de C/C++ de Windows 292
 - creación en C++ de UNIX 170
 - creación en C de UNIX 163

N

- NOCONVERT, opción 257

O

- Object Linking and Embedding
 - automatización
 - con Visual Basic 267
 - con Visual C++ 282
 - ejemplos 96
 - funciones de tabla de base de datos
 - archivos de ejemplo 97

- Object Linking and Embedding *(continuación)*
 - funciones de tabla de base de datos *(continuación)*
 - descripción 258
 - soporte de DB2 AD Client 3
- Object REXX para Windows 309
 - ejemplos 100
- objetos de datos ActiveX
 - creación con Visual Basic 260
 - creación con Visual C++ 281
 - ejemplos de Visual Basic 105
 - ejemplos de Visual C++ 108
 - soporte de DB2 AD Client 3
- Objetos de datos remotos
 - creación con Visual Basic 266
 - ejemplos de Visual Basic 105

P

- Perl
 - creación de aplicaciones 153
 - ejemplos 97
 - soporte de DB2 8
- PHP
 - creación de aplicaciones 155
 - ejemplos 98
 - soporte de DB2 8
- plugins
 - ejemplos de Java 93
 - ejemplos de seguridad 102
- portabilidad
 - en la migración de aplicaciones 58
- precompilación
 - procedimientos de SQL 149
- precompiladores
 - soporte de DB2 AD Client 3
- problemas, determinación
 - guías de aprendizaje 336
 - información en línea 336
- procedimientos almacenados
 - archivos de configuración de C++ de AIX 196
 - automatización OLE con Visual Basic 267
 - automatización OLE con Visual C++ 282
 - sentencia CALL 144
- procedimientos de SQL
 - aplicaciones cliente 148
 - archivos de programas de ejemplo 102
 - copia de seguridad y restauración 151
 - creación 147
 - opciones de vinculación y precompilación 149
 - revinculación 152
 - sentencia CALL 144
- procesador de línea de mandatos (CLP)
 - archivos de ejemplo 82
 - ejecución de scripts 143
 - soporte de DB2 AD Client 3
- programas
 - ejemplos 65
- programas de salida de usuario
 - archivos de ejemplo 94

- proveedor OLE DB
 - con Visual Basic 260
 - con Visual C++ 281
- publicaciones de DB2
 - impresión de archivos PDF 331
- puntos de entrada para rutinas, AIX 181

R

- resolución de problemas
 - guías de aprendizaje 336
 - información en línea 336
- restauración
 - procedimientos de SQL 151
- revinculación
 - procedimientos de SQL 152
- rutinas
 - archivos de creación 110
 - archivos de programas de ejemplo
 - procedimientos de SQL 102
 - cargar un biblioteca compartida COBOL en AIX 183
 - puntos de entrada de AIX para 181
 - revinculación de bibliotecas compartidas 27
 - sentencia CREATE en AIX 182

S

- scripts
 - ejecución del Procesador de línea de mandatos (CLP) 143
- seguridad
 - ejemplos 102
- sentencia CALL
 - Procesador de línea de mandatos 144
- servidores
 - soportado por DB2 7
- sistemas operativos
 - soportado por DB2 8
 - versiones soportadas de AIX 9
 - versiones soportadas de HP-UX 11
 - versiones soportadas de Linux 13
 - versiones soportadas de Solaris 18
 - versiones soportadas de Windows 20
 - vías de acceso de instalación de DB2 para 53
- sistemas principales
 - creación de bases de datos de ejemplo 50
 - servidores soportados 7
- Solaris Operating Environment
 - aplicaciones
 - opciones de compilación y enlace en C 244
 - opciones de compilación y enlace en C++ 249
 - aplicaciones C multihebra
 - creación 247
 - aplicaciones COBOL de Micro Focus
 - opciones de compilación y enlace 254
 - configuración de Java 41
 - rutinas
 - opciones de compilación y enlace en C 246

- Solaris Operating Environment
(*continuación*)
 - rutinas (*continuación*)
 - opciones de compilación y enlace en C++ 251
 - rutinas COBOL de Micro Focus
 - opciones de compilación y enlace 256
- solicitud de manuales de DB2 332
- SQL, ayuda para sentencias de invocación 335
- SQL incorporado
 - archivos de configuración de C++ de AIX 195
 - soporte de DB2 AD Client 3
- SQLJ
 - aplicaciones
 - opciones de compilación en UNIX 133
 - opciones de compilación en Windows 135
 - rutinas
 - opciones de compilación en UNIX 139
 - opciones de compilación en Windows 141
- SQLJ (SQL incorporado para Java)
 - aplicaciones
 - creación 131
 - applets
 - creación 129
 - applets temas para utilizar 122
 - archivos de creación 110
 - creación de rutinas 136
 - ejemplos 90
 - programas
 - creación 128
 - soporte de DB2 AD Client 3

T

- transacciones emparejadas débilmente
 - ejemplos de Visual Basic 105
 - Visual Basic
 - creación en Windows 262
 - resolución de problemas 264

U

- UNIX
 - aplicaciones C
 - creación 161
 - aplicaciones C++
 - creación 168
 - aplicaciones COBOL de Micro Focus
 - creación 176
 - aplicaciones SQLJ
 - opciones de compilación 133
- C
 - creación de aplicaciones
 - multiconexión 163
- C++
 - creación de aplicaciones
 - multiconexión 170
 - configuración de Java 35

- UNIX (*continuación*)
 - desarrollo de aplicaciones
 - configurar 34
 - valores de variables de entorno 34
 - rutinas C
 - creación 165
 - rutinas C++
 - creación 172
 - rutinas COBOL de Micro Focus
 - creación 177
 - rutinas SQLJ
 - opciones de compilación 139

V

- variables de entorno
 - UNIX 34
- vinculación
 - procedimientos de SQL 149
- vinculación, programas de utilidad
 - base de datos de ejemplo 51
- Visual Basic
 - automatización OLE 267
 - creación de aplicación RDO 266
 - creación de aplicaciones ADO 260
 - ejemplos 105
 - transacciones emparejadas débilmente
 - creación en Windows 262
 - resolución de problemas 264
 - versiones soportadas de Windows 20
- Visual Basic. NET
 - aplicaciones
 - opciones de compilación y enlace 274
 - archivos de proceso por lotes 110
 - creación de aplicaciones 272
 - ejemplos 107
- Visual C++
 - automatización OLE 282
 - creación de aplicaciones ADO 281
 - ejemplos 108

W

- wchar_t, tipo de datos
 - convert, opción de precompilación 257
- WCHARTYPE CONVERT
 - opción del precompilador 257
- WebSphere MQ, funciones definidas por el usuario
 - configurar 30
- Windows
 - aplicaciones C/C++
 - creación 283
 - opciones de compilación y enlace 286
 - aplicaciones COBOL de IBM
 - creación 296
 - opciones de compilación y enlace 299
 - aplicaciones COBOL de Micro Focus
 - creación 304
 - opciones de compilación y enlace 306

- Windows (*continuación*)
 - aplicaciones SQLJ
 - opciones de compilación 135
 - desarrollo de aplicaciones
 - configurar entorno 42
 - Java
 - configurar 45
 - Management Instrumentation 259
 - ejemplos 109
 - rutinas C/C++
 - creación 287
 - opciones de compilación y enlace 291
 - rutinas COBOL de IBM
 - creación 300
 - opciones de compilación y enlace 302
 - rutinas COBOL de Micro Focus
 - creación 307
 - opciones de compilación y enlace 309
 - rutinas SQLJ
 - opciones de compilación 141

Cómo ponerse en contacto con IBM

En los EE.UU., puede ponerse en contacto con IBM llamando a uno de los siguientes números:

- 1-800-IBM-SERV (1-800-426-7378) para servicio al cliente
- 1-888-426-4343 para obtener información sobre las opciones de servicio técnico disponibles
- 1-800-IBM-4YOU (426-4968) para marketing y ventas de DB2

En Canadá, puede ponerse en contacto con IBM llamando a uno de los siguientes números:

- 1-800-IBM-SERV (1-800-426-7378) para servicio al cliente
- 1-800-465-9600 para obtener información sobre las opciones de servicio técnico disponibles
- 1-800-IBM-4YOU (1-800-426-4968) para marketing y ventas de DB2

Para localizar una oficina de IBM en su país o región, consulte IBM Directory of Worldwide Contacts en el sitio Web <http://www.ibm.com/planetwide>

Información sobre productos

La información relacionada con productos DB2 Universal Database se encuentra disponible por teléfono o a través de la World Wide Web en el sitio <http://www.ibm.com/software/data/db2/udb>

Este sitio contiene la información más reciente sobre la biblioteca técnica, pedidos de manuales, descargas de productos, grupos de noticias, FixPaks, novedades y enlaces con recursos de la Web.

Si vive en los EE.UU., puede llamar a uno de los números siguientes:

- 1-800-IBM-CALL (1-800-426-2255) para solicitar productos u obtener información general.
- 1-800-879-2755 para solicitar publicaciones.

Para obtener información sobre cómo ponerse en contacto con IBM desde fuera de los EE.UU., vaya a la página IBM Worldwide en el sitio www.ibm.com/planetwide



Número Pieza: CT2TUES

SC10-3733-01



(1P) P/N: CT2TUES



Spine information:



IBM® DB2 Universal Database™

Creación y ejecución de aplicaciones

Versión 8.2