
CUBRID 2008 R3.1

릴리스 노트

목차

1. 개요	4
릴리스 노트 정보	4
릴리스 노트 개정 내역	4
참고 문서	4
버그 리포트 및 사용자 피드백 제공 방법 안내	4
추가 정보 안내	4
2. CUBRID 2008 R3.1 정보	5
CUBRID 2008 R3.1 릴리스 특징	5
지원 플랫폼 및 설치 권장 사양	5
라이선스 안내	5
버전 호환성과 운용성	6
응용 프로그램 및 CUBRID 매니저의 호환성	6
데이터베이스 호환성	6
상호 운용성	6
CUBRID 2008 R3.1의 설치 방법	6
Linux에서 설치	6
Windows에서 설치	7
CUBRID 환경 변수 및 OS 환경 변수 설정	7
CUBRID 2008 R3.1로 업그레이드하는 방법	7
2008 R3.1 Beta 버전을 사용 중인 경우 업그레이드 방법 <i>New</i>	7
업그레이드 주의 사항	7
데이터베이스 마이그레이션 절차	8
HA 환경에서 데이터베이스 마이그레이션 절차	9
복제 기능 사용 시 주의사항	10
3. CUBRID 2008 R3.1에서 변경된 사항	11
새로 추가된 기능	11
CUBRIDSUS-3652 BLOB/CLOB 타입의 추가 및 관련 API 지원	11
CUBRIDSUS-434 JDBC/CCI에서 Foreign Key 정보를 얻을 수 있는 기능 추가	12
CUBRIDSUS-3788 ODBC 드라이버에 스키마 정보를 얻을 수 있는 기능 추가	12
CUBRIDSUS-4091 ODBC 드라이버에 UTF-8 문자셋 지원 <i>New</i>	12
CUBRIDSUS-3879, 4183 브로커의 상태 정보 출력 기능에 브로커 응용 서버(CAS) 정보 포함 옵션 추가 <i>New</i>	12
개선된 기능	13
CUBRIDSUS-3761 CCI에서 응용과 브로커 간 네트워크 상태를 확인하는 방식 개선	13
CUBRIDSUS-3343 브로커와 JDBC 응용 프로그램 간 최대 응답 시간을 줄이도록 개선 <i>New</i>	13
CUBRIDSUS-4027 브로커의 ACCESS_MODE를 동적으로 변경할 수 있도록 개선	13
CUBRIDSUS-3835 보존할 보관 로그 파일의 최대 개수를 동적으로 변경할 수 있도록 개선 <i>New</i>	14
CUBRIDSUS-3771 deadlock 및 lock timeout 발생 시 데이터베이스 서버의 에러 로그 메시지 개선 <i>New</i>	14
CUBRIDSUS-4011 Debian 패키지 빌드가 가능하도록 개선	14
수정된 사항	14
CUBRIDSUS-3698 IF 함수의 인자에 NULL이 주어지면 항상 NULL을 반환하는 오류 수정	14
CUBRIDSUS-4047 RIGHT OUTER JOIN이 MERGE 조인으로 실행될 경우 결과가 잘못 나오는 오류 수정	14
CUBRIDSUS-4193 LIKE 조건식에서 지정한 escape 문자가 검색 문자열 내에 존재하면 발생하는 오류 수정 <i>New</i>	15
CUBRIDSUS-4176 계층 질의문에서 출력할 컬럼의 타입이 CHAR 혹은 NUMERIC이면 발생하는 오류 수정 <i>New</i>	15
CUBRIDSUS-4177 계층 질의문에서 INLINE VIEW 외부에 WHERE 조건이 있을 경우 잘못된 결과를 출력하는 오류 수정 <i>New</i>	15
CUBRIDSUS-4178 계층 질의문에서 ORDER SIBLINGS BY 정렬이 잘못되는 오류 수정 <i>New</i>	16
CUBRIDSUS-4004 기본 키 생성 실패 후 COMMIT을 수행하면 해당 테이블 조회 시 에러가 발생할 수 있는 오류 수정	16
CUBRIDSUS-3538 메타 데이터를 얻어오는 JDBC 메소드에서 ArrayIndexOutOfBoundsException 예외가 발생하는 오류 수정	16

CUBRIDSUS-3991 HA 환경에서 TRUNCATE TABLE 질의 수행 시 복제가 중단되는 오류 수정	17
CUBRIDSUS-3911 HA 환경에서 온라인 백업 후 보관 로그 동기화 오류로 인해 복제가 중단되는 오류 수정	17
CUBRIDSUS-3910 HA 환경에서 applylogdb 프로세스가 재시작 후 비정상 종료되면서 데이터 불일치가 발생하는 오류 수정	17
CUBRIDSUS-4295 HA 환경에서 인덱스 이름만을 명시한 DROP INDEX 구문이 슬레이브 DB에 반영되지 않는 오류 수정 New	17
CUBRIDSUS-4347 HA 환경에서 DB 연결을 설정하는 중에 발생하는 오류 수정 New	17
CUBRIDSUS-3804 다중 동시 접속 및 삽입/삭제 질의 수행 시 특정 조건에서 연결이 종료되지 않는 오류 수정	17
CUBRIDSUS-4272 -327번 에러가 발생할 수 있는 오류 수정 New	18
CUBRIDSUS-4218 다량의 페이지를 가지는 내부 파일 생성 요구가 실패하는 오류 수정 New	18
CUBRIDSUS-4217 문자열 중간에 NULL이 포함된 데이터를 내보내기(unload)하는 경우 데이터 파일이 잘못 생성되는 오류 수정 New	18
CUBRIDSUS-4084 64bit Windows 시스템에 32bit CUBRID 설치 시 ODBC 드라이버가 등록되지 않는 오류 수정	18
CUBRIDSUS-3926 Windows의 명령 프롬프트 창에서 CUBRID 제어 시 정상 수행되지 않는 오류 수정 New	18
CUBRIDSUS-4015 영문 Windows에서 CUBRID Service Tray와 관련된 오류 메시지 출력 시 한글이 출력되는 오류 수정	18
4. CUBRID 매니저 2008 R3.1에서 변경된 사항	19
새로 추가된 기능	19
HA 모니터링 대시보드 기능 추가	19
BLOB, CLOB, 길이가 100 이상인 BIT 타입의 입력 기능 추가	19
정의된 SQL 실행 기능 추가	19
DDL/DML을 클립보드로 복사하는 기능 추가	20
개선된 기능	20
데이터 올리기/내려받기 할 때 문자 집합 설정 및 확인할 수 있도록 개선	20
질의 편집기에서 현재 사용중인 문자 집합을 확인할 수 있도록 개선	21
자동 볼륨 추가 설정 시 최대값을 사용자가 설정 할 수 있도록 개선	21
DELETE ALL, TRUNCATE TABLE 등을 여러 개의 테이블에 한번에 적용할 수 있도록 개선	21
브로커 상태 모니터 차트에서 SESSION과 ACTIVE SESSION이 구분되도록 개선	22
다중 네트워크 사용 환경에서 데이터베이스 서버 접속에 실패하는 문제 개선 New	22
수정된 사항	23
호스트 상태 모니터에서 메모리 정보 출력 시 물리적 메모리 외의 영역을 포함하는 오류 수정	23
Windows XP 서비스팩 3에서 CUBRID 매니저로 데이터베이스 시작 시 발생 가능한 오류 수정	23
Windows 7에서 데이터베이스 상태 정보의 일부를 확인할 수 없는 오류 수정	23
문법에 맞지 않는 일부 질의에 대해 에러가 출력되지 않는 오류 수정	23
5. 주의 사항	24
새로 추가된 주의 사항	24
CUBRIDSUS-3826 GLO 클래스 지원 중단에 따른 주의 사항	24
CUBRIDSUS-4172 BLOB, CLOB 타입 사용 시 제약 사항	24
CUBRIDSUS-3926 Windows에서 브로커 파라미터의 동적 변경 제약 사항	24
CUBRIDSUS-4186 Windows Vista 이상 버전에서 CUBRID 유틸리티를 사용한 서비스 제어 시 권장 사항 Updated	24
기존 주의 사항	25
CUBRIDSUS-3217 JDBC에서 연결 정보를 URL 스트링으로 입력하는 경우 물음표를 반드시 명시	25
CUBRIDSUS-3564 마스터 프로세스와 서버 프로세스 간 프로토콜 변경 및 두 개 버전을 동시에 운영하는 경우 포트 설정 필요	25
CUBRIDSUS-2828 데이터베이스 이름에 @를 포함할 수 없음	25
CUBRIDSUS-3267 Windows 환경에서 디렉터리 경로 설정 시 주의 사항	26
CUBRIDSUS-3553 CUBRID 소스 빌드 후 실행 시, 매니저 서버 프로세스 관련 오류 발생	26
CUBRID 매니저에서 [볼륨 자동 추가 기능 사용] 옵션을 선택하고 DB를 생성하는 경우 주의 사항	26

1. 개요

릴리스 노트 정보

본 문서는 CUBRID 2008 R3.1 버전에 관한 유용한 정보를 포함한다. 릴리스 노트의 최신 버전은 CUBRID 오픈 소스 프로젝트 사이트(<http://dev.naver.com/projects/cubrid>)에서 확인할 수 있다.

릴리스 노트 개정 내역

CUBRID 2008 R3.1 버전의 릴리스 이후 릴리스 노트의 변경 사항은 아래와 같다.

작성 날짜	설명
2010년 11월	CUBRID 2008 R3.1 Beta 릴리스
2010년 12월	CUBRID 2008 R3.1 릴리스 (Beta 버전 이후 추가 및 변경된 사항 뒤 New 또는 Updated 추가)

참고 문서

CUBRID 2008 R3.1 제품과 함께 배포되는 문서는 아래와 같다.

문서	설명
릴리스 노트	CUBRID 릴리스 버전의 특징 및 이전 버전에서 변경된 사항과 관련된 정보를 포함한다.
매뉴얼	Quick Start Guide, CUBRID Architecture, SQL 설명서, 튜닝 안내서, 관리자 안내서, CUBRID 매니저 안내서, API References를 포함한다.

버그 리포트 및 사용자 피드백 제공 방법 안내

CUBRID 프로젝트에서는 사용자의 거침없는 버그 리포트와 솔직한 피드백을 기다리고 있으며, 아래 사이트에서 등록할 수 있다.

문서	설명
버그 리포트	CUBRID 오픈 소스 프로젝트: http://dev.naver.com/projects/cubrid/issue
사용자 피드백	CUBRID 오픈 소스 프로젝트: http://dev.naver.com/projects/cubrid/forum CUBRID 공식 사이트: http://www.cubrid.com

추가 정보 안내

CUBRID에 관한 유용한 정보는 아래 사이트에서 찾을 수 있다.

정보	사이트
CUBRID 제품 정보	http://cubrid.com/zbxe/product
CUBRID 라이선스 정보	http://cubrid.com/zbxe/bbs_oss_guide/32249
CUBRID 사용자 문서	http://cubrid.com/zbxe/developer
CUBRID 교육 서비스	http://cubrid.com/zbxe/education_overview

2. CUBRID 2008 R3.1 정보

CUBRID 2008 R3.1 릴리스 특징

CUBRID 2008 R3.1 릴리스는 이미지 또는 긴 텍스트 등의 Large Object를 저장하는 BLOB, CLOB 타입을 추가하고, 이를 사용하는 JDBC 표준 메소드 및 CCI API, SQL 함수를 추가하였다. 또한, 서비스 운영 모니터링의 편의성을 위하여 CUBRID 매니저 유틸리티에 HA 상태 모니터링 기능을 추가하였다.

CUBRID 2008 R3.1 버전의 주요 특징은 다음과 같다.

- **BLOB, CLOB 타입 추가**

이미지 또는 긴 텍스트 등의 Large Object를 저장할 수 있는 BLOB, CLOB 타입을 지원한다. BLOB, CLOB 데이터는 데이터베이스 볼륨이 아닌 외부 파일 시스템에 파일 형태로 저장된다. JDBC 표준 메소드인 getBlob/setBlob, getClob/setClob을 지원하며 CLOB_FROM_FILE, BLOB_FROM_FILE 등의 SQL 함수를 지원하여 응용 개발에 편의성을 제공한다.

기존의 GLO 클래스를 이용하여 Large Object를 저장하는 방법은 이번 버전부터 지원하지 않는다. GLO 클래스 지원 중단에 따른 주의 사항은 [CUBRIDSUS-3826](#)을 참고한다.

- **HA 상태 모니터링 기능 추가**

CUBRID 매니저의 HA 상태 모니터링 기능을 통해 데이터베이스 서버, 브로커, HA의 운영 상태 및 시스템 자원을 감시할 수 있도록 하여 데이터베이스의 운영 편의성을 보다 강화하였다.

보다 자세한 내용은 아래의 [CUBRID 2008 R3.1에서 변경된 사항](#)을 참고한다.

지원 플랫폼 및 설치 권장 사양

CUBRID 2008 R3.1 버전이 지원하는 플랫폼과 설치를 위한 하드웨어/소프트웨어 요구 사항은 아래 표와 같다.

지원 플랫폼	메모리 크기	디스크 크기	필요 소프트웨어
Windows 계열 32/64 Bit (XP, 2003, Vista)	1G 이상	500MB 이상	- Microsoft Visual C++ 2008 또는 2008 SP1 재배포 가능 패키지 - JRE 또는 JDK 1.6 이상 CUBRID 매니저 또는 Java 저장 프로시저를 사용하는 경우 필요 - .NET Framework 2.0 이상 CUBRID Service Tray를 사용하는 경우 필요
Linux 계열 32/64 Bit (Linux kernel 2.4 및 glibc 2.3.4 이상)	1G 이상	500MB 이상	- JRE 또는 JDK 1.6 이상 CUBRID 매니저 또는 Java 저장 프로시저를 사용하는 경우 필요

Microsoft Visual C++ 2008 SP1 재배포 가능 패키지는 아래의 링크에서 언어별 다운로드 가능 (언어 확인 필요)

x86용 <http://www.microsoft.com/downloads/details.aspx?FamilyID=a5c84275-3b97-4ab7-a40d-3802b2af5fc2>

x86_64용 <http://www.microsoft.com/downloads/details.aspx?FamilyID=ba9257ca-337f-4b40-8c14-157cfdffee4e>

라이선스 안내

CUBRID의 서버 엔진에는 GNU GPL v2 or later 가 적용되고 CUBRID 매니저 및 인터페이스(API)에는 BSD 라이선스가 적용된다. 보다 상세한 정보는 CUBRID 공식 사이트의 [라이선스 가이드](#)를 참고한다.

버전 호환성과 운용성

응용 프로그램 및 CUBRID 매니저의 호환성

- 이전 버전의 JDBC, PHP, CCI API 등을 사용하는 응용 프로그램은 CUBRID 2008 R3.1 데이터베이스에 접근할 수 있다. 다만, JDBC, PHP, CCI 인터페이스에 추가/개선된 기능을 사용하기 위해서는 CUBRID 2008 R3.1 버전의 라이브러리를 링크하여야 한다.
- CUBRID 매니저는 CUBRID 2008 R2.1 이상 버전의 서버에 대해서 하위 호환성을 보장하며, 각 서버 버전과 일치하는 CUBRID JDBC 라이브러리를 링크하여야 한다. CUBRID JDBC 드라이버는 CUBRID 설치 시 \$CUBRID/jdbc 디렉터리에 포함되어 있다.¹ (2008 R2.2 patch 5 이상 버전의 드라이버는 CUBRID 오픈 소스 프로젝트 사이트에서 별도로 받을 수 있다.)
- 또한, CUBRID 매니저의 Bit 버전과 JRE의 Bit 버전은 서로 동일하여야 한다. 예를 들어, 64Bit 버전 DB 서버라도 CUBRID Manager 32Bit 버전을 사용한다면 JRE 또는 JDK 32Bit 버전을 설치하여야 한다.
- 새로운 예약어 추가 및 일부 질의에 대한 스펙 변경으로 인해 질의 결과가 이전 버전과 다를 수 있으므로 주의한다. 보다 상세한 사항은 업그레이드 주의 사항을 참고한다.
- GLO 클래스를 이용하여 개발된 응용은 BLOB, CLOB 타입에 맞는 응용 및 스키마로 변환하여 사용하여야 한다.

데이터베이스 호환성

서버 버전을 업그레이드하는 경우, 2008 R3.0 사용자는 데이터베이스 마이그레이션을 하지 않아도 된다. 단, GLO 클래스를 사용하는 경우 데이터베이스의 마이그레이션 작업을 수행하여야 한다. 보다 상세한 사항은 [데이터베이스 마이그레이션 절차](#)를 참고한다.

상호 운용성

CUBRID DB 서버와 브로커 서버를 분리하여 운영하는 경우, 서버 장비의 운영 체제가 다르더라도 상호 운용성을 보장한다. 단, DB 서버의 Bit 버전과 브로커 서버의 Bit 버전은 서로 동일하여야 한다. 예를 들어, Linux용 64Bit 버전 DB 서버는 Windows용 64Bit 버전 브로커 서버와 상호 운용이 가능하지만, 32Bit 버전 브로커 서버와는 상호 운용이 불가능하다.

CUBRID 2008 R3.1의 설치 방법

Linux에서 설치

Linux용 설치 패키지는 바이너리를 포함하는 스크립트, tar.gz 압축 파일, Linux RPM 패키지 형태로 제공되며, 설치 방법은 매뉴얼의 [CUBRID 시작> 설치와 실행> Linux에서의 설치와 실행](#)을 참고한다.

¹ \$CUBRID는 CUBRID가 설치된 곳을 지정하는 Linux용 환경변수이며, Windows 환경에서는 %CUBRID%와 같은 형식으로 사용된다.

Windows에서 설치

Windows용 설치 파일이 제공되며, 설치 마법사를 이용하여 설치할 수 있다. 설치 방법은 매뉴얼의 [CUBRID 시작> 설치와 실행> Windows에서의 설치와 실행](#)을 참고한다.

Windows 환경에서 설치 디렉터리 설정에 관한 주의 사항은 [CUBRIDSUS-3267](#)을 참고한다.

CUBRID 환경 변수 및 OS 환경 변수 설정

CUBRID를 사용하기 위해서는 CUBRID 환경 변수와 관련 OS 환경 변수를 설정하여야 한다. ([관련 매뉴얼 참고](#)) 특히, CUBRID 매니저와 Java 저장 프로시저를 사용하기 위해서는 Java 환경 변수를 설정하여야 한다. ([관련 매뉴얼 참고](#))

CUBRID 2008 R3.1로 업그레이드하는 방법

2008 R3.1 Beta 버전을 사용 중인 경우 업그레이드 방법 NEW

2008 R3.1 Beta 버전을 사용하는 사용자의 경우, 정식 버전으로의 업그레이드를 적극 권장하며 업그레이드 시 별도의 데이터베이스 마이그레이션 작업은 요구되지 않는다.

업그레이드 주의 사항

- 기존 환경 설정 파일 보관

이전 버전의 \$CUBRID/conf 디렉터리의 환경 설정 파일(**cubrid.conf**, **cubrid_broker.conf**, **cm.conf**)과 \$CUBRID_DATABASES 디렉터리의 데이터베이스 위치 정보 파일(**databases.txt**)을 보관한다. ²

- 데이터베이스 마이그레이션

GLO 기능을 사용하지 않는 CUBRID 2008 R3.0 사용자는 마이그레이션을 하지 않아도 된다. CUBRID 2008 R3.1은 GLO를 지원하지 않으며 LOB 타입이 GLO 기능을 대체하게 되었으므로, GLO를 이용한 응용 및 스키마는 LOB 타입에 맞게 수정하여야 한다. (아래의 [2008 R3.0에서 2008 R3.1로 마이그레이션 참고](#))

CUBRID 2008 R3.1은 2008 R2.x 이하 버전의 데이터베이스와 호환하지 않으므로, 이전 데이터베이스를 CUBRID 2008 R3.1로 마이그레이션하여야 한다. (아래의 [2008 R2.x에서 2008 R3.1로 마이그레이션 참고](#))

- 복제 또는 HA 환경 재구성

이전 버전의 복제 기능을 사용하는 시스템에서는 보다 안정적인 운영을 위해 DB 마이그레이션 및 HA 환경으로 재구성할 것을 권장한다. 또한, CUBRID 2008 R2.0, 2008 R2.1에서 제공된 Linux Heartbeat 기반의 HA 기능을 사용하는 시스템도 보다 안정적인 운영을 위해 DB 마이그레이션 및 CUBRID Heartbeat 기반의 HA 환경으로 재구성할 것을 권장한다. (아래의 [HA 환경에서 데이터베이스 마이그레이션 절차 참고](#))

- 새로 추가된 예약어 검사

CUBRID 설치 패키지 또는 [다운로드 페이지](#)에서 배포되는 CUBRID 2008 R3.1용 예약어 검출 스크립트인 **check_reserved.sql** 을 이용하여 예약어 사용 여부를 검사 할 수 있으며, 예약어로 지정된 식별자를 사용하고 있을 경우 식별자를 수정해야 한다. ([관련 매뉴얼 참고](#))

² \$CUBRID 또는 \$CUBRID_DATABASES 형식의 환경변수는 Linux 환경에서 사용하는 형식이며 Windows 환경에서는 %CUBRID% 또는 %CUBRID_DATABASES%와 같은 형식으로 사용해야 한다.

데이터베이스 마이그레이션 절차

[2008 R3.0에서 업그레이드하는 사용자를 위한 절차 바로 가기](#)

[2008 R2.x에서 업그레이드하는 사용자를 위한 절차 바로 가기](#)

● 2008 R3.0에서 2008 R3.1로 마이그레이션

- GLO 클래스 사용 여부에 따라 다음 작업을 수행

GLO 클래스를 사용하지 않는 경우, 별도의 작업 없이 2008 R3.0의 데이터베이스 볼륨을 2008 R3.1에서 그대로 사용할 수 있다.

그러나 GLO 클래스를 사용하는 경우, 2008 R3.1부터는 GLO 클래스를 지원하지 않으므로 BLOB 또는 CLOB 타입을 사용하도록 응용과 스키마를 변경해야 한다. 변경 작업이 용이하지 않다면 마이그레이션을 보류할 것을 권장한다.

- BLOB 또는 CLOB 타입을 사용하려면 다음 작업을 수행

2008 R3.1부터 지원하는 BLOB 또는 CLOB 타입을 사용하기 위해서는 아래의 예제와 같이 LOB 타입의 데이터를 저장할 디렉토리를 생성한 후, 이 경로를 \$CUBRID_DATABASES/databases.txt에 추가하여야 한다.³

```
// lob 디렉터리 생성하기
$ cd $CUBRID_DATABASES/olddb
$ mkdir lob

// 생성한 lob 디렉터리 경로를 databases.txt 에 lob-base-path 로 추가하기
$ cd $CUBRID_DATABASES/
$ vi databases.txt
#db-name          vol-path          db-host          log-path          lob-base-
path
r30db             /home/CUBRID/databases/olddb localhost         /home/CUBRID/databases/olddb
file:/home/CUBRID/databases/olddb/lob
```

● 2008 R2.x에서 2008 R3.1로 마이그레이션

2008 R2.x 사용자는 아래의 표와 같이 2008 R3.1으로 데이터베이스 마이그레이션을 해야 한다.

또한, 기존의 GLO 클래스 사용 여부에 따라 필요한 추가 작업을 하도록 한다. (위의 [2008 R3.0에서 2008 R3.1로 마이그레이션 참고](#))

아래는 \$CUBRID/bin/migrate_r30 유틸리티 및 [다운로드 페이지](#)에서 별도 배포되는 **check_reserved.sql** 예약어 검출 스크립트를 이용하여 마이그레이션을 수행하는 방법이다.³

다른 방법으로는 **cubrid unloaddb/loaddb** 유틸리티를 사용하여 마이그레이션을 수행할 수 있다. ([관련 매뉴얼](#)과 본 문서의 [5. 주의 사항> CUBRIDSUS-3826](#)을 참고)

단계	Linux 환경	Windows 환경
C1 단계: CUBRID Service 종료	% cubrid service stop	CUBRID Service Tray>[Exit]를 선택한다.
C2 단계: 예약어 검출 스크립트 실행	예약어 검출 스크립트가 위치하는 디렉터리에서 아래 명령을 실행한다. 검출 결과를 확인하여 마이그레이션 진행 또는 식별자 수정 작업을 진행한다. (허용되는 식별자는 관련 매뉴얼 참고) % csql -S -u dba -i check_reserved.sql <r22_db_name>	
C3 단계: 이전 버전 DB 백업	이전 버전으로 복구하는 상황에 대비하기 위해 백업을 수행하고, 백업 파일을 별도 디렉터리(R22_backup)에 보관한다. (C3a) % mkdir R22_backup % cubrid backupdb -S -D R22_backup <r22_db_name> 이전 버전의 databases.txt 및 conf 디렉터리 내 설정 파일을 별도 디렉터리에 보관한다. (C3b)	

³ \$CUBRID 또는 \$CUBRID_DATABASES 형식의 환경변수는 Linux 환경에서 사용하는 형식이며 Windows 환경에서는 %CUBRID% 또는 %CUBRID_DATABASES%와 같은 형식으로 사용해야 한다.

		이전 버전의 CUBRID를 제거한다.
C4 단계: 2008 R3.1 설치	본 문서의 CUBRID 2008 R3.1의 설치 방법 을 참고한다.	
C5 단계: 마이그레이션 도구 실행		CUBRID Service Tray> [CUBRID Server]>[Stop] 을 선택하여 서버를 종료한다.
	(C3b)에서 보관한 databases.txt를 2008 R3.1의 설치 디렉터리에 복사한다. (C5a) 아래와 같이 migrate_r30 유틸리티를 실행한다. (C5b) <pre>% migrate_r30 <db name></pre>	
C6 단계: 현재 버전 DB 백업	이전 데이터베이스 버전이 2008 R1.x인 경우에만 수행한다. 이전 버전이 2008 R2.x인 경우, 이 단계를 생략할 수 있다. <pre>% cubrid backupdb -S <db name></pre>	
C7 단계: CUBRID 환경 설정 및 CUBRID Service 구동	환경 설정 파일을 수정한다. 이때, (C3b)에서 보관한 이전 버전의 환경 설정 파일을 사용할 수 있다. <pre>% cubrid service start % cubrid server start <db name></pre>	CUBRID Service Tray> [CUBRID Server]-> [Start]를 선택하여 서버를 시작한다.

HA 환경에서 데이터베이스 마이그레이션 절차

[2008 R2.2 이상 버전에서 업그레이드 하는 사용자를 위한 절차 바로 가기](#)

[2008 R2.0 또는 2008 R2.1에서 업그레이드 하는 사용자를 위한 절차 바로 가기](#)

● 2008 R2.2 이상 버전에서 2008 R3.1로 HA 마이그레이션

아래는 브로커, 마스터 DB, 슬레이브 DB를 각각 별도 서버에 구축한 환경에서 현재 서비스를 중지하고 업그레이드를 수행하기 위한 가이드이다. 서비스 무정지 업그레이드 시나리오는 별도 가이드 문서를 참고한다.

단계	설명
H1~H6 단계: 마스터 서버에서 C1~C6 단계를 수행	마스터 서버에서 CUBRID 업그레이드 및 데이터베이스 마이그레이션을 수행하고, 2008 R3.1 데이터베이스를 백업한다.
H7 단계: 슬레이브 서버에 CUBRID 2008 R3.1 설치	설치 방법은 본 문서의 CUBRID 2008 R3.1의 설치 방법 을 참고한다.
H8 단계: 마스터 백업본을 슬레이브 서버에서 복구	H6에서 생성된 마스터 서버의 2008 R3.1 데이터베이스 백업본(예: testdb_bk*)을 슬레이브 서버에서 복구한다. <pre>% scp user1@master:~/DB/testdb/testdb_bk0v000 . % scp user1@master:~/DB/testdb/log/testdb_bkvinf ./log/. % cubrid restoredb testdb</pre>
H9단계: HA 환경 재구성 후 HA모드 구동	이 단계의 작업은 root 계정으로 수행한다. HA구동 스크립트(cubrid-ha) 및 환경 설정 파일(cubrid.conf)을 설정한다. (관련 매뉴얼 참고) 참고한다. 마스터 및 슬레이브 서버에서 HA모드로 DB를 구동한다. (관련 매뉴얼 참고) <pre>[root@master ~]# service cubrid-ha start [root@slave ~]# service cubrid-ha start</pre>
H10 단계: 브로커 서버에 CUBRID 2008 R3.1 설치 및 브로커 구동	설치 방법은 본 문서의 CUBRID 2008 R3.1의 설치 방법 을 참고한다. 브로커 설정 후 브로커를 시작한다. (관련 매뉴얼 참고) <pre>% cat cubrid broker.conf ... ACCESS MODE=RW % cubrid broker start</pre>

● 2008 R2.0 또는 2008 R2.1에서 2008 R3.1로 HA 마이그레이션

CUBRID 2008 R2.0 또는 2008 R2.1의 HA 기능을 사용하는 경우, 해당 버전에서 사용되었던 Linux Heartbeat 패키지를 제거하고, 서버 버전 업그레이드, 데이터베이스 마이그레이션을 수행한 후 HA 환경을 새롭게 구축하여야 한다.

CUBRID 2008 R3.1 릴리스 노트

아래의 H0 단계를 수행한 후, 위의 H1~H10 단계를 수행한다.

단계	설명
H0 단계: HA 관련 서비스 종료 및 기존 Linux heartbeat 제거	<p>구동 중인 브로커를 중지한다.</p> <pre>% cubrid broker stop</pre> <p>이하의 작업은 마스터 및 슬레이브 서버에서 root 계정으로 수행한다.</p> <pre>[root@master ~]# service heartbeat stop [root@master ~]# chkconfig --del heartbeat [root@master ~]# pkill -u user1 -f "cub_master"</pre> <p>// 슬레이브 서버에서 동일 작업 수행</p>

복제 기능 사용 시 주의사항

CUBRID 2008 R3.0 이상 버전부터 복제 기능에 관한 개발은 중단(Outdated and Deprecated Feature)되었으므로, 복제 기능을 사용 중인 환경에서는 보다 안정적인 운영을 위하여 HA 기능을 사용하여 이중화 환경을 구축할 것을 권장한다. 서버 버전 업그레이드 및 데이터베이스 마이그레이션을 수행한 후, HA 환경을 새롭게 구축할 수 있다.

3. CUBRID 2008 R3.1에서 변경된 사항

새로 추가된 기능

CUBRIDSUS-3652 BLOB/CLOB 타입의 추가 및 관련 API 지원

이미지 또는 긴 텍스트 등의 Large Object를 저장할 수 있는 데이터 타입으로 BLOB/CLOB을 제공한다. BLOB은 바이너리 데이터를 저장하기 위한 타입이며 CLOB은 문자열 데이터를 저장하기 위한 타입이다.

BLOB/CLOB 타입의 데이터가 삽입되면 데이터는 외부 파일 시스템에 파일로 저장되고, 해당 파일의 경로 정보(LOB locator)는 CUBRID 데이터베이스에 저장된다.

BLOB/CLOB 데이터가 저장되는 최상위 디렉터리는 데이터베이스 생성 시 지정할 수 있으며 지정하지 않을 경우 데이터베이스 볼륨이 생성되는 경로의 lob 디렉터리 아래에 저장된다. ([관련 매뉴얼 참고](#))

BLOB/CLOB 타입과 관련하여 다음의 표와 같은 SQL 함수, JDBC 메소드, CCI API가 추가되었다.

대상	목록	비고
SQL 함수	CLOB_TO_CHAR BLOB_TO_BIT CHAR_TO_CLOB BIT_TO_BLOB CHAR_TO_BLOB CLOB_FROM_FILE BLOB_FROM_FILE CLOB_LENGTH BLOB_LENGTH	관련 매뉴얼 참고
JDBC	CUBRIDConnection.createBlob ResultSet.setBlob ResultSet.getBlob	관련 매뉴얼 참고
CCI	cci_blob_new cci_blob_size cci_blob_write cci_blob_read cci_blob_free	관련 매뉴얼 참고

아래는 BLOB/CLOB 타입을 이용한 SQL 예제이다.

```
// 테이블 생성
CREATE TABLE lob_tbl (id int PRIMARY KEY, content CLOB, image BLOB);
// 데이터 입력
INSERT INTO lob_tbl VALUES (1, CHAR_TO_CLOB('This is a Dog'),
BLOB_FROM_FILE('/home1/data1/doc_image.jpg'));
// 데이터 조회
SELECT id, content, image FROM lob_tbl
id content image
=====
1 file:/data1/ces 522/lob tbl.00001282208855807171 7329
file:/data1/ces 393/lob tbl.00001282208855809474 7474
```

BLOB/CLOB 관련 제약사항은 본 문서의 [CUBRIDSUS-3826 GLO 클래스 지원 중단에 따른 주의 사항](#)을 참고한다.

CUBRIDSUS-434 JDBC/CCI에서 Foreign Key 정보를 얻을 수 있는 기능 추가

JDBC와 CCI API에 다음과 같이 Foreign Key 관련 정보를 얻을 수 있는 기능을 추가하였다.

대상	목록	비고
JDBC	DatabaseMetaData 클래스의 관련 메소드 지원 getImportedKeys getExportedKeys getCrossReference	관련 매뉴얼 참고
CCI	cci_schema_info 함수의 관련 타입 확장 CCI_SCH_IMPORTED_KEYS CCI_SCH_EXPORTED_KEYS CCI_SCH_CROSS_REFERENCE	관련 매뉴얼 참고

CUBRIDSUS-3788 ODBC 드라이버에 스키마 정보를 얻을 수 있는 기능 추가

ODBC 드라이버에 다음과 같이 Primary Key, Foreign Key 등 스키마 관련 정보를 얻을 수 있는 기능을 추가하였다.

대상	목록	비고
ODBC	SQLForeignKeys SQLPrimaryKeys SQLProcedures SQLProcedureColumns SQLTablePrivileges	관련 매뉴얼 참고

CUBRIDSUS-4091 ODBC 드라이버에 UTF-8 문자셋 지원 **NEW**

ODBC 드라이버에 UTF-8 문자셋 지원 기능을 추가하여 UTF-8 문자열로 입력 및 조회가 가능하도록 개선하였다. UTF-8 이외에 다른 문자셋은 지정할 수 없으며, 아래의 예와 같이 연결 문자열에 "CHARSET=utf-8" 속성을 추가하는 방식으로 사용할 수 있다.

```
// 연결 문자열에 CHARSET 속성을 UTF-8 로 설정
DRIVER=CUBRID
Driver;UID=public;FETCH SIZE=100;PORT=30000;SERVER=localhost;DB NAME=demodb;DESCRIPTION=cubrid odbc;C
HARSET=utf-8
```

CUBRIDSUS-3879, 4183 브로커의 상태 정보 출력 기능에 브로커 응용 서버(CAS) 정보 포함 옵션 추가 **NEW**

브로커 상태 정보 출력을 위한 `cubrid broker status -b` 명령 사용 시 `-f` 옵션을 추가로 사용하면 브로커 응용 서버(CAS) 정보를 포함하도록 개선하였다.

`-f` 옵션에 `-i` 옵션을 추가하면 지정한 초 동안 대기 상태인 응용 서버의 개수(Ns-W)와 수행 상태인 브로커 응용 서버(CAS)의 개수(Ns-B)를 출력한다. (`-i` 옵션을 생략하면 기본값은 1초이다.)

새로 추가한 브로커 응용 서버(CAS) 정보는 다음과 같다.

항목 이름	설명
T	실행 중인 응용 서버(CAS)의 전체 개수
W	현재 클라이언트 대기(Waiting) 상태인 응용 서버(CAS) 개수
B	현재 클라이언트 수행(Busy) 상태인 응용 서버(CAS) 개수

Ns-W	N초 동안 클라이언트 대기(Waiting) 상태였던 응용 서버(CAS) 개수
Ns-B	N초 동안 클라이언트 수행(Busy) 상태였던 응용 서버(CAS) 개수
CANCELED	브로커 구동 후 사용자 인터럽트로 취소된 질의 개수 (-s 옵션 추가 시 지정한 초 동안 누적된 개수)
ACCESS_MODE	브로커의 ACCESS_MODE 설정
SQL_LOG	브로커의 SQL_LOG 설정

```
// 브로커 상태 정보 출력 시 -f 옵션 추가. -l 옵션은 N 초 동안의 Ns-W, Ns-B 를 출력하도록 초를 설정
% cubrid broker status -b -f -l 2
@ cubrid broker status
NAME      PID    PSIZE PORT  AS(T W B 2s-W 2s-B) JQ REQ TPS QPS LONG-T LONG-Q ERR-Q CANCELED
ACCESS_MODE SQL_LOG
=====
query_editor 16784 56700 38000  5 0 0 0  0  0 0 0 0/60.0 0/60.0 0  0
RW      ALL
```

개선된 기능

CUBRIDSUS-3761 CCI에서 응용과 브로커 간 네트워크 상태를 확인하는

방식 개선

CCI(C API)를 이용한 응용과 브로커 간의 네트워크 연결 상태를 주기적으로 검사하기 위하여 기존에는 Echo 포트를 사용하였으나, 브로커의 포트를 사용하도록 변경하였다.

CUBRIDSUS-3343 브로커와 JDBC 응용 프로그램 간 최대 응답 시간을

줄이도록 개선 New

브로커 응용 서버(CAS)와 응용 프로그램 사이에서 사용하는 네트워크 버퍼 크기를 줄여 JDBC 응용 프로그램의 최대 응답 시간이 줄어들도록 개선하였다.

JDBC 드라이버는 네트워크 버퍼를 통해 얻은 질의 수행 결과를 Java 객체로 생성하여 관리한다. 이때 생성된 객체가 차지하는 메모리 영역으로 인해 자주 JVM의 Full GC(Garbage Collection)가 발생하여 응답 시간이 늘어날 수 있으므로 네트워크 버퍼의 크기를 기존의 128K에서 16K로 줄여 Full GC의 발생 빈도가 낮아지도록 하였다.

이 수정은 CCI, PHP, ODBC 응용 프로그램에도 적용되지만, JDBC 응용 프로그램과 같이 성능 상의 이익이 발생하지는 않는다.

CUBRIDSUS-4027 브로커의 ACCESS_MODE를 동적으로 변경할 수

있도록 개선

broker_changer 유틸리티를 사용하여 특정 브로커의 ACCESS_MODE 값을 변경하고 해당 브로커는 자동으로 reset 되도록 개선하였다.

```
//ACCESS_MODE 를 RO 로 변경하고 브로커 reset
% broker_changer broker_name access_mode ro

//ACCESS_MODE 를 RW 로 변경하고 브로커 reset
% broker_changer broker_name access_mode rw
```

Windows 환경에서는 브로커 파라미터의 동적 변경 기능을 사용할 수 없다. ([3. CUBRID 2008 R3.1에서 변경된 사항](#)> [CUBRIDSUS-3926](#) 참고)

CUBRIDSUS-3835 보존할 보관 로그 파일의 최대 개수를 동적으로 변경할

수 있도록 개선 New

데이터베이스를 재시작하지 않고 보존할 보관 로그 파일의 최대 개수를 동적으로 변경할 수 있도록 개선하였다. 데이터베이스 구동 시에는 cubrid.conf에서 사용된 log_max_archives 파라미터의 값이 보존할 보관 로그 파일의 최대 개수로 설정된다.

아래의 예와 같이 csql을 이용하여 변경하는 경우에는 데이터베이스의 dba 계정으로 접속해야 한다.

```
//dba 계정으로 실행한 csql에서 log_max_archives 값을 동적으로 변경
csql>;set log_max_archives=5
```

CUBRIDSUS-3771 deadlock 및 lock timeout 발생 시 데이터베이스

서버의 에러 로그 메시지 개선 New

데이터베이스 서버의 에러 로그 파일에 deadlock 발생 시 deadlock 메시지를 추가하고, lock timeout 발생 시 기존의 lock timeout 메시지에 lock timeout의 원인을 제공한 프로세스 이름과 프로세스 아이디를 출력하도록 개선하였다.

```
// deadlock 이 발생하는 경우, deadlock 메시지와 함께 이를 유발한 호스트/프로세스 정보 출력
Time: 08/23/10 12:06:11.215 - NOTIFICATION *** CODE = -993, Tran = 0, CLIENT = (unknown):(unknown) (-1), EID = 7
deadlock cycle is detected. public@cdb006.cub|csql(19502), public@cdb006.cub|csql(19501),
public@cdb006.cub|csql(19500).

// lock timeout 이 발생하는 경우, 에러 메시지의 제일 끝에 프로세스 이름과 프로세스 아이디를 출력
Time: 08/23/10 12:06:11.215 - ERROR *** ERROR CODE = -967, Tran = 3, CLIENT = cdb006.cub:csql(19502),
EID = 8
Your transaction (index 3, public@cdb006.cub|19502) timed out waiting on IS LOCK lock on class tb1
because of deadlock. You are waiting for user(s) public@cdb006.cub|csql(19500) to finish.
```

CUBRIDSUS-4011 Debian 패키지 빌드가 가능하도록 개선

Ubuntu를 포함한 Debian 계열 Linux에서 deb 패키지 빌드가 가능하도록 관련 파일을 추가하였다. 그리고, Ubuntu Linux의 최신 버전(10.10 maverick)에서 기본 제공하는 bison 2.4는 CUBRID 빌드 시 사용하는 bison 2.3과 호환되지 않기 때문에 bison 2.3의 소스를 CUBRID 소스 패키지에 포함하여 빌드 시 이를 이용하도록 개선하였다.

수정된 사항

CUBRIDSUS-3698 IF 함수의 인자에 NULL이 주어지면 항상 NULL을

반환하는 오류 수정

IF 함수에서 첫 번째 인자로 주어진 조건식에 컬럼이 포함되고 두 번째 혹은 세 번째 인자로 주어진 결과값에 NULL이 주어지면, 조건의 참, 거짓에 관계없이 항상 NULL을 반환하는 오류를 수정하였다.

```
CREATE TABLE t1( i1 INTEGER);
INSERT INTO t1 VALUES(1), (2);
SELECT i1, IF ( i1 > 1, i1, NULL) FROM t1; -- 첫 번째 레코드는 거짓, 두 번째 레코드는 참이나 모두 NULL을 반환
```

CUBRIDSUS-4047 RIGHT OUTER JOIN이 MERGE 조인으로 실행될 경우

결과가 잘못 나오는 오류 수정

아래의 예와 같이 RIGHT OUTER JOIN 질의를 MERGE 조인으로 실행하는 경우, 결과가 잘못 나오는 오류를 수정하였다.

```
select /*+ USE_MERGE */ * from foo right outer join bar on foo.i = bar.i
```

CUBRIDSUS-4193 LIKE 조건식에서 지정한 escape 문자가 검색 문자열

내에 존재하면 발생하는 오류 수정 New

LIKE 조건식에 ESCAPE 구문이 있는 경우, wildcard(%, _) 이외의 문자 앞에 escape 문자가 존재하면 오류가 발생하였으나, wildcard(%, _)를 포함한 모든 문자 앞에 escape 문자가 존재하면 뒤의 문자를 일반 문자로 처리하여 오류가 발생하지 않도록 수정하였다.

```
// escape 문자 \를 사용하여 질의를 수행하는 예. path 컬럼의 값이 "c:\path_"로 시작하는 레코드를 검색
SELECT * FROM tbl WHERE path LIKE 'c:\\path\_ ' ESCAPE '\';
```

CUBRIDSUS-4176 계층 질의문에서 출력할 컬럼의 타입이 CHAR 혹은

NUMERIC이면 발생하는 오류 수정 New

계층 질의문에서 출력할 컬럼의 타입이 CHAR 혹은 NUMERIC처럼 고정 길이를 지정하는 타입인 경우, Query execution failure #13511라는 에러 메시지가 발생하는 오류를 수정하였다.

```
// 데이터 생성
CREATE TABLE tbl (id VARCHAR(20), parent_id VARCHAR(20), use_yn CHAR(1));
INSERT INTO tbl VALUES ('AFAC002', 'AFA', 'Y'), ('AFAC018', 'AFAC02', 'Y'), ('AFAC019', 'AFAC002', 'Y'),
('AFAC020', 'AFAC002', 'Y'), ('AFAC022', 'AFAC02', 'Y');

// 계층 질의문 수행
SELECT id, parent id, use yn FROM tbl
START WITH ID='AFAC002' CONNECT BY NOCYCLE PRIOR id=parent_id;

// 기존 버전에서 다음의 에러 발생
Execute: Query execution failure #13511
```

CUBRIDSUS-4177 계층 질의문에서 INLINE VIEW 외부에 WHERE 조건이

있을 경우 잘못된 결과를 출력하는 오류 수정 New

INLINE VIEW가 포함된 계층 질의문에 WHERE 절이 있는 경우, INLINE VIEW 외부의 WHERE 조건이 내부에 적용되어 잘못된 결과를 출력하는 오류를 수정하였다.

아래의 예를 보면, 기존 버전에서는 inline_view 외부에 WHERE id <> 1 이라는 조건이 inline_view 내부에 적용되어 inline_view의 결과에 id가 1인 레코드가 빠지게 되고, 이 결과를 가지고 START WITH id=1 인 조건을 수행하면서 최종 결과는 0건이 되는 오류가 존재한다.

```
// 테이블 생성 및 데이터 입력
CREATE TABLE tree(ID INT, MgrID INT, Name VARCHAR(32));
INSERT INTO tree VALUES (1,NULL,'Kim');
INSERT INTO tree VALUES (2,NULL,'Moy');
INSERT INTO tree VALUES (3,1,'Jonas');
INSERT INTO tree VALUES (4,1,'Smith');
INSERT INTO tree VALUES (5,2,'Verma');
INSERT INTO tree VALUES (6,2,'Foster');
INSERT INTO tree VALUES (7,6,'Brown');

// 계층 질의문 수행 시 WHERE id <> 1 조건이 inline_view 안에서 수행되던 오류를 수정함
SELECT id,mgrid,name
FROM (
    SELECT id,mgrid,name FROM tree
)inline view
WHERE id <> 1
START WITH id = 1
CONNECT BY PRIOR id = mgrid;

// 기존 버전에서 수행 시 결과 0 건
// 수정된 버전에서 수행 시 결과 2 건
3 1 Jonas
4 1 Smith
```

CUBRIDSUS-4178 계층 질의문에서 ORDER SIBLINGS BY 정렬이

잘못되는 오류 수정 **NEW**

계층 질의문에서 ORDER SIBLINGS BY에 의한 정렬을 수행하는 경우, 같은 계층의 레코드, 즉 SIBLINGS가 제대로 정렬이 되지 않는 오류를 수정하였다.

ORDER SIBLINGS BY <컬럼>에 의한 정렬은 최상위 레벨의 레코드에서 <컬럼> 값의 순서가 가장 먼저 오는 것을 출력하고 그 뒤를 이어 자식 레코드를 출력하는 깊이 우선 탐색을 할 때 같은 부모 레코드를 가지는 자식 레코드들을 <컬럼> 값의 순서대로 출력하도록 한다.

```
// ORDER SIBLINGS BY seq 에 의해 정렬되는 예
// 같은 최상위 레벨의 레코드가인 A01, A02, A03 중 seq 값이 가장 작은 A02 를 먼저 출력하고,
// 이어서 A02 의 자식 레코드를 한다. 이때 같은 레벨의 자식 레코드 중 seq 값이 작은 A0201 을 먼저 출력하고,
// A0202 를 나중에 출력한다.
```

```
CREATE TABLE tbl(id VARCHAR(10), pid VARCHAR(15), name VARCHAR(15) , seq INT);
```

```
INSERT INTO tbl VALUES('A01', 'A00', 'MENU01', 3);
INSERT INTO tbl VALUES('A02', 'A00', 'MENU02', 1);
INSERT INTO tbl VALUES('A0101', 'A01', 'MENU0101', 2);
INSERT INTO tbl VALUES('A0102', 'A01', 'MENU0102', 1);
INSERT INTO tbl VALUES('A0202', 'A02', 'MENU0202', 2);
INSERT INTO tbl VALUES('A0201', 'A02', 'MENU0201', 1);
INSERT INTO tbl VALUES('A03', 'A00', 'MENU03', 2);
```

```
SELECT id, pid, name, LEVEL, seq
FROM tbl
START WITH pid = 'A00'
CONNECT BY PRIOR id = pid
ORDER SIBLINGS BY seq;
```

id	pid	name	LEVEL	seq
'A02'	'A00'	'MENU02'	1	1
'A0201'	'A02'	'MENU0201'	2	1
'A0202'	'A02'	'MENU0202'	2	2
'A03'	'A00'	'MENU03'	1	2
'A01'	'A00'	'MENU01'	1	3
'A0102'	'A01'	'MENU0102'	2	1
'A0101'	'A01'	'MENU0101'	2	2

CUBRIDSUS-4004 기본 키 생성 실패 후 COMMIT을 수행하면 해당

테이블 조회 시 에러가 발생할 수 있는 오류 수정

기본 키(Primary Key) 제약 조건을 추가하는 ALTER TABLE문이 실패한 상태에서 해당 트랜잭션을 커밋하면 서버에 캐시된 변경 내역을 불완전 롤백하게 되고, 이후 해당 테이블에 UPDATE문을 실행하면 데이터가 잘못 수정되는 오류가 발생한다. 이로 인해 데이터베이스 서버를 재시작한 이후 해당 테이블을 조회할 때 `ERROR: Unknown representation identifier`라는 에러 메시지가 출력되었으나, 이를 수정하였다.

CUBRIDSUS-3538 메타 데이터를 얻어오는 JDBC 메소드에서

ArrayIndexOutOfBoundsException 예외가 발생하는 오류 수정

JDBC의 DatabaseMetaData 클래스 내에 정의된 getProcedure, getProcedureColumns, getColumns, getColumnPrivileges, getTablePrivileges, getImportedKeys, getExportedKeys, getCrossReference, getTypeInfo, getIndexInfo, getAttributes 메소드를 통해 얻은 ResultSet에서 사용하는 내부 정보가 잘못 기록되어, getMetaData 메소드를 호출하면 ArrayIndexOutOfBoundsException 예외가 발생하는 경우가 있었으나, 이를 수정하였다.

CUBRIDSUS-3991 HA 환경에서 TRUNCATE TABLE 질의 수행 시 복제가 중단되는 오류 수정

HA 환경에서 마스터 서버에 TRUNCATE TABLE 문이 실행된 경우, 복제 정보 처리의 오류로 인해 슬레이브 서버의 로그 반영 프로세스가 비정상 종료되는 현상이 발생하였으나, 이를 수정하였다.

CUBRIDSUS-3911 HA 환경에서 온라인 백업 후 보관 로그 동기화 오류로 인해 복제가 중단되는 오류 수정

HA 환경에서 마스터 DB를 온라인 백업한 이후 복제가 지연되는 특정 상황에서 copylogdb 프로세스가 빈 보관 로그를 생성하는 오류로 인해, 특정 시점부터 copylogdb 프로세스가 반복적으로 재시작되면서 복제가 중단되는 문제가 발생하였으나 이를 수정하였다.

CUBRIDSUS-3910 HA 환경에서 applylogdb 프로세스가 재시작 후 비정상 종료되면서 데이터 불일치가 발생하는 오류 수정

HA 환경에서 데이터베이스의 페이지 크기(--page-size 옵션 사용)가 로그 페이지 크기(--log-page-size 옵션 사용)보다 크게 설정되어 생성된 경우, HA 운영 중에 로그 레코드가 미리 할당한 메모리 영역을 초과하여 기록되면서 로그를 슬레이브 DB로 반영할 때 applylogdb 프로세스가 비정상 종료되고, 이로 인해 데이터 불일치가 발생하는 오류가 있었으나 이를 수정하였다.

CUBRIDSUS-4295 HA 환경에서 인덱스 이름만을 명시한 DROP INDEX 구문이 슬레이브 DB에 반영되지 않는 오류 수정 New

HA 환경의 마스터 서버에서 아래의 예와 같이 테이블 명과 컬럼 명을 생략하고 인덱스 명만을 명시하여 DROP INDEX 구문을 수행하는 경우, 슬레이브 DB에 반영되지 않는 오류를 수정하였다.

```
// 인덱스 명만 명시하여 DROP INDEX 수행
DROP INDEX idx_abc;
```

CUBRIDSUS-4347 HA 환경에서 DB 연결을 설정하는 중에 발생하는 오류 수정 New

HA 환경에서 브로커 서버의 DB 호스트 접속 순서가 databases.txt에 host1:host2로 되어 있고 마스터 DB는 host1, 슬레이브 DB는 host2인 환경에서 브로커의 ACCESS_MODE=SO(Slave Only)로 설정되어 있는 경우, 슬레이브 DB를 찾는 과정에서 처음에 연결했던 host1은 슬레이브 DB가 아니므로 브로커 응용 서버(CAS)와 DB 서버 간의 연결을 종료하여야 하나, 연결이 종료되지 않고 남아 있는 오류를 수정하였다.

또한, HA 환경에서 DB 서버가 허용하는 최대 클라이언트 개수(cubrid.conf의 max_client)를 초과하는 경우, 브로커 서버의 databases.txt에 있는 순서에 따라 다음 호스트로 DB 연결을 시도하여야 하나 그렇지 않은 오류를 수정하였다.

CUBRIDSUS-3804 다중 동시 접속 및 삽입/삭제 질의 수행 시 특정 조건에서 연결이 종료되지 않는 오류 수정

다중으로 동시 접속하여 삽입 및 삭제 질의를 수행하는 경우, 응용 프로그램이 종료되지 않고 브로커 응용 서버(CAS) 중 일부는 연결 상태가 CLOSE_WAIT로 대기하는 현상이 발생하는 경우가 있었으나 이를 수정하였다. 이 현상은 브로커 설정 파일(cubrid_broker.conf)의 MIN_NUM_APPL_SERVER와 MAX_NUM_APPL_SERVER의 값이 서로 다르게 설정된 경우에 발생하였다.

CUBRIDSUS-4272 -327번 에러가 발생할 수 있는 오류 수정 New

동시 사용자 환경에서 다음과 같이 -327번 에러가 발생할 수 있는 오류를 수정하였다.

```
Time: 12/01/10 17:14:13.728 - ERROR *** ERROR CODE = -327, Tran = 5, EID = 3
Cannot create MOP with NULL OID.
```

CUBRIDSUS-4218 다량의 페이지를 가지는 내부 파일 생성 요구가

실패하는 오류 수정 New

인덱스 생성과 같이 한번에 많은 양의 페이지를 요구하는 작업을 수행하는 경우, 내부 파일 관리 맵이 잘못 되어 다음과 같은 에러를 발생시키면서 페이지 할당이 실패할 수 있는 오류를 수정하였다.

```
Internal error: fetching deallocated pageid -2146798529 of volume "/home/qa/db/testdb/testdb_t32763".
```

CUBRIDSUS-4217 문자열 중간에 NULL이 포함된 데이터를

내보내기(unload)하는 경우 데이터 파일이 잘못 생성되는 오류 수정 New

VARCHAR, CHAR와 같은 문자 타입 컬럼이 있는 레코드 값의 중간에 ASCII CODE 값 0(즉, NULL)이 포함되어 있는 경우, unloaddb 유틸리티가 데이터 파일을 잘못 생성하여 이를 loaddb 유틸리티로 로딩하는데 실패하였으나, 데이터 파일을 정상적으로 생성하여 데이터 파일 로딩이 성공하도록 오류를 수정하였다.

CUBRIDSUS-4084 64bit Windows 시스템에 32bit CUBRID 설치 시

ODBC 드라이버가 등록되지 않는 오류 수정

64bit Windows 시스템에 32bit CUBRID를 설치하는 경우, ODBC 관리자에 32bit용 ODBC 드라이버가 등록되지 않는 오류가 있었으나, 이를 수정하였다.

CUBRIDSUS-3926 Windows의 명령 프롬프트 창에서 CUBRID 제어 시

정상 수행되지 않는 오류 수정 New

Windows의 명령 프롬프트 창에서 cubrid 유틸리티 명령어로 브로커나 서버 또는 매니저 서버를 시작/종료 시도하는 경우, SYSTEM 계정이 아닌 일반 사용자 계정으로 실행을 시도하면서 정상 동작하지 않는 오류가 있었으나, SYSTEM 계정으로 정상 수행이 가능하도록 수정하였다.

이로 인해 Windows Vista 이상 버전에서는 cubrid 유틸리티나 CUBRID Service Tray를 관리자 권한으로 구동하지 않는 경우 권한 상승 확인을 위한 UAC(User Account Control) 대화 상자가 활성화되고 이를 승인하여야 정상적인 제어가 가능하다.

cubrid 유틸리티를 사용하는 경우 명령 프롬프트 창을 관리자 권한으로 구동하는 것을 권장하며 구동 방법은 [5.주의사항 > CUBRIDSUS-4186](#)을 참고한다.

이 수정으로 인해 Windows 환경에서 브로커 파라미터의 동적 변경 기능이 동작하지 않는 제약 사항이 추가되었다. ([5.주의사항 > CUBRIDSUS-3926](#) 참고)

CUBRIDSUS-4015 영문 Windows에서 CUBRID Service Tray와 관련된

오류 메시지 출력 시 한글이 출력되는 오류 수정

한글 폰트가 설치되지 않은 영문 Windows에서 CUBRID Service Tray와 관련된 오류 메시지 출력 시 한글 메시지가 출력되어 메시지가 비정상적으로 보였으나, 이 메시지를 영문으로 출력하도록 수정하였다.

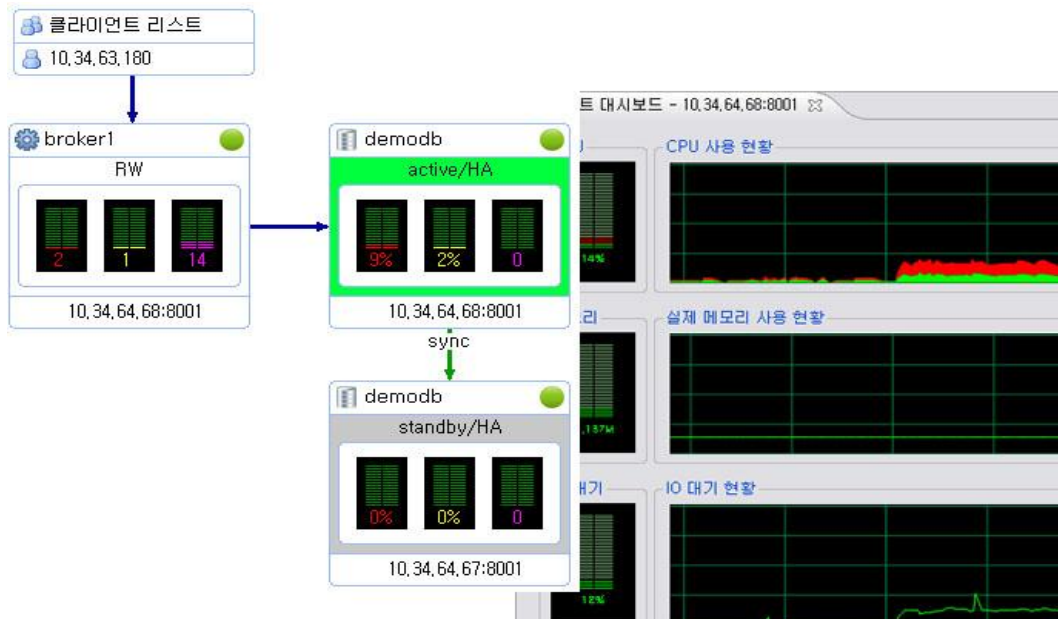
4. CUBRID 매니저 2008 R3.1에서 변경된 사항

새로 추가된 기능

HA 모니터링 대시보드 기능 추가

운영 중인 데이터베이스 서버 및 브로커, HA의 상태를 모니터링 할 수 있는 모니터링 대시보드 기능을 추가하였다.

모니터링 대시보드를 통해 데이터베이스를 구동하고 있는 호스트, 데이터베이스 서버, 브로커의 실시간 운영 및 장애 현황을 감시할 수 있으며, 각각의 상세한 세부 정보도 별도의 창을 통해 볼 수 있다.

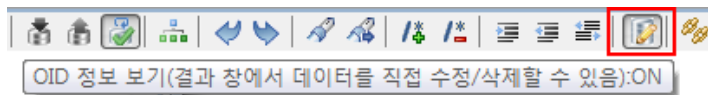


BLOB, CLOB, 길이가 100 이상인 BIT 타입의 입력 기능 추가

BLOB, CLOB, 그리고 길이가 100 이상인 BIT 타입의 데이터는 기존의 [데이터 내려받기 및 내보내기] 메뉴에서 데이터를 입력할 수 없어 [질의 편집기> 질의 결과> 상세 보기] 메뉴에서 입력할 수 있도록 추가하였다. 이와 같은 데이터는 질의 결과 창에 데이터를 직접 표시하지 않고 (BLOB), (CLOB), (BIT)와 같은 태그만 표시된다.

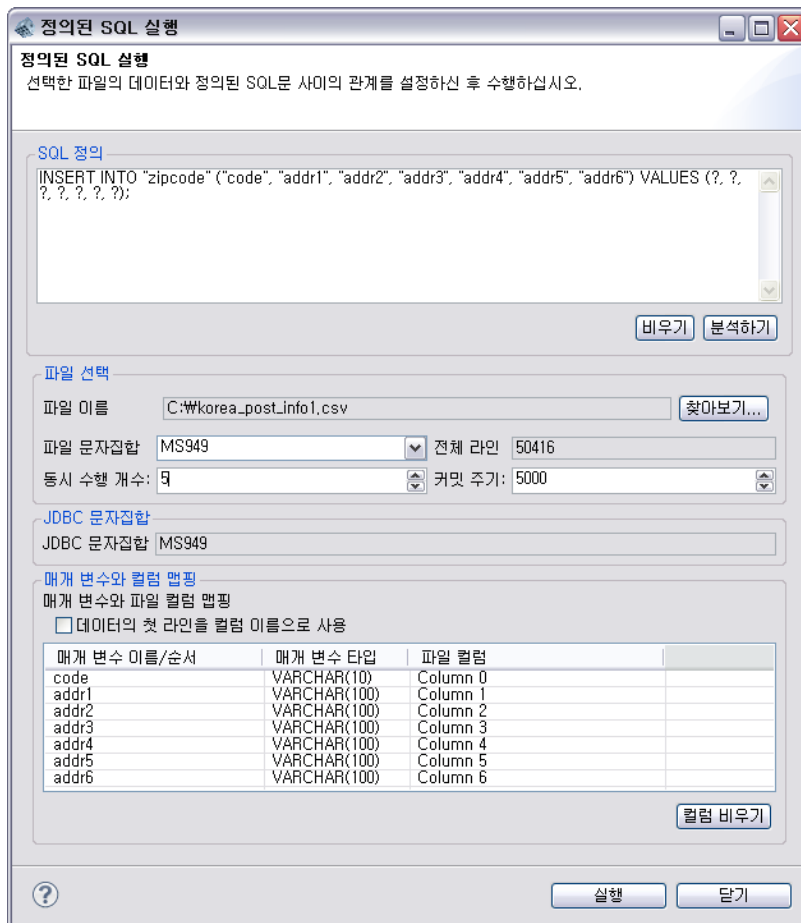
질의 편집기에서 데이터를 수정하려면, [질의 결과] 탭의 해당 컬럼에서 마우스 오른쪽 버튼을 클릭하고 [상세 보기]를 선택한 후 입력할 데이터 파일을 선택하면 된다.

단, 해당 기능은 REUSE_OID 옵션을 사용하여 생성한 테이블에서는 사용할 수 없으며, [OID 정보 보기] 기능을 ON했을 경우에 한해서만 지원한다. [OID 정보 보기]를 ON하는 방법은 아래의 그림과 같이 질의 편집기 툴 바에서 선택할 수 있으며, 기본값으로 설정하고자 할 때는 해당 호스트의 [속성> 질의 편집기] 옵션에서 [OID 정보 보기]를 선택한다.



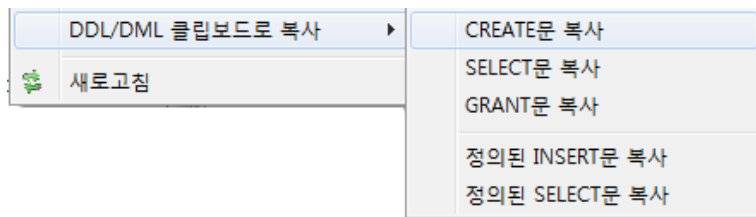
정의된 SQL 실행 기능 추가

CUBRID 매니저에서 사용자가 작성한 정의된 SQL(Prepared Statement)문을 이용하여 데이터를 입력하고 조회할 수 있는 기능을 지원한다. 이를 통해 대량의 데이터를 좀 더 쉽고 빠르게 데이터베이스에 입력할 수 있으며 사용자가 문자 집합과 동시 수행 개수 그리고 커밋 주기 등을 직접 설정할 수 있다.



DDL/DML을 클립보드로 복사하는 기능 추가

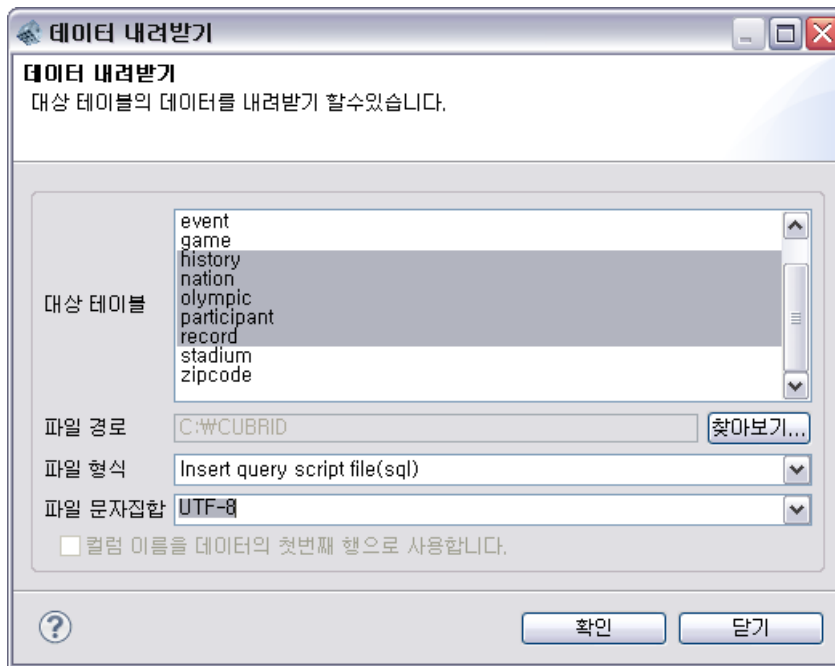
[DDL/DML 클립보드로 복사] 기능은 선택한 테이블의 DDL과 DML을 클립보드로 복사하여 다른 편집도구에서 사용하거나 저장할 수 있도록 지원한다. 제공하는 기능으로는 CREATE문, SELECT문, GRANT문과 Prepared Statement문으로 작성된 INSERT문과 SELECT문이 있다.



개선된 기능

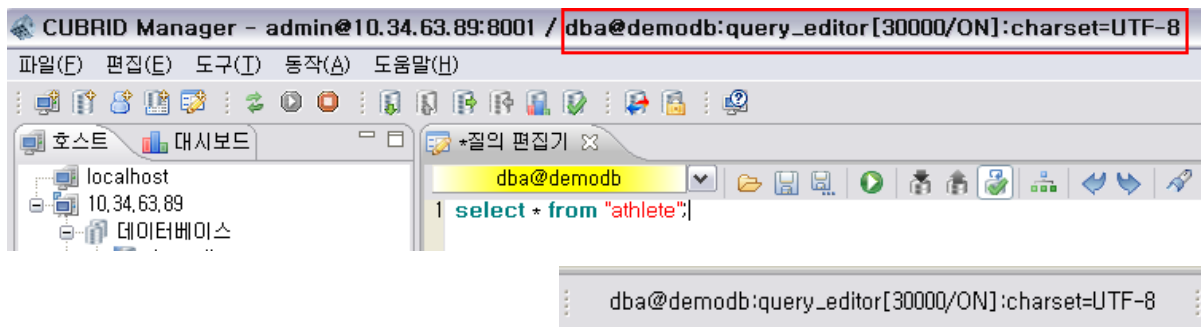
데이터 올리기/내려받기 할 때 문자 집합 설정 및 확인할 수 있도록 개선

데이터 올리기 및 내려받기에서 데이터의 문자 집합을 지정할 수 있도록 개선하였다. 뿐만 아니라, 다중 테이블을 선택하여 한번에 내려받을 수 있다.



질의 편집기에서 현재 사용중인 문자 집합을 확인할 수 있도록 개선

여러 개의 질의 편집기를 사용할 때 현재 편집기에서 사용중인 문자 집합 값을 쉽게 확인할 수 있도록, 매니저의 타이틀 바와 상태 바에 현재 질의 편집기에서 접속한 데이터베이스의 문자 집합 정보를 출력하도록 개선되었다.

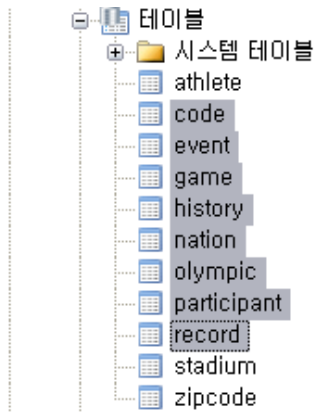


자동 볼륨 추가 설정 시 최대값을 사용자가 설정 할 수 있도록 개선

이전 버전은 자동 볼륨 추가 설정의 최대값이 2G로 내부 설정되어 있었으나, 최대값을 사용자가 설정할 수 있도록 개선하였다.

DELETE ALL, TRUNCATE TABLE 등을 여러 개의 테이블에 한번에 적용할 수 있도록 개선

테이블 탐색창에서 테이블을 다중 선택하여 DELETE ALL, TRUNCATE TABLE, 데이터 내려받기, 테이블 삭제, DDL/DML 클립보드로 복사 기능을 수행할 수 있도록 개선하였다.



브로커 상태 모니터 차트에서 SESSION과 ACTIVE SESSION이 구분되도록 개선

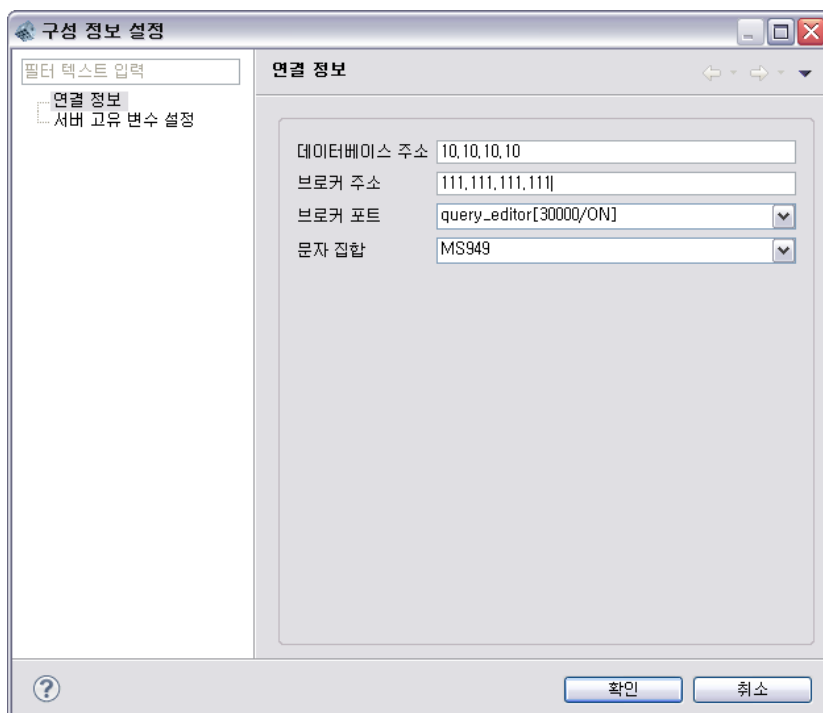
브로커 상태 모니터 차트 설정 시 기존에는 Active Session만 있던 항목을 Active Session과 Session 두 항목으로 구분되도록 개선하였다.

기존 버전에서 Active Session은 브로커 프로세스(이하 CAS) 상태가 IDLE이 아닌 모든 것을 의미하였으나, 2008 R3.1에서 Active Session은 CAS 상태가 BUSY인 것만 나타내며, Session은 CAS 상태가 IDLE이 아닌 모든 것을 의미하도록 수정하였다.

다중 네트워크 사용 환경에서 데이터베이스 서버 접속에 실패하는 문제 개선 New

CUBRID 매니저 클라이언트에서 접속할 때 사용하는 IP와 데이터베이스 서버에 접속할 때 사용하는 IP의 망(Network)이 분리되어 있는 경우, CUBRID 매니저에서 호스트 접속을 하면 서버 내부에서도 데이터베이스 접속을 같은 IP로 시도하게 되어 데이터베이스에 접근하지 못하는 문제가 발생하는데, 데이터베이스 서버에서 사용하는 IP를 호스트 속성에 추가하여 네트워크 망이 분리되어 있는 환경에서도 CUBRID 매니저를 정상 수행할 수 있도록 하였다.

호스트 > 데이터베이스 > 데이터베이스 명에서 마우스 오른쪽 버튼을 클릭하고, [속성] 메뉴를 선택하면 연결 정보에서 데이터베이스 주소를 설정할 수 있다.



수정된 사항

호스트 상태 모니터에서 메모리 정보 출력 시 물리적 메모리 외의 영역을 포함하는 오류 수정

호스트 상태 모니터의 사용 메모리 영역에 물리적으로 사용한 메모리 영역 외에 캐시와 버퍼 영역을 포함하여 출력하던 오류가 있었으나, 실제 물리적으로 사용한 메모리만 출력되도록 수정하였다.

Windows XP 서비스팩 3에서 CUBRID 매니저로 데이터베이스 시작 시 발생 가능한 오류 수정

서비스팩 3가 설치된 Windows XP에서 CUBRID 매니저로 데이터베이스를 시작하면 매우 드물게 프로세스 정보 수집 오류로 인해 "Database(demodb) is running in standalone mode" 라는 메시지가 발생하는 경우가 있었으나, 이를 수정하였다.

Windows 7에서 데이터베이스 상태 정보의 일부를 확인할 수 없는 오류 수정

Windows 7에서 CUBRID 매니저로 데이터베이스 상태 화면을 활성화 시키면 수직 스크롤 바가 생성되지 않아 DB 볼륨 정보 그래프를 볼 수 없었던 오류를 수정하였다.

문법에 맞지 않는 일부 질의에 대해 에러가 출력되지 않는 오류 수정

CUBRID 매니저의 질의 분석기 오류로 인해 SQL 문법에 맞지 않는 일부 질의가 입력된 경우 에러 메시지가 출력되지 않는 문제를 수정하였다.

5. 주의 사항

새로 추가된 주의 사항

CUBRIDSUS-3826 GLO 클래스 지원 중단에 따른 주의 사항

CUBRID 2008 R3.0 이하 버전은 GLO(Generalized Large Object) 클래스를 사용하여 Large Object를 처리하였으나, CUBRID 2008 R3.1은 GLO 클래스를 제거하고 BLOB, CLOB 타입(이하 LOB)을 지원한다. ([관련 매뉴얼 참고](#))

기존의 GLO 클래스 사용자는 다음과 같이 작업할 것을 권장한다.

- GLO 데이터를 파일로 저장한 후 어플리케이션 및 DB 스키마에서 GLO를 사용하지 않도록 수정한다.
- 데이터베이스 마이그레이션을 한다. (본 문서의 [데이터베이스 마이그레이션 절차 참고](#))
- 변경한 어플리케이션에 맞게 파일을 LOB 데이터로 로딩하는 작업을 수행하도록 한다.
- 수정한 어플리케이션이 정상 동작하는지 확인한다.

참고로, cubrid loaddb 유틸리티는 GLO 클래스를 상속받거나 GLO 클래스 타입을 가진 테이블을 로딩하려는 경우, `Error occurred during schema loading` 에러 메시지와 함께 데이터 로딩을 중지한다.

CUBRIDSUS-4172 BLOB, CLOB 타입 사용 시 제약 사항

BLOB, CLOB 타입(이하 LOB)에 대하여 다음과 같은 제약 사항이 있으므로 사용에 주의한다.

- LOB 타입 컬럼 간 비교 연산(=, <>, IN, NOT IN 등)을 할 수 없으며, 이를 위해서는 문자열 또는 비트열로 타입을 변환한 후 사용해야 한다. 단, IS NULL, IS NOT NULL은 지원한다.
- PRIMARY KEY, FOREIGN KEY, UNIQUE, NOT NULL 제약 조건 또는 인덱스를 정의할 수 없다.
- 테이블 생성 및 수정 시, SHARED 속성을 정의할 수 없으며 DEFAULT 속성은 NULL 값에 대해서만 정의할 수 있다.
- 데이터베이스에는 파일의 위치(LOB Locator)가 저장되고 데이터는 파일로 저장되는 구조이므로, 장애가 발생하여 특정 시점으로 복구할 때 LOB Locator와 LOB 데이터의 매핑이 유효하지 않아 에러가 발생할 수 있다.
- ALTER TABLE DROP 문을 사용하여 컬럼을 삭제하거나, DROP TABLE 문을 사용하여 테이블을 삭제하는 경우 LOB Locator만 삭제되고 LOB 컬럼이 참조하는 외부 파일 시스템의 LOB 파일은 삭제되지 않고 남아있다.
- CUBRID가 제공하는 API나 CUBRID 매니저, csq를 사용하지 않고 사용자 임의로 LOB 타입의 데이터 파일을 직접 수정하면 내용이 일치됨을 보장할 수 없다.

(자세한 설명은 [관련 매뉴얼 참고](#))

CUBRIDSUS-3926 Windows에서 브로커 파라미터의 동적 변경 제약 사항

Windows 환경에서는 broker_changer를 사용하여 브로커 파라미터를 동적으로 변경할 수 없으므로 주의한다. (본 문서의 [3. CUBRID 2008 R3.1에서 변경된 사항 > CUBRIDSUS-3926 참고](#))

```
// LONG_TRANSACTION_TIME 파라미터를 동적으로 변경하는 예
% broker_changer broker1 LONG_TRANSACTION_TIME 30.00
```

CUBRIDSUS-4186 Windows Vista 이상 버전에서 CUBRID 유틸리티를

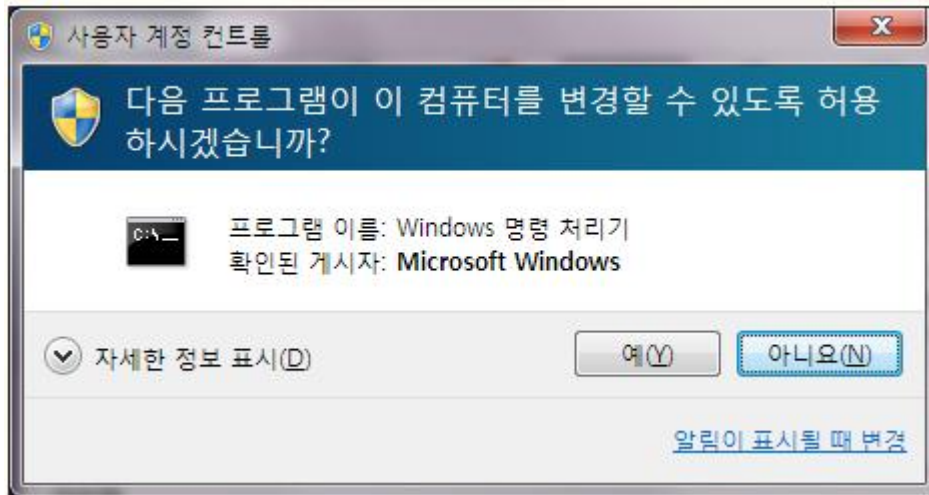
사용한 서비스 제어 시 권장 사항 Updated

Windows Vista 이상 버전에서 cubrid 유틸리티를 사용하여 서비스를 제어하려면 명령 프롬프트 창을 관리자 권한으로 구동한 후 사용하는 것을 권장한다.

명령 프롬프트 창을 관리자 권한으로 구동하지 않고 cubrid 유틸리티를 사용하는 경우 UAC(User Account Control) 대화 상자를 통하여 관리자 권한으로 수행될 수 있으나 수행 결과 메시지를 확인할 수 없다.

Windows Vista 이상 버전에서 명령 프롬프트 창을 관리자 권한으로 구동하는 방법은 다음과 같다.

- [시작> 모든 프로그램> 보조 프로그램> 명령 프롬프트]에서 마우스 오른쪽 버튼을 클릭한다.
- [관리자 권한으로 실행(A)]을 선택하면 아래와 같은 권한 상승을 확인하는 대화 상자가 활성화된다.



- “예”를 클릭하면 명령 프롬프트 창이 관리자 권한으로 구동된다.

기존 주의 사항

CUBRIDSUS-3217 JDBC에서 연결 정보를 URL 스트링으로 입력하는 경우

물음표를 반드시 명시

JDBC에서 URL 스트링으로 연결 정보를 입력하는 경우, 이전 버전에서는 물음표(?)를 입력하지 않더라도 속성(Property) 정보가 적용되었으나, CUBRID 2008 R3.0부터는 문법에 따라 반드시 물음표를 명시하여야 하고 이를 생략할 경우 에러를 출력한다. 또한, 연결 정보 중 USERNAME과 PASSWORD가 없더라도 반드시 콜론(:)을 명시하여야 한다.

```
URL=jdbc:CUBRID:127.0.0.1:31000:db1::althosts=127.0.0.2:31000,127.0.0.3:31000 -에러 처리
URL=jdbc:CUBRID:127.0.0.1:31000:db1::?althosts=127.0.0.2:31000,127.0.0.3:31000 -정상 처리
```

CUBRIDSUS-3564 마스터 프로세스와 서버 프로세스 간 프로토콜 변경 및

두 개 버전을 동시에 운영하는 경우 포트 설정 필요

마스터 프로세스(cub_master)와 서버 프로세스(cub_server) 간 통신 프로토콜 변경으로 인해 CUBRID 2008 R3.0 이상 버전의 마스터 프로세스는 하위 버전의 서버 프로세스와 통신할 수 없고, 하위 버전의 마스터 프로세스도 2008 R3.0 이상 버전의 서버 프로세스와 통신할 수 없다. 따라서, 이미 하위 버전이 설치되어 있는 환경에서 새 버전을 추가 설치하여, 두 개 버전의 CUBRID를 동시에 운영하는 경우, 각각 서로 다른 포트를 사용하도록 **cubrid.conf**의 **cubrid_port_id** 파라미터를 수정하여야 한다.

CUBRIDSUS-2828 데이터베이스 이름에 @를 포함할 수 없음

데이터베이스 이름에 @이 포함되는 경우 호스트 이름이 명시된 것으로 해석될 수 있으므로, 이를 방지하기 위하여 **cubrid createdb**, **cubrid renamedb**, **cubrid copydb** 유틸리티 실행 시 데이터베이스 이름에 @를 포함할 수 없도록 수정하였다.

CUBRIDSUS-3267 Windows 환경에서 디렉터리 경로 설정 시 주의 사항

Windows 환경에서 CUBRID 설치 디렉터리 경로에 공백을 포함하는 경우 정상 설치가 되지 않으므로 주의한다. 또한, DB 언로드/로드/백업 등의 작업 대상 디렉터리 경로에도 공백을 포함할 수 없다.

CUBRIDSUS-3553 CUBRID 소스 빌드 후 실행 시, 매니저 서버 프로세스

관련 오류 발생

사용자가 직접 빌드하여 설치하는 경우, CUBRID와 CUBRID 매니저를 각각 빌드하여 설치하여야 한다. 만약, CUBRID 소스만 checkout하여 빌드 후 **cubrid service** 또는 **cubrid manager**를 실행하면, `cubrid manager server is not installed`라는 오류가 발생한다.

CUBRID 매니저에서 [볼륨 자동 추가 기능 사용] 옵션을 선택하고 DB를

생성하는 경우 주의 사항

[볼륨 자동 추가 기능 사용] 옵션은 DB 볼륨이 사용자가 설정한 "여유 공간 비율" 값 이하가 될 경우 해당 볼륨을 자동으로 추가하는 기능이며, 이 옵션이 선택되면 여유 공간을 모니터링하기 위해 주기적(디폴트 5초)으로 **cubrid spacedb** 유틸리티를 수행하므로 에러 로그(<dbname>_spacedb.err)가 증가할 수 있다. 사용자는 **cm.conf** 파일의 **monitor_interval** 파라미터를 설정하여 모니터링 주기를 조정할 수 있다.