

---

# CUBRID 2008 R4.1 Patch 1

## 릴리스 노트

---

## 목차

1. 개요.....	6
릴리스 노트 정보.....	6
릴리스 노트 개정 내역.....	6
참고 문서.....	6
버그 리포트 및 사용자 피드백 제공 방법 안내.....	6
추가 정보 안내.....	6
2. CUBRID 2008 R4.1 정보.....	7
CUBRID 2008 R4.1 PATCH 1 릴리스 특징.....	7
CUBRID 2008 R4.1 릴리스 특징.....	7
대량의 트랜잭션 테스트에서 INSERT와 UPDATE 성능 70% 향상.....	8
다양한 SQL 함수 및 정규 표현식 추가.....	8
브로커 파라미터 및 상태 정보 추가.....	8
50여 개의 발견된 버그 수정.....	8
자동 커밋 모드 기본값, IFNULL 등 NULL 조건 함수들의 반환 타입, 날짜 값으로 '0000-00-00' 지원 등의 동작 방식 변경.....	8
지원 플랫폼 및 설치 권장 사양.....	8
라이선스 안내.....	9
버전 호환성과 운용성.....	9
응용 프로그램의 호환성.....	9
CUBRID 매니저의 호환성.....	9
데이터베이스 호환성.....	9
상호 운용성.....	10
CUBRID 2008 R4.1의 설치 방법.....	10
제품 설치.....	10
CUBRID 환경 변수 및 OS 환경 변수 설정.....	10
CUBRID 2008 R4.1로 업그레이드하는 방법.....	10
2008 R4.0에서 2008 R4.1로 업그레이드하기.....	10
2008 R4.0 베타 이전 버전에서 2008 R4.1로 업그레이드하기.....	10
업그레이드 주의 사항.....	10
데이터베이스 마이그레이션 절차.....	11
HA 환경에서 데이터베이스 마이그레이션 절차.....	12
복제 기능 사용 시 주의 사항.....	13
3. CUBRID 2008 R4.1 PATCH 1에서 변경된 사항.....	14
새로 추가된 기능.....	14
CUBRIDSUS-6070 JDBC 연결 URL에 디버깅용 로그를 기록하도록 설정하는 기능 추가.....	14
개선된 성능.....	14
CUBRIDSUS-6648 CCI, PHP 인터페이스에서 한번의 호출로 PREPARE와 EXECUTE를 수행할 수 있게 개선하고, 요청 핸들 종료 함수 호출 시 메시지 전송횟수를 줄이도록 수정.....	14
CUBRIDSUS-6906 JDBC 드라이버의 성능 개선.....	14
수정된 사항.....	14
CUBRIDSUS-6678 데이터베이스 서버 프로세스 재시작 시 데이터베이스 볼륨이 정상적으로 복구되지 못하는 오류 수정.....	14
CUBRIDSUS-6881 IFNULL/NVL/NVL2/COALESCE 함수에 호스트 변수를 사용하여 INSERT 수행 시 발생하는 오류 수정.....	14
CUBRIDSUS-6689 NOT 뒤에 단항 연산자 사용 시 'unknown opcode' 오류 수정.....	15
CUBRIDSUS-6778 인덱스 스캔 도중 키 잠금을 획득/해제하는 과정에서 라이브락 현상이 발생할 수 있는 문제 수정.....	15
CUBRIDSUS-6914 CCI 응용이 DATASOURCE를 이용하여 데이터베이스에 접속한 상태에서 연결 종료 함수를 호출하는 경우 응용 프로그램이 비정상 종료될 수 있는 현상 수정.....	15
CUBRIDSUS-6925 cci_set_query_timeout() 함수로 설정한 질의 타임아웃 동작 오류 수정.....	15
CUBRIDSUS-6958 CCI, PHP 응용 프로그램에서 접속 실패 시 fallback이 되어도 첫 번째로 설정한 호스트에는 접속을 시도하지 않는 문제 수정.....	15

CUBRIDSUS-6753 인덱스에서 B+트리 노드의 분할/병합과 INSERT/DELETE/UPDATE가 동시에 일어날 때 "Unknown key" 오류 수정 .....	15
CUBRIDSUS-6897 CCI 기반 응용이 두 개 이상의 브로커에 접속하는 환경에서, 어떤 질의가 5초 이상 수행되면 접속 오류가 발생할 수 있는 문제 수정 .....	15
CUBRIDSUS-6779 응용 프로그램과 브로커 응용 서버(CAS) 연결 직후 둘 사이의 상태 정보가 일치하지 않는 현상 수정 .....	16
CUBRIDSUS-6754 브로커 응용 서버(CAS)의 개수를 100으로 설정하는 경우 브로커 구동에 실패하는 문제 수정 .....	16
CUBRIDSUS-6364 브로커 응용 서버(CAS)가 SQL 로그를 쓰는 도중 멈출(hang) 수 있는 현상 수정 .....	16
CUBRIDSUS-6889 HA 복제 반영 프로세스가 복제를 진행하지 못할 수 있는 문제 수정 .....	16
CUBRIDSUS-6905 JDBC에서 집합형 데이터 타입을 사용하면 브로커 응용 서버(CAS)와 JDBC 응용 프로그램이 멈추는(hang) 현상 수정 .....	16
<b>4. CUBRID 2008 R4.1에서 변경된 사항 .....</b>	<b>17</b>
<b>새로 추가된 기능 .....</b>	<b>17</b>
CUBRIDSUS-5860 ADDTIME 함수 추가 .....	17
CUBRIDSUS-6265 ASCII 함수 추가 .....	17
CUBRIDSUS-5860 BIN 함수 추가 .....	17
CUBRIDSUS-6233 CONV 함수 추가 .....	17
CUBRIDSUS-5860 FIND_IN_SET 함수 추가 .....	17
CUBRIDSUS-6233 HEX 함수 추가 .....	17
CUBRIDSUS-6190 현재의 시리얼 값과 하나 이상의 시리얼 값을 반환하는 함수들 추가 .....	17
CUBRIDSUS-5903 정규 표현식에 대한 연산자 추가 .....	17
CUBRIDSUS-5616 질의 실행 시간을 제한하는 브로커 파라미터 추가 .....	18
CUBRIDSUS-6197 JDBC 연결 URL에 질의 타임아웃 시간을 설정하는 기능 추가 .....	18
CUBRIDSUS-6198 JDBC 연결 URL에 데이터베이스 연결 타임아웃 시간을 설정하는 기능 추가 .....	18
CUBRIDSUS-5388 브로커 응용 서버의 상태를 출력할 때 트랜잭션 시작 시간을 추가 .....	18
CUBRIDSUS-6199 브로커 응용 서버의 상태를 출력할 때 응용 프로그램이 CAS에 접속한 회수, CAS의 재구동 회수를 추가 .....	18
CUBRIDSUS-6128 여러 심각성 수준이 NOTIFICATION일 때 교착상태가 발생하면 서버 여러 로그 파일에 잠금 관련 정보를 기록하는 기능 추가 .....	18
<b>개선된 성능 .....</b>	<b>19</b>
CUBRIDSUS-5300 트랜잭션 로그의 동시 처리 성능 개선 .....	19
CUBRIDSUS-5687 메모리 버퍼의 데이터 페이지를 디스크에 쓰는 방식을 수정하여 쓰기 성능 개선 .....	19
CUBRIDSUS-5297 현재의 날짜/시간 정보를 구하는 함수의 실행 성능 개선 .....	19
CUBRIDSUS-6570 ORDER BY 컬럼에 대해 BETWEEN ... AND 조건인 경우에도 다중 키 범위 조건의 정렬 수행 최적화가 가능하도록 개선 .....	19
CUBRIDSUS-5295 질의 실행 계획 캐시에 INSERT 질의가 포함되도록 개선 .....	19
CUBRIDSUS-5850 시스템의 부하가 매우 큰 경우 트랜잭션 처리 응답 시간의 최저 값과 최고 값의 편차를 줄이도록 개선 .....	19
CUBRIDSUS-5824 HA 환경에서 복제 지연 시 성능 개선 .....	19
<b>수정된 사항 .....</b>	<b>20</b>
CUBRIDSUS-5882 날짜/시간 타입에 대해 '0000-00-00' 또는 '0000-00-00 00:00:00' 형식의 입력 지원 .....	20
CUBRIDSUS-5296 시리얼(SERIAL)을 포함하는 질의문의 질의 계획을 질의 실행 계획 캐시에 저장할 수 있도록 수정 .....	20
CUBRIDSUS-6138 ABS 등의 수치 연산 함수들에서 입력 인자가 호스트 변수이고 문자열을 바인딩하는 경우 질의 결과가 잘못되는 오류 수정 .....	20
CUBRIDSUS-5988 TO_CHAR 함수의 첫번째 인자가 호스트 변수이고 DATE 타입의 값이 바인딩되면 발생하는 오류 수정 .....	20
CUBRIDSUS-6280 STR_TO_DATE 함수에서 '%r' 포맷 지정 시 PM 시각 문자열이 타입 변환이 되지 않거나 '12:XX:XX AM' 문자열이 '12:XX:XX PM'으로 잘못 변환되는 오류 수정 .....	20
CUBRIDSUS-6023 조건절에 AND .. OR 조건이 포함된 경우 OR의 왼쪽 항이 AND로 연결된 조건이면 질의 결과가 잘못될 수 있는 오류 수정 .....	20
CUBRIDSUS-6231 NULL 입력 여부에 따라 결과 값을 정하는 SQL 함수들에서 입력 타입에 따른 결과 타입의 변환 규칙 변경 .....	21
CUBRIDSUS-6033 커버링 인덱스로 WHERE 절에 IN 조건식이 있는 질의의 수행 도중 발생하는 메모리 누수 현상 수정 .....	21
CUBRIDSUS-6345 커버링 인덱스가 사용되며 연관 부절의 존재하는 질의의 경우 잘못된 결과가 출력될 수 있는 오류 수정 .....	21
CUBRIDSUS-6649 조인 질의문에 LIMIT 조건이 주어지고 인덱스를 이용한 LIMIT 최적화가 적용되면 잘못된 결과를 출력할 수 있는 문제 수정 .....	21
CUBRIDSUS-6322 계층 질의의 수행 시 서버가 비정상 종료될 수 있는 문제 수정 .....	21
CUBRIDSUS-6282 파티션 테이블 또는 부모 테이블에서 인덱스 관련 제약조건을 DROP 후 재생성하면 서버 프로세스가 비정상 종료되는 문제	

수정 .....	22
CUBRIDSUS-6352 연관 부질의 포함 질의가 잘못된 결과를 출력하는 문제 수정 .....	22
CUBRIDSUS-6404 INSERT 또는 UPDATE 도중 인덱스 변경과 관련하여 발생하는 오류 수정 .....	22
CUBRIDSUS-5808 서로 다른 DB 연결들이 하나의 세션 ID를 참조할 수 있는 오류 수정 .....	22
CUBRIDSUS-6054 "SELECT COUNT(DISTINCT 상수) FROM tbl"의 결과를 1건으로 출력하도록 수정 .....	22
CUBRIDSUS-6370 다중 컬럼 인덱스를 구성하는 컬럼의 NULL 값을 UPDATE하면 오류가 발생하는 문제 수정 .....	22
CUBRIDSUS-6347 PREPARE 실패 시 cubrid broker status의 에러 카운트를 증가하도록 수정 .....	23
CUBRIDSUS-5879 CCI 인터페이스에서 자동 커밋 모드의 기본값이 ON이 되도록 수정 .....	23
CUBRIDSUS-6220 CCI, PHP, ODBC 응용 프로그램에서 어떤 질의 요청 핸들러(request handler)를 CLOSE 할 때 다른 질의 요청 핸들러에 대한 커밋에 영향을 주는 오류 수정 .....	23
CUBRIDSUS-6327 CCI, PHP, ODBC, OLE DB 등의 응용 프로그램에서 PREPARE 함수 수행 중 오류가 발생하면 이후 트랜잭션 롤백을 실행해도 롤백되지 않는 현상 수정 .....	23
CUBRIDSUS-6366 Linux 환경의 CCI 인터페이스 기반 응용 프로그램에서 네트워크 소켓 fd 값이 1024를 초과하는 경우 비정상 동작하는 문제 수정 .....	23
CUBRIDSUS-6491 cci_datasource_borrow 함수 호출 시 연결을 가져오는 대기 시간까지 기다리지 않고 오류를 반환할 수 있는 문제 수정 .....	23
CUBRIDSUS-6606 CCI 인터페이스의 DATASOURCE 사용 시 statement 풀링을 설정하면 발생하는 메모리 누수 현상 수정 .....	23
CUBRIDSUS-6673 CCI 인터페이스로 개발된 응용에서 COMMUNICATION ERROR 발생 시 비정상 동작하는 현상 수정 .....	24
CUBRIDSUS-6318 JDBC에서 Java의 BigDecimal 클래스를 사용하여 BIGINT 타입의 값을 가져오는 경우 잘못된 값을 가져오는 오류 수정 .....	24
CUBRIDSUS-6290 JDBC 드라이버에서 한 번의 요청으로 입력 가능한 데이터의 크기 제한을 제거 .....	24
CUBRIDSUS-6453 독립 모드로 실행한 CSQL 인터프리터에서 사용자 세션 변수로 PREPARE 문을 수행하면 비정상 종료하는 현상 수정 .....	24
CUBRIDSUS-6106 addvoldb, spacedb 유틸리티를 SA 모드로 수행할 때 16MB의 데이터 버퍼를 사용하도록 수정 .....	24
CUBRIDSUS-6235 CUBRID 매니저와 같은 응용 프로그램에서 장시간 동안 요청이 없는 경우에도 연결이 종료되지 않도록 수정 .....	24
CUBRIDSUS-5472 질의 수행 중 인터럽트가 발생하는 경우 CSQL 또는 브로커 응용 서버(CAS)가 비정상 종료되는 문제 수정 .....	24
CUBRIDSUS-5816 크기를 지정하는 CUBRID 유틸리티의 옵션 혹은 시스템 파라미터에 소수점 이하의 숫자 입력을 허용하도록 수정 .....	25
CUBRIDSUS-6275 인덱스가 삭제되는 작업 도중에 인터럽트가 발생하면 나타날 수 있는 인덱스 깨짐 현상 수정 .....	25
CUBRIDSUS-6418 온라인 백업 수행 도중 DDL을 동시에 수행하면 서버 프로세스가 동작을 멈추는(hang) 문제 수정 .....	25
CUBRIDSUS-6279 인덱스 리프 노드(leaf node)의 이전 링크가 잘못되는 오류 수정 .....	25
CUBRIDSUS-6001 다른 트랜잭션이 입력 도중 실패한 공간을 사용하기 위해 대기하게 되는 현상 수정 .....	25
CUBRIDSUS-6396 UPDATE 도중 인터럽트가 발생하면 데이터베이스 볼륨 깨짐(crash) 현상 수정 .....	25
CUBRIDSUS-6720 부울리언 표현식 처리 과정에서 서버 프로세스의 메모리 누수 현상 수정 .....	25
CUBRIDSUS-6072 응용프로그램에서 트랜잭션을 종료하고 질의 요청 핸들을 CLOSE한 이후 일정 시간 동안 요청이 없는 경우 연결이 종료되는 문제 수정 .....	25
CUBRIDSUS-6342 체크포인트 오류로 인해 서버가 비정상 종료된 이후 데이터베이스 복구에 실패할 수 있는 문제 수정 .....	26
CUBRIDSUS-6513 서버 프로세스의 오류 로그에 소스의 파일 이름과 라인 번호가 항상 출력되도록 수정 .....	26
CUBRIDSUS-6240 SLOW SQL 로그 기록 시 질의 수행 시작 시간과 완료 시간이 모두 기록되도록 수정 .....	26
CUBRIDSUS-5956 브로커 응용 서버(CAS) 프로세스의 메모리 사용량이 급격히 증가하는 경우를 차단하기 위한 방법 제공 .....	26
CUBRIDSUS-5851 replica 노드에서는 ASYNC 모드로만 트랜잭션 로그를 복사하도록 수정 .....	26
CUBRIDSUS-4817 HA 환경에서 한 트랜잭션 내 변경 레코드 개수가 매우 많은 경우 메모리 부족으로 인해 복제 반영이 되지 않는 문제 수정 .....	26
CUBRIDSUS-6117 HA 환경에서 2008 R4.0 Patch2 버전 이후 발생한 applylogdb 프로세스의 성능 퇴행 개선 .....	26
CUBRIDSUS-6172 HA 환경에서 failover 수행 이후, 슬레이브에서 마스터로 변경된 노드에 복제된 보관 로그 파일이 삭제되지 않고 계속 남아 있을 수 있는 현상 수정 .....	27
CUBRIDSUS-6183 HA 환경에서 applylogdb 프로세스가 쓰기가 끝나지 않은 로그 페이지를 버퍼에 캐싱하여 읽고 반영하면서 오동작하는 문제 수정 .....	27
CUBRIDSUS-6193 HA 환경에서 마스터 서버 테이블의 기본 키를 변경한 직후 온라인 백업본으로 슬레이브 서버를 재구축하면 슬레이브 서버 프로세스가 비정상 종료될 수 있었으나 오류 처리하도록 수정 .....	27
CUBRIDSUS-5475 HA 환경의 슬레이브 서버에서 INCR/DECR 함수를 호출하면 해당 데이터를 업데이트할 수 없도록 수정 .....	27
CUBRIDSUS-5939 HA 환경에서 standby 상태의 노드를 maintenance 상태로 변경할 때 진행 중이던 트랜잭션이 정상 종료될 때까지 대기하는 시간을 설정하는 기능 제공 .....	27
CUBRIDSUS-6154 데이터 버퍼의 더티 페이지가 디스크에 저장되는 주기를 설정하는 시스템 파라미터의 값이 -1일 때의 동작 방식 변경 .....	27
CUBRIDSUS-6133 교착상태를 감지하는 주기를 설정하는 시스템 파라미터 값의 입력 단위를 소수점 이하 한자리까지 가능하도록 변경 .....	28

CUBRIDSUS-6083 스택 크기를 결정하는 기준 및 크기를 설정하는 시스템 파라미터의 기본값 변경.....	28
CUBRIDSUS-6668 데이터 캐싱 버퍼의 크기를 설정하는 시스템 파라미터의 최소값 변경.....	28
CUBRIDSUS-6035 Linux 용 CUBRID 패키지를 기존 CUBRID 버전이 설치되어 있는 경로에 설치하면 파일들이 잘못 설치되는 오류 수정.....	28
CUBRIDSUS-6061 JDK1.7로 JDBC 드라이버 소스를 빌드하지 못하는 문제 수정.....	28
CUBRIDSUS-5942 Windows 시스템을 재부팅한 이후 CUBRID Service Tray가 자동으로 구동될 수 있는 환경인지 CUBRID 설치 중에 미리 확인이 가능하도록 수정.....	28
CUBRIDSUS-6028 Windows 용 CUBRID 패키지 설치 후 시스템을 재부팅해야 cubrid 유틸리티가 실행되는 문제 수정.....	28
CUBRIDSUS-6278 Windows용 CCI 라이브러리(cascci.dll)를 사용한 응용 프로그램 수행 시 CRT DLL이 필요하지 않도록 수정.....	28
<b>5. 주의 사항.....</b>	<b>29</b>
<b>2008 R4.1 신규 주의 사항.....</b>	<b>29</b>
CUBRIDSUS-5879 2008 R4.1 버전부터 CCI_DEFAULT_AUTOCOMMIT 의 기본값이 ON으로 바뀜.....	29
CUBRIDSUS-5238 2008 R4.1 버전은 2008 R4.0 Beta 및 그 이전 버전과 데이터베이스 볼륨이 호환되지 않음.....	29
<b>기존 주의 사항.....</b>	<b>29</b>
CUBRIDSUS-5597 CCI, PHP, JDBC 연결 URL에서 구분자를 암호에 포함할 수 없음.....	29
CUBRIDSUS-5136 페이지 단위의 옵션 제거 예정.....	29
CUBRIDSUS-4222 데이터베이스 볼륨 크기 설정 시 주의 사항.....	30
CUBRIDSUS-4222 페이지 단위의 시스템 파라미터 제거 예정.....	30
CUBRIDSUS-4095 일부 시스템 파라미터들의 기본값 변경.....	30
CUBRIDSUS-5375 시스템 파라미터를 잘못 설정하면 데이터베이스 서비스, 유틸리티 및 응용 프로그램이 구동되지 않음.....	30
CUBRIDSUS-4524 복제 기능 제거.....	30
CUBRIDSUS-5228 CUBRID 매니저 설치 패키지 별도 제공.....	30
CUBRIDSUS-5097 컬럼 크기보다 큰 문자열을 INSERT/UPDATE 할 때 문자열이 절삭되어 입력됨.....	31
CUBRIDSUS-5349 CUBRID 32bit 버전에서 data_buffer_size에 2G를 초과하는 값을 설정하면 데이터베이스 구동에 실패함.....	31
CUBRIDSUS-4059 VARCHAR 타입의 컬럼에서 값을 가져올 때 커버링 인덱스가 적용되는 경우 뒤에 따르는 공백 문자열이 무시됨.....	31
CUBRIDSUS-3757 HA 관련 주의 사항.....	31
CUBRIDSUS-5071 데이터베이스 백업/복구 시 LOB 타입 저장소는 복구되지 않음.....	31
CUBRIDSUS-3826 GLO 클래스 지원 중단에 따른 주의 사항.....	31
CUBRIDSUS-4172 BLOB, CLOB 타입 사용 시 제약 사항.....	32
CUBRIDSUS-4186 Windows Vista 이상 버전에서 CUBRID 유틸리티를 사용한 서비스 제어 시 권장 사항.....	32
CUBRIDSUS-3217 JDBC에서 연결 정보를 URL 스트링으로 입력하는 경우 물음표를 반드시 명시.....	33
CUBRIDSUS-3564 마스터 프로세스와 서버 프로세스 간 프로토콜 변경 및 두 개 버전을 동시에 운영하는 경우 포트 설정 필요.....	33
CUBRIDSUS-2828 데이터베이스 이름에 @를 포함할 수 없음.....	33
CUBRIDSUS-3267 Windows 환경에서 디렉터리 경로 설정 시 주의 사항.....	33
CUBRIDSUS-3553 CUBRID 소스 빌드 후 실행 시, 매니저 서버 프로세스 관련 오류 발생.....	33

## 1. 개요

### 릴리스 노트 정보

본 문서는 CUBRID 2008 R4.1 Patch 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.

CUBRID 2008 R4.1 Patch 1 이전 버전에 대한 자세한 내용은 CUBRID 2008 R4.0의 최신 패치 버전 릴리스 노트를 참조한다.

### 릴리스 노트 개정 내역

CUBRID 2008 R4.1 버전 릴리스 노트의 변경 사항은 아래와 같다.

작성 날짜	설명
2012년 2월	CUBRID 2008 R4.1 Patch 1 릴리스 (8.4.1.1018)
2012년 1월	CUBRID 2008 R4.1 릴리스 (8.4.1.0564)

### 참고 문서

CUBRID 2008 R4.1 제품과 함께 배포되는 문서는 아래와 같다.

문서	설명
릴리스 노트	CUBRID 릴리스 버전의 특징 및 이전 버전에서 변경된 사항과 관련된 정보를 포함한다.
<a href="#">매뉴얼</a>	CUBRID 소개, 시작, CSQL 인터프리터, SQL 설명서, 관리자 안내서, 성능 튜닝, API 레퍼런스, CUBRID 매니저, CUBRID 매니저 관리자 안내서를 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: <a href="http://dev.naver.com/projects/cubrid/issue">http://dev.naver.com/projects/cubrid/issue</a>
사용자 피드백	CUBRID 오픈 소스 프로젝트: <a href="http://dev.naver.com/projects/cubrid/forum">http://dev.naver.com/projects/cubrid/forum</a> CUBRID 공식 사이트: <a href="http://www.cubrid.com">http://www.cubrid.com</a>

### 추가 정보 안내

CUBRID에 관한 유용한 정보는 아래 사이트에서 찾을 수 있다.

정보	사이트
CUBRID 제품 정보	<a href="http://cubrid.com/zbxe/product">http://cubrid.com/zbxe/product</a>
CUBRID 라이선스 정보	<a href="http://cubrid.com/zbxe/bbs_oss_guide/32249">http://cubrid.com/zbxe/bbs_oss_guide/32249</a>
CUBRID 사용자 문서	<a href="http://cubrid.com/zbxe/developer">http://cubrid.com/zbxe/developer</a>
CUBRID 교육 서비스	<a href="http://cubrid.com/zbxe/education_overview">http://cubrid.com/zbxe/education_overview</a>

## 2. CUBRID 2008 R4.1 정보

### CUBRID 2008 R4.1 Patch 1 릴리스 특징

CUBRID 2008 R4.1 Patch 1은 하위 패치 버전의 수정 사항들을 모두 포함하며 아래의 수정 사항들이 추가로 반영되었다. 특히, CUBRID 2008 R4.1에서 발견된 데이터베이스 복구 관련 오류를 포함하여, 데이터베이스 엔진의 심각한 오류 및 CCI 및 JDBC의 오류들이 수정되었으므로, 2008 R4.1 사용자는 2008 R4.1 Patch 1으로 반드시 업그레이드해야 한다. 이외에도, PHP의 `cubrid_query()`, `cubrid_execute()` 함수를 개선하였으며, 응용 프로그램과 브로커 응용 서버(CAS) 사이의 메시지를 줄여, SELECT 위주의 워크로드에 대하여 이전 버전 대비 약 45%의 성능 향상을 보였다.

이슈
<a href="#">CUBRIDSUS-6070 JDBC 연결 URL에 디버깅용 로그를 기록하도록 설정하는 기능 추가</a>
<a href="#">CUBRIDSUS-6648 CCI, PHP 인터페이스에서 한번의 호출로 PREPARE와 EXECUTE를 수행할 수 있게 개선하고, 요청 핸들 종료 함수 호출 시 메시지 전송횟수를 줄이도록 수정</a>
<a href="#">CUBRIDSUS-6906 JDBC 드라이버의 성능 개선</a>
<a href="#">CUBRIDSUS-6678 데이터베이스 서버 프로세스 재시작 시 데이터베이스 볼륨이 정상적으로 복구되지 못하는 오류 수정</a>
<a href="#">CUBRIDSUS-6881 IFNULL/NVL/NVL2/COALESCE 함수에 호스트 변수를 사용하여 INSERT 수행 시 발생하는 오류 수정</a>
<a href="#">CUBRIDSUS-6689 NOT 뒤에 단항 연산자 사용 시 'unknown opcode' 오류 수정</a>
<a href="#">CUBRIDSUS-6778 인덱스 스캔 도중 키 잠금을 획득/해제하는 과정에서 라이브 잠금 현상이 발생할 수 있는 문제 수정</a>
<a href="#">CUBRIDSUS-6914 CCI 응용이 DATASOURCE를 이용하여 데이터베이스에 접속한 상태에서 연결 종료 함수를 호출하는 경우 응용 프로그램이 비정상 종료될 수 있는 현상 수정</a>
<a href="#">CUBRIDSUS-6925 cci_set_query_timeout() 함수로 설정한 질의 타임아웃 동작 오류 수정</a>
<a href="#">CUBRIDSUS-6958 CCI, PHP 응용 프로그램에서 접속 실패 시 fallback이 되어도 첫 번째로 설정한 호스트에는 접속을 시도하지 않는 문제 수정</a>
<a href="#">CUBRIDSUS-6753 인덱스에서 B+트리 노드의 분할/병합과 INSERT/DELETE/UPDATE가 동시에 일어날 때 "Unknown key" 오류 수정</a>
<a href="#">CUBRIDSUS-6897 CCI 기반 응용이 두 개 이상의 브로커에 접속하는 환경에서, 어떤 질의가 5초 이상 수행되면 접속 오류가 발생할 수 있는 문제 수정</a>
<a href="#">CUBRIDSUS-6779 응용 프로그램과 브로커 응용 서버(CAS) 연결 직후 둘 사이의 상태 정보가 일치하지 않는 현상 수정</a>
<a href="#">CUBRIDSUS-6754 브로커 응용 서버(CAS)의 개수를 100으로 설정하는 경우 브로커 구동에 실패하는 문제 수정</a>
<a href="#">CUBRIDSUS-6364 브로커 응용 서버(CAS)가 SQL 로그를 쓰는 도중 멈춤(hang) 수 있는 현상 수정</a>
<a href="#">CUBRIDSUS-6889 HA 복제 반영 프로세스가 복제를 진행하지 못할 수 있는 문제 수정</a>
<a href="#">CUBRIDSUS-6905 JDBC에서 집합형 데이터 타입을 사용하면 브로커 응용 서버(CAS)와 JDBC 응용 프로그램이 멈추는(hang) 현상 수정</a>

보다 자세한 내용은 [CUBRID 2008 R4.1 Patch 1에서 변경된 사항](#)을 참고한다.

### CUBRID 2008 R4.1 릴리스 특징

대량의 트랜잭션 부하 테스트 시 INSERT와 UPDATE 성능이 이전 버전 대비 70% 향상되었고, 다양한 SQL 함수를 추가하였으며 조건 검색 시 정규 표현식을 지원한다. 또한, 브로커 파라미터의 추가 및 브로커 모니터링이 개선되었다. 아울러 50여개의 크고 작은 버그들을 수정하였다.

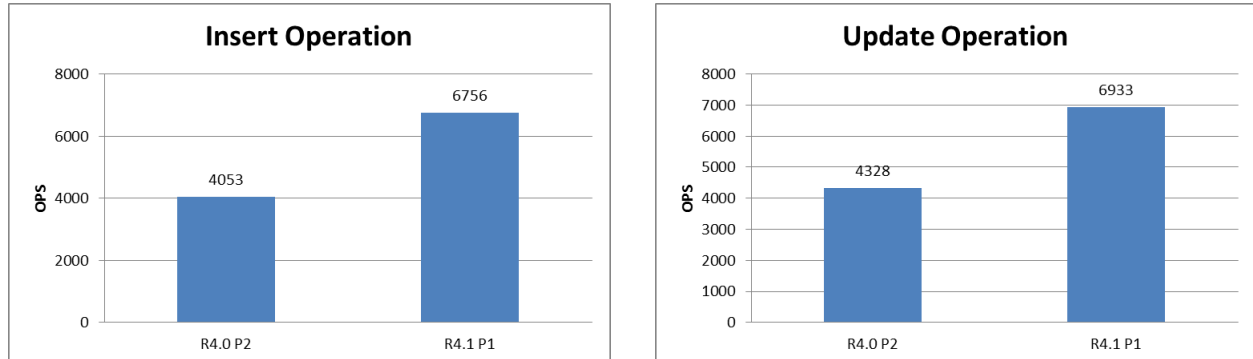
동작에 영향을 미치는 변경 중 특히 주의할 사항으로, 브로커 파라미터인 CCI\_DEFAULT\_AUTOCOMMIT의 기본값이 ON으로 변경되었다. 따라서 CCI 및 CCI로 개발된 PHP, ODBC, OLE DB 등의 드라이버 사용자가 자동 커밋 모드 기본값을 2008 R4.1 이전 버전과 같이 유지하려면 cubrid\_broker.conf의 CCI\_DEFAULT\_AUTOCOMMIT 값을 OFF로 설정해야 한다.

CUBRID 2008 R4.1 릴리스는 CUBRID 2008 R4.0 Patch 2 및 이전 버전의 수정 사항을 모두 포함한다.

CUBRID 2008 R4.1 버전의 주요 특징은 다음과 같다.

## 대량의 트랜잭션 테스트에서 INSERT와 UPDATE 성능 70% 향상

트랜잭션 로그의 동시 처리, 메모리 버퍼의 디스크 쓰기, HA 복제 등에서 내부 구현을 개선하였다. 그 결과, YCSB (<https://github.com/brianfrankcooper/YCSB/wiki>)에 기반한 대량의 트랜잭션 테스트 중 INSERT와 UPDATE에서 2008 R4.0 Patch 2 버전 대비 70%라는 매우 높은 성능 향상을 보였다. 성능 테스트와 관련한 자세한 내용은 "CUBRID 2008 R4.1 Patch1 QA Completion Report"를 참고한다. (아래 그림에서 Y축은 초당 질의 개수)



## 다양한 SQL 함수 및 정규 표현식 추가

ADDTIME, ASCII, BIN, CONV, FIND\_IN\_SET, HEX 등의 함수와 한 번의 호출로 여러 개의 시리얼 값을 얻을 수 있는 SERIAL\_NEXT\_VALUE 함수를 추가하였다.

또한, REGEXP 연산자의 지원으로 정규 표현식을 이용한 검색이 가능해졌다.

## 브로커 파라미터 및 상태 정보 추가

브로커 파라미터 MAX\_QUERY\_TIMEOUT을 통해 프로그램의 질의 실행 시간을 제한할 수 있게 하고, APPL\_SERVER\_MAX\_SIZE\_HARD\_LIMIT으로 브로커에서 응용 응용 서버(CAS)의 메모리 사용량이 초과되면 진행 중인 트랜잭션을 중단하고 CAS를 강제로 재시작할 수 있게 하였다.

또한, cubrid broker status 유틸리티의 출력 정보에 브로커의 각 CAS ID에 대한 트랜잭션 시작 시간, 응용 프로그램 접속 회수와 재구동 회수를 추가하였다.

## 50여 개의 발견된 버그 수정

SQL 함수, 인덱스, 질의문, DB 프로세스, HA 프로세스 등에서 발견된 50여 개의 버그를 수정하였다.

## 자동 커밋 모드 기본값, IFNULL 등 NULL 조건 함수들의 반환 타입, 날짜 값으로 '0000-00-00' 지원 등의 동작 방식 변경

브로커 파라미터인 CCI\_DEFAULT\_AUTOCOMMIT의 기본값이 ON으로 변경되었다.

IFNULL, NVL, NVL2, COALESCE 함수들의 입력 인자 타입들이 서로 달라도 이를 수용하고 결과 타입으로 VARCHAR를 반환하게 되었다.

DATE, DATETIME, TIMESTAMP 타입에 '0000-00-00' 또는 '0000-00-00 00:00:00' 입력을 지원한다.

보다 자세한 변경 사항은 아래의 [CUBRID 2008 R4.1에서 변경된 사항](#)을 참고한다.

## 지원 플랫폼 및 설치 권장 사양

CUBRID 2008 R4.1 버전이 지원하는 플랫폼과 설치를 위한 하드웨어/소프트웨어 요구 사항은 아래 표와 같다.

지원 플랫폼	메모리 여유 공간	디스크 여유 공간	필요 소프트웨어
--------	-----------	-----------	----------



- Windows 32/64 Bit XP, 2003, Vista, Windows7  - Linux 계열 32/64 Bit Linux kernel 2.4 및 glibc 2.3.4 이상	1G 이상	2GB 이상 <sup>1</sup>	- JRE 또는 JDK 1.6 이상  Java 저장 프로시저를 사용하는 경우 필요
---	-------	---------------------	---

2008 R4.0부터는 CUBRID 패키지 설치 시 CUBRID 매니저 클라이언트가 같이 설치되지 않는다. 따라서 CUBRID 매니저를 사용하려면 이를 추가로 설치해야 한다. CUBRID 설치 패키지는 [다운로드 페이지](#)에서 받을 수 있다.

CUBRID 매니저 및 CUBRID 쿼리 브라우저 설치 패키지와 JDBC, PHP, ODBC, OLE DB 등의 드라이버들은 [cubrid.org의 다운로드 페이지](#)에서 받을 수 있다.

## 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 프로그램의 호환성

- 이전 버전의 JDBC, PHP, CCI API 등을 사용하는 응용 프로그램은 CUBRID 2008 R4.1 데이터베이스에 접근할 수 있다. 다만, JDBC, PHP, CCI 인터페이스에 추가/개선된 기능을 사용하기 위해서는 CUBRID 2008 R4.1 버전의 라이브러리를 링크하거나 드라이버를 사용해야 한다.
- 새로운 예약어 추가 및 일부 질의에 대한 스펙 변경으로 인해 질의 결과가 이전 버전과 다를 수 있으므로 주의한다. 보다 상세한 사항은 [업그레이드 주의 사항](#)을 참고한다.
- GLO 클래스를 이용하여 개발된 응용은 BLOB, CLOB 타입에 맞는 응용 및 스키마로 변환하여 사용해야 한다.

### CUBRID 매니저의 호환성

주의: 2008 R4.0부터는 CUBRID 패키지 설치 시 CUBRID 매니저 클라이언트가 같이 설치되지 않는다. 따라서 CUBRID 매니저를 사용하기 위해서는 이를 추가로 설치해야 한다.

CUBRID 매니저는 CUBRID 2008 R2.1 이상 버전의 서버에 대해서 하위 호환성을 보장하며, 각 서버 버전과 일치하는 CUBRID JDBC 드라이버를 사용한다. 하지만 CUBRID 매니저에서 제공하는 모든 기능을 제대로 사용하기 위해서는 CUBRID 서버 버전보다 높은 버전의 CUBRID 매니저를 사용해야 한다. CUBRID JDBC 드라이버는 CUBRID 설치 시 \$CUBRID/jdbc 디렉터리에 포함되어 있다.<sup>2</sup>

(2008 R2.2 이상 버전의 드라이버는 CUBRID 매니저에 기본으로 내장되어 있으며, [cubrid.org](#)에서 별도로 받을 수도 있다.)

### 데이터베이스 호환성

서버 버전을 하위 버전에서 업그레이드하는 경우, 데이터베이스의 마이그레이션 작업을 수행해야 한다. 보다 상세한 사항은 [데이터베이스 마이그레이션 절차](#)를 참고한다.

<sup>1</sup> 처음 설치 시 약 500MB의 디스크 공간이 필요하며, 하나의 데이터베이스를 기본 옵션으로 생성할 경우 약 1.5GB의 디스크 공간이 필요하다.

<sup>2</sup> \$CUBRID는 CUBRID가 설치된 곳을 지정하는 Linux용 환경변수이며, Windows 환경에서는 %CUBRID%와 같은 형식으로 사용된다.

## 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 동일해야 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 R4.1의 설치 방법

### 제품 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.

Windows용 설치 패키지는 설치 마법사를 이용하여 설치할 수 있다. 설치 방법은 매뉴얼의 'CUBRID 시작> 설치와 실행> Windows에서의 설치와 실행'을 참고한다. 이와 함께 [Windows 환경에서 설치 디렉터리 설정에 관한 주의 사항](#)을 참고한다.

### CUBRID 환경 변수 및 OS 환경 변수 설정

CUBRID를 사용하기 위해서는 CUBRID 환경 변수와 관련 OS 환경 변수를 설정해야 한다. (매뉴얼의 'CUBRID 시작 > 환경 변수 설정 및 CUBRID 서비스 시작 > 환경 변수 설정' 참고) 특히, CUBRID 매니저와 Java 저장 프로시저를 사용하기 위해서는 Java 환경 변수를 설정해야 한다. (매뉴얼의 'CUBRID SQL 설명서 > Java 저장 함수/프로시저 > Java 저장 함수/프로시저 사용을 위한 환경 설정' 참고)

## CUBRID 2008 R4.1로 업그레이드하는 방법

### 2008 R4.0에서 2008 R4.1로 업그레이드하기

CUBRID 2008 R4.0, 2008 R4.0 Patch X 버전 사용자는 2008 R4.1을 설치한 후 기존의 환경 설정 파일에서 파라미터들의 값을 다음과 같이 변경해야 한다.

- cubrid.conf
  - thread\_stacksize의 기본값이 100K에서 1M으로 변경되었으므로 이 값을 설정하지 않은 사용자는 CUBRID 관련 프로세스들의 메모리 사용량을 살펴볼 것을 권장한다.
  - data\_buffer\_size의 최소값이 64K에서 16M으로 변경되었으므로 이 값을 16M 미만으로 설정한 사용자는 16M 이상으로 설정해야 한다.
- cubrid\_broker.conf
  - CCI\_DEFAULT\_AUTOCOMMIT의 기본값이 ON으로 변경되었으므로 이를 설정하지 않은 응용 프로그램 사용자가 기존과 같은 자동 커밋 모드를 유지하고 싶다면 OFF로 설정해야 한다.
  - APPL\_SERVER\_MAX\_SIZE\_HARD\_LIMIT의 최소값이 1024M이므로 APPL\_SERVER\_MAX\_SIZE의 값을 설정한 사용자는 APPL\_SERVER\_MAX\_SIZE\_HARD\_LIMIT의 값을 이보다 크게 설정할 것을 권장한다(이 문서의 [CUBRIDSUS-5956](#) 참고).

### 2008 R4.0 베타 이전 버전에서 2008 R4.1로 업그레이드하기

CUBRID 2008 R4.0 Beta 버전 및 그 이전 버전에서 업그레이드하는 사용자는 아래의 CUBRID 2008 R4.1로 업그레이드하는 방법을 참고하도록 한다.

### 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.<sup>3</sup>

#### ● 새로 추가된 예약어 검사

CUBRID 설치 패키지에 포함 또는 [다운로드 페이지](#)에서 배포되는 CUBRID 2008 R4.1용 예약어 검출 스크립트인 **check\_reserved.sql**을 이용하여 예약어 사용 여부를 검사할 수 있으며, 예약어로 지정된 식별자를 사용하고 있을 경우 식별자를 수정해야 한다. (매뉴얼의 'CUBRID SQL 설명서 > 식별자' 참고)

#### ● 데이터베이스 마이그레이션

CUBRID 2008 R3.1부터 GLO를 지원하지 않으며 LOB 타입이 GLO 기능을 대체하게 되었으므로, GLO를 이용한 응용 및 스키마는 LOB 타입에 맞게 수정해야 한다. (아래의 [GLO 클래스 사용자의 마이그레이션 참고](#))

CUBRID 2008 R4.1은 2008 R4.0 Beta 및 그 이전 버전과 데이터베이스 볼륨이 호환되지 않으므로, 이전 데이터베이스를 CUBRID 2008 R4.1로 마이그레이션해야 한다. (아래의 [데이터베이스 마이그레이션 절차](#) 참고)

2008 R3.1 및 그 이전 버전에서 마이그레이션하기 위해서는 **cubrid unloaddb/loaddb** 유틸리티를 사용한다.

2008 R4.0 Beta에서 마이그레이션하기 위해서는 다운로드 페이지에서 제공하는 **migrate\_r40beta2ga** 유틸리티를 사용할 수 있다.

#### ● 복제 또는 HA 환경 재구성

2008 R4.0부터는 복제 기능을 더 이상 지원하지 않으므로, 이전의 복제 기능을 사용하는 시스템에서는 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 재구성해야 한다. (아래의 [HA 환경에서 데이터베이스 마이그레이션](#) 참고)

HA 환경 구성은 매뉴얼의 '관리자 안내서 > CUBRID HA'를 참고하여 재설정해야 한다.

### 데이터베이스 마이그레이션 절차

[2008 R3.x 및 그 이전 버전에서 업그레이드하는 사용자를 위한 절차 바로 가기](#)

[2008 R4.0 Beta 버전에서 업그레이드하는 사용자를 위한 절차 바로 가기](#)

[GLO 클래스 사용자를 위한 절차 바로 가기](#)

#### ● 2008 R3.x 및 그 이전 버전에서 2008 R4.1로 마이그레이션

2008 R3.x 및 그 이전 버전의 사용자는 아래의 표와 같이 2008 R4.1로 데이터베이스 마이그레이션을 해야 한다.

또한, 기존의 GLO 클래스를 사용하고 있는 경우에는 추가 작업이 필요하다. (아래의 [GLO 클래스 사용자의 마이그레이션](#) 참고)

아래는 **cubrid unloaddb/loaddb** 유틸리티와 [다운로드 페이지](#)에서 별도 배포되는 **check\_reserved.sql** 예약어 검출 스크립트를 이용하여 마이그레이션을 수행하는 방법이다. (매뉴얼의 '관리자 안내서 > 데이터베이스 마이그레이션'과 본 문서의 [주의 사항> GLO 클래스 지원 중단에 따른 주의 사항](#) 참고)

단계	Linux 환경	Windows 환경
C1 단계: CUBRID Service 종료	% cubrid service stop	CUBRID Service Tray를 종료한다.
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 <a href="#">관련 매뉴얼</a> 참고) % csq1 -S -u dba -i check_reserved.sql testdb	
C3 단계:	이전 버전의 databases.txt 및 conf 디렉터리 내 설정 파일을 별도 디렉터리에 보관한다. (C3a)	

<sup>3</sup> \$CUBRID 또는 \$CUBRID\_DATABASES 형식의 환경변수는 Linux 환경에서 사용하는 형식이며 Windows 환경에서는 %CUBRID% 또는 %CUBRID\_DATABASES%와 같은 형식으로 사용해야 한다.

이전 버전 DB 언로드	cubrid unloaddb 유틸리티를 실행하고 이때 생성된 파일을 별도 디렉터리에 보관한다. (C3b) % cubrid unloaddb -S testdb 기존 데이터베이스를 삭제한다. (C3c) % cubrid deletedb testdb	
		이전 버전의 CUBRID를 제거한다.
C4 단계: 2008 R4.1 설치	본 문서에서 <a href="#">CUBRID 2008 R4.1의 설치 방법</a> 을 참고한다.	
C5 단계: 데이터베이스 생성 및 데이터 로딩	데이터베이스를 생성할 디렉터리로 이동한 후, 데이터베이스를 생성한다. (C5a) % cd \$CUBRID/databases/testdb % cubrid createdb testdb (C3b)에서 보관한 파일을 가지고 cubrid loaddb 유틸리티를 실행한다. (C5b) % cubrid loaddb -s testdb_schema -d testdb_objects -i testdb_indexes testdb	
C6 단계: 새 버전 DB 백업	% cubrid backupdb -S testdb	
C7 단계: CUBRID 환경 설정 및 CUBRID Service 구동	환경 설정 파일을 수정한다. 이때, (C3a)에서 보관한 이전 버전의 환경 설정 파일을 새 버전 에 맞게 수정한다. (시스템 파라미터 설정은 주의 사항 및 관련 매뉴얼 참고) % cubrid service start % cubrid server start testdb	CUBRID Service Tray> [Service Start]를 선택하여 서비스를 시작한다. 명령 프롬프트 창에서 데이터베이스 서버를 구동한다. % cubrid server start testdb

### ● 2008 R4.0 Beta 버전에서 마이그레이션

2008 R4.0 Beta 버전 사용자는 아래의 표와 같이 데이터베이스 마이그레이션을 해야 한다.

다운로드 페이지에서 제공되는 **migrate\_r40beta2ga** 유틸리티를 이용하여 마이그레이션을 수행할 수 있다. 그러나 데이터베이스의 페이지 크기가 4K 미만인 볼륨은 **cubrid unloaddb/loaddb**를 이용하여 마이그레이션을 수행해야 한다. ([2008 R3.x 및 그 이전 버전에서 2008 R4.1로 마이그레이션](#) 참고)

마이그레이션 수행 중 실패하는 경우 백업해 놓았던 이전 버전의 DB를 복구(**cubrid restoredb**)하여 재시도한다.

단계	Linux 환경	Windows 환경
D1 단계: CUBRID Service 종료	% cubrid service stop	CUBRID Service Tray를 종료한다.
D2 단계: 이전 버전 DB 백업	이전 버전으로 복구하는 상황에 대비하기 위해 백업을 수행하고, 백업 파일을 별도 디렉터리(backup)에 보관한다. (D2a) % mkdir backup % cubrid backupdb -S -D backup testdb 이전 버전의 databases.txt 및 conf 디렉터리 내 설정 파일을 별도 디렉터리에 보관한다. (D2b)	
		이전 버전의 CUBRID를 제거한다.
D3 단계: 2008 R4.1 설치	본 문서의 <a href="#">CUBRID 2008 R4.1의 설치 방법</a> 을 참고한다.	
D4 단계: 마이그레이션 도구 실행	(D2b)에서 보관한 databases.txt를 2008 R4.1의 설치 디렉터리에 복사한다. (D4a) 아래와 같이 migrate_r40beta2ga 유틸리티를 실행한다. (D4b) % migrate_r40beta2ga testdb	
D5 단계: 새 버전 DB 백업	% cubrid backupdb -S testdb	
D6 단계: CUBRID 환경 설정 및 CUBRID Service 구동	보관한 이전 버전의 환경 설정 파일을 새 버전 에 맞게 수정한다. (시스템 파라미터 설정은 주의 사항 및 <a href="#">관련 매뉴얼</a> 참고) % cubrid service start % cubrid server start testdb	CUBRID Service Tray> [Service Start]를 선택하여 서비스를 시작한다. 명령 프롬프트 창에서 데이터베이스 서버를 구동한다. % cubrid server start testdb

### ● GLO 클래스 사용자의 마이그레이션

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.

## 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 R4.1로 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: 버전에 따라 마스터 서버에서 C1~C6 단계 또는 D1~D5 단계를 수행	마스터 서버에서 CUBRID 업그레이드 및 데이터베이스 마이그레이션을 수행하고, 2008 R4.1 데이터베이스를 백업한다.
H7 단계: 슬레이브 서버에 CUBRID 2008 R4.1 설치	슬레이브 서버에서 이전 버전의 데이터베이스는 삭제하고, 새 버전을 설치한다. 설치 방법은 본 문서의 <a href="#">CUBRID 2008 R4.1의 설치 방법</a> 을 참고한다.
H8 단계: 마스터 서버 백업본을 슬레이브 서버에서 복구	H6 단계에서 생성된 마스터 서버의 2008 R4.1 데이터베이스 백업본(예: testdb_bk*)을 슬레이브 서버에서 복구한다. <pre>% scp user1@master:\$CUBRID/databases/databases.txt \$CUBRID/databases/. % cd ~/DB/testdb % scp user1@master:~/DB/testdb/testdb_bk0v000 . % scp user1@master:~/DB/testdb/testdb_bkvinf . % cubrid restoredb testdb</pre>
H9 단계: HA 환경 재구성 후 HA모드 구동	마스터 및 슬레이브 서버에서 CUBRID 환경 설정 파일(cubrid.conf) 및 HA 환경 설정 파일(cubrid_ha.conf)을 설정한다. ( <a href="#">관련 매뉴얼</a> 참고) <pre>% vi \$CUBRID/conf/cubrid.conf ha_mode=on</pre> <pre>% vi \$CUBRID/conf/cubrid_ha.conf [common] ha_port_id=59901 ha_node_list=cubrid-ha@master:slave ha_db_list=testdb</pre> 마스터 및 슬레이브 서버에서 HA모드로 DB를 구동한다. ( <a href="#">관련 매뉴얼</a> 참고) <pre>% cubrid heartbeat start</pre>
H10 단계: 브로커 서버에 CUBRID 2008 R4.1 설치 및 브로커 구동	설치 방법은 본 문서의 <a href="#">CUBRID 2008 R4.1의 설치 방법</a> 을 참고한다. 브로커 서버에 있는 브로커를 시작한다. ( <a href="#">관련 매뉴얼</a> 참고) <pre>% cubrid broker start</pre>

### ● 2008 R2.0 또는 2008 R2.1에서 2008 R4.1로 HA 마이그레이션

CUBRID 2008 R2.0 또는 2008 R2.1의 HA 기능을 사용하는 경우, 서버 버전 업그레이드, 데이터베이스 마이그레이션을 수행하고 HA 환경을 새롭게 구축한 후 해당 버전에서 사용되었던 Linux Heartbeat 자동 시작 설정을 변경해야 한다. (Linux Heartbeat 패키지가 불필요한 경우 삭제한다.)

위의 H1~H10 단계를 수행한 후, 아래의 H11 단계를 수행한다.

단계	설명
H11 단계: 기존 Linux heartbeat 자동 시작 설정 변경	이하의 작업은 마스터 및 슬레이브 서버에서 root 계정으로 수행한다. <pre>[root@master ~]# chkconfig --del heartbeat // 슬레이브 서버에서 동일 작업 수행</pre>

## 복제 기능 사용 시 주의 사항

CUBRID 2008 R4.1 버전부터 복제 기능이 제거되었으므로, 기존의 복제 기능을 사용 중인 환경에서 이중화 환경을 구축하려면 HA 기능을 사용해야 한다. 서버 버전 업그레이드 및 데이터베이스 마이그레이션을 수행한 후, HA 환경을 새롭게 구축할 수 있다. HA 환경 구축과 관련하여, 매뉴얼의 '[관리자 안내서 > CUBRID HA](#)'를 참고한다.

### 3. CUBRID 2008 R4.1 Patch 1에서 변경된 사항

#### 새로 추가된 기능

##### CUBRIDSUS-6070 JDBC 연결 URL에 디버깅용 로그를 기록하도록 설정하는 기능 추가

JDBC 연결 URL에 logOnException 프로퍼티로 예외 처리를, logSlowQueries 프로퍼티와 slowQueryThresholdMillis 프로퍼티로 슬로우 쿼리를 로그 파일에 기록하도록 설정하는 기능을 추가하였다.

```
url =
"jdbc:cubrid:localhost:33000:demodb:::logOnException=true&logSlowQueries=true&slowQueryThresholdMillis=1000"
```

#### 개선된 성능

##### CUBRIDSUS-6648 CCI, PHP 인터페이스에서 한번의 호출로 PREPARE와 EXECUTE를 수행할 수 있게 개선하고, 요청 핸들 종료 함수 호출 시 메시지 전송횟수를 줄이도록 수정

CCI와 PHP에서 한 번의 호출로 PREPARE와 EXECUTE를 수행하도록 cci\_prepare\_and\_execute()를 추가하고, PHP 인터페이스의 cubrid\_query(), cubrid\_execute()가 이를 사용하게 하였다. 단, cubrid\_execute()는 입력인자로 연결 객체와 질의문을 전달하는 경우에만 cci\_prepare\_and\_execute()를 사용한다.

이와 함께, cci\_close\_req\_handle()을 호출할 경우 종료 요청을 브로커 응용서버(CAS)에게 보내지 않고 다음 번 PREPARE시에 종료할 요청 핸들(request handle) 번호를 같이 보내줌으로써 메시지 전송 횟수를 줄이도록 수정하였다. 또한, 이 수정으로 인해 자동 커밋 모드가 ON일 때 CAS가 재시작된 후 cci\_close\_req\_handle()를 호출하면 -4 (COMMUNICATION ERROR) 오류를 반환하는 현상이 해결되었다.

SELECT 위주의 워크로드에 대하여 이전 버전 대비 약 45%의 성능 향상을 보였다.

##### CUBRIDSUS-6906 JDBC 드라이버의 성능 개선

JDBC 드라이버에서 헤더 정보를 버퍼에 모아서 한번에 전송하도록 수정하여 2008 R4.1 버전 JDBC 드라이버의 성능 저하를 개선하였다.

#### 수정된 사항

##### CUBRIDSUS-6678 데이터베이스 서버 프로세스 재시작 시 데이터베이스 볼륨이 정상적으로 복구되지 못하는 오류 수정

서버 프로세스의 비정상 종료를 포함하여 종료 후 재시작 시에 데이터베이스 복구(recovery)가 비정상적으로 수행되어 데이터베이스 볼륨이 손상될 수 있는 오류를 수정하였다.

##### CUBRIDSUS-6881 IFNULL/NVL/NVL2/COALESCE 함수에 호스트 변수를 사용하여 INSERT 수행 시 발생하는 오류 수정

IFNULL/NVL/NVL2/COALESCE 함수에 호스트 변수를 사용하여 INSERT 문을 수행하는 경우 "ERROR: Execute: Cannot evaluate ' ifnull( ? :0 , 1)'" 오류가 발생하였으나 이를 수정하였다.

## CUBRIDSUS-6689 NOT 뒤에 단항 연산자 사용 시 'unknown opcode' 오류 수정

NOT 뒤에 단항(unary) 연산자를 사용하는 경우 "ERROR: 'unknown opcode' operator is not defined on types integer and integer." 오류가 발생하였으나 이를 수정하였다.

```
SELECT * FROM tab WHERE NOT - col0 = - col0;
```

## CUBRIDSUS-6778 인덱스 스캔 도중 키 잠금을 획득/해제하는 과정에서 라이브락 현상이 발생할 수 있는 문제 수정

인덱스 스캔 도중 잠금(lock)을 해제했다가 획득하려는 시도를 무한 반복하는 라이브락(livelock) 현상이 발생할 수 있는 문제를 수정하였다.

## CUBRIDSUS-6914 CCI 응용이 DATASOURCE를 이용하여 데이터베이스에 접속한 상태에서 연결 종료 함수를 호출하는 경우 응용 프로그램이 비정상 종료될 수 있는 현상 수정

cci\_datasource\_create() 함수로 DATASOURCE를 생성하여 데이터베이스에 접속한 상태에서 cci\_disconnect()를 호출하는 경우 응용 프로그램이 비정상 종료될 수 있는 현상을 수정하였다.

## CUBRIDSUS-6925 cci\_set\_query\_timeout() 함수로 설정한 질의 타임아웃 동작 오류 수정

cci\_set\_query\_timeout() 함수로 설정한 질의 타임아웃이 정상적으로 동작하지 않는 문제를 수정하였다.

## CUBRIDSUS-6958 CCI, PHP 응용 프로그램에서 접속 실패 시 fallback이 되어도 첫 번째로 설정한 호스트에는 접속을 시도하지 않는 문제 수정

CCI, PHP 응용 프로그램에서 접속에 실패하면 연결 URL의 althost 앞에 설정한 첫 번째 호스트에는 접속을 시도하지 않는 문제로 인해, fallback이 되어도 마스터 서버로 재접속하지 않는 현상이 존재하였으나 이 문제를 수정하였다.

```
cci:cubrid:10.0.0.1:30000:demodb::?althosts=10.0.0.2:30000
```

## CUBRIDSUS-6753 인덱스에서 B+트리 노드의 분할/병합과 INSERT/DELETE/UPDATE가 동시에 일어날 때 "Unknown key" 오류 수정

인덱스에서 B+트리 노드의 분할/병합과 INSERT/DELETE/UPDATE가 동시에 일어날 때 "Unknown key '???' referenced in B+tree index {vfid: (40, 3), rt\_pgid: 2520, key\_type: character varying}" 오류가 발생하였으나 이를 수정하였다.

## CUBRIDSUS-6897 CCI 기반 응용이 두 개 이상의 브로커에 접속하는 환경에서, 어떤 질의가 5초 이상 수행되면 접속 오류가 발생할 수 있는 문제 수정

CCI 기반(CCI, PHP, ODBC, OLE DB, Perl, Ruby, Python) 응용 프로그램에서 여러 개의 쓰레드들이 두 개 이상의 브로커에 접속 할 때 적어도 하나의 쓰레드에서 질의가 5초 이상 수행되면 -4(COMMUNICATION ERROR) 오류가 발생할 수 있는 문제를 수정하였다. 해당 오류 발생 시 응용 프로그램이 비정상 종료되거나 응용 프로그램에 의해 1G 이상 메모리가 불필요하게 할당되는 등의 이상 현상이 존재하였다.



---

**CUBRIDSUS-6779 응용 프로그램과 브로커 응용 서버(CAS) 연결 직후  
둘 사이의 상태 정보가 일치하지 않는 현상 수정**

응용 프로그램이 CAS에 연결한 직후 `cubrid broker status` 명령으로 상태 정보를 확인하는 경우 트랜잭션 상태 정보(STATUS)가 CLOSE WAIT로 바뀌지 않고 CLIENT\_WAIT인 현상을 수정하였다.

---

**CUBRIDSUS-6754 브로커 응용 서버(CAS)의 개수를 100으로 설정하는  
경우 브로커 구동에 실패하는 문제 수정**

CAS의 개수를 설정하는 브로커 파라미터인 MIN\_NUM\_APPL\_SERVER와 MAX\_NUM\_APPL\_SERVER의 값을 100으로 설정하는 경우 브로커 구동에 실패하는 문제를 수정하였다.

---

**CUBRIDSUS-6364 브로커 응용 서버(CAS)가 SQL 로그를 쓰는 도중  
멈출(hang) 수 있는 현상 수정**

CAS가 SQL 로그를 쓰는 도중 인터럽트 발생했을 때 멈출(hang) 수 있는 현상을 수정하였다.

---

**CUBRIDSUS-6889 HA 복제 반영 프로세스가 복제를 진행하지 못할 수  
있는 문제 수정**

HA 복제 반영 프로세스인 `applylogdb`에서 내부 오류로 인해 복제 반영을 정상적으로 진행하지 못하는 오류를 수정하였다.

---

**CUBRIDSUS-6905 JDBC에서 집합형 데이터 타입을 사용하면 브로커  
응용 서버(CAS)와 JDBC 응용 프로그램이 멈추는(hang) 현상 수정**

JDBC에서 SET, MULTISSET, LIST 등 집합형 데이터 타입을 사용하면 브로커 응용 서버(CAS)와 JDBC 응용 프로그램이 멈추는(hang) 현상을 수정하였다.



## 4. CUBRID 2008 R4.1에서 변경된 사항

### 새로 추가된 기능

#### CUBRIDSUS-5860 ADDTIME 함수 추가

날짜/시간 타입의 값에 시간 값을 더하는 ADDTIME 함수를 추가하였다.

```
SELECT ADDTIME('09:00:13 am', '9:33:17')
'06:33:30 PM'
```

#### CUBRIDSUS-6265 ASCII 함수 추가

입력 문자를 ASCII 코드 값으로 반환하는 ASCII 함수를 추가하였다.

```
SELECT ASCII('a');
97
```

#### CUBRIDSUS-5860 BIN 함수 추가

숫자를 이진 문자열로 표현하는 BIN 함수를 추가하였다.

```
SELECT BIN(12);
'1100'
```

#### CUBRIDSUS-6233 CONV 함수 추가

숫자를 나타내는 입력 문자열을 진수가 다른 숫자로 변환하는 CONV 함수를 추가하였다.

```
// 다음은 'a'인 16 진수를 2 진수로 바꾼다.
SELECT CONV('a',16,2);
'1010'
```

#### CUBRIDSUS-5860 FIND\_IN\_SET 함수 추가

침표(,)를 구분자로 하는 문자 집합에서 지정한 문자의 개수를 반환하는 FIND\_IN\_SET 함수를 추가하였다.

```
SELECT FIND_IN_SET('1', '1,4,1,8');
2
```

#### CUBRIDSUS-6233 HEX 함수 추가

입력 인자가 문자열인 경우 각 문자에 해당하는 아스키 코드를 2자리 16진수 문자열로 반환하며, 입력인자가 숫자인 경우 10진수를 16진수 문자열로 변환하는 HEX 함수를 추가하였다.

```
SELECT HEX('abc'), HEX(255);
'616263' 'FF'
```

#### CUBRIDSUS-6190 현재의 시리얼 값과 하나 이상의 시리얼 값을 반환하는 함수들 추가

현재의 시리얼 값을 반환하는 SERIAL\_CURRENT\_VALUE(serial\_name)와, 하나 이상의 시리얼 값을 반환하는 SERIAL\_NEXT\_VALUE(serial\_name, number) 를 추가하였다.

SERIAL\_CURRENT\_VALUE(serial\_name)는 serial\_name.current\_value와 동일하게 동작한다.

#### CUBRIDSUS-5903 정규 표현식에 대한 연산자 추가

정규 표현식(regular expression)에 대한 연산자인 REGEXP를 추가하였다. REGEXP에서 사용하는 패턴은 대소문자를 구분하지 않으며, 대소문자를 구분하려면 REGEXP BINARY를 사용한다.

```
SELECT name FROM athlete where name REGEXP '^[a-d]';
```

```
name
=====
'Dziouba Irina'
'Dzieciol Iwona'
'Crosta Daniele'
'Bukovec Brigita'
'Abdullayev Namik'
```

### CUBRIDSUS-5616 질의 실행 시간을 제한하는 브로커 파라미터 추가

브로커로 접속하는 응용 프로그램의 질의 실행 시간을 브로커 파라미터로 제한하기 위해 질의 타임아웃 값을 설정하는 MAX\_QUERY\_TIMEOUT 파라미터를 추가하였다. broker\_changer 유틸리티로 동적 변경이 가능하며, 값의 범위는 0~86400초(1일)이다.

### CUBRIDSUS-6197 JDBC 연결 URL에 질의 타임아웃 시간을 설정하는

#### 기능 추가

JDBC 연결 URL에 queryTimeout 프로퍼티를 사용하여 질의 수행에 대한 타임아웃 시간을 초 단위로 설정하는 기능이 추가되었다(기본값: 0, 무제한). 이 값은 DriverManager.setQueryTimeout() 메소드에 의해 변경될 수 있다.

```
url = "jdbc:cubrid:localhost:33000:demodb:::?rctime=600&queryTimeout=1"
```

### CUBRIDSUS-6198 JDBC 연결 URL에 데이터베이스 연결 타임아웃

#### 시간을 설정하는 기능 추가

JDBC 연결 URL에 connectTimeout 프로퍼티를 사용하여 데이터베이스 연결에 대한 타임아웃 시간을 초 단위로 설정하는 기능이 추가되었다(기본값: 0). DriverManager.setLoginTimeout() 메소드로도 이 값을 설정할 수 있으나, 연결 URL에 설정하면 해당 메소드로 설정하는 값은 무시된다.

```
url = "jdbc:cubrid:localhost:33000:demodb:::~?rctime=600&connectTimeout=5"
```

### CUBRIDSUS-5388 브로커 응용 서버의 상태를 출력할 때 트랜잭션 시작

#### 시간을 추가

cubrid broker status -f 명령으로 브로커 응용 서버(CAS)의 상태를 출력할 때 트랜잭션 시작 시간을 추가하였다.

### CUBRIDSUS-6199 브로커 응용 서버의 상태를 출력할 때 응용

#### 프로그램이 CAS에 접속한 회수, CAS의 재구동 회수를 추가

cubrid broker status -f 명령으로 브로커 응용 서버(CAS)의 상태를 출력할 때 응용 프로그램이 CAS에 접속한 회수, CAS의 재구동 회수를 추가하였다. 이러한 CAS의 재구동 회수가 많은 경우, CAS의 메모리 사용량의 최대 크기를 지정하는 브로커 파라미터인 APPL\_SERVER\_MAX\_SIZE의 설정 값을 현재보다 크게 변경하는 것을 고려할 수 있다.

### CUBRIDSUS-6128 에러 심각성 수준이 NOTIFICATION일 때

#### 교착상태가 발생하면 서버 에러 로그 파일에 잠금 관련 정보를 기록하는 기능

#### 추가

에러 심각성 수준을 설정하는 시스템 파라미터인 error\_log\_level의 값을 NOTIFICATION으로 설정했을 때 교착 상태가 발생하면 서버 에러 로그 파일에 잠금 관련 정보를 기록하는 기능을 추가하였다.

다음의 에러 로그 파일 정보에서 (1)은 교착상태를 유발한 테이블 이름을, (2)는 인덱스 이름을 나타낸다.

```
demodb_20111102_1811.err
```

```
...
OID = -532| 520| 1
(1) Object type: Index key of class ( 0| 417| 7) = tbl.
BTID = 0| 123| 530
(2) Index Name : i_tbl_col1
Total mode of holders = NS_LOCK, Total mode of waiters = NULL_LOCK.
```

```
Num holders= 1, Num blocked-holders= 0, Num waiters= 0
LOCK HOLDERS:
Tran_index = 2, Granted_mode = NS_LOCK, Count = 1
...
```

## 개선된 성능

### CUBRIDSUS-5300 트랜잭션 로그의 동시 처리 성능 개선

여러 응용 프로그램들에 의해 동시에 트랜잭션 로그 버퍼를 사용하는 작업들이 수행되어야 하는 경우, 이들이 순서대로 진행되도록 잠금을 유지하는 구간(critical section) 내의 작업을 최소화하여, 대량의 트랜잭션 부하 테스트에서 특히 INSERT와 UPDATE에 대해 2008 R4.0 Patch 2 버전 대비 70% 의 성능이 향상되었다.

### CUBRIDSUS-5687 메모리 버퍼의 더티 페이지를 디스크에 쓰는 방식을

#### 수정하여 쓰기 성능 개선

메모리 버퍼의 더티 페이지를 디스크에 무작위로 쓰는 방식에서 일정 분량의 페이지를 모아 정렬하여 쓰도록 하여, 쓰기 트랜잭션의 성능이 개선되었다.

### CUBRIDSUS-5297 현재의 날짜/시간 정보를 구하는 함수의 실행 성능

#### 개선

SYSDATE, SYSTIME, SYSDATETIME, SYSTIMESTAMP 등 현재의 날짜/시간 정보를 구하는 함수 실행 시 이전 버전에서는 매번 서버에 값을 요청했으나 최근 1분 이내 요청된 값으로부터 현재 값을 계산하도록 하여 이들 함수의 실행 성능이 개선되었다.

### CUBRIDSUS-6570 ORDER BY 컬럼에 대해 BETWEEN ... AND

#### 조건인 경우에도 다중 키 범위 조건의 정렬 수행 최적화가 가능하도록 개선

ORDER BY 절의 컬럼에 대해 동등(=) 조건인 경우에만 중간 값의 정렬을 진행하면서 결과를 수집하는 다중 범위 최적화(multi-range optimization)를 수행할 수 있었으나, 해당 컬럼에 대해 BETWEEN ... AND ... 조건인 경우에도 수행할 수 있도록 개선하였다.

```
CREATE INDEX i_tbl_a_b_c ON tbl(a, b, c);
SELECT a, b, c
FROM tbl
WHERE a='1' AND b IN (1,2,3) AND
c BETWEEN 5 AND 10
USING INDEX i_tbl_a_b_c(+)
ORDER BY c LIMIT 10;
```

### CUBRIDSUS-5295 질의 실행 계획 캐시에 INSERT 질의가 포함되도록

#### 개선

이전 버전에서는 SELECT, UPDATE, DELETE 질의만이 질의 실행 계획 캐시에 저장되었으나, INSERT 질의도 포함되도록 개선하였다.

### CUBRIDSUS-5850 시스템의 부하가 매우 큰 경우 트랜잭션 처리 응답

#### 시간의 최저 값과 최고 값의 편차를 줄이도록 개선

시스템의 부하가 매우 큰 경우에도 일정한 성능을 유지하게 하기 위해 트랜잭션 처리 응답 시간의 최저 값과 최고 값의 편차를 줄이도록 개선하였다.

### CUBRIDSUS-5824 HA 환경에서 복제 지연 시 성능 개선

HA 환경에서 복제 지연 발생 시에 저장 로그(archive log) 볼륨 파일에 접근할 때 존재하던 오버헤드를 제거하여 성능을 개선하였다.

## 수정된 사항

### CUBRIDSUS-5882 날짜/시간 타입에 대해 '0000-00-00' 또는 '0000-00-00 00:00:00' 형식의 입력 지원

이전 버전에서는 DATETIME, TIMESTAMP 타입에서만 '0000-00-00 00:00:00' 형식의 입력을 지원하였으나 DATE 타입도 '0000-00-00' 형식의 입력을 지원하도록 수정하였다. 그리고 이전 버전에서는 DATETIME, TIMESTAMP 타입에서 '0000-00-00 00:00:00'을 입력하면 최소값으로 변환되어 저장되었으나 '0000-00-00 00:00:00'이 저장되도록 수정하였다.

이로 인해 TIMESTAMP 타입의 최소값이 '1970-01-01 00:00:00' (GMT)에서 '1970-01-01 00:00:01' (GMT)로 변경되었다.

보다 자세한 사항은 매뉴얼의 'CUBRID SQL 설명서 > 데이터 타입 > 날짜/시간 데이터 타입'을 참고한다.

### CUBRIDSUS-5296 시리얼(SERIAL)을 포함하는 질의문의 질의 계획을 질의 실행 계획 캐시에 저장할 수 있도록 수정

시리얼을 포함하는 질의문의 질의 계획을 질의 실행 계획 캐시에 저장할 수 있도록 수정하였다.

### CUBRIDSUS-6138 ABS 등의 수치 연산 함수들에서 입력 인자가 호스트 변수이고 문자열을 바인딩하는 경우 질의 결과가 잘못되는 오류 수정

ABS, FLOOR, CEIL 등 수치 연산 함수들에서 입력 인자가 호스트 변수이고 문자열을 바인딩하는 경우 질의 결과가 잘못되는 오류를 수정하였다.

```
SELECT a FROM tbl WHERE a < ABS( ? );
setString (1, "1.5");
```

### CUBRIDSUS-5988 TO\_CHAR 함수의 첫번째 인자가 호스트 변수이고 DATE 타입의 값이 바인딩되면 발생하는 오류 수정

TO\_CHAR 함수의 첫번째 인자가 호스트 변수이고 DATE 타입의 값이 바인딩되면 "Semantic: Cannot coerce host var to type double" 오류가 발생하였으나 이를 수정하였다.

```
SELECT * FROM tbl WHERE TO_CHAR(col,'yyyymmdd') = TO_CHAR(?,'yyyymmdd')
```

### CUBRIDSUS-6280 STR\_TO\_DATE 함수에서 '%r' 포맷 지정 시 PM 시각 문자열이 타입 변환이 되지 않거나 '12:XX:XX AM' 문자열이 '12:XX:XX PM'으로 잘못 변환되는 오류 수정

STR\_TO\_DATE 함수에서 '시:분:초 PM' 형태의 문자열에 '%r' 포맷을 지정하는 경우 타입 변환이 되지 않는 오류와, 오전 12시인 '12:XX:XX AM' 형태의 문자열에 '%r' 포맷을 지정하는 경우 오후 12시인 '12:XX:XX PM'으로 잘못 변환되는 오류를 수정하였다.

```
SELECT STR_TO_DATE('11:41:10 PM', '%r');
SELECT STR_TO_DATE('12:41:10 AM', '%r');
```

### CUBRIDSUS-6023 조건절에 AND .. OR 조건이 포함된 경우 OR의 왼쪽 항이 AND로 연결된 조건이면 질의 결과가 잘못될 수 있는 오류 수정

조건절에 AND .. OR 조건이 포함된 경우 OR의 왼쪽 항이 AND로 연결된 조건이고 이 조건이 항상 거짓이면 OR 조건 전체의 결과도 거짓으로 판단하여 질의 결과가 잘못될 수 있는 오류를 수정하였다.

```
SELECT * FROM tbl WHERE col0 BETWEEN 81 AND 23 OR col0 = 92;
```

## CUBRIDSUS-6231 NULL 입력 여부에 따라 결과 값을 정하는 SQL

### 함수들에서 입력 타입에 따른 결과 타입의 변환 규칙 변경

IFNULL, NVL, NVL2, COALESCE 와 같이 NULL 입력 여부에 따라 결과 값을 정하는 SQL 함수들에서 입력 타입에 따른 결과 타입의 변환 규칙을 변경하였다. 입력 인자들 사이에서 우선순위가 가장 큰 타입이 결과 타입이 되며, 타입이 호환되지 않는 경우 VARCHAR 타입이 결과 타입이 된다.

변환 가능한 타입과 우선순위는 아래와 같다.

- CHAR < VARCHAR
- NCHAR < NCHAR VARYING
- BIT < VARBIT
- SHORT < INTEGER < BIGINT < NUMERIC < FLOAT < DOUBLE
- DATE < TIMESTAMP < DATETIME

예를 들어, 아래와 같이 COALESCE 함수에서 입력 인자의 타입인 INTEGER, DATE, TIME 사이에 서로 타입 변환이 될 수 없는 경우 이전 버전에서는 "ERROR: 'coalesce ' operator is not defined on types date and time." 오류가 발생했으나, 2008 R4.1 버전부터는 VARCHAR 타입의 결과 값을 반환한다.

```
CREATE TABLE tbl(a INTEGER, b DATE, c TIME);
INSERT INTO tbl(a) values(1);
INSERT INTO tbl(b) values(SYSDATE);
INSERT INTO tbl(c) values(SYSTIME);
SELECT COALESCE (a, b, c) FROM tbl;
```

## CUBRIDSUS-6033 커버링 인덱스로 WHERE 절에 IN 조건식이 있는

### 질의 수행 도중 발생하는 메모리 누수 현상 수정

커버링 인덱스(covering index)로 WHERE 절에 IN 조건식이 있는 질의 수행 도중 발생하는 메모리 누수 문제를 수정하였다. 커버링 인덱스에 대한 자세한 내용은 매뉴얼의 'CUBRID SQL 설명서 > 질의 최적화 > 인덱스 활용 > 커버링 인덱스'를 참조한다.

## CUBRIDSUS-6345 커버링 인덱스가 사용되며 연관 부질의가 존재하는

### 질의의 경우 잘못된 결과가 출력될 수 있는 오류 수정

질의 결과 집합에 대한 모든 데이터를 포함하는 커버링 인덱스(covering index)가 사용되며 연관 부질의(correlated subquery)가 존재하는 질의의 경우 잘못된 결과가 출력될 수 있는 오류를 수정하였다. 이 오류는 2008 R4.0 버전 ~ 2008 R4.0 Patch 2 버전에서만 발생하였다.

```
CREATE TABLE foo (i int, j int, k int);
CREATE INDEX ON foo (i, j);
SELECT count(*) FROM foo x, foo y WHERE x.i = y.i
AND EXIST(SELECT y.k FROM foo z WHERE z.i = x.i+1 and z.j = y.j+1)
```

## CUBRIDSUS-6649 조인 질의문에 LIMIT 조건이 주어지고 인덱스를

### 이용한 LIMIT 최적화가 적용되면 잘못된 결과를 출력할 수 있는 문제 수정

조인 질의문에 LIMIT 조건이 주어지고 인덱스를 이용한 LIMIT 최적화가 적용되면, Nested Loop 조인의 외부 테이블과 내부 테이블에 각각 LIMIT 최적화를 적용하는 오류로 인해 기대보다 적은 수의 레코드를 출력하는 문제를 수정하였다. 이 문제는 2008 R4.0 ~ R4.0 Patch 2 버전에서만 발생하였다.

```
SELECT * FROM x, y WHERE x.id > 10 AND x.id = y.id LIMIT 4;
```

## CUBRIDSUS-6322 계층 질의 수행 시 서버가 비정상 종료될 수 있는

### 문제 수정

레코드 개수가 많은 테이블에 대해 계층 질의를 수행하면 비정상 종료되는 문제를 수정하였다.

```
SELECT * FROM tree1 CONNECT BY PRIOR id=PARENTID ORDER BY id
```

### CUBRIDSUS-6282 파티션 테이블 또는 부모 테이블에서 인덱스 관련 제약조건을 DROP 후 재생성하면 서버 프로세스가 비정상 종료되는 문제 수정

파티션 테이블 또는 자식 테이블을 가진 부모 테이블에서 PRIMARY KEY, UNIQUE, FOREIGN KEY 등 인덱스 관련 제약조건을 DROP 후 재생성하면 서버 프로세스가 비정상 종료되는 문제를 수정하였다.

```
CREATE TABLE bar (id int, name VARCHAR(200), PRIMARY KEY(id)) PARTITION BY RANGE(id) (PARTITION aa VALUES LESS THAN(2000), partition bbb VALUES LESS THAN MAXVALUE);
ALTER TABLE bar DROP CONSTRAINT pk_bar_id;
ALTER TABLE bar ADD CONSTRAINT PRIMARY KEY(name, id);
```

### CUBRIDSUS-6352 연관 부질의를 포함한 질의가 잘못된 결과를 출력하는 문제 수정

아래와 같은 연관 부질의를 포함한 질의가 잘못된 결과를 출력하는 문제를 수정하였다.

```
SELECT i FROM t
WHERE
    NOT EXISTS (
        SELECT 1
        FROM (SELECT t1.i
              FROM t1
              UNION
              SELECT t2.i
              FROM t2
             ) a
        WHERE t.i = a.i
    )
ORDER BY 1;
```

### CUBRIDSUS-6404 INSERT 또는 UPDATE 도중 인덱스 변경과 관련하여 발생하는 오류 수정

INSERT 또는 UPDATE로 인해 인덱스 키 하나의 크기가 데이터베이스 한 페이지 크기의 1/8을 초과할 때 발생할 수 있는 다음과 같은 오류를 수정하였다.

```
FATAL ERROR *** file /home1/cubrid/src/storage/btree.c, line 8481 ERROR
CODE = -2 Tran = 1, EID = 4
Internal system failure: no more specific information is available.
```

### CUBRIDSUS-5808 서로 다른 DB 연결들이 하나의 세션 ID를 참조할 수 있는 오류 수정

서로 다른 DB 연결들이 하나의 세션 ID를 참조할 수 있는 오류로 인해 사용자 정의 변수, PREPARE 문, LAST\_INSERT\_ID, ROW\_COUNT 등의 값이 잘못될 수 있는 문제를 수정하였다. 다만, HA failover 시에는 수정 이후 버전에서도 해당 현상이 여전히 발생할 수 있다.

### CUBRIDSUS-6054 "SELECT COUNT(DISTINCT 상수) FROM tbl"의 결과를 1건으로 출력하도록 수정

"SELECT COUNT(DISTINCT 상수) FROM tbl"의 결과로 이전에는 레코드 전체 개수를 출력하였으나 1건을 출력하도록 수정하였다.

### CUBRIDSUS-6370 다중 컬럼 인덱스를 구성하는 컬럼의 NULL 값을 UPDATE하면 오류가 발생하는 문제 수정

다중 컬럼 인덱스를 구성하는 컬럼에 존재하는 NULL 값을 UPDATE하면 다음의 오류가 발생하는 문제를 수정하였다.

```
Unknown key {NULL, '45', NULL, '45'} referenced in B+tree index {vfid: (151, 0), rt_pgid: 550, key_type: midxkey}
```

## CUBRIDSUS-6347 PREPARE 실패 시 cubrid broker status의

### 에러 카운트를 증가하도록 수정

PREPARE 실패 시 cubrid broker status의 에러 카운트(ERR-Q)를 증가하도록 수정하였다.

## CUBRIDSUS-5879 CCI 인터페이스에서 자동 커밋 모드 기본값이

### ON이 되도록 수정

CCI 인터페이스에서 브로커의 CCI\_DEFAULT\_AUTOCOMMIT 파라미터를 통해 설정하는 자동 커밋 모드 기본값이 ON이 되도록 수정하였다.

## CUBRIDSUS-6220 CCI, PHP, ODBC 응용 프로그램에서 어떤 질의

### 요청 핸들러(request handler)를 CLOSE 할 때 다른 질의 요청 핸들러에

### 대한 커밋에 영향을 주는 오류 수정

CCI 응용 프로그램 및 PHP, ODBC 등 CCI 기반 인터페이스로 개발된 응용 프로그램에서 어떤 질의 요청 핸들러(request handler) req1이 CLOSE 되는 시점에 다른 질의 요청 핸들러 req2의 커밋에 영향을 주는 오류를 수정하였다.

다음은 이전 버전에서 문제가 발생하였던 시나리오로, EXECUTE(req1) 이후에는 자동 커밋 모드가 OFF로 바뀌었음에도 불구하고 CLOSE(req1) 시점에 req1에 잘못 기록되었던 자동 커밋 모드를 참조하여 커밋이 수행되면서 req2의 변경내역도 반영되었다.

```
AUTOCOMMIT ON
req1 = PREPARE
      EXECUTE(req1)
AUCOMMIT OFF
req2 = PREPARE
      EXECUTE(req2)
CLOSE(req1)
```

## CUBRIDSUS-6327 CCI, PHP, ODBC, OLE DB 등의 응용

### 프로그램에서 PREPARE 함수 수행 중 오류가 발생하면 이후 트랜잭션

### 롤백을 실행해도 롤백되지 않는 현상 수정

CCI 응용 프로그램 및 PHP, ODBC, OLE DB 등 CCI 기반 인터페이스로 개발된 응용 프로그램에서 PREPARE 함수 수행 중 오류가 발생하면 이후 트랜잭션 롤백을 실행해도 롤백되지 않는 현상을 수정하였다.

## CUBRIDSUS-6366 Linux 환경의 CCI 인터페이스 기반 응용

### 프로그램에서 네트워크 소켓 fd 값이 1024를 초과하는 경우 비정상

### 동작하는 문제 수정

Linux 환경의 CCI 인터페이스 기반 응용 프로그램에서 네트워크 소켓 fd(file descriptor) 값이 1024를 초과하는 경우 CCI 인터페이스 구현에 사용된 네트워크 함수인 select()의 한계로 인해 응용 프로그램이 잘못된 메모리를 참조하게 되어, 비정상 종료 또는 통신 오류가 발생하는 문제를 수정하였다.

## CUBRIDSUS-6491 cci\_datasource\_borrow 함수 호출 시 연결을

### 가져오는 대기 시간까지 기다리지 않고 오류를 반환할 수 있는 문제 수정

cci\_datasource\_borrow 함수 호출 시 연결을 가져오기 위해 대기하는 최대 시간(max\_wait)까지 기다리지 않고 "All connections are used"라는 오류를 반환할 수 있는 문제를 수정하였다.

## CUBRIDSUS-6606 CCI 인터페이스의 DATASOURCE 사용 시

### statement 풀링을 설정하면 발생하는 메모리 누수 현상 수정

cci\_datasource\_create 함수로 DATASOURCE를 생성할 때 statement 풀링 기능을 사용하도록 설정하면 발생하는 메모리 누수 현상을 수정하였다.



## CUBRIDSUS-6673 CCI 인터페이스로 개발된 응용에서 COMMUNICATION ERROR 발생 시 비정상 동작하는 현상 수정

CCI 인터페이스로 개발된 응용에서 "COMMUNICATION ERROR net\_read\_header()" 오류 발생 시 비정상 동작하는 현상이 존재하였으나 네트워크 함수 오류에 대한 검사를 추가하여 이 문제를 수정하였다.

## CUBRIDSUS-6318 JDBC에서 Java의 BigDecimal 클래스를 사용하여 BIGINT 타입의 값을 가져오는 경우 잘못된 값을 가져오는 오류 수정

JDBC에서 Java의 BigDecimal 클래스를 사용하여 BIGINT 타입의 값을 가져오는 경우 잘못된 값을 가져오는 오류를 수정하였다.

```
BigDecimal nInt = rs.getBigDecimal(1);
```

## CUBRIDSUS-6290 JDBC 드라이버에서 한 번의 요청으로 입력 가능한 데이터의 크기 제한을 제거

JDBC 드라이버에서 한 번의 요청(예를 들어 Java 응용 프로그램에서 execute 혹은 executeBatch와 같은 메소드를 통한 요청)으로 입력 가능한 데이터의 크기가 이전 버전에서는 100MB로 제한되었으나, 크기 제한이 제거되었다. 2008 R4.0 Patch 2 이하 버전에서는 크기 제한을 넘을 경우 다음의 오류가 발생하였다.

```
Java.lang.ArrayIndexOutOfBoundsException : 1024
At cubrid.jdbc.jci.UOutputBuffer.writeByte(UOutputBuffer.java : 607)
```

## CUBRIDSUS-6453 독립 모드로 실행한 CSQL 인터프리터에서 사용자 세션 변수로 PREPARE 문을 수행하면 비정상 종료하는 현상 수정

독립 모드(-S)로 실행한 CSQL 인터프리터에서 사용자 세션 변수를 사용하여 PREPARE 문을 수행하면 비정상 종료하는 현상을 수정하였다.

```
prepare st from 'select ?';
set @a='a';
execute st using @a;
```

## CUBRIDSUS-6106 addvoldb, spacedb 유틸리티를 SA 모드로 수행할 때 16MB의 데이터 버퍼를 사용하도록 수정

cubrid addvoldb, spacedb 유틸리티를 SA 모드로 수행할 때 이전 버전에서는 512M의 데이터 버퍼를 사용하였으나, 16MB의 데이터 버퍼를 사용하도록 수정하였다.

## CUBRIDSUS-6235 CUBRID 매니저와 같은 응용 프로그램에서 장시간 동안 요청이 없는 경우에도 연결이 종료되지 않도록 수정

CUBRID 브로커가 설치된 장비 또는 방화벽 장비에서 일정 시간 동안 데이터의 전송이 없으면 소켓 연결을 종료하게 설정되어 있는 경우, 연결 종료 이후 재연결 과정에서 많은 시간이 소요된다. 이와 같이 설정된 장비에 연결한 CUBRID 매니저의 질의 편집기를 장시간 동안 방치하였다가 질의를 수행하면, 결과를 출력하기 전까지 장시간이 소요되는 현상을 수정하였다.

## CUBRIDSUS-5472 질의 수행 중 인터럽트가 발생하는 경우 CSQL 또는 브로커 응용 서버(CAS)가 비정상 종료되는 문제 수정

클라이언트/서버 모드로 실행한 CSQL 혹은 질의 실행 함수(cci\_execute/cubrid\_execute)를 ASYNC 모드로 설정한 CCI/PHP 인터페이스 프로그램에서 질의 수행 중 인터럽트(ctrl+C, 또는 query cancel)가 발생하는 경우, CSQL 또는 브로커 응용 서버(CAS)가 비정상 종료되는 문제를 수정하였다.



## CUBRIDSUS-5816 크기를 지정하는 CUBRID 유틸리티의 옵션 혹은 시스템 파라미터에 소수점 이하의 숫자 입력을 허용하도록 수정

크기를 지정하는 CUBRID 유틸리티의 옵션 혹은 시스템 파라미터에 소수점 이하의 숫자 입력을 허용하였다.

```
cubrid createdb --db-volume-size=0.5G --log-volume-size=0.5G testdb
cubrid addvoldb -p data --db-volume-size=0.25G testdb

# cubrid.conf
db_volume_size=0.5G
log_volume_size=0.5G
data_buffer_size=0.5G
log_buffer_size=2.5M
sort_buffer_size=2.5M
```

## CUBRIDSUS-6275 인덱스가 삭제되는 작업 도중에 인터럽트가 발생하면 나타날 수 있는 인덱스 깨짐 현상 수정

인덱스가 삭제되는 작업 도중에 질의 타임아웃 등의 인터럽트가 발생하면 나타날 수 있는 인덱스 깨짐 현상을 수정하였다.

## CUBRIDSUS-6418 온라인 백업 수행 도중 DDL을 동시에 수행하면 서버 프로세스가 동작을 멈추는(hang) 문제 수정

온라인 백업 수행 도중 DDL을 동시에 수행하면 서버 프로세스가 동작을 멈추는(hang) 문제를 수정하였다. 이 문제는 2008 R4.0 Patch 2 버전에서만 발생하였다.

## CUBRIDSUS-6279 인덱스 리프 노드(leaf node)의 이전 링크가 잘못되는 오류 수정

인덱스 병합(merge) 또는 분할(split) 시 리프 노드(leaf node)의 이전 링크가 잘못되어, 역방향 검색 결과가 잘못 출력되거나 임시 볼륨이 급격히 증가하는 문제를 수정하였다. 이 문제는 2008 R4.0 GA 및 그 이후 버전에서 발생하였다.

## CUBRIDSUS-6001 다른 트랜잭션이 입력 도중 실패한 공간을 사용하기 위해 대기하게 되는 현상 수정

트랜잭션 A가 고유 키 위반(unique key violation)등으로 특정 레코드의 INSERT에 실패한 경우 트랜잭션이 종료되기 전까지 해당 슬롯의 잠금을 유지함으로 인해, 트랜잭션 B는 해당 슬롯의 잠금이 해제되기를 기다리는 현상이 존재하였으나, 트랜잭션 B가 새로운 슬롯을 찾아 대기 과정 없이 INSERT하도록 수정하였다.

## CUBRIDSUS-6396 UPDATE 도중 인터럽트가 발생하면 데이터베이스 볼륨 깨짐(crash) 현상 수정

UPDATE 도중 인터럽트가 발생한 이후 인덱스 복구 과정에서 오류가 존재하여 데이터베이스 볼륨이 깨지는 현상을 수정하였다.

## CUBRIDSUS-6720 부울리언 표현식 처리 과정에서 서버 프로세스의 메모리 누수 현상 수정

부울리언(Boolean) 표현식 처리 과정에서 서버 프로세스(cub\_server)의 메모리 누수 현상을 수정하였다.

## CUBRIDSUS-6072 응용프로그램에서 트랜잭션을 종료하고 질의 요청 핸들을 CLOSE한 이후 일정 시간 동안 요청이 없는 경우 연결이 종료되는 문제 수정

응용프로그램에서 트랜잭션을 종료하고 질의 요청 핸들을 CLOSE한 이후, 일정 시간 동안 요청이 없으면 세션 타임아웃(브로커 파라미터인 SESSION\_TIMEOUT으로 설정)이 발생하여 브로커 응용 서버(CAS)와 응용 프로그램 간의 연결이 종료되는 문제를 수정하였다.

## CUBRIDSUS-6342 체크포인트 오류로 인해 서버가 비정상 종료된 이후 데이터베이스 복구에 실패할 수 있는 문제 수정

체크포인트 오류로 인해 서버가 비정상 종료되고, 이후 데이터베이스 복구에 실패할 수 있는 문제를 수정하였다.

```
Unable to mount disk volume "/data/DB/demodb/log/demodb_lgar592".... No such file or directory
```

## CUBRIDSUS-6513 서버 프로세스의 오류 로그에 소스의 파일 이름과 라인 번호가 항상 출력되도록 수정

서버 프로세스(cub\_server)에서 기록하는 오류 로그에 소스의 파일 이름과 라인 번호가 항상 출력되도록 수정하였다.

## CUBRIDSUS-6240 SLOW SQL 로그 기록 시 질의 수행 시작 시각과 완료 시각이 모두 기록되도록 수정

SLOW SQL 로그 기록 시 질의 수행 완료 시점의 시각이 일괄적으로 기록되었으나, 질의 수행 시작 시각과 완료 시각이 모두 기록되도록 수정하였다.

```
11/29 11:10:43.477 (10) execute srv_h_id 1 select 1 from db_class a, db_class b, db_class c, db_class d
11/29 11:10:45.076 (10) execute error:-4 tuple 0 time 1.634, EID = 8
```

## CUBRIDSUS-5956 브로커 응용 서버(CAS) 프로세스의 메모리 사용량이 급격히 증가하는 경우를 차단하기 위한 방법 제공

브로커 응용 서버(CAS) 프로세스의 메모리 사용량이 급격히 증가하는 경우를 차단하기 위해 지정한 메모리 사용량을 초과하면 강제로 CAS 프로세스를 재구동하게 하였다.

이와 관련하여 새로 추가된 브로커 파라미터는 APPL\_SERVER\_MAX\_SIZE\_HARD\_LIMIT로서, CAS 프로세스가 이 파라미터에 의해 지정된 메모리 사용량을 초과하면 진행 중인 트랜잭션이 강제로 종료(롤백)되고 해당 CAS 프로세스가 재구동된다. 기본값은 1024MB이다. 이전 버전부터 있던 APPL\_SERVER\_MAX\_SIZE 브로커 파라미터는 이와 비슷한 역할을 하지만, CAS 프로세스가 이 파라미터에 의해 지정된 메모리 사용량을 초과해도 진행 중인 트랜잭션이 있으면 사용자에 의해 정상 종료(커밋 혹은 롤백)되기를 기다린 후 재구동된다는 점이 다르다.

일반적으로 메모리 사용량이 지정된 크기를 초과해도 어느 정도까지는 트랜잭션이 정상 종료되기를 기다리기 위해서 APPL\_SERVER\_MAX\_SIZE\_HARD\_LIMIT의 값을 APPL\_SERVER\_MAX\_SIZE의 값보다 크게 설정할 것을 권장한다.

## CUBRIDSUS-5851 replica 노드에서는 ASYNC 모드로만 트랜잭션 로그를 복사하도록 수정

HA 환경에서 replica 노드인 경우, cubrid\_ha.conf의 ha\_copy\_sync\_mode 파라미터 설정과 관계 없이 ASYNC 모드로 트랜잭션 로그를 복사하도록 수정하였다.

## CUBRIDSUS-4817 HA 환경에서 한 트랜잭션 내 변경 레코드 개수가 매우 많은 경우 메모리 부족으로 인해 복제 반영이 되지 않는 문제 수정

HA 환경에서 한 트랜잭션 내 변경 레코드 개수가 매우 많은 경우 복제 로그를 반영하는 applylogdb 프로세스의 메모리 사용량이 프로세스에 허용된 최대 메모리 사용량을 초과하면서 복제 반영이 되지 않는 문제를 수정하였다.

## CUBRIDSUS-6117 HA 환경에서 2008 R4.0 Patch2 버전 이후 발생한 applylogdb 프로세스의 성능 퇴행 개선

HA 환경에서 2008 R4.0 Patch2 버전 이후 발생한 복제 로그 반영 프로세스(applylogdb)의 성능 퇴행을 개선하였다.

---

### **CUBRIDSUS-6172 HA 환경에서 failover 수행 이후, 슬레이브에서 마스터로 변경된 노드에 복제된 보관 로그 파일이 삭제되지 않고 계속 남아 있을 수 있는 현상 수정**

HA 환경에서 failover 수행 이후, 슬레이브에서 마스터로 변경된 노드에 보관 로그(archive log) 파일의 유지 개수를 설정하는 시스템 파라미터인 `log_max_archives`를 설정하였음에도 불구하고, 복제된 보관 로그 파일이 삭제되지 않고 계속 남아 있을 수 있는 현상을 수정하였다.

---

### **CUBRIDSUS-6183 HA 환경에서 applylogdb 프로세스가 쓰기가 끝나지 않은 로그 페이지를 버퍼에 캐싱하여 읽고 반영하면서 오동작하는 문제 수정**

HA 환경에서 copylogdb 프로세스에 의해 쓰기가 끝나지 않은 로그 페이지를 applylogdb 프로세스가 페이지 버퍼에 캐싱한 후 이를 읽고 반영하면서 잘못된 로그 페이지 반영, 프로세스 비정상 종료 등 오동작할 수 있는 문제를 수정하였다.

---

### **CUBRIDSUS-6193 HA 환경에서 마스터 서버 테이블의 기본 키를 변경한 직후 온라인 백업본으로 슬레이브 서버를 재구축하면 슬레이브 서버 프로세스가 비정상 종료될 수 있었으나 오류 처리하도록 수정**

HA 환경에서 마스터 서버의 기본 키(primary key)를 변경한 직후 온라인 백업본으로 슬레이브 서버를 재구축하는 경우, 기본 키 변경 이전의 로그를 기본 키 변경이 완료된 테이블에 반영하려고 시도하면서 슬레이브 서버 프로세스가 비정상 종료될 수 있었으나, 로그 반영 과정에서 기본 키 구성이 로그 레코드와 일치하지 않으면 오류로 처리하도록 수정하였다.

---

### **CUBRIDSUS-5475 HA 환경의 슬레이브 서버에서 INCR/DECR 함수를 호출하면 해당 데이터를 업데이트할 수 없도록 수정**

HA 환경의 슬레이브 서버에서 INCR/DECR 함수를 호출하면 이전 버전에서는 함수가 수행되면서 마스터 서버와 슬레이브 서버 간 데이터 불일치가 발생하였으나, 해당 데이터를 업데이트할 수 없도록 수정하였다.

---

### **CUBRIDSUS-5939 HA 환경에서 standby 상태의 노드를 maintenance 상태로 변경할 때 진행 중이던 트랜잭션이 정상 종료될 때까지 대기하는 시간을 설정하는 기능 제공**

HA 환경에서 changemode 유틸리티를 실행하여 standby 상태의 노드를 maintenance 상태로 변경할 때, 진행 중이던 트랜잭션이 정상 종료될 때까지 대기하는 시간을 설정하는 `-t` 옵션(생략 시 기본값: 5초)을 제공하였다. 설정한 시간까지 트랜잭션이 진행 중이면 강제 종료 후 maintenance 상태로 변경하고, 설정한 시간 이내에 모든 트랜잭션이 정상 종료되면 즉시 maintenance 상태로 변경한다.

```
// 진행 중이던 트랜잭션의 수행 시간이 changemode 실행 시점 이후 10 초를 넘으면 트랜잭션이 롤백되고, 노드 상태는 maintenance
// 상태로 변경된다.
cubrid changemode -m maintenance -t 10 demodb
```

---

### **CUBRIDSUS-6154 데이터 버퍼의 더티 페이지가 디스크에 저장되는 주기를 설정하는 시스템 파라미터의 값이 -1일 때의 동작 방식 변경**

데이터 버퍼의 더티 페이지(dirty page)가 디스크에 저장되는 주기를 설정하는 시스템 파라미터인 `page_flush_interval_in_msecs`의 값이 -1인 경우 체크포인트 또는 페이지가 스왑(swap)되는 시점에만 더티 페이지가 디스크로 저장되었으나, 주기 값이 0인 경우와 동일하게 바로 저장되도록 변경하였다.

### CUBRIDSUS-6133 교착상태를 감지하는 주기를 설정하는 시스템

#### 파라미터 값의 입력 단위를 소수점 이하 한자리까지 가능하도록 변경

교착상태(deadlock)를 감지하는 주기를 설정하는 시스템 파라미터인 `deadlock_detection_interval_in_secs` 값의 입력 단위를 소수점 이하 한자리까지 가능하도록 변경하였다. 최소값은 1초에서 0.1초로 변경되었다. 이 값은 0.1초 단위로 올림하여 동작한다. 즉, 입력값이 0.12초이면 0.2초를 입력한 것과 같이 동작한다.

### CUBRIDSUS-6083 스레드의 스택 크기를 결정하는 기준 및 크기를

#### 설정하는 시스템 파라미터의 기본값 변경

OS에 설정된 스레드의 스택 크기가 시스템 파라미터인 `thread_stacksize` 보다 큰 경우 이전 버전에서는 OS의 설정 값에 의해 크기가 정해졌으나, 항상 `thread_stacksize`의 값에 의해 크기가 정해지도록 변경하였으며, `thread_stacksize`의 기본값을 100KB에서 1MB로 변경하였다.

### CUBRIDSUS-6668 데이터 캐싱 버퍼의 크기를 설정하는 시스템

#### 파라미터의 최소값 변경

데이터 캐싱 버퍼의 크기를 설정하는 시스템 파라미터인 `data_buffer_size`의 최소값을 64K에서 16M로 변경하였다.

### CUBRIDSUS-6035 Linux 용 CUBRID 패키지를 기존 CUBRID

#### 버전이 설치되어 있는 경로에 설치하면 파일들이 잘못 설치되는 오류 수정

Linux 용 CUBRID 패키지(.sh)를 기존 CUBRID 버전이 설치되어 있는 경로에 설치하는 경우 파일들이 잘못 설치되는 오류를 수정하였다.

### CUBRIDSUS-6061 JDK1.7로 JDBC 드라이버 소스를 빌드하지 못하는

#### 문제 수정

JDK1.7로 JDBC 드라이버 소스를 빌드하지 못하는 문제를 수정하였다. 참고로, JDK 1.7로 JDBC 드라이버 소스를 빌드하기 위해서는 ANT 1.8 이상의 버전이 필요하다.

### CUBRIDSUS-5942 Windows 시스템을 재부팅한 이후 CUBRID

#### Service Tray가 자동으로 구동될 수 있는 환경인지 CUBRID 설치 중에

#### 미리 확인이 가능하도록 수정

Windows 시스템을 재부팅한 이후 CUBRID Service Tray가 자동으로 구동될 수 있는 환경인지 CUBRID 설치 중에 미리 검사하여, 문제 발견 시 주의 메시지를 출력하도록 수정하였다. 시스템 재부팅 시 CUBRID Service Tray가 자동으로 구동되려면 "제어판 > 관리 도구 > 서비스"의 Task Scheduler가 시작된 상태에서 "관리 도구 > 작업 스케줄러"에 CUBRID Service Tray가 등록되어야 한다.

### CUBRIDSUS-6028 Windows 용 CUBRID 패키지 설치 후 시스템을

#### 재부팅해야 cubrid 유틸리티가 실행되는 문제 수정

Windows 용 CUBRID 패키지 설치 후 시스템을 재부팅하기 전까지 CUBRID 관련 환경 변수가 시스템에 적용되지 않는 현상으로 인해 cubrid 유틸리티가 실행되지 않는 문제를 수정하였다.

### CUBRIDSUS-6278 Windows용 CCI 라이브러리(cascci.dll)를

#### 사용한 응용 프로그램 수행 시 CRT DLL이 필요하지 않도록 수정

Windows용 CCI 라이브러리(cascci.dll)를 사용한 응용 프로그램 수행 시 CRT 동적 연결 라이브러리(C RunTime DLL)가 필요하지 않도록 수정하였다.

## 5. 주의 사항

### 2008 R4.1 신규 주의 사항

#### CUBRIDSUS-5879 2008 R4.1 버전부터

##### CCI\_DEFAULT\_AUTOCOMMIT 의 기본값이 ON으로 바뀜

2008 R4.1 버전부터 CCI 인터페이스로 개발된 응용 프로그램의 자동 커밋 모드에 영향을 주는 브로커 파라미터인 CCI\_DEFAULT\_AUTOCOMMIT의 기본값이 ON으로 변경되었다. 따라서 CCI 및 CCI로 개발된 인터페이스(PHP, ODBC, OLE DB 등) 사용자는 응용 프로그램의 자동 커밋 모드가 이에 적합한지 살펴보아야 한다.

#### CUBRIDSUS-5238 2008 R4.1 버전은 2008 R4.0 Beta 및 그 이전

##### 버전과 데이터베이스 볼륨이 호환되지 않음

2008 R4.0 버전 이상 사용자는 2008 R4.1과 데이터베이스 볼륨이 호환된다.

2008 R4.1 버전은 2008 R4.0 Beta 및 그 이전 버전과 데이터베이스 볼륨이 호환되지 않으므로 데이터베이스를 업그레이드하려면 데이터베이스 볼륨을 마이그레이션해야 한다.

2008 R3.x 및 이전 버전 사용자는 **cubrid unloaddb/loaddb**를 이용한다.

2008 R4.0 Beta 사용자는 다운로드 페이지에서 제공되는 **migrate\_r40beta2ga** 유틸리티를 이용할 수 있으나, 페이지 크기가 4K 미만인 볼륨은 **cubrid unloaddb/loaddb**를 이용하여야 한다.

보다 자세한 사항은 [데이터베이스 마이그레이션 절차](#)를 참고한다.

## 기존 주의 사항

#### CUBRIDSUS-5597 CCI, PHP, JDBC 연결 URL에서 구분자를 암호에

##### 포함할 수 없음

CCI, PHP, JDBC 연결 URL에서 구분자로 사용되는 ? 또는 :을 암호에 포함할 수 없다. 다음은 암호에 ?가 있어 연결 URL로 사용할 수 없는 예이다.

```
// CCI, PHP의 경우
url = "cci:jdbc:CUBRID:192.168.0.10:33000:tdb:dba:12?:?charset=UTF-8";
// JDBC의 경우
url = "jdbc:CUBRID:192.168.0.10:33000:tdb:dba:12?:?charset=UTF-8";
```

암호에 ? 또는 :을 포함한 경우에는 암호를 별도의 인자로 전달하여 사용할 수 있다.

```
// CCI의 경우
url = "cci:CUBRID:192.168.0.10:33000:tdb:::?charset=UTF-8";
cci_connect_with_url(url, "dba", "12?");
// PHP의 경우
url = "cci:CUBRID:192.168.0.10:33000:tdb:::~?charset=UTF-8";
cubrid_connect_with_url(url, "dba", "12?");
// JDBC의 경우
url = "jdbc:CUBRID:192.168.0.10:33000:tdb:::~?charset=UTF-8";
conn = DriverManager.getConnection(url, "dba", "12?");
```

이 주의 사항은 모든 버전에 해당한다.

#### CUBRIDSUS-5136 페이지 단위의 옵션 제거 예정

**cubrid createdb** 유틸리티의 데이터베이스 볼륨 크기 및 로그 볼륨 크기를 지정할 때 페이지 단위를 사용하는 옵션들(-p, -l, -s)은 제거될 예정이므로, 2008 R4.0 Beta 이후 새로 추가된 옵션들(--db-volume-size, --log-volume-size, --db-page-size, --log-page-size)을 사용한다.

**cubrid addvolddb** 유틸리티의 데이터베이스 볼륨 크기를 지정하는 경우에도 페이지 단위를 사용하지 않고 2008 R4.0 Beta 이후 새로 추가된 옵션(--db-volume-size)을 사용한다.

### CUBRIDSUS-4222 데이터베이스 볼륨 크기 설정 시 주의 사항

2008 R4.0 Beta 버전부터 데이터베이스 생성 시 데이터 페이지 및 로그 페이지의 크기 기본값이 4KB에서 16KB로 변경되었으므로, 데이터베이스 볼륨을 페이지 개수로 지정하여 생성하는 경우 볼륨의 바이트 크기가 기대와 다를 수 있음에 주의한다. 아무런 옵션도 주지 않을 경우 이전 버전에서는 4KB의 페이지 크기로 100MB의 데이터베이스 볼륨을 생성하였으나, 2008 R4.0 버전부터는 16KB의 페이지 크기로 512MB의 데이터베이스 볼륨을 생성하게 된다.

그리고, 데이터베이스 볼륨의 생성 가능한 최소 크기를 20MB로 제한하였으므로 이보다 작은 크기의 데이터베이스 볼륨은 생성할 수 없다.

### CUBRIDSUS-4222 페이지 단위의 시스템 파라미터 제거 예정

페이지 단위의 시스템 파라미터들은 추후 제거될 예정이므로 바이트 단위의 새로운 시스템 파라미터를 사용할 것을 권장한다. 관련 시스템 파라미터들에 대한 내용은 아래를 참고한다.

### CUBRIDSUS-4095 일부 시스템 파라미터들의 기본값 변경

2008 R4.0부터 다음 시스템 파라미터들의 기본값이 변경되었다.

DB 서버가 허용하는 동시 접속 개수를 설정하는 `max_clients`의 기본값, 인덱스 페이지 생성 시 향후 업데이트를 대비하여 확보하는 여유 공간 비율을 설정하는 `index_unfill_factor`의 기본값이 변경되었으며, 바이트 단위 시스템 파라미터의 기본값이 기존 페이지 단위 시스템 파라미터의 기본값보다 커져서 별도의 설정을 하지 않는 경우 더 많은 메모리를 사용하게 되었다.

기존 시스템 파라미터	추가된 시스템 파라미터	기존 기본값	변경된 기본값 (단위: 바이트)
<code>max_clients</code>	-	50	100
<code>index_unfill_factor</code>	-	0.2	0.05
<code>data_buffer_pages</code>	<code>data_buffer_size</code>	100M(페이지 크기=4K)	512M
<code>log_buffer_pages</code>	<code>log_buffer_size</code>	200K(페이지 크기=4K)	4M
<code>sort_buffer_pages</code>	<code>sort_buffer_size</code>	64K(페이지 크기=4K)	2M
<code>index_scan_oid_buffer_pages</code>	<code>index_scan_oid_buffer_size</code>	16K(페이지 크기=4K)	64K

또한, **cubrid createdb**로 데이터베이스 생성 시 데이터 페이지 크기와 로그 페이지 크기의 최소값이 1K에서 4K로 변경되었다.

### CUBRIDSUS-5375 시스템 파라미터를 잘못 설정하면 데이터베이스

서비스, 유틸리티 및 응용 프로그램이 구동되지 않음

`cubrid.conf` 또는 `cubrid_ha.conf`에 정의되지 않은 시스템 파라미터를 설정하거나, 페이지 단위의 시스템 파라미터와 바이트 단위의 시스템 파라미터가 동시에 사용되거나, 시스템 파라미터 값이 허용 범위를 벗어나면 이와 관련된 데이터베이스 서비스, 유틸리티 및 응용 프로그램이 구동되지 않는다.

### CUBRIDSUS-4524 복제 기능 제거

CUBRID 2008 R4.0 버전부터 복제 기능이 제거되었으므로, 이중화 환경을 구축하려면 HA 기능을 사용해야 한다. 서버 버전 업그레이드 및 데이터베이스 마이그레이션을 수행한 후, HA 환경을 새롭게 구축할 수 있다. HA 환경 구축과 관련하여, 매뉴얼의 '관리자 안내서 > CUBRID HA'를 참고한다.

### CUBRIDSUS-5228 CUBRID 매니저 설치 패키지 별도 제공

CUBRID 2008 R4.0부터는 CUBRID 매니저 설치 패키지를 별도로 제공하며, CUBRID 매니저를 사용하려면 CUBRID 패키지 설치 후 CUBRID 매니저를 별도로 설치해야 한다. .

## CUBRIDSUS-5097 컬럼 크기보다 큰 문자열을 INSERT/UPDATE 할 때 문자열이 절삭되어 입력됨

컬럼 크기보다 큰 CHAR, VARCHAR, NCHAR, VARNCHAR 타입의 문자열을 INSERT/UPDATE하면 컬럼 크기를 초과하는 문자열 부분을 절삭한다. (매뉴얼의 'CUBRID SQL 설명서 > 데이터 타입 > 문자열 데이터 타입' 참고)

## CUBRIDSUS-5349 CUBRID 32bit 버전에서 data\_buffer\_size에 2G를 초과하는 값을 설정하면 데이터베이스 구동에 실패함

CUBRID 32bit 버전에서 data\_buffer\_size가 2G를 초과하는 값으로 설정되는 경우 데이터베이스 구동에 실패한다. 32bit 버전에서는 OS의 한계로 인해 설정값이 2G를 초과할 수 없음에 주의한다.

## CUBRIDSUS-4059 VARCHAR 타입의 컬럼에서 값을 가져올 때 커버링 인덱스가 적용되는 경우 뒤에 따르는 공백 문자열이 무시됨

VARCHAR 타입의 컬럼에서 값을 가져올 때 커버링 인덱스가 적용되는 경우 뒤에 따라오는 공백 문자열은 절삭된다. 질의 수행 시 커버링 인덱스가 적용되면 질의 결과 값을 인덱스에서 가져오는데, 인덱스에는 뒤이어 나타나는 공백 문자열을 제거한 채로 값을 저장하기 때문이다. 이러한 현상을 원하지 않을 경우에는 NO\_COVERING\_IDX 힌트를 지정하면 된다. (매뉴얼의 'CUBRID SQL 설명서 > 질의 최적화 > 인덱스 활용 > 커버링 인덱스' 참고)

```
CREATE TABLE tab(c VARCHAR(32));
INSERT INTO tab VALUES('abcd'),('abcd '),('abcd ');
CREATE INDEX ON tab(c);

-- 아래 질의는 커버링 인덱스가 적용되어 3 개의 데이터가 모두 같은 조건으로 인식된다.
SELECT * FROM tab WHERE c='abcd ' USING INDEX i_tab_c(+);
c
=====
'abcd'
'abcd'
'abcd'
```

## CUBRIDSUS-3757 HA 관련 주의 사항

CUBRID HA에서 트리거 및 자바 저장 프로시저를 사용할 경우 마스터 노드에서 이미 수행된 트리거 또는 자바 저장 프로시저를 슬레이브 노드에서 중복 수행하여 CUBRID HA 그룹 내의 노드 간 데이터 불일치가 발생할 수 있으므로, CUBRID HA에서는 트리거 및 자바 저장 프로시저를 사용하지 않도록 한다.

CUBRID HA는 복제 로그를 기반으로 CUBRID HA 그룹 내의 노드 간 데이터를 동기화하므로 복제 로그를 생성하지 않는 메소드를 사용하거나 CUBRID 매니저를 통해 NOT NULL 옵션 설정 작업 수행 시 CUBRID HA 그룹 내 노드 간 데이터 불일치가 발생할 수 있으므로, CUBRID HA는 메소드를 사용할 수 없고, CUBRID 매니저를 통해 작업할 수 없다.

## CUBRIDSUS-5071 데이터베이스 백업/복구 시 LOB 타입 저장소는 복구되지 않음

CUBRID에서 LOB 타입의 데이터는 데이터베이스 볼륨이 아닌 별도의 저장소에 존재하므로 데이터베이스의 백업/복구 과정에 포함되지 않는다. 즉, 데이터베이스의 백업 시 LOB 타입 저장소는 같이 백업되지 않으므로, 복구 시 LOB 타입 저장소는 복구되지 않는다. LOB 타입 저장소는 별도로 관리해야 한다.

## 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 R3.1 이상 버전 GLO 클래스를 제거하고 BLOB, CLOB 타입(이하 LOB)을 지원한다. (매뉴얼의 'CUBRID SQL 설명서 > 데이터 타입 > BLOB/CLOB 데이터 타입' 참고)

기존의 GLO 클래스 사용자는 다음과 같이 작업할 것을 권장한다.

- GLO 데이터를 파일로 저장한 후 어플리케이션 및 DB 스키마에서 GLO 를 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 loadddb** 유틸리티는 GLO 클래스를 상속받거나 GLO 클래스 타입을 가진 테이블을 로딩하려는 경우 **Error occurred during schema loading** 오류 메시지와 함께 데이터 로딩을 중지한다.

GLO 클래스의 지원 중단에 따라 각 인터페이스 별로 삭제한 함수는 다음과 같다.

인터페이스	삭제한 함수
CCI	cci_glo_append_data cci_glo_compress_data cci_glo_data_size cci_glo_delete_data cci_glo_destroy_data cci_glo_insert_data cci_glo_load cci_glo_new cci_glo_read_data cci_glo_save cci_glo_truncate_data cci_glo_write_data
JDBC	CUBRIDConnection.getNewGLO CUBRIDOID.loadGLO CUBRIDOID.saveGLO
PHP	cubrid_new_glo cubrid_save_to_glo cubrid_load_from_glo cubrid_send_glo

## 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 NULL 은 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 Locator 만 삭제되고 LOB 컬럼이 참조하는 외부 파일 시스템의 LOB 파일은 삭제되지 않고 남아있다.
- CUBRID 가 제공하는 API 나 CUBRID 매니저, CSQL 을 사용하지 않고 사용자 임의로 LOB 타입의 데이터 파일을 직접 수정하면 내용이 일치됨을 보장할 수 없다.

(매뉴얼의 'CUBRID SQL 설명서 > 데이터 타입 > BLOB/CLOB 데이터 타입' 참고)

## CUBRIDSUS-4186 Windows Vista 이상 버전에서 CUBRID

### 유틸리티를 사용한 서비스 제어 시 권장 사항

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 권한 상승을 확인하는 대화 상자가 활성화되고, “예”를 클릭하여 관리자 권한으로 구동한다.

### 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 명시해야 하고 이를 생략할 경우 에러를 출력한다. 또한, 연결 정보 중 USERNAME과 PASSWORD가 없더라도 반드시 콜론(:)을 명시해야 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** 시스템 파라미터를 수정해야 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, **cubrid unloaddb**, **cubrid loaddb**, **cubrid backupdb** 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, CUBRID 와 CUBRID 매니저를 각각 빌드하여 설치해야 한다. 만약, CUBRID 소스만 checkout 하여 빌드 후 **cubrid service start** 또는 **cubrid manager start** 를 실행하면, **cubrid manager server is not installed** 라는 오류가 발생한다.