

트러블슈팅 가이드			솔루션 번호	1
CATEGORY			솔루션 식별번호 (Code)	
서비스영역	기술분야	세부기술분야		
플랫폼 및 기반구조	데이터베이스	DBMS	SW 3-3-01-001	
솔루션 정보				
솔루션 명칭	CUBRID			
목록	<ol style="list-style-type: none"> 1. CUBRID 사용 포트는 뭔가요? 2. libjvm.so 찾지 못하여 서버 구동이 되지 않은 경우 3. CSQL 에서 한글을 입력하는 방법 4. CUBRID 백업은 어떻게? 5. 백업파일을 이용하여 데이터베이스가 없는 다른 서버에서 복구 6. 데이터베이스 볼륨 자동 증가 설정 7. 브로커 상태를 확인할 때 status 값의 의미는 뭔가요? 8. CUBRID 복제 시 마스터 데이터베이스 서버 장애로 중지되었을 때 9. CUBRID 제거 절차 			
1	문제	CUBRID DBMS가 사용하는 포트는 뭔가요?		
	이론과 배경	CUBRID 데이터베이스는 데이터베이스 서버, 브로커, CUBRID 매니저로 구성되어 있으며, 각각 사용하는 포트를 알고 있어야하며 방화벽 설정 시 오픈하여야 한다.		
	처리방법	<ol style="list-style-type: none"> 1. 데이터베이스 서버 <ul style="list-style-type: none"> o cub_master: 마스터 프로세스는 클라이언트 프로세스가 서버 프로세스에 접속하여 통신할 수 있게 하는 중계 프로세스로서, 호스트별로 한 개씩 동작한다. TCP/IP 사용 port 1523(default) 이며, Parameter CUBRID/conf/cubrid.comf/cubrid_port_id=1523 값을 수정하여 port를 변경할 수 있다. o cub_server: 데이터베이스 서버로 데이터베이스별로 한 개의 서버 프로세스가 존재한다. 외부 클라이언트 프로세스와 데이터베이스 접속은 마스터 프로세스가 지정된 데이터베이스 명에 따라 해당 서버 프로세스로 소켓 통신을 옮겨주어 접속 처리 한다, 데이터베이스 서버에 port는 사용자 변경 대상이 아니다. 		

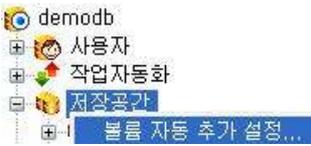
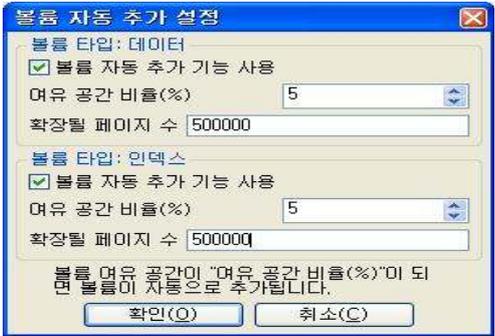
	<p>2. 브로커</p> <ul style="list-style-type: none"> o cub_broker, cub_cas: 브로커는 cas, broker Process로 구분되고 cub_broker는 응용 클라이언트와 cub_cas 사이의 연결을 관리하고 중계하는 기능을 수행하며 cub_cas 관리한다. 또한 연결된 cub_cas는 DB서버의 클라이언트로 동작하여 클라이언트의 요청에 의해 데이터베이스 서버와 연결을 제공한다. <p>기본적으로 port 30000번, 33000번 이 설정 되어 있으나 이는 변경이 가능하며, \$CUBRID/conf/cubrid_broker.conf Parameter BROKER_PORT=30000 값을 수정하여 port를 변경할 수 있다.</p> <p>[%query_editor]</p> <pre> SERVICE = ON BROKER_PORT = 30000 => APPL_SERVER_PORT 가 별도로 등록되어 있지 않다면 30000~30040번까지 사용하게 된다. MIN_NUM_APPL_SERVER = 5 => 초기 cub_cas 기동 개수이다. MAX_NUM_APPL_SERVER = 40 => 최대로 사용 가능한 port 개수이다. APPL_SERVER_PORT = 30100 => 이 항목을 등록할 경우 30101~30140번까지 사용하게 된다. </pre> <p>3. CUBRID 매니저</p> <ul style="list-style-type: none"> o cub_auto, cub_js: CUBRID Manager Process로 매니저 사용이 필요 없는 경우 정지해도 CUBRID DB 사용에 문제는 없다. CUBRID 매니저 서버와 클라이언트 간의 통신 포트로 기본값은 8001이다. 연결 시 실사용 포트는 설정한 포트와 1을 증가한 포트 두 개가 사용된다. 예를 들어, \$CUBRID/conf/cm.conf Parameter cm_port를 8001로 설정하면 8001, 8002 두 개의 포트가 사용된다. 이 부분은 CUBRID 매니저 서버와 클라이언트 사이에 방화벽이 있고 해당 포트만 오픈할 경우 주의해야 한다. <p>정리하면 마스터(1523), 브로커(30000,33000), 매니저(8001.8002) 포트가 열려 있어야 CUBRID 사용하는데 문제가 없다.</p>
<p>관련파일/명령어</p>	<p>cubrid_broker.conf</p>
<p>기타</p>	

2	문제	libjvm.so 찾지 못하여 서버 구동이 되지 않은 경우
	이론과 배경	<p>“Unable to load native library:libjvm.so:cannot open shared object file:No such file or directory”와 같은 오류가 발생하여 서버가 구동이 되지않는다.</p> <p>원인은 Java Stored Procedure를 수행하기 위한 환경으로 JRE 1.4이상이 설치되어 있어야 하는데 다른 리눅스 계열에서는 설치가 안되어 있을 경우 Java Stored Procedure만 수행할 수 없으나, Fedora Core 5에서는 설치가 안되어 있을 경우 CUBRID 서버를 사용할 수 없고 위와 같은 오류가 발생한다.</p>
	처리방법	<p>Java SDK 1.4이상이나, JRE 1.4이상을 설치하고 아래와 같이 Library Path을 설정해야 한다.</p> <ol style="list-style-type: none"> Windows 경우 <pre>set JAVA_HOME=C:\jdk1.5.0 set PATH=%PATH%;%JAVA_HOME%\jre\bin\client</pre> Linux Bash의 경우 <pre>JAVA_HOME=/home/cubrid/jdk1.5.0 ;export JAVA_HOME PATH=.:\$JAVA_HOME/bin:\$PATH ;export PATH LD_LIBRARY_PATH=\$LD_LIBRARY_PATH:\$JAVA_HOME/ jre/lib/i386:\$JAVA_HOME/jre/lib/i386/client ; export LD_LIBRARY_PATH</pre> Linux csh의 경우 <pre>setenv JAVA_HOME \$HOME/jdk.1.5.0 setenv LD_LIBRARY_PATH \$JAVA_HOME/jre/lib/i386:\$JAVA_HOME/jre/lib/i386/client: \$LD_LIBRARY_PATH set path=(\$path \$JAVA_HOME/bin .)</pre>
	관련파일/명령어	
	기타	
3	문제	CSQL 에서 한글을 입력하는 방법
	이론과 배경	CSQL 에서는 한글 입력을 지원 하고 있지 않지만 질의나 명령에 반드시 한글을 입력 하셔야 할 경우, 아래의 절차를 따르면 한글을 입력 할 수 있다.
	처리방법	1. csql> 프롬프트에서 ;edit 명령을 입력한다.

		<p>2. 나타나는 에디터 (리눅스 : VI, 윈도우 : notepad) 에서 수행할 명령 또는 질의를 입력한다.</p> <p>3. 저장을 하고 빠져 나온다.</p> <p>4. ;x 또는 ;ru 를 실행하여, 저장된 명령 또는 질의를 수행한다.</p>
	관련파일/명령어	csql
	기타	
4	문제	CUBRID 백업은 어떻게?
	이론과 배경	CUBRID 백업 종류는 backupdb, unloaddb 두가지 방법이 있다.
	처리방법	<p>1. cubrid 데이터베이스 볼륨, 제어 파일, 로그 파일을 백업하는 방법으로 cubrid backupdb 명령과 cubrid 매니저를 이용하여 수행할 수 있다. 온/오프라인 모두 지원되며 미디어 고장과 데이터베이스 파일들의 손상으로부터 데이터베이스 복구에 사용되고 특정 시점복구가 가능하다.</p> <p>예) cubrid backupdb -C -r -z demodb -> 운영중인 demodb full backup 진행.</p> <p>※ [참고사항] cubrid backupdb는 증분 백업으로 관리할 수 있다, 증분 백업(incremental backup)은 0, 1, 2 레벨의 백업 수준을 제공하며, 0레벨은 full backup, 1레벨은 0레벨(full backup) 이후 증분(incremental) backup, 2레벨은 1레벨 이후 증분 백업으로 낮은 백업을 순차적으로 수행해야 한다,</p> <p>2. 데이터베이스를 아스키 파일(text 형태)로 생성하는 유틸리티로 cubrid unloaddb 명령을 사용하는데 시스템 교체, 제품 버전 업그레이드, 또는 데이터베이스 볼륨의 재구성을 위해 스키마, 데이터, 인덱스등을 내려 받는다, cubrid unloaddb 명령은 온/오프라인 모두 지원하며, 옵션에 따라 스키마, 데이터 별로 내려 받을 수 있다,</p> <p>예) cubrid unloaddb -C demodb -> 운영중인 demodb 모든 자료를 내려 받는다.</p> <p>※ 일반적인 백업방법은 cubrid backupdb 사용을 권장하고, 특정 테이블 데이터 이관이나 데이터베이스 재구성 필요할 경우 cubrid unloaddb와 cubrid loaddb을 이용한다.</p>
	관련파일/명령어	cubrid backupdb, cubrid unloaddb
	기타	

	문제	백업파일을 이용하여 데이터베이스가 없는 다른 서버에서 복구
	이론과 배경	데이터베이스가 삭제되었거나, 다른 서버에 데이터베이스를 복사하고자 할때 백업파일만을 이용하여 데이터베이스를 복구할 수가 있다.
5	처리방법	<p>※ 전제조건</p> <ol style="list-style-type: none"> 백업당시와 동일한 DBMS 버전 원본 데이터베이스와 동일한 디렉토리 구성 <p>※ Unix(Linux 동일) 환경으로 설명하며, Windows 환경과 다른점은 아래와 같다.</p> <ul style="list-style-type: none"> 데이터베이스 등록정보 파일명 <ul style="list-style-type: none"> databases.txt 파일 패스 <ul style="list-style-type: none"> UNIX : 예) /database/demodb WINDOWS : 예) D:\database\demodb 환경 변수 <ul style="list-style-type: none"> UNIX : 예) \$CUBRID WINDOWS : 예) %CUBRID% <p>※ 백업파일만을 이용하여 데이터베이스 복구한다.</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <ul style="list-style-type: none"> 데이터베이스 : CUBRID 서버이름 : db_server 데이터베이스 이름 : demodb 설치 위치 : /database/demodb, 로그는 /database/demodb/log </div> <ol style="list-style-type: none"> 백업당시와 동일한 DBMS 버전 환경 마련 이미 설치되어 있다면 상관없고, 그렇지 않다면 제품을 설치한다. 원 데이터베이스가 있던 위치와 동일한 디렉토리를 구성한다. <pre>% mkdir /database % mkdir /database/demodb % mkdir /database/demodb/log</pre> <ul style="list-style-type: none"> ※ UNIX(LINUX) 의 경우 적절한 권한을 주어야 할 수도 있다. 데이터베이스 등록정보 파일을 편집하여 데이터베이스 정보를 입력한다. 이 정보는 원 데이터베이스의 위치 정보이며, 원 데이터베이스가 있던 서버의 \$CUBRID_DATABASESES/databases 의 내용을 그대로 옮기면 되며, 다만 서버가 달라지는 경우 앞에서 세번째 내용이 서버이름이므로 이 값만 현재 서버의 서버이름으로 적으면 된다.

		<pre>% vi \$CUBRID_DATABASES/databases.txt demodb /database/demodb db_server /database/demodb/log</pre> <p>4. 데이터베이스 백업파일을 임의의 위치에 복사한다. 이때 가장 마지막 백업본을 사용한다. 단 마지막 백업이 풀(level 0) 백업이 아닌 경우 현재 레벨 이전의 모든 백업본이 필요하다. 예를 들어 현재 백업본이 level 2 라면 level 0, level 1 모두 필요하다. 디폴트로 백업시 백업파일명은 demodb_bk0v000 형식이며, 이때 bk 다음 v 앞의 숫자가 level 이다. 0v 이면 level 0 백업 파일이다. /tmp 아래에 복사한 것으로 가정함</p> <p>5. 데이터베이스 복구 명령을 수행한다. 마지막 백업이 level 0 일 경우에는 아래와 같이 수행한다. % cubrid restoredb -B /tmp -p demodb 만약 마지막 백업이 level 2 라면 아래와 같이 수행한다. % cubrid restoredb -l 2 -B /tmp -p demodb</p> <p>6. csq1 를 수행하여 정상 복구 여부를 확인한다. % csq1 -S demodb csq1> ;exit</p>
	관련파일/명령어	cubrid, csq1
	기타	
6	문제	데이터베이스 볼륨 자동 증가 설정
	이론과 배경	<p>데이터베이스의 볼륨(데이터 저장 공간)은 데이터베이스 서버에 의해 공간이 부족해지면 자동으로 증가가 된다. 일반적으로 데이터베이스 생성 시 데이터볼륨, 인덱스볼륨을 별도로 생성하여 사용하게 되는데 데이터베이스 서버가 자동으로 볼륨을 추가하게 되는 경우 데이터나 인덱스 볼륨이 아니라 범용볼륨을 추가하게 된다. 보통 범용볼륨은 그 크기를 작게 생성하게 되므로 자동으로 추가되는 볼륨은 그 크기가 작기 때문에 볼륨 증가가 자주 발생하게 되어 성능에도 영향을 준다. 따라서 데이터공간이 부족할 경우에는 데이터공간을 적정크기로 증가시켜주는 설정이 필요하며 이러한 설정은 CUBRID 매니저를 이용하여 설정할 수 있다.</p>
	처리방법	매니저에 접속후 데이터베이스 아래 저장공간을 우 클릭후 "볼륨자동추가"를 선택합니다.

		 <p>1. 볼륨타입:데이터 부분에 "볼륨 자동 추가 기능 사용"을 체크후, 확장될 페이지 수를 입력한다. 페이지 수는 기존 데이터베이스의 데이터 볼륨 크기와 동일하게 합니다. 여유공간 비율은 전체 데이터볼륨의 여유공간이 해당 비율보다 작아졌을때 데이터볼륨이 추가되도록 한다.</p> <p>2. 볼륨타입:인덱스 부분에 "볼륨 자동 추가 기능 사용"을 체크후, 확장될 페이지 수를 입력한다. 페이지수는 기존 데이터베이스의 인덱스 볼륨 크기와 동일하게 하며 "확인"을 클릭하여 설정을 저장한다.</p> 
	관련파일/명령어	
	기타	
7	문제	브로커 상태를 확인할 때 status 값의 의미는 뭔가요?
	이론과 배경	cubrid의 유틸리티로 broker의 상태를 확인할 수 있다.
	처리방법	<p>1. cubrid의 유틸리티 중 broker의 상태를 알기 위해 다음과 같이 실행할 수 있다.</p> <pre>\$ cubrid broker status options [args] options : [-b -q -s secs -t -f] args : broker_name</pre> <p>실행하면 다음과 같이 출력된다.</p> <pre>\$ cubrid broker status</pre>

```

% query_editor - cub_cas [28433,40820]
/home/CUBRID/log/broker/query_editor.access /home/CUBRID/
JOB QUEUE:0, AUTO_ADD_APPL_SERVER:ON,
SQL_LOG_MODE:ALL:100000
LONG_TRANSACTION_TIME:60, LONG_QUERY_TIME:60,
SESSION_TIMEOUT:300
KEEP_CONNECTION:AUTO, ACCESS_MODE:RW

```

ID	PID	QPS	LQS	PSIZE	STATUS
1	28434	0	0	50144	IDLE
2	28435	0	0	50144	IDLE
3	28436	0	0	50144	IDLE
4	28437	0	0	50140	IDLE
5	28438	0	0	50144	IDLE

```

% broker1 - cub_cas [28443,40821]
/home/CUBRID/log/broker/broker1.access /home/CUBRID/
JOB QUEUE:0, AUTO_ADD_APPL_SERVER:ON,
SQL_LOG_MODE:ALL:100000
LONG_TRANSACTION_TIME:60, LONG_QUERY_TIME:60,
SESSION_TIMEOUT:300
KEEP_CONNECTION:AUTO, ACCESS_MODE:RW

```

ID	PID	QPS	LQS	PSIZE	STATUS
1	28444	0	0	50144	IDLE
2	28445	0	0	50140	IDLE
3	28446	0	0	50144	IDLE
4	28447	0	0	50144	IDLE
5	28448	0	0	50144	IDLE

이 때 'STATUS'라는 항목이 있는데 자세한 의미는 다음과 같다.

※ 응용프로그램에서 jdbc를 통해 broker와 연결 했을 경우

IDLE - jdbc와 연결을 맺지 않은 상태

BUSY - jdbc와 연결을 맺은 상태이며, jdbc가 보낸 요청을
처리하고 있는 중인 상태

CLIENT_WAIT - jdbc와 연결을 맺은 상태이며, 트랜잭션이
진행중인 상태에서 jdbc가 request를 보내길
기다리고 있는 상태

CLOSE_WAIT - jdbc와 연결을 맺은 상태이며, 트랜잭션을
종료한 후 jdbc가 request를 보내길 기다리고 있는 상태

		(이 상태에서 broker가 restart되거나, 다른 jdbc에 할당되거나, jdbc와 연결이 끊어지는 경우에 jdbc는 reconnect를 시도하기 때문에 reconnect에 성공할 경우 에러가 발생하지 않음)
	관련파일/명령어	
	기타	
8	문제	Cubrid 복제 시 마스터 데이터베이스 서버 장애로 중지되었을 때
	이론과 배경	마스터 데이터베이스 서버가 장애로 중지된 경우에는 복제 서버나 복제 에이전트는 중단되지 않고 장애 발생 이전에 변경된 데이터를 슬레이 시스템에 복제한다. 그러나 슬레이브 데이터베이스 서버가 장애로 중지된 경우에는 복제 에이전트도 같이 중지된다.
	처리방법	<ol style="list-style-type: none"> 1. 슬레이브 데이터베이스 서버가 재시작한 후 수동으로 복제 에이전트를 재시작 해야 한다 2. 가급적 5개에서 10개 정도의 트랜잭션 보관(아카이브) 로그는 삭제하지 않는 것이 좋다. <ul style="list-style-type: none"> o 복제 서버 구동 <pre>\$ cubrid repl_server restart [마스터데이터베이스이름][복제서버포트] \$ cubrid repl_server start masterdb 5627 \$ cubrid repl_server stop masterdb ;중지</pre> o 복제 에이전트 구동 <pre>\$ cubrid repl_agent start [배포데이터베이스이름][배포데이터베이스dba암호] \$ cubrid repl_agent start distdb distdb_passwd \$ cubrid repl_agent stop distdb ; 중지</pre>
	관련파일/명령어	
	기타	
9	문제	CUBRID제거 절차
	이론과 배경	CUBRID 제거하고자 한다.
	처리방법	<ol style="list-style-type: none"> 1. Windows에서 CUBRID제거 <ul style="list-style-type: none"> o cmd창에서 “cubrid service stop” 명령어를 수행하거나, 시스템 tray에서 CBURID Service Tray를 종료시킨다. o 작업관리자에서 cub_ 로 시작하는 프로세스가 있는지 확인 하고 있으면 프로세스를 종료 시킨다.

		<ul style="list-style-type: none"> o 제어판에서 설치된 리스트 중 CUBRID 선택하여 제거한다. o 데이터베이스 및 백업이 존재할 수 있으므로 보전여부를 검토하고 CUBRID 홈 경로의 모든 파일 및 폴더를 삭제한다. <p>2. Linux에서 CUBRID 제거</p> <ul style="list-style-type: none"> o “cubrid service stop” 명령을 수행한다. o 프로세스들이 존재하는지 “ps -ef grep cub_*” 명령어를 수행하여 확인 한다. 만약 존재할 경우 kill명령어를 이용하여 종료 시킨다. o 데이터베이스 및 백업이 존재할 수 있으므로 보전여부를 검토하여 rm명령어로 CUBRID 홈 디렉토리를 삭제 한다. o 환경변수(.bash_profile)에서 CUBRID 환경 설정파일 등록부분을 삭제 및 주석처리 한다. o CUBRID 환경설정 파일 .cubrid.sh를 삭제한다. o 자동 구동/종료 설정(init.d)이 되어 있을 경우 해당 설정을 제거한다. o cron을 통해 백업 및 기타 주기적은 CUBRID 관련 동작을 등록하였을 경우 crontab -e 명령을 통해 해당 내용을 삭제한다.
	관련파일/명령어	
	기타	