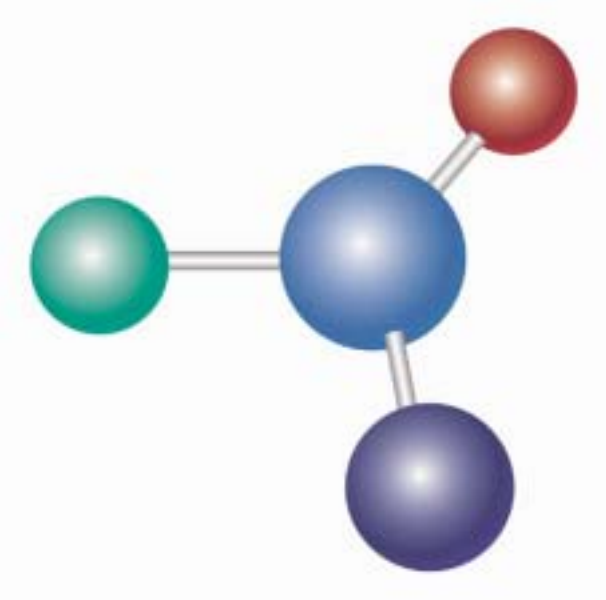
UniSQL 설치 가이드



UniSQL

Contents

1

Unix 용 설치 방법

이 장은 호스트의 OS 가 Unix 인 경우 UniSQL/X, UniCAS, UniTool 을 함께 설치하는 방법을 기술한다. 서버 패키지들 (UniSQL/X, UniCAS, UniTool 의 Easy-Manager Server)은 Unix 머신에 설치되지만, UniTool 의 클라이언트 패키지들(Easy-Manager, Visual-SQL)은 Window 머신에 설치되어야 하므로 “서버 패키지 설치” 와 “클라이언트 패키지 설치” 로 나누어 설명한다.

서버 패키지 설치

Step	*Action*
Step 1	설치하고자 하는 사용자 계정으로 로그인 한다. 본 예제에서는 “unisql” 이라는 사용자 계정과 그룹을 만들고, “unisql” 사용자의 홈 디렉토리(본 예에서는 “/usr/unisql” 사용)를 생성한 후 “unisql” 사용자 계정으로 로그인 한다.

Step 2	제품 CD 로부터 홈디렉토리에 **UniSQL/X** 를 아래와 같이 추출한다. 실제 제품 CD 에 있는 UniSQL/X 제품 파일명은 UniSQL<version>_<platform>-<os>.tar.Z 로 이루어져 있다. 즉 사용하는 머신이 SUN-SPARC 플랫폼이고, OS 가 Solaris7 일 경우 UniSQL/X 버전 6.0 을 설치하고자 한다면 “UniSQL6.0_SUN-SPARC-Solaris7.tar.Z” 제품 파일이 필요하다. 제품 CD 에 UniSQL 로 시작하는 파일이 하나라면 아래와 같이 “UniSQL*.tar.Z” 라 명세할 수 있다.
	
	`% cd $HOME`
	`% cat /cdrom/UniSQL*.tar.Z | uncompress | tar xvf -`
	
	추출이 완료 되면 홈디렉토리 밑에 “unisql” 이라는 디렉토리가 생긴다. (본 예에서는 “/usr/unisql/unisql” 디렉토리가 생기게 됨).

Step 3	제품 CD 로부터 홈디렉토리에 **UniCAS** 를 아래와 같이 추출한다. 실제 제품 CD 에 있는 UniCAS 제품 파일명은 UniCAS<version>_<platform>-<os>_TCL<version>.tar.Z 로 이루어져 있다. 즉 사용하는 머신이 SUN-SPARC 플랫폼이고, OS 가 Solaris7 이고, TCL 8.0 을 사용하고자 하는 경우 UniCAS 버전 4.6.2 를 설치하고자 한다면 “UniCAS4.6.2_SUN-SPARC-Solaris6_TCL8.0.tar.Z” 제품 파일이 필요하다. UniSQL/X 버전에 따라 사용할 수 있는 UniCAS 버전이 다르므로 Appendix 의 “제품 연관 관계”를 참조하여 설치하는 UniSQL/X 와 연동 가능한 UniCAS 버전을 선택해야 한다. 제품 CD 에 UniCAS 로 시작하는 파일이 하나라면 아래와 같이 “UniCAS*.tar.Z” 라 명세할 수 있다.
	
	`% cd $HOME`
	`% cat /cdrom/UniCAS*.tar.Z | uncompress | tar xvf -`
	
	추출이 완료 되면 홈디렉토리 밑에 “unicas” 라는 디렉토리가 생긴다. (본 예에서는 “/usr/unisql/unicas” 디렉토리가 생기게 됨).

Step 4	제품 CD 로부터 홈디렉토리에 **UniTool** 중 **Easy-Manager Server** 를 아래

	와 같이 추출한다. 실제 제품 CD 에 있는 Easy-Manager Server 제품 파일명은 EMGR<version>_<platform>-<os>.tar.Z 로 이루어져 있다. 즉 사용하는 머신이 SUN-SPARC 플랫폼이고, OS 가 Solaris7 일 경우 Easy-Manager Server 버전 1.0 을 설치하고자 한다면 “EMGR1.0_SUN-SPARC-Solaris7.tar.Z” 제품 파일이 필요하다. 제품 CD 에 EMGR 로 시작하는 파일이 하나라면 아래와 같이 “EMGR*.tar.Z” 라 명세할 수 있다.
	
	`% cd $HOME`
	`% cat /cdrom/EMGR*.tar.Z | uncompress | tar xvf -`
	
	추출이 완료 되면 홈디렉토리 밑에 “emgr” 이라는 디렉토리가 생긴다. (본 예에서는 “/usr/unisql/emgr” 디렉토리가 생기게 됨).

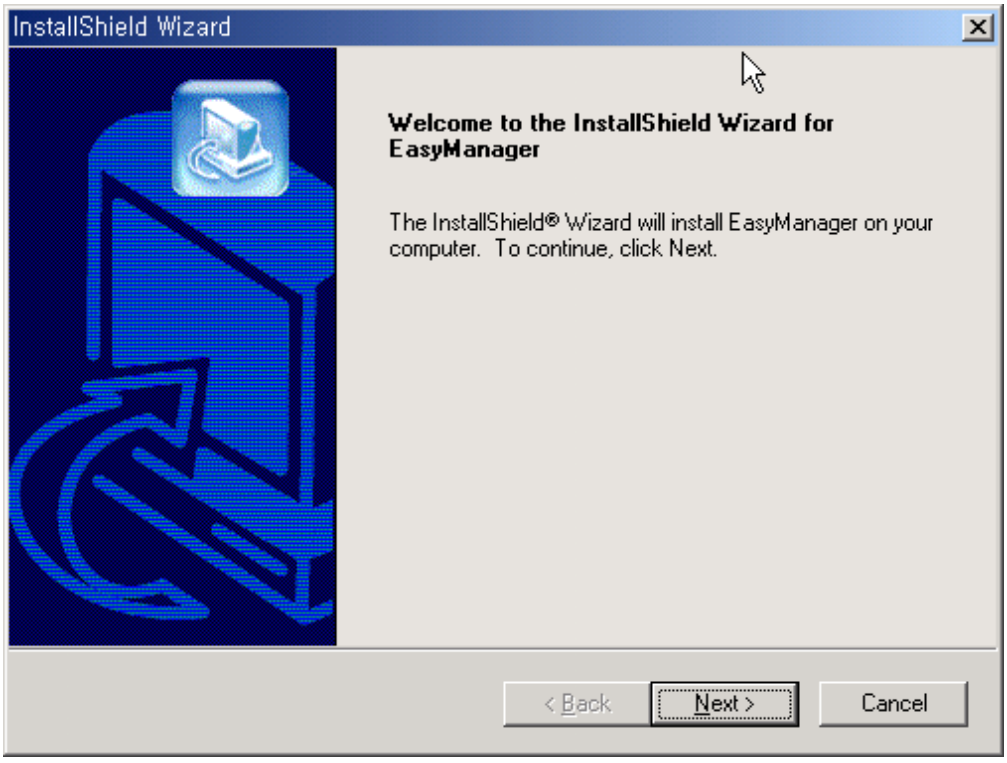
Step 5	아래의 환경 변수를 사용하는 shell 의 종류에 따라 설정한다.
	■ 만일 “csh" 을 사용한다면, “.cshrc” 파일에 아래의 내용을 추가한다.
	
	`setenv UNISQLX $HOME/unisql`
	`setenv UNISQLX_MODE client`
	`setenv UNISQLX_DATABASES $HOME/unisql`
	`setenv UNISQLX_LANG Ko_KR`
	`setenv UNICAS $HOME/unicas`
	`setenv UNITOOL_EMGR $HOME/emgr`
	
	`setenv LD_LIBRARY_PATH $UNISQLX/lib:$UNICAS/lib:$UNITOOL_EMGR/bin:/usr/lib`
	`setenv SHLIB_PATH $LD_LIBRARY_PATH`
	`setenv LIBPATH $LD_LIBRARY_PATH`
	`setenv PATH $UNISQLX/bin:$UNISQLX/utilities:$UNICAS/bin:$UNITOOL_EMGR/bin:$PATH`
	
	■ 만일 “sh" 을 사용한다면, “.profile” 파일에 아래의 내용을 추가한다.
	
	`UNISQLX=$HOME/unisql`
	`export UNISQLX`
	`UNISQLX_MODE=client`
	`export UNISQLX_MODE`
	`UNISQLX_DATABASES=$HOME/unisql`
	`export UNISQLX_DATABASES`
	`UNISQLX_LANG=Ko_KR`
	`export UNISQLX_LANG`
	`UNICAS=$HOME/unicas`
	`export UNICAS`
	`UNITOOL_EMGR=$HOME/emgr`
	`export UNITOOL_EMGR`

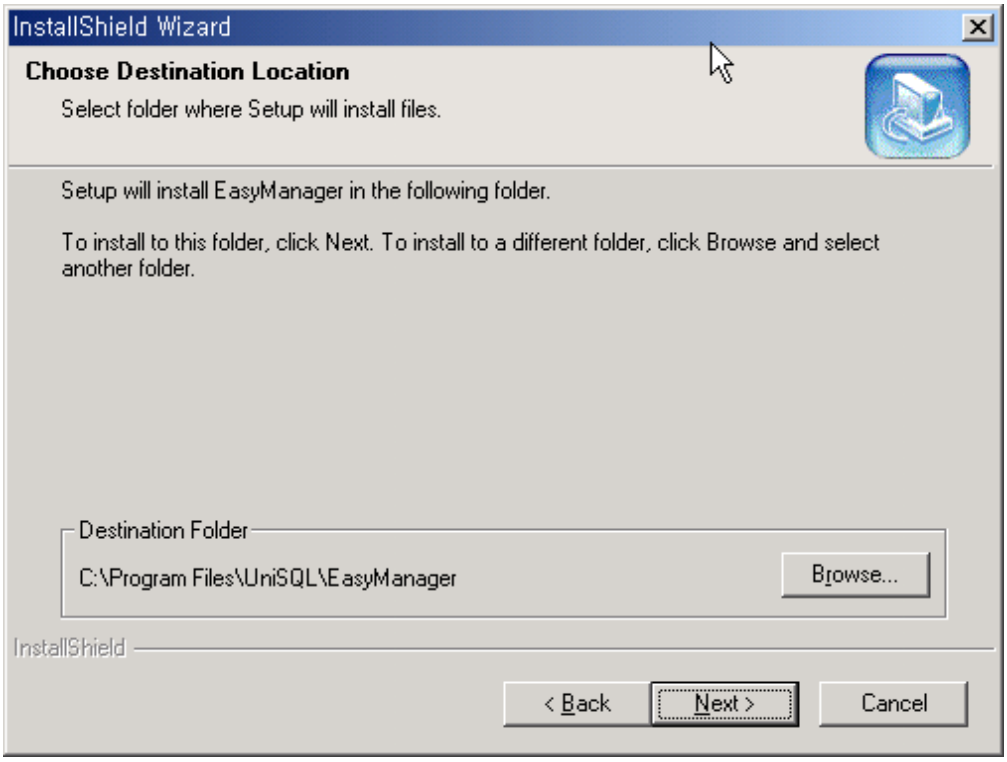
	LD_LIBRARY_PATH=$UNISQLX/lib:$UNICAS/lib:$UNITOOL_EMGR/bin:/usr/lib export LD_LIBRARY_PATH SHLIB_PATH=$LD_LIBRARY_PATH export SHLIB_PATH LIBPATH=$LD_LIBRARY_PATH export LIBPATH PATH=$UNISQLX/bin:$UNISQLX/utilities:$UNICAS/bin:$UNITOOL_EMGR/bin:$PATH export PATH
Step 6	앞에서 설정한 환경 변수 파일의 내용이 반영되도록 “unisql” 사용자 계정으로 다시 로그인 한다.
Step 7	“$UNISQLX/admin” 디렉토리에 “hostkeys” 파일을 만들어 호스트 키 값을 삽입한다. 이 때 key 값의 앞 뒤에 공백이나 다른 문자가 들어가면 안된다 # vi $UNISQLX/admin/hostkeys
Step 8	**UniSQL/X** 가 올바로 설치되었는지를 확인하기 위해서는 데이터베이스를 생성해 보면 알 수 있다. 개별 데이터베이스가 “$HOME/db” 라는 디렉토리에 들어가도록 하기 위해서 먼저 mkdir 시스템 명령어를 사용해 디렉토리를 만들고 cd 시스템 명령어로 “$HOME/db” 디렉토리로 들어가 createdb 명령을 수행하면 현재 디렉토리에 데이터베이스 파일이 생성된다. % mkdir $HOME/db % cd $HOME/db 설치가 올바로 되었다면 createdb 시 아래와 같은 메시지를 얻게 된다. % createdb testdb Creating database with 1000 pages. *** UniSQL/X Standalone Release 6.0.0 *** Generated Jan 17 2003 at 11:48:22 만일 “$UNISQLX/admin/hostkeys” 파일에 입력한 hostkey 값이 잘못 입력되었거나 evaluation 기간이 지난 경우는 아래와 같은 메시지를 얻게 된다. % createdb testdb Creating database with 1000 pages. Couldn't create database.

	Host "BIG" is not authorized to run database.
Step 9	**UniCAS** 가 올바로 설치되었는지를 확인하기 위해서는 uc 명령어로 UniCAS 를 구동해 보면 알 수 있다. % cd $UNICAS/bin % uc start OK:start
Step 10	**UniTool** 의 **Easy-Manager Server** 가 올바로 설치되었는지를 확인하기 위해서는 ems 명령어로 Easy-Manager Server 를 구동해 보면 알 수 있다. % cd $UNITOOL_EMGR/bin % ems start . . . Checking /usr/unisql/databases.txt ... found. OK. Checking /usr/unisql/emgr/conf/emgr.pass ... found. OK. Checking /usr/unisql/emgr/conf/emgr_db.pass ... found. OK. Checking /usr/unisql/emgr/logs ... OK Checking /usr/unisql/emgr/conf ... OK Checking /usr/unisql/emgr/tmp ... OK . . .

클라이언트 패키지 설치

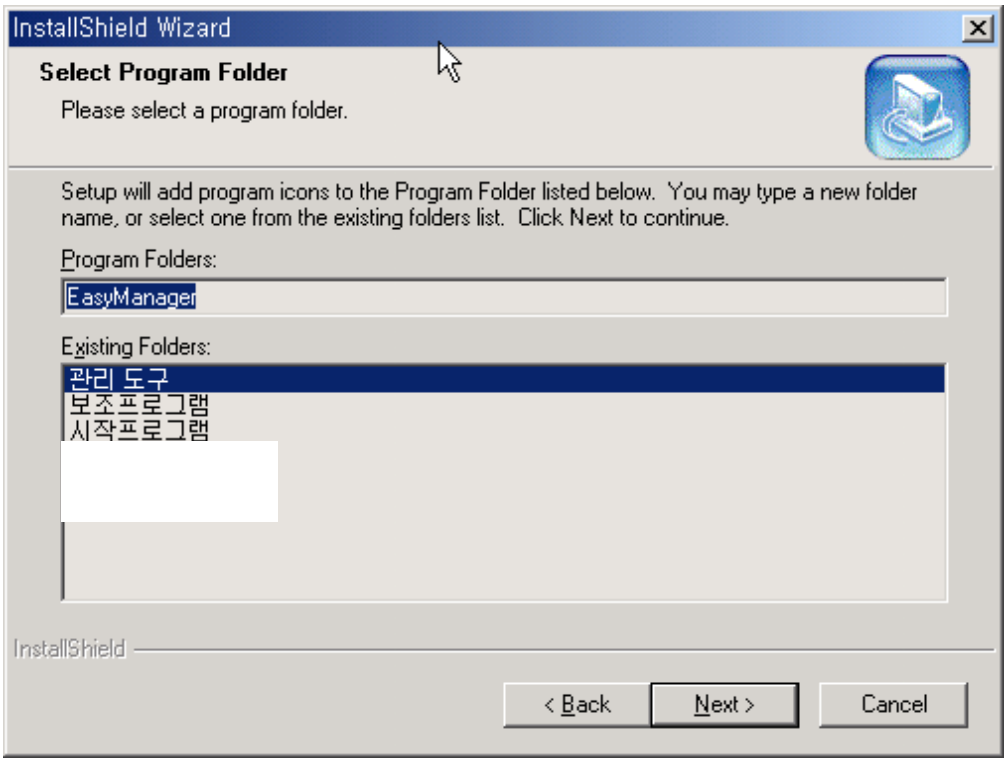
Step 1 ~ 5 까지는 Easy-Manager Client 설치를, Step 6 ~ 10 까지는 Visual-SQL 설치 과정을 보여준다. 마지막 Step 10 의 화면을 통해 최종적으로 Easy-Manager 와 Visual-SQL 이 성공적으로 설치되었음을 확인할 수 있다.

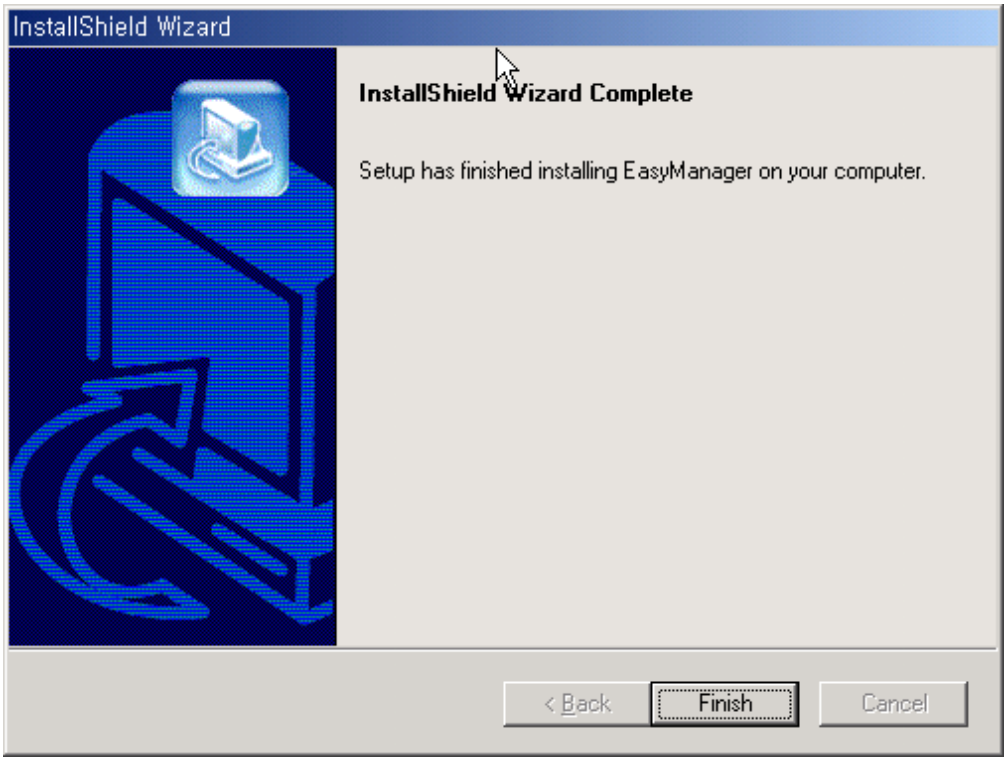
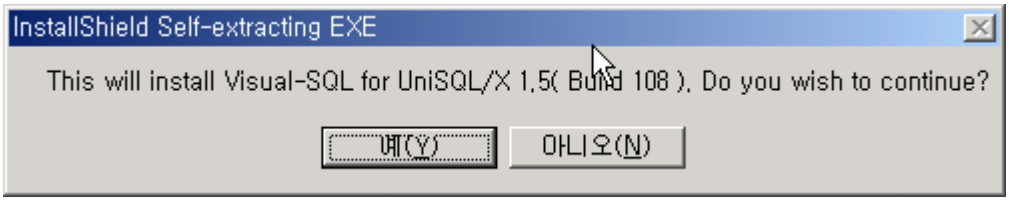
Step	***Action***
Step 1	제품 CD 로부터 EasyManager<version>.zip 파일의 압출을 풀어 EasyManager<version>.exe 을 추출한 뒤 EasyManager<version>.exe 를 실행한다. (예를 들어 EasyManager100.zip 파일에서 EasyManager100.exe 파일을 추출한 뒤 실행한다.)
Step 2	아래의 화면은 설치를 시작할 것인지를 묻는 것으로 설치하려면 “Next” 버튼을 클릭한다. 
Step 3	아래의 화면은 Easy-Manager 설치 디렉토리를 지정하는 화면으로 지정한 후 “Next” 버튼을 클릭한다. (디폴트 디렉토리는 “C:\Program Files\UniSQL\EasyManager”)

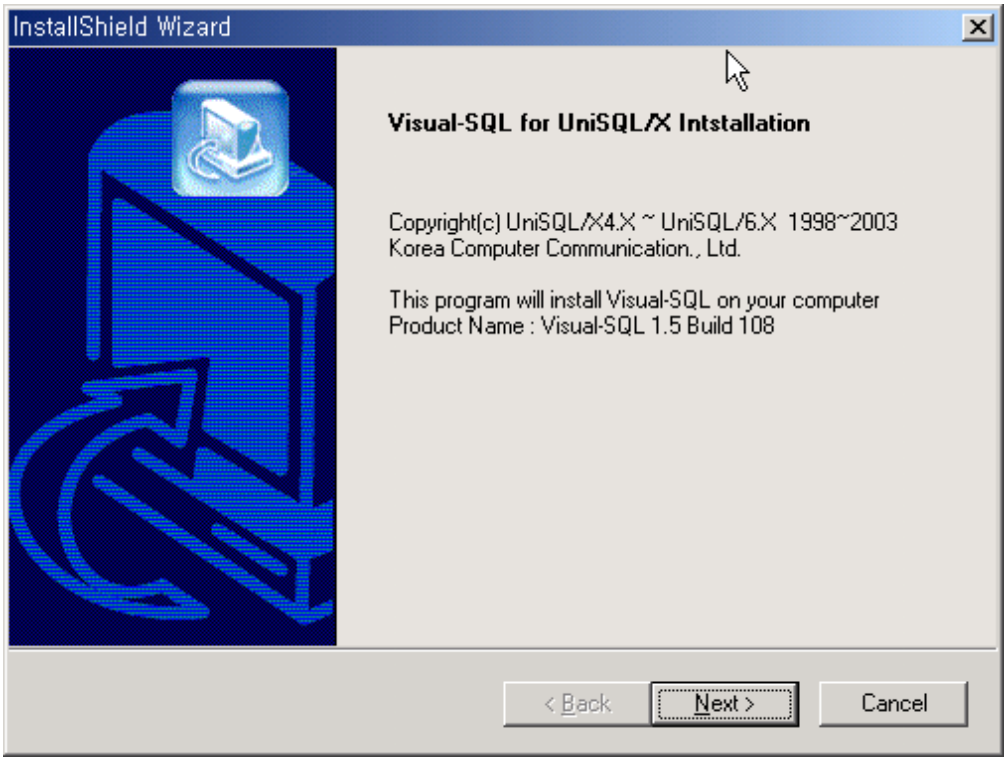


Step 4

아래의 화면은 Easy-Manager 프로그램 폴더를 지정하는 화면으로 폴더명을 지정한 후 “Next” 버튼을 클릭한다. (디폴트 폴더는 “EasyManager”)

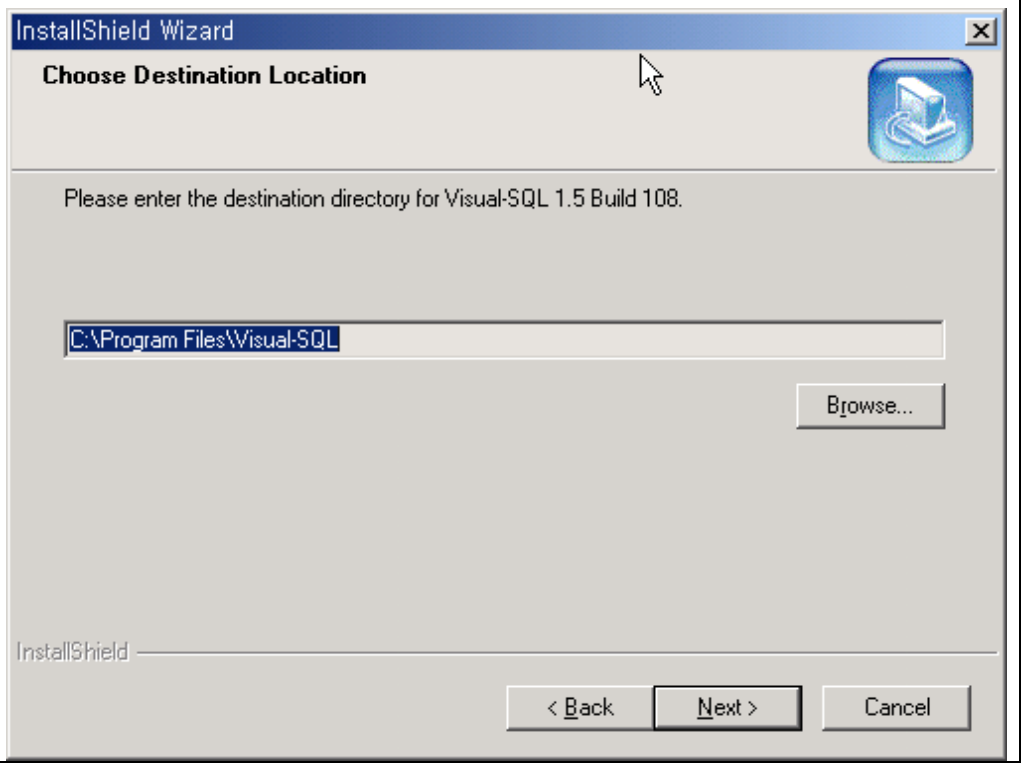


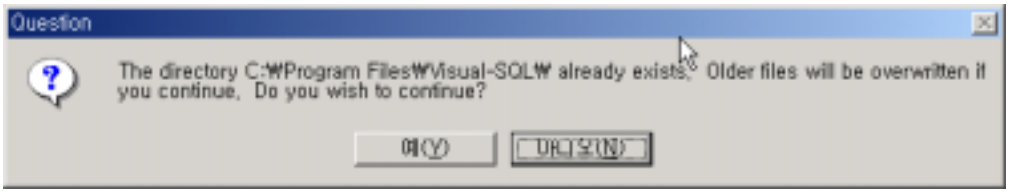
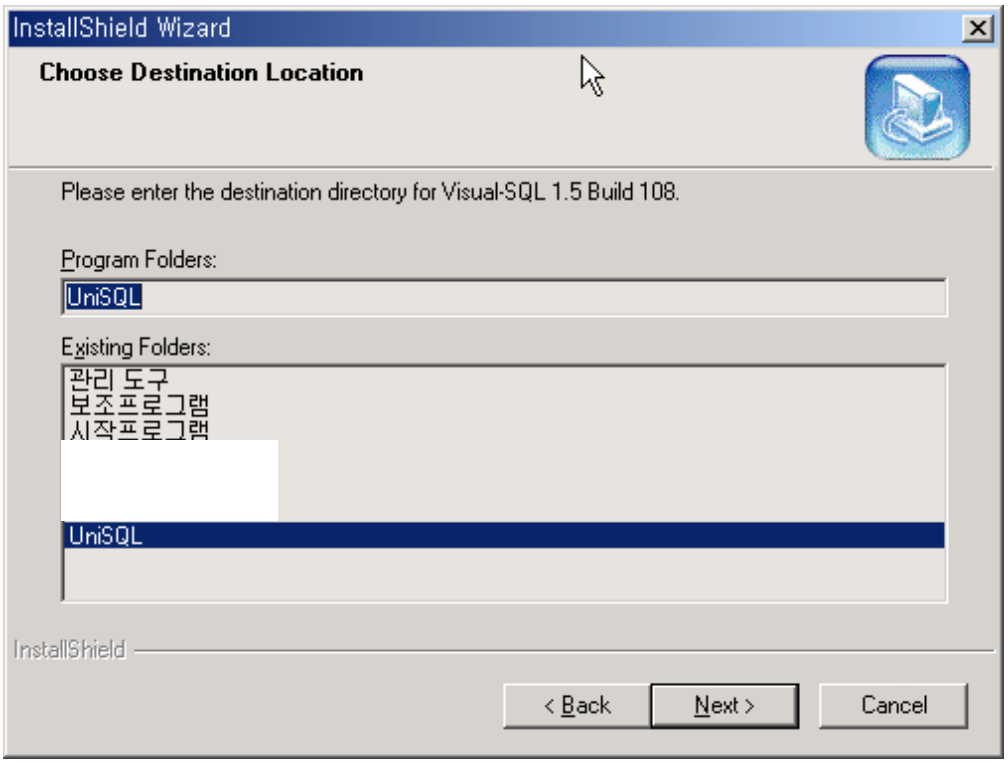
Step 5	아래의 화면이 나오면 Easy-Manager 는 정상적으로 설치가 된 것이므로 Finish 버튼을 클릭하여 Easy-Manager Client 설치 작업을 끝낸다. 
Step 6	제품 CD 로부터 VisualSQL<version>.zip 파일의 압출을 풀어 VisualSQL<version>.exe 을 추출한 뒤 VisualSQL<version>.exe 를 실행한다. (예를 들어 VisualSQL15108.zip 파일에서 VisualSQL15108.exe 파일을 추출한 뒤 실행한다.)
Step 7	아래의 화면은 설치를 시작할 것인지를 묻는 것으로 설치하려면 “예(Y)” 버튼을 클릭한다. 
Step 8	아래의 화면은 저작권과 버전 정보를 보여주는 것으로 읽은 후 “Next” 버튼을 클릭한다.

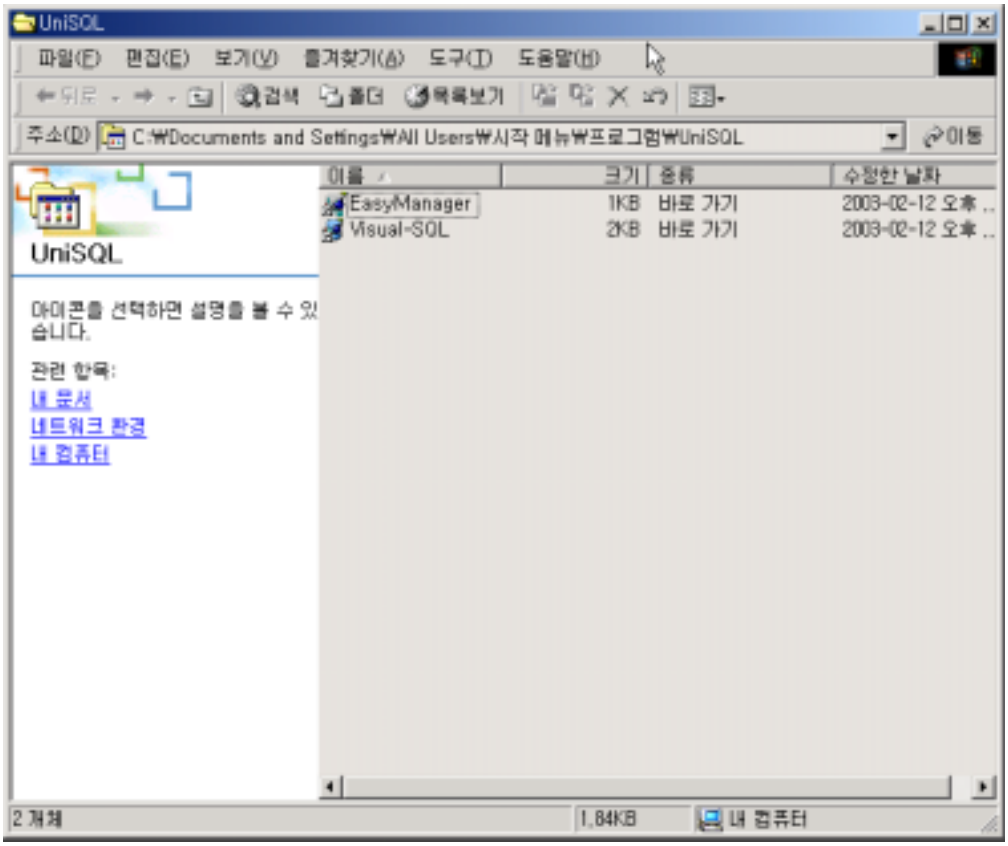


Step 8

아래의 화면은 Visual-SQL 설치 디렉토리를 지정하는 화면으로 지정한 후 “Next” 버튼을 클릭한다. (디폴트 디렉토리는 “C:\Program Files\Visual-SQL”)



Step 8	만일 Visual-SQL 을 재설치 하고 있다면 아래의 윈도우가 팝업될 수 있다. 이 때 “예(Y)” 를 클릭한다. 
Step 9	아래의 화면은 Visual-SQL 프로그램 폴더를 지정하는 화면으로 지정한 후 “Next” 버튼을 클릭한다. (디폴트 폴더는 “UniSQL”) 
Step 10	Visual-SQL 설치가 완료되면 아래의 윈도우가 팝업된다. 이 윈도우는 EasyManager 와 Visual-SQL 가 시작메뉴->프로그램->UniSQL 폴더로 접근될 수 있음을 나타낸다.

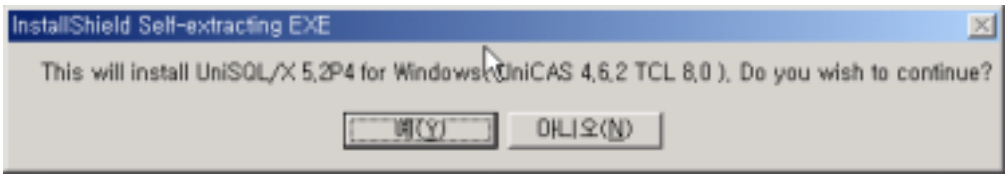

UniSQL
파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)
검색 폴더 목록보기
주소(D) C:₩Documents and Settings₩All Users₩시작 메뉴₩프로그램₩UniSQL
이동
UniSQL
아이콘을 선택하면 설명을 볼 수 있습니다.
관련 항목:
내 문서
네트워크 환경
내 컴퓨터
이름
크기
종류
수정한 날짜
EasyManager
1KB
바로 가기
2003-02-12 오후 ..
Visual-SQL
2KB
바로 가기
2003-02-12 오후 ..
2 개체
1.84KB
내 컴퓨터

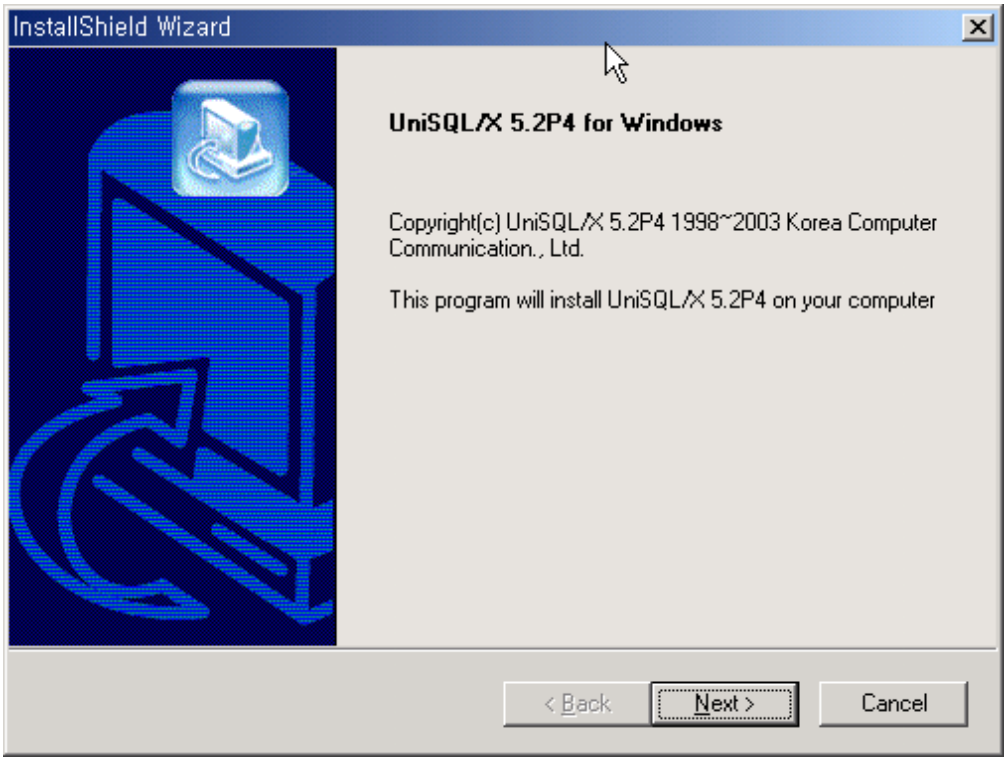
2

Windows 용 설치 방법

이 장은 호스트의 OS 가 Windows 인 경우 UniSQL/X, UniCAS, UniTool 을 함께 설치하는 방법을 기술한다.

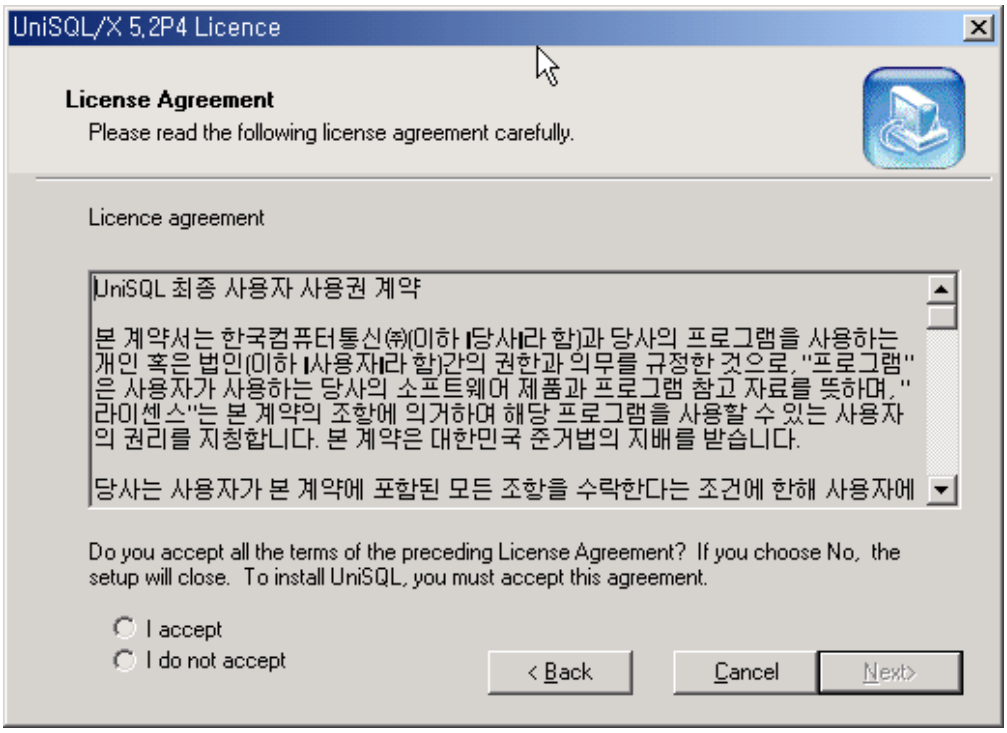
서버/클라이언트 패키지 설치

Step	*Action*
Step 1	제품 CD 로부터 UniSQL<version>_<platform>(UniCAS<version>-TCL<version>).zip 파일의 압축을 풀어 UniSQL<version>_<platform>(UniCAS<version>-TCL<version>).exe 을 추출한 뒤 실행한다. (예를 들어 UniSQL5.2P4_INTEL-x86-W2K(UniCAS4.6.2-TCL8.0).zip 파일에서 UniSQL5.2P4_INTEL-x86-W2K(UniCAS4.6.2-TCL8.0).exe 파일을 추출한 뒤 실행한다.) 이 파일을 실행하면 UniSQL/X, UniCAS, UniTool 모두를 한번에 Windows 에 설치할 수 있다. (c:\UniSQL 디렉토리 밑에 설치됨)
Step 2	아래의 화면은 설치를 시작할 것인지를 묻는 것으로 설치하려면 “예(Y)” 버튼을 클릭한다. 
Step 3	아래의 화면은 저작권과 버전 정보를 보여주는 화면으로 “Next” 버튼을 클릭한다.

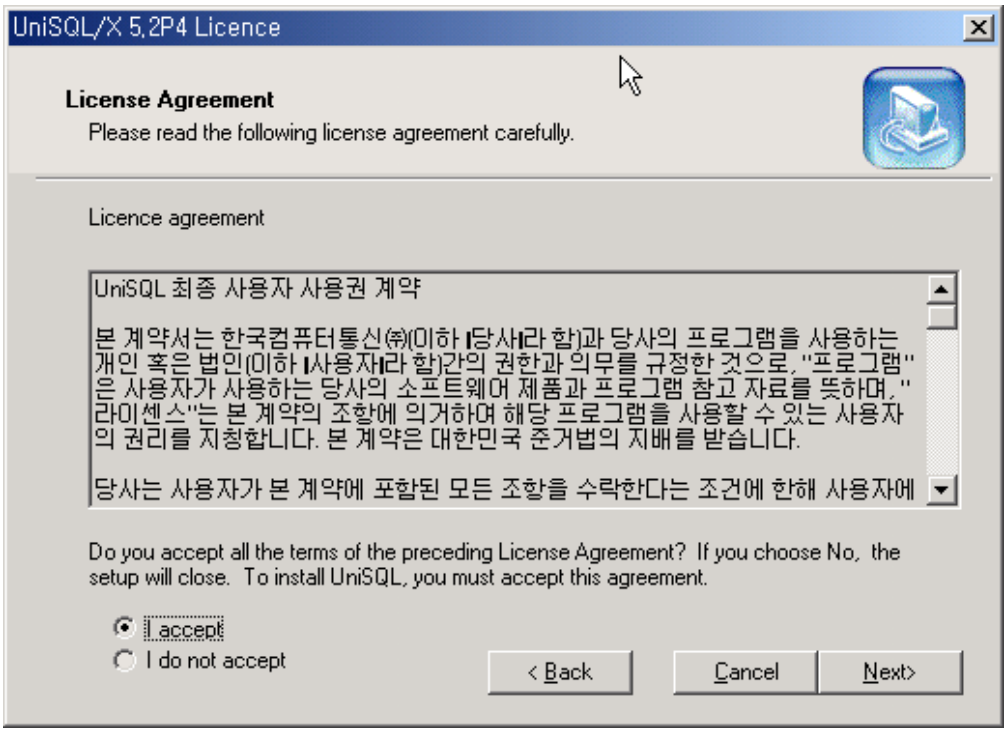
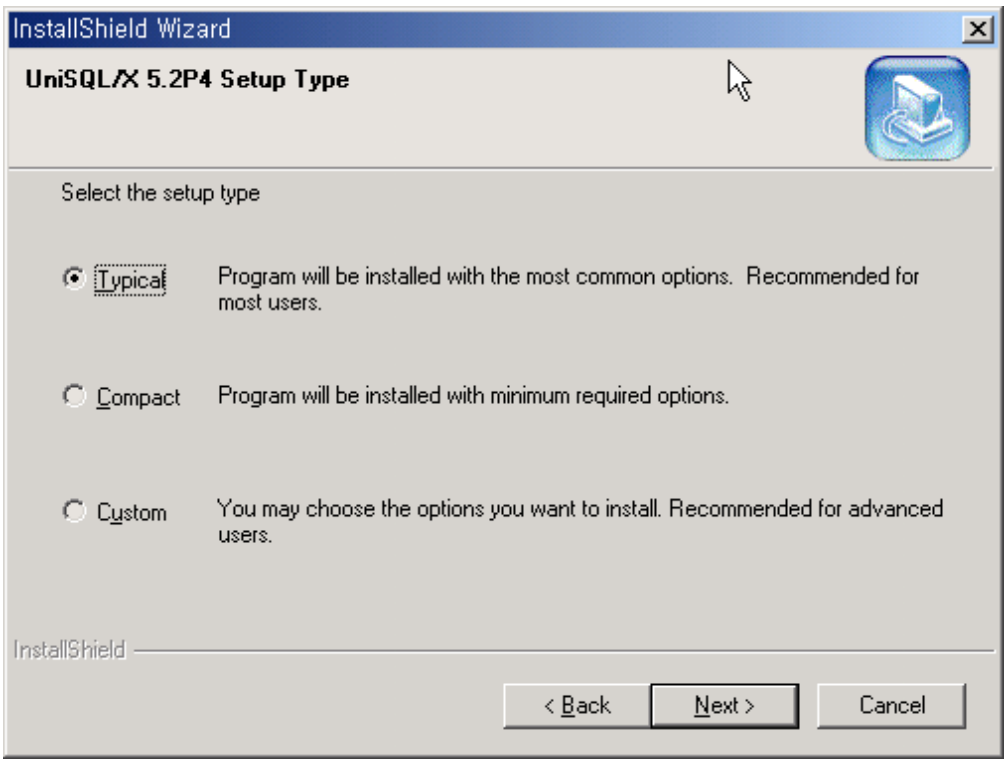


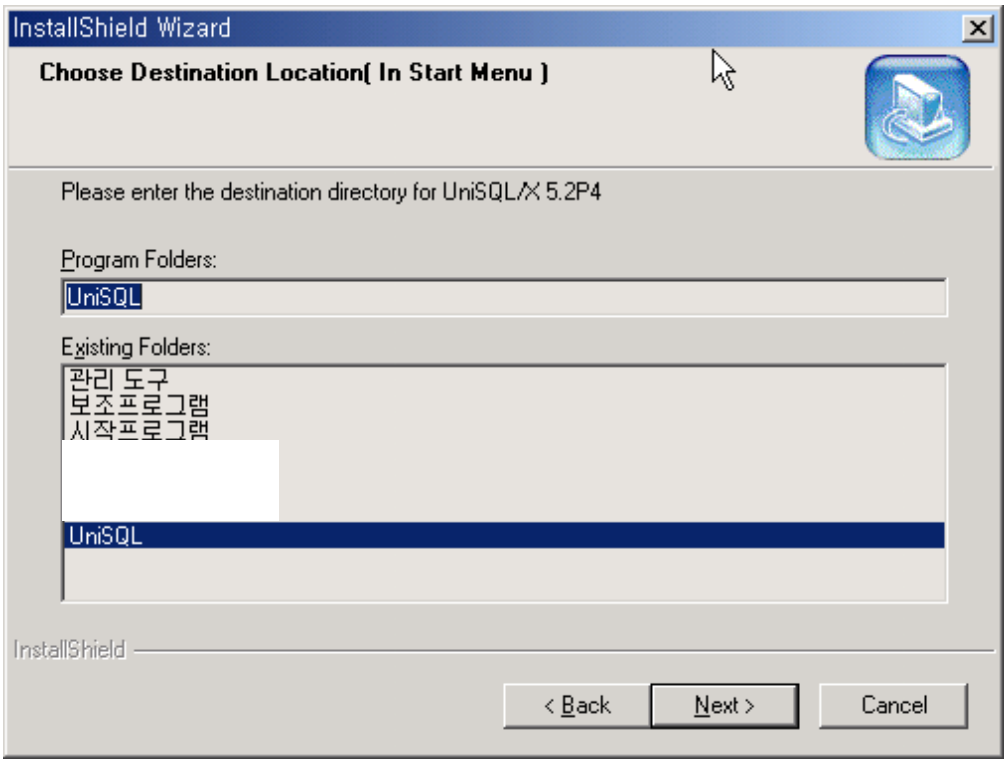
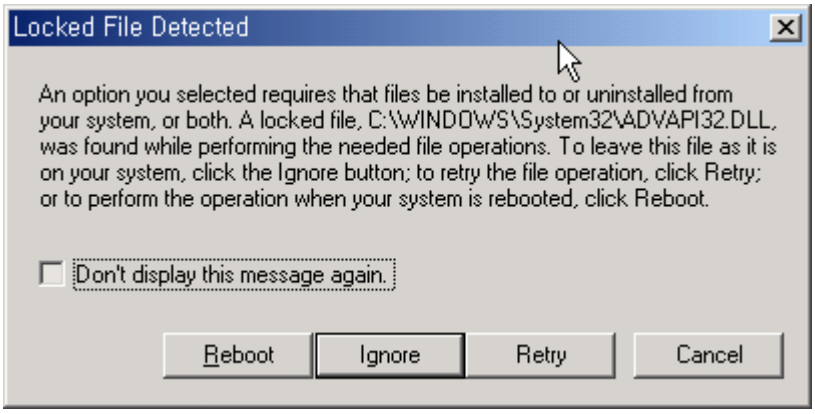
Step 4

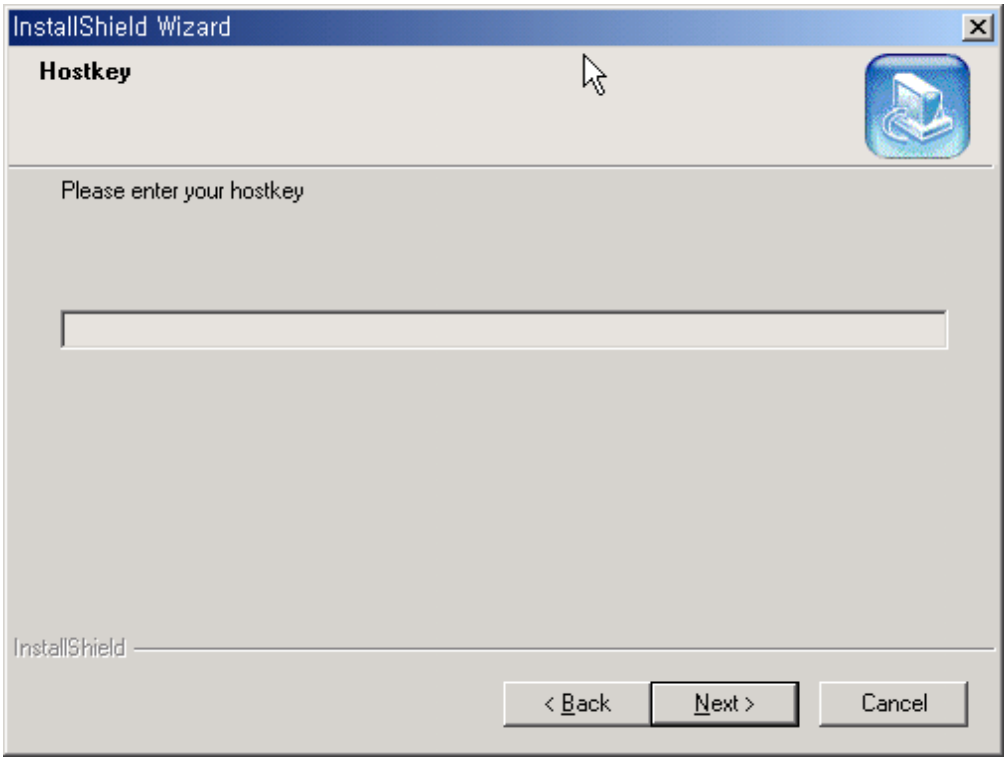
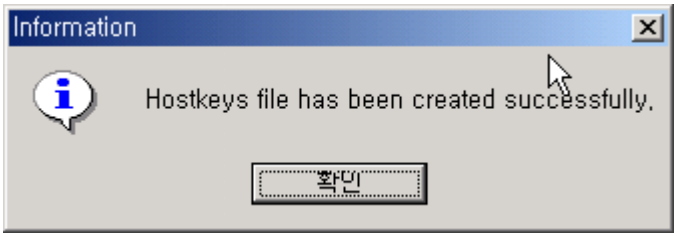
아래의 화면은 Licence Agreement 를 보여주는 화면으로 진행하기 위해서는 반드시 "I accept" 를 선택해야 한다.

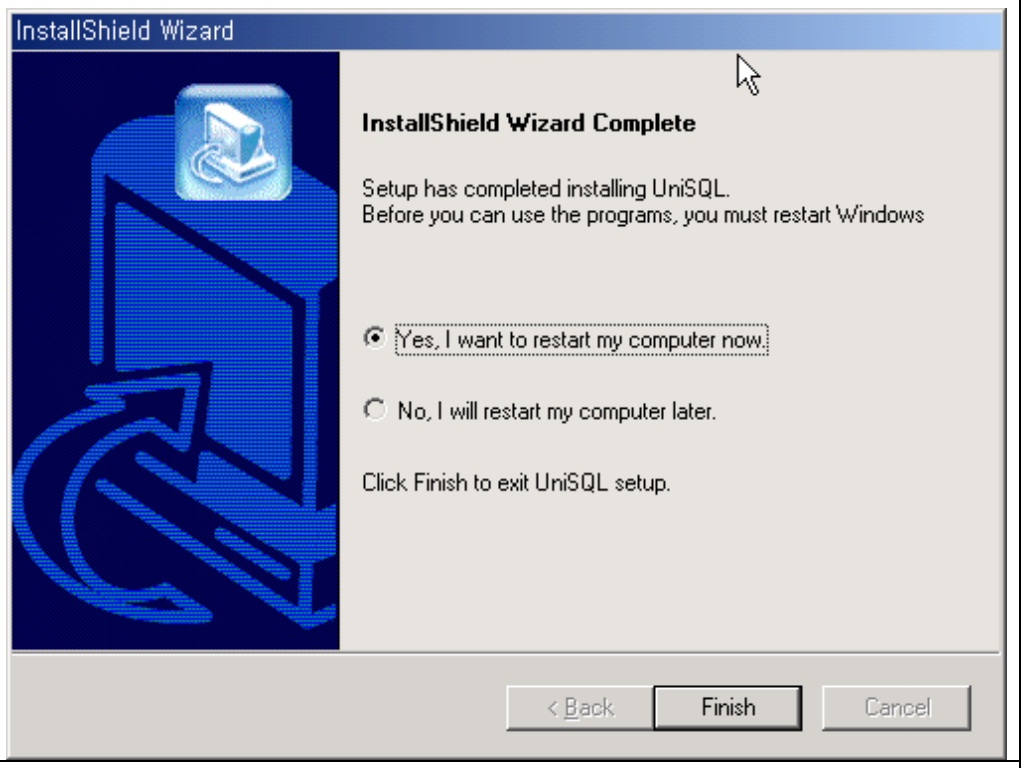


"I accept" 를 선택하면 아래의 화면과 같이 "Next" 버튼이 활성화 된다.

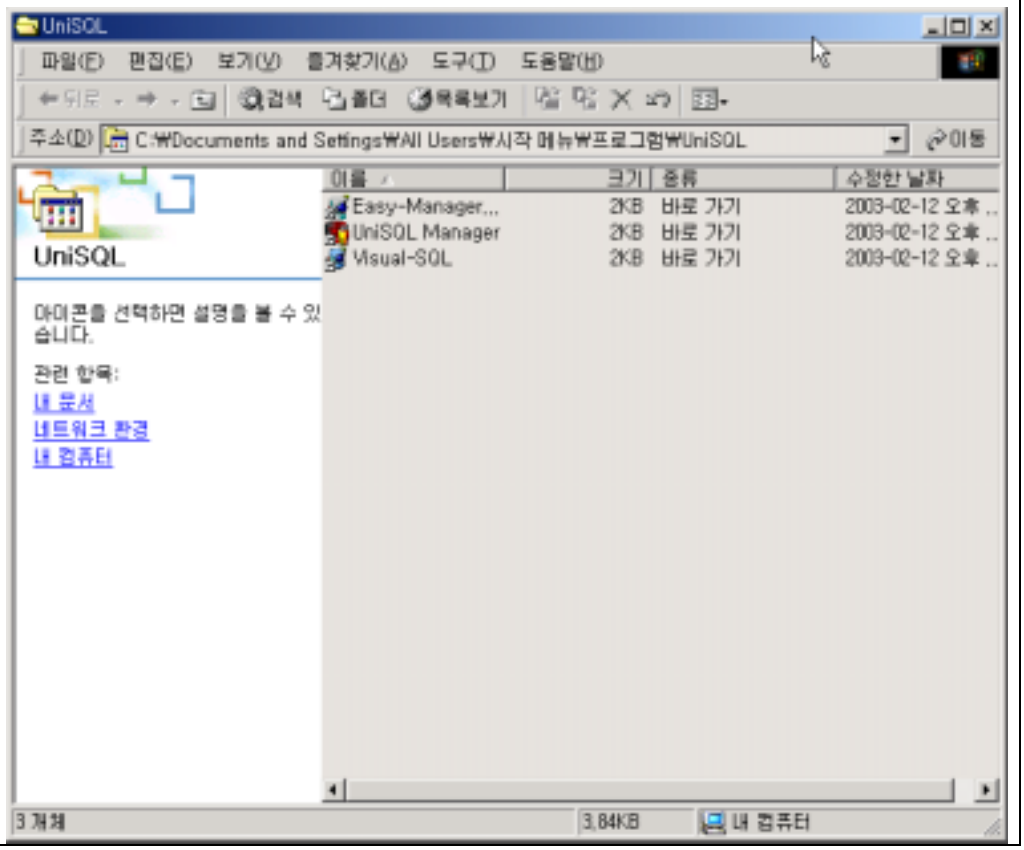
	이 때 “Next” 버튼을 클릭하여 다음으로 진행한다. 
Step 5	아래의 화면은 설치 유형을 선택하는 화면으로 선택 후 “Next” 버튼을 클릭한다. (디폴트 선택인 “Typical” 권장) 

Step 6	아래의 화면은 프로그램 폴더를 지정하는 화면으로 지정한 후 “Next” 버튼을 클릭한다. (디폴트 폴더는 “UniSQL”) 
Step 7	만일 설치할 DLL 들이 이미 존재한다면 아래의 윈도우를 만날 수 있다. “Don’t display this message again.” 에 체크한 뒤 “Ignore” 버튼을 클릭하여 다음으로 진행한다. 
Step 8	아래의 화면은 Hostkey 를 넣기 위한 화면으로 키 값을 넣은 뒤 “Next” 버튼을 클릭한다.

	
Step 9	키 값이 맞으면 아래의 화면이 나타난다. “확인” 버튼을 클릭한다. 
Step 10	아래의 화면이 나오면 일차적인 설치를 완료된 상태이지만, 컴퓨터를 재부팅하여야 필요한 나머지 설정을 마칠 수 있으므로 “Yes, I want to restart my computer now.” 가 선택된 채로 “Finish” 버튼을 클릭한다.



Step 11 재부팅을 하고 나면 아래의 화면을 볼 수 있는데, 이렇게 되면 설치가 모두 완료된 상태이다.



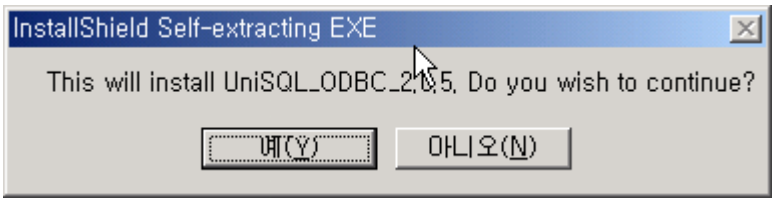
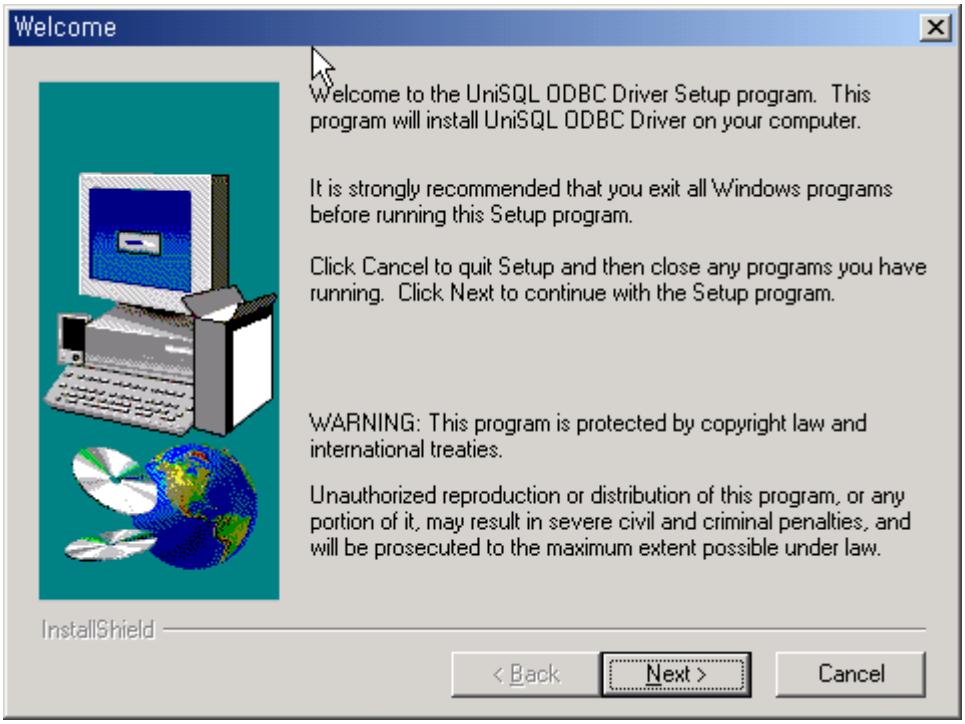
Step 12	위의 화면에서 Easy-Manager 나 Visual-SQL 을 클릭하면 화면에 해당 화면이 나타나지만, UniSQL Manager 를 클릭하면 아래와 같이 하단에 있는 트레이에 UniSQL Manager 아이콘이 나타난다. 
Step 13	트레이에 있는 UniSQL Manager 아이콘 상에서 오른쪽 버튼을 클릭하면 아래의 메뉴가 팝업된다. 이를 통해 UniSQL 데이터베이스, UniCAS, Easy-Manager Server 를 start/stop 할 수 있으며, UniTool 인 Easy-Manager Client, Visual-SQL 도 구동할 수 있다. 

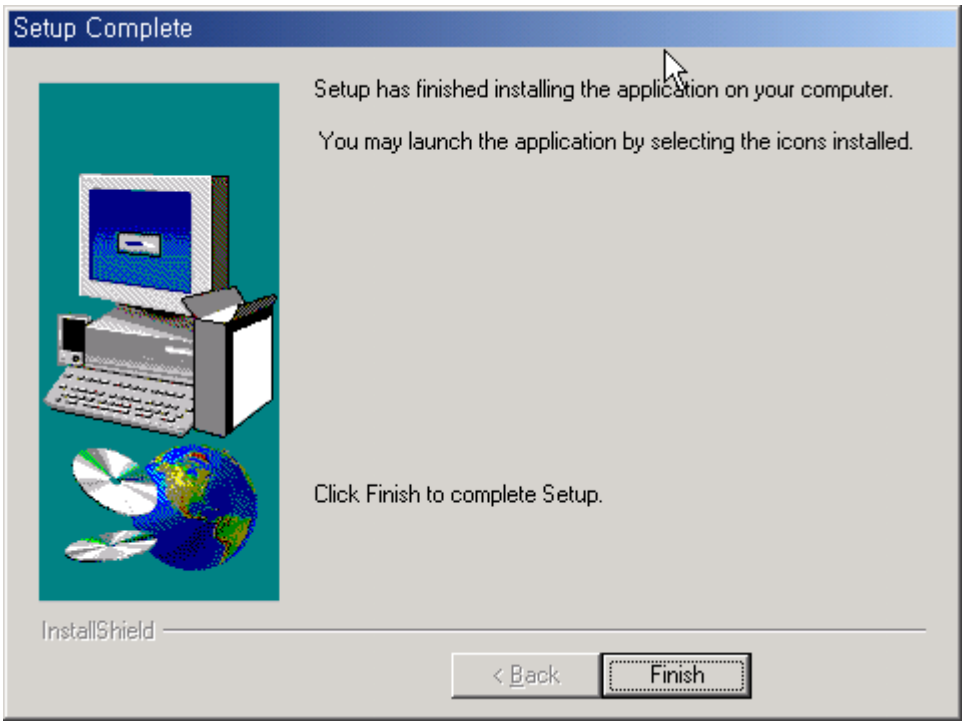
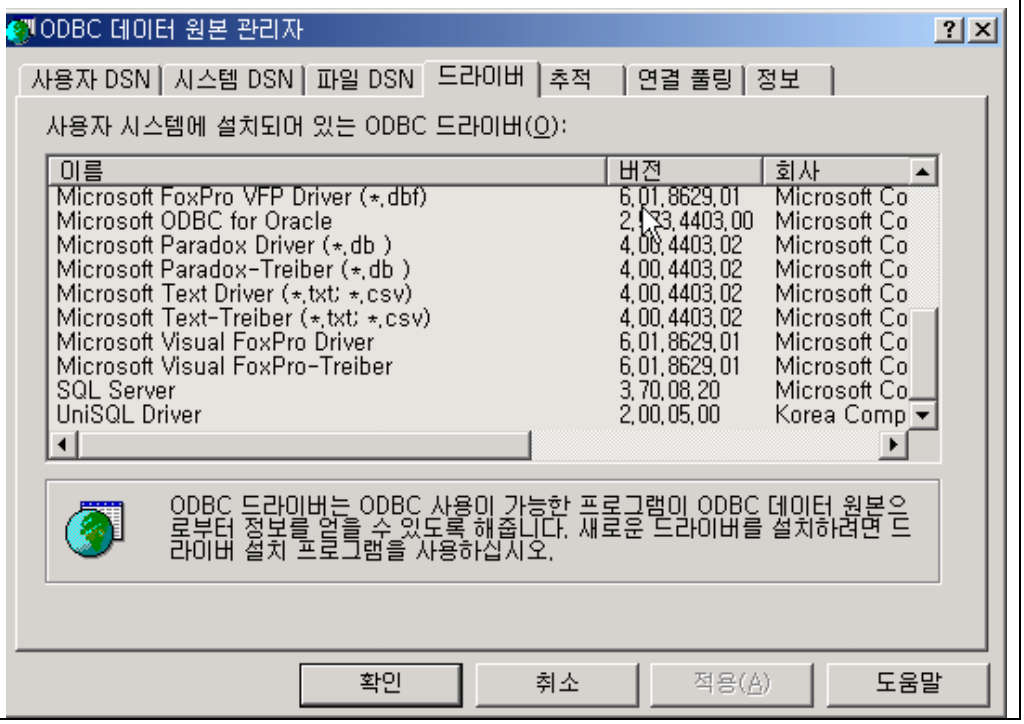
3

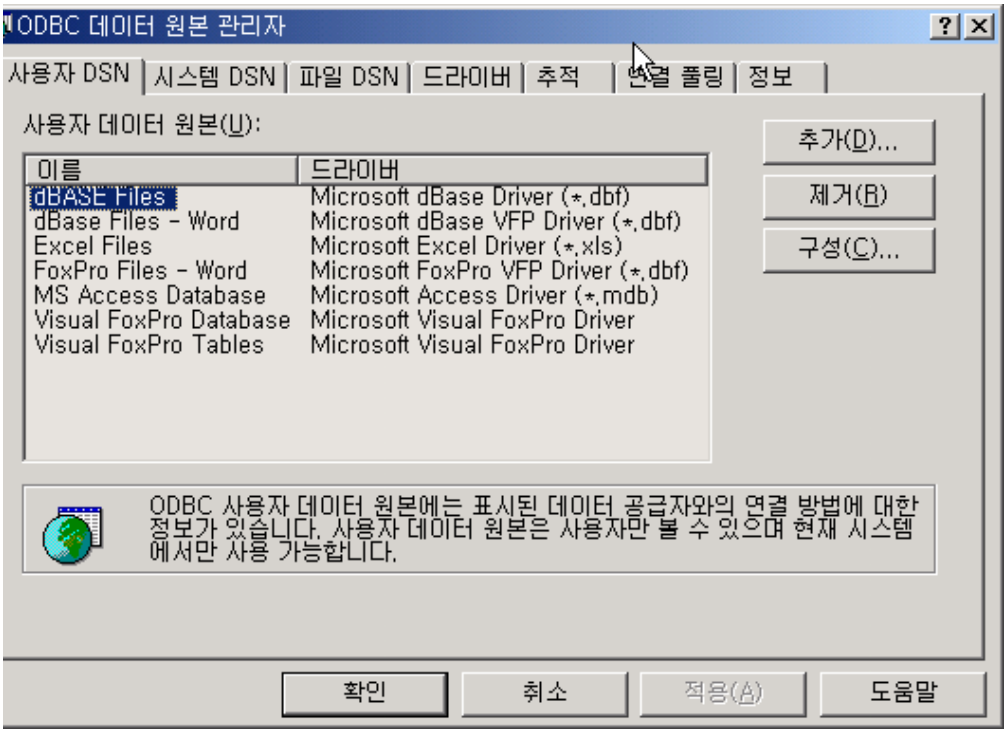
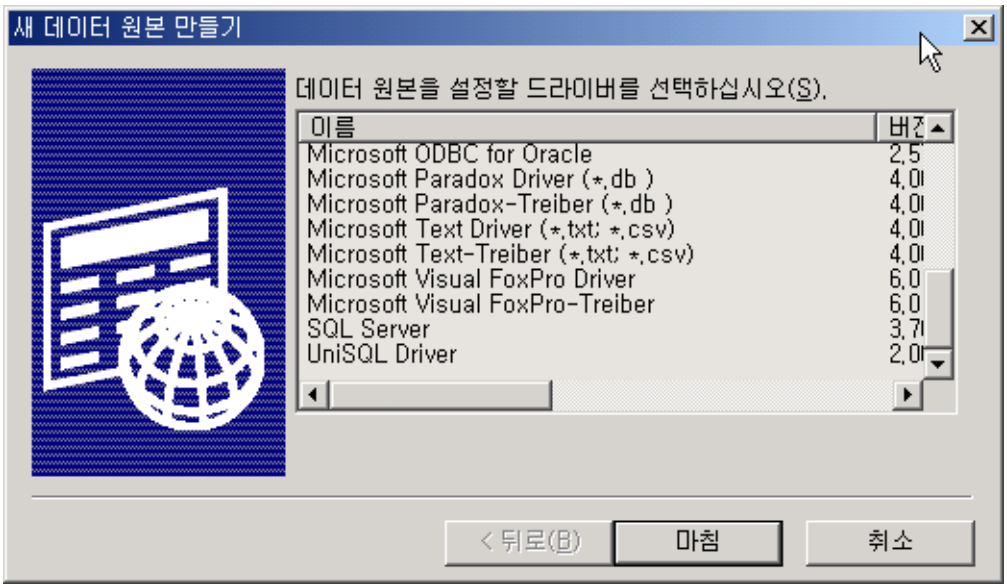
Driver 설치 방법

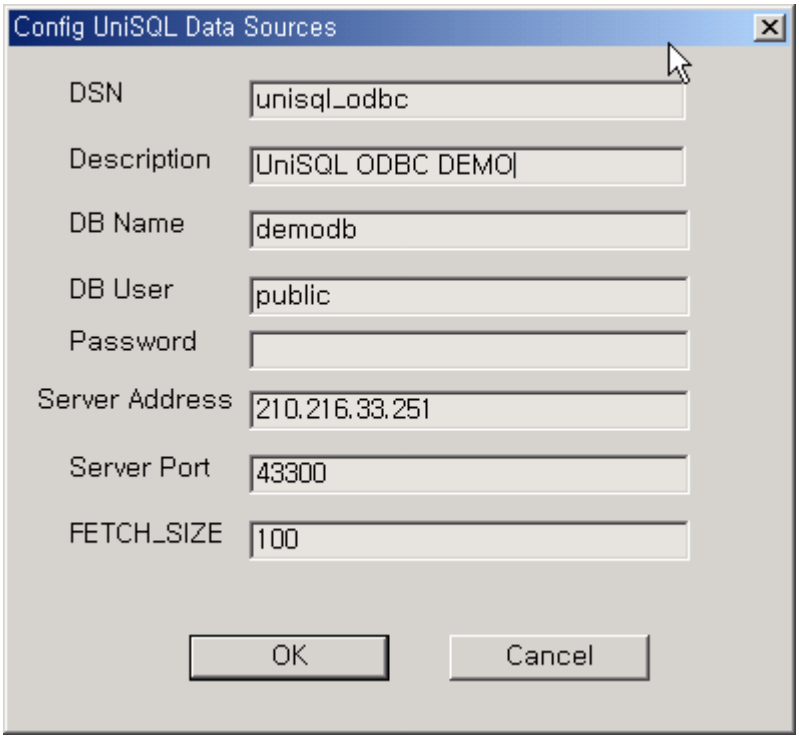
이 장은 ODBC, JDBC, PHP 드라이버를 설치하는 방법을 기술한다.

ODBC 드라이버 설치

Step	Action
Step 1	제품 CD 로부터 UniSQL_ODBC_<version>.zip 파일의 압축을 풀어 UniSQL_ODBC_<version>.exe 을 추출한 뒤 실행한다. (예를 들어 UniSQL_ODBC_2.0.5.zip 파일에서 UniSQL_ODBC_2.0.5.exe 파일을 추출한 뒤 실행한다.)
Step 2	아래의 화면은 설치를 시작할 것인지를 묻는 것으로 설치하려면 “예(Y)” 버튼을 클릭한다. 
Step 3	아래의 화면은 저작권과 버전 정보를 보여주는 화면으로 읽은 후 “Next” 버튼을 클릭한다. 

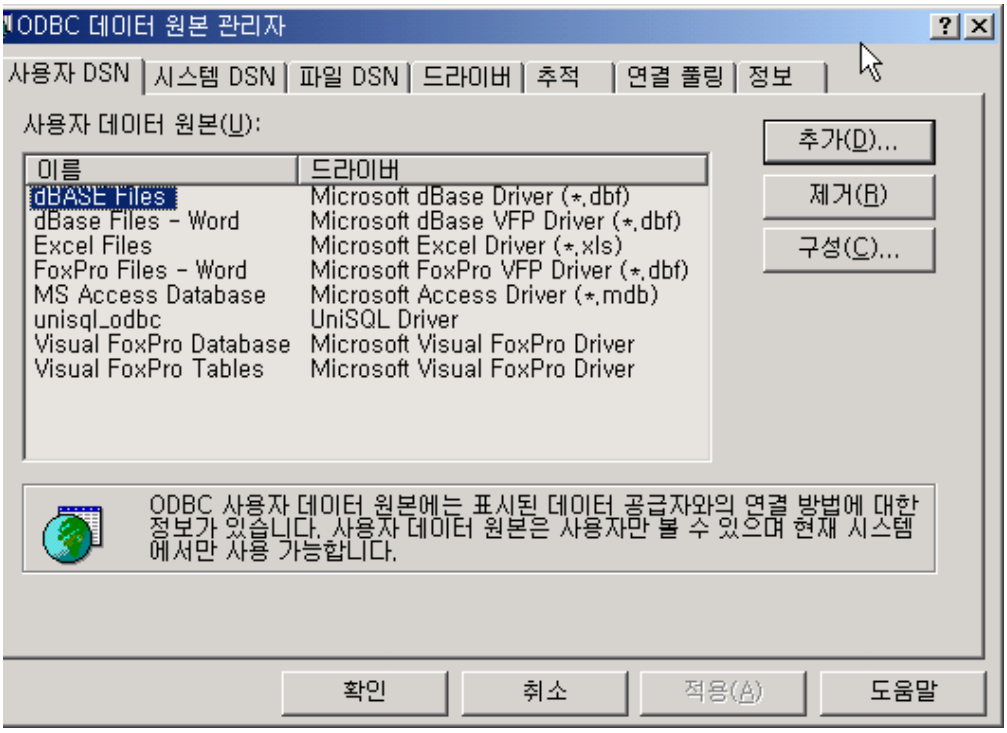
Step 4	설치가 완료되면 아래의 화면이 나온다. “Finish” 버튼을 클릭하여 설치를 종료한다. 
Step 5	드라이버가 올바로 설치되었음을 확인하기 위해 “제어판”->”관리도구”->”ODBC 데이터 원본 관리자” 를 실행시킨 후 “드라이버” 탭을 클릭하여 “UniSQL Driver” 를 확인한다. 

Step 6	"사용자 DSN" 탭을 클릭하면 아래의 화면이 나타난다. 새로운 DSN 을 만들기 위해 오른쪽에 있는 "추가" 버튼을 클릭한다. 
Step 7	아래의 화면이 나타나면 "UniSQL Driver" 를 클릭한다. 
Step 8	아래의 데이터 원본 입력 창을 채우고 "OK" 버튼을 클릭한다.



Step 9

새로운 DSN 이 올바로 생성되었으면 "사용자 DSN" 탭에 해당 DSN 을 가진 레코드를 볼 수 있다.



JDBC 드라이버 설치

Step	*Action*

Step 1	UniSQL JDBC 드라이버 제품 파일명은 unisqljdbc<version>.jar 로 이루어져 있다. 버전이 2.0.2 (JDBC 버전이 아니고 드라이버 버전임) 인 제품을 설치하려면 unisqljdbc202.jar 파일이 필요하다. 제품 CD 에 unisqljdbc 로 시작하는 파일이 하나라면 아래와 같이 “unisqljdbc*.jar” 로 명세할 수 있다. 아래의 예는 $HOME/driver 디렉토리를 만들고 거기에 UniSQL JDBC 드라이버를 복사한다.
	
	`% mkdir $HOME/driver`
	`% cp /cdrom/unisqljdbc*.jar $HOME/driver`
	
	Windows 의 경우에는 아래와 같이 된다.
	
	`C:> mkdir \UniSQL\driver`
	`C:> copy E:\unisqljdbc*.jar \UniSQL\driver`

Step 2	환경변수 CLASSPATH 에 “unisqljdbc*.jar” 의 절대 경로를 추가하여 설치를 완료한다.
	■ “csh" 사용시 “.cshrc” 파일에서
	
	`setenv CLASSPATH $HOME/driver/unisqljdbc202.jar:$HOME/jdk1.3/lib`
	
	■ “sh" 사용시 “.profile” 파일에서
	
	`CLASSPATH=$HOME/driver/unisqljdbc202.jar:$HOME/jdk1.3/lib`
	`export CLASSPATH`
	
	■ Windows 사용시 “autoexec.bat” 파일에서
	
	`set CLASSPATH=C:\UniSQL\driver\unisqljdbc202.jar:C:\jdk1.3\lib`

PHP 드라이버 설치

Step	*Action*

Step 1	UniSQL JDBC 드라이버 제품 파일명은 php_unisql_<platform>_<os>_<version>.tar.Z 로 이루어져 있다. 사용하는 머신이 SUN-SPARC 플랫폼이고 OS 가 Solaris2.6 이고 드라이버 버전이 1.2.0 (PHP 버전이 아니고 드라이버 버전임) 인 제품을 설치하려면 php_unisql_sparc_solaris26_1.2.0.tar.Z 파일이 필요하다. 제품 CD 에 php_unisql 로 시작하는 파일이 하나라면 아래와 같이 “php_unisql*.tar.Z” 로 명세할 수 있다. 아래의 예는 $HOME/driver 디렉토리를 만들고 거기에 UniSQL PHP 드라이버 제품의 압축을 풀고 추출한다.
	
	`% mkdir $HOME/driver`
	`% cd $HOME/driver`
	`% cat /cdrom/php_unisql*.tar.Z | uncompress | tar xvf -`

Step 2	$HOME/driver 디렉토리에는 드라이버 명이 unisql<php-version>.so 로 구성된 .so 파일들이 다양한 PHP 버전에 따라 존재하게 되므로 사용중인 PHP 버전에 따라 적합한 .so 파일을 선택하여 “php.ini” 파일에 아래의 내용과 같이 설정하여 설치를 완료한다.
	■ “extension_dir” 환경변수를 드라이버가 존재하는 디렉토리로 설정한다.
	
	`extension_dir=/usr/unisql/driver`
	
	■ “extension” 환경변수에 드라이버 파일명을 설정한다. 여기서는 unisql4.0.1pl2.so 를 사용한다.
	
	`extension=unisql4.0.1pl2.so`
	
	■ 에러 메시지 파일을 위해 [UniSQL] 섹션을 추가하고 그 아래 unisql.err_path 환경변수에 에러 메시지 파일의 절대 경로를 설정한다.
	
	`[UniSQL]`
	`unisql.err_path=/usr/unisql/driver/unisql_err.msg`

A

Appendix

Appendix 에서는 아래의 내용들을 기술한다.

- 제품 연관 관계
- 지원 플랫폼
- 시스템 튜닝

제품 연관 관계

아래의 테이블은 함께 연동될 수 있는 제품들의 버전들을 나타낸다. 예를 들어, UniSQL 6.0 을 사용하는 사용자에게는 UniCAS 4.6, JDBC 2.0, ODBC 2.1, PHP 2.0, TCL 8.0/8.3 을 함께 연동해서 사용할 수 있으며 UniWEB 은 Not Applicable 하다. 여기서 버전은 Spec 버전이 아니고 제품 버전이다.

<table>
<tr><th>UniSQL</th><th>UniCAS</th><th>JDBC</th><th>ODBC</th><th>PHP</th><th>TCL</th><th>UniWEB</th></tr>
<tr><td>6.0</td><td rowspan="2">4.6.x</td><td rowspan="5">2.0</td><td rowspan="3">2.0</td><td rowspan="3">2.0</td><td rowspan="4">8.0/8.3</td><td rowspan="3">N/A</td></tr>
<tr><td>5.2</td></tr>
<tr><td>5.1</td><td rowspan="2">4.5.x</td></tr>
<tr><td>5.0</td><td rowspan="2">1.5/2.0</td><td rowspan="2">1.2/2.0</td><td rowspan="6">3.8.x</td></tr>
<tr><td>4.3</td><td>4.2</td></tr>
<tr><td rowspan="4">4.1</td><td>4.1.6</td><td rowspan="3">1.0.[2-6]</td><td rowspan="4">1.0</td><td rowspan="2">1.1.0</td><td rowspan="2">8.3</td></tr>
<tr><td>4.1.[3-5]</td></tr>
<tr><td>4.1.[0-2]</td><td rowspan="2">1.0.x</td></tr>
<tr><td>4.0.x</td><td>1.0/1.0.1</td><td>8.0</td></tr>
<tr><td>4.0.x</td><td>N/A</td><td>N/A</td><td>N/A</td><td>N/A</td><td>N/A</td></tr>
</table>

지원 플랫폼

UniSQL/X, UniCAS, UniTool 은 아래의 플랫폼에 설치 될 수 있다.

- Sun Microsystems SPARC with Solaris 2.6 or 7
- HP with HP-UX 11.0
- IBM RS/6000 with AIX 4.3.X
- OpenUnix 8
- Intel Pentium-compatible with Red Hat Linux 7.2
- Intel Pentium-compatible with Windows2000/XP

일반적으로 UniSQL/X, UniCAS, UniTool 을 설치하기 위해서는 메인 메모리 128 MB 이상이 필요하며, 300 MB 이상의 디스크 공간이 필요하다.

시스템 튜닝

5.0 이상의 version 을 사용하고자 한다면 반드시 각 platform 에 맞게 튜닝을 해 주고 환경을 설정하는 것이 필요하다.

Sun Solaris2.6, 7

- Sun Solaris2.6 이나 Solaris7 의 경우 UniSQL 5.0 이상을 사용하기 위하여 특별히 튜닝 해 주어야 할 부분은 없다. 다만, Solaris7 의 libthread patch13 이상으로 반드시 upgrade 해주어야 한다. 필요한 patch cluster 는 SunSolve Homepage (http://sunsolve.sun.com) 에서 다운로드 받을 수 있다. (가장 최근의 patch cluster 를 다운로드 받는다.)

libthread patch version 확인

% showrev –p | grep 106980 (Solaris7, patch level 13 이상, 예: 106980-13)

% showrev –p | grep 105568 (Solaris2.6 , patch level 20 이상)

HP-UX 11.0

- HP-UX 에서 UniSQL 5.0 이상을 운용하기 위해서는 kernel parameter 값을 변경해 주어야 한다. 그렇지 않은 경우 사용 가능한 자원의 한계로 인하여 DB Server 가 뜨지 않는 현상이 발생할 수 있고 기타 많은 문제점에 직면할 수 있다. 각 kernel parameter 의 값은 HP-UX 의 **sam** 을 이용하여 변경할 수 있다. (root user)

[HP-UX11.0 의 관리자 메뉴얼 참고]

- Memory 관련 Parameters

Kernel Parameters	Value	Default	Description

maxdsiz	UniSQL 5.0 이상: (unisqlx_num_data_buffers * unisql_page_size) +(unisqlx_max_clients + unisqlx_active_requests) * 1024 이상	64MB	Maximum process data segment size
maxssiz	적어도 16MB 이상	8MB	Maximum process stack segment size
maxtsiz	적어도 32MB 이상	64MB	Maximum process text segment size
maxfiles	1024 이상	64	Maximum file process can open

■ Thread 관련 Parameters

Kernel Parameters	Value
nkthread	(((NPROC*7)/4)+unisql_max_clients+ unisqlx_active_requests)
nproc	((MAXUSERS*3)+가능한 큰 값)

■ Kernel 값을 rebuild 한 후에 반드시 user resource limit 를 확인하여 제대로 변경되었는지 보고 UniSQL 을 사용하도록 한다.

csh 사용

```
% limit
cputime              unlimited
filesize             unlimited
datasize             3711104 kbytes
stacksize            386048 kbytes
coredumpsize         2097151 kbytes
memoryuse            unlimited
descriptors          2048
```

sh 사용시

```
$ ulimit –a
time(seconds)        unlimited
file(blocks)         unlimited
data(kbytes)         3711104
```

```
stack(kbytes)              386048
memory(kbytes)         unlimited
coredump(blocks)       4194303
nofiles(descriptors)       2048
```

IBM AIX 4.3.X

- SMP 가 지원되는 AIX(CPU 가 2 개 이상인 machine)를 사용하는 경우에는 thread tuning 을 고려해 주어야 한다. 환경변수를 적절히 설정하지 않고 UniSQL 5.0 이상을 사용할 경우 예상하지 못한 문제(server 가 CPU 를 할당 받지 못하는 문제 등) 가 발생할 수 있으므로 사용자는 이 점을 숙지하고 system 에 적합한 환경변수를 설정해 주어야 한다. AIX4.3.X 에서 UniSQL 5.0 이상의 경우 두 가지 모드, SCOPE SYSTEM, SCOPE PROCESS 에서 운영될 수 있다. 여기서 전자는 system 전반에 걸친 thread 들과 함께 자원경쟁 및 스케줄링을 한다는 것을 의미하고 후자는 process 범주 내에서 thread 를 스케줄링 하는 것을 의미한다. UniSQL 5.0 이상은 default 로 PROCESS SCOPE mode 로 운영되고, SYSTEM SCOPE 로 변경하고 싶은 경우에는 sqlx.init 파일에 is_pthread_scope_process 값을 0 으로 설정해 주면 된다. System scope 를 사용할지 Process scope 를 사용할 지는 각 운영환경에 따라 달라질 수 있으니 사용자는 테스트를 통하여 좀 더 나은 쪽을 선택하는 것이 좋을 것이다. 일반적으로 SYSTEM SCOPE 모드가 권고된다.

- [AIX 관리자 매뉴얼 및 아래 site 참고]

http://www.rs6000.ibm.com/doc_link/en_US/a_doc_lib/aixbman/prftungd/2365c35.htm

[CASE 1] SCOPE_SYSTEM

[$UNISQLX/admin/sqlx.init]

is_pthread_scope_process=0

Environment variables	Default	Range	Recommended Value
YIELDLOOPTIME	0	A positive value	10

```
% vi $UNISQLX/admin/sqlx.init
is_pthread_scope_process=0      # 이 줄을 파일에 추가한다

csh
```

```
% vi ~unisql/.cshrc
setenv YIELDLOOPTIME    10

sh
$ vi ~unisql/.profile
YIELDLOOPTIME=10
export    YIELDLOOPTIME
```

[CASE 2] SCOPE_PROCESS

Environment variables	Default	Range	Recommended Value
YIELDLOOPTIME	0	A positive value	10
AIXTHREAD_MNRATIO	8:1	Two positive values (p:k), where k is the number of kernel threads that should be employed to handle p runnable pthreads	1:1

```
csh
% vi ~unisql/.cshrc
setenv YIELDLOOPTIME    10
setenv AIXTHREAD_MNRATIO    1:1

sh
$ vi ~unisql/.profile
YIELDLOOPTIME=10
export    YIELDLOOPTIME
AIXTHREAD_MNRATIO=1:1
export AIXTHREAD_MNRATIO
```

■ **[참고] Thread Support Tunable Parameters**

AIXTHREAD_MNRATIO (Version 4.3 and later)

Purpose:

Controls the scaling factor of the library. This ratio is used when creating and terminating pthreads.

Values:

Default: 8:1; Range: Two positive values (*p:k*), where *k* is the number of kernel threads that should be employed to handle *p* runnable pthreads

Display:

echo $AIXTHREAD_MNRATIO (this is turned on internally, so the initial default value will not be seen with the **echo** command)

Change:

AIXTHREAD_MNRATIO=*p:k*
export AIXTHREAD_MNRATIO

Change takes effect immediately in this shell. Change is effective until logging out of this shell. Permanent change is made by adding **AIXTHREAD_MNRATIO**=*p:k* command to the **/etc/environment** file.

Diagnosis:

N/A

Tuning:

May be useful for applications with a very large number of threads. However, always test a ratio of 1:1 because it may provide for better performance.

Refer to:

Variables for Process-Wide Contention Scope

YIELDLOOPTIME

Purpose:

Controls the number of times to yield the processor before blocking on a busy lock (only for libpthreads). The processor is yielded to another kernel thread, assuming there is another runnable kernel thread with sufficient priority.

Values:

Default: 0; Range: A positive value

Display:

echo $YIELDLOOPTIME (this is turned on internally, so the initial default value will not be seen with the **echo** command)

Change:

YIELDLOOPTIME=*n*
export YIELDLOOPTIME

Change takes effect immediately in this shell. Change is effective until logging out of this shell. Permanent change is made by adding **YIELDLOOPTIME**=*n* command to the **/etc/environment** file.

Diagnosis:

If threads are going to sleep often (lot of idle time), then the **YIELDLOOPTIME** may not be high enough.

Tuning:

Increasing the value from default value of 0 may benefit if you do not want the threads to go to sleep when waiting for locks.

Refer to:

Thread Environment Variables

OpenUnix 8

- OpenUnix8 operating system 설치와 동시에 설정되는 기본 Kernel Parameter 값은 UniSQL 5.0 이상을 운영하기에 부적합하다. 이러한 문제로 UniSQL 5.0 이상 버전이 올바르게 동작하지 않을 가능성이 크므로 Kernel Parameter 값을 적절히 튜닝해 주어야 한다.

- 파라미터 값은 크게 현재(current)값과 초기(default)값으로 구분된다. OpenUnix8 은 현재값과 초기값을 시스템 운용에 따라서 수정한 후 시스템 커널을 재생성함으로써 시스템 튜닝(tuning)이 완료된다. OpenUnix8 은 초기값을 /etc/conf/cf.d/stune 파일로 저장하고 있다. 사용자가 수정해야 할 시스템 파라미터들은 다음 표와 같다.

파라미터 이름	설명	설정 값(MAX)
SCORLIM	The soft limit specifying the largest size, in bytes, of a core file that can be created. A soft limit of ``0'' prevents the creation of core files.	0X7FFFFFFF
HCORLIM	The maximum value of SCORLIM	0X7FFFFFFF
SDATLIM	The soft limit specifying the maximum size, in bytes, of a process's heap.	0X7FFFFFFF
HDATLIM	The maximum value of SDATLIM.	0X7FFFFFFF

SSTKLIM	The soft limit specifying the maximum size, in bytes, of the stack segment for a process.	0X7FFFFFFF
HSTKLIM	The maximum value of SSTKLIM.	0X7FFFFFFF
SVMMLIM	The soft limit specifying the maximum address space that can be mapped to a process.	0X7FFFFFFF
HVMMLIM	The maximum value of SVMMLIM.	0X7FFFFFFF
SFSZLIM	The soft limit specifying the largest offset, in bytes, of any single file that can be created by the process.	0X7FFFFFFF
HFSZLIM	The maximum value of SFSZLIM.	0X7FFFFFFF
SFNOLIM	The soft limit specifying the maximum number of open files the process can have.	2048
HFNOLIM	The maximum value of SFNOLIM	2048
NPROC	An upper bound on the number of processes in the system	12500
MAXUP	Specifies the number of concurrent processes a user without P_SYSOPS privilege can run.	5000
MAXULWP	Maximum number of additional lightweight processes (LWPs) per user created explicitly using _lwp_create.	65000
ULIMIT	virtual memory resource limit	unlimited
stream socket		256 의 배수

■ Parameter 값 변경과 적용

사용자는 다음 순서대로 시스템 파라미터를 설정하고 시스템 커널을 재생성함으로써 시스템 튜닝을 완성할 수 있다.

1) root user 로 login.

■ vi 명령을 사용하여 /etc/conf/cf.d/stune 파일을 수정한다.

```
# cd   /etc/conf/cf.d
# vi stune
```

■ idtune 명령어를 사용하여 현재값을 수정한다.

```
# cd   /etc/conf/bin
#./idtune SCORLIM 0x7FFFFFFF
```

```
#./idtune HCORLIM 0x7FFFFFFF
```

- ULIMIT 파라미터값을 설정한다. (/etc/default/login)

```
# cd   /etc/default
# vi login
# ULIMIT 필드를 COMMENT(#)로 막는다.
```

- stream socket 의 최대값을 256 의 배수로 설정한다. (/etc/conf/sdevice.d/ticots)

```
# cd   /etc/conf/sdevice.d
# vi ticots
```

- Rebuild System Kernel

```
# cd   /etc/conf/bin
# ./idbuild   –B
# init 6                          Reboot
```

Redhat Linux 7.2

- 리눅스용 UniSQL 6.X 는 Redhat Linux7.2 (Kernel 2.4.7, glibc2.2.4) 에서 작동한다. 특별히 튜닝해 주어야 하는 부분은 없고 Kernel version 과 glibc version 이 UniSQL 6.X 의 운영환경을 만족하는지 확인해 보아야 한다.

Check the Kernel version
% uname –r
2.4.7-10smp

Check the glibc version
% rpm –qa | grep glibc
glibc-common-**2.2.4**-13
glibc-**2.2.4**-13
glibc-devel-**2.2.4**-13