

[솔루션 성능 테스트] MySQL 클러스터 성능 테스트 절차서

**한국소프트웨어진흥원
공개SW기술지원센터**

<Revision 정보>

일자	VERSION	변경내역	작성자
2007. 11. 20	0.1	초기 작성	손명선

목 차

1. 문서 개요	4
가. 문서의 목적	4
나. 본 문서의 사용방법	4
2. 테스트 절차 내역	5
가. WAS(Tomcat5) + MySQL 환경설정	5
나. Rational Robot 테스트	12

1. 문서 개요

본 문서는 핵심 업무시스템에서 운영되는 공개SW 지원 솔루션의 부족 현상을 극복하고, 다양한 공개SW 지원 솔루션 확보 가속화를 위해 발굴된 MySQL DBMS 고성능 / 고가용성 솔루션인 MySQL Cluster의 성능을 검증하기 위한 테스트 수행 절차를 기술하기 위해 작성되었으며, MySQL Cluster의 도입을 검토하는 관련 업체의 참고자료로 활용하기 위해 제작되었다.

가. 문서의 목적

다음과 같은 세부적인 목적을 달성하기 위하여 작성되었다.

- 0 WAS(Tomcat5)와 단일 노드 MySQL 연동 환경의 성능 테스트 절차 및 결과 기술
- 0 WAS(Tomcat5)와 MySQL Cluster 연동 환경의 성능 테스트 절차 및 결과 기술
- 0 진행 중 문제 발생 사항과 각각의 진행사항
- 0 MySQL Cluster 도입을 검토하는 업체의 참고 자료로 제공
- 0 공개SW 지원 솔루션 확보 확대

나. 본 문서의 사용방법

다음과 같은 방법으로 사용할 수 있다.

- 0 리눅스 OS(Booyo2.0, Asianux3.0, RedHat EL4 AS) 기반하에 WAS(Tomcat5)와 단일 노드 MySQL 연동 환경의 성능 테스트 절차 및 결과를 확인한다.
- 0 리눅스 OS(Booyo2.0, Asianux3.0, RedHat EL4 AS) 기반하에 WAS(Tomcat5)와 MySQL Cluster 연동 환경의 성능 테스트 절차 및 결과를 확인한다.
- 0 MySQL Cluster 도입 검토시 참고 자료로 사용한다.

2. 테스트 절차 내역

가. WAS(Tomcat5) + MySQL 환경설정

단계	항목/시험/결과	
1	시험항목	Tomcat5와 MySQL 연동 환경 설정
	시험절차	1. JDBC 드라이버 설정 2. MySQL 데이터베이스 구성 3. 테스트 JSP 코드 작성
	시험결과	1. JDBC 드라이버 설정 MySQL JDBC 드라이버 mysql-connector-java-5.0.8-bin.jar 파일을 Tomcat의 설치된 해당 디렉토리의 ../common/lib/ 라이브러리 폴더에 복사 2. mysql 데이터베이스 구성 - test 유저 생성 - test 데이터베이스 생성 3. 테스트 JSP 코드 작성 /usr/local/tomcat/webapps/ROOT 경로에 dbbmt-mysql.jsp 코드 작성 # vi dbbmt-mysql.jsp <pre> <% int nRandom; String rt_uid; rt_uid = request.getParameter("uid"); //out.println("uid = [" + rt_uid + "]
"); if (rt_uid == null) { nRandom = (int)(Math.random()*100000); rt_uid = String.valueOf(nRandom); } %> <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"> <html> <head> <title>::: DB BMT :::</title> <meta http-equiv="Content-Type" content="text/html; charset=euc-kr"> </pre>

		<pre> <style type="text/css"> <!-- td { font-family: "Verdana", "Arial", "Helvetica", "sans-serif"; font-size: 12px; } --> </style> </head> <body> <table width="500" border="0" cellspacing="0" cellpadding="0"> <tr> <td height="30">DataBase Benchmarking Page</td> </tr> <tr> <td height="1" bgcolor="#CCCCCC"></td> </tr> <tr> <td height="100" valign="top"> <form action="dbbmt-mysql.jsp" method="post" name="form" id="form"> <input name="uid" type="hidden" id="uid" value="<%=rt_uid%>"> <input name="mode" type="hidden" value="active"> <table width="500" border="0" cellspacing="0" cellpadding="10"> <tr> <td width="102">DB UserName</td> <td><input type="text" name="dbuser" value="root"></td> <td width="67"></td> </tr> <tr> <td>DB Password</td> <td><input type="password" name="dbpwd" value=""></td> <td></td> </tr> <tr> <td>Total Row</td> <td><input type="text" name="totalrow" value="1000"></td> <td></td> </tr> <tr> <td>Select Command</td> <td><select name="choice_command"> <option value="18" selected>=====</option> <option value="1">CREATE-TBL</option> <option value="2">INSERT</option> <option value="3">SELECT</option> <option value="4">UPDATE</option> </td> </tr> </table> </form> </td> </tr> </table> </pre>
--	--	--

		<pre> <option value="5">DELETE</option> <option value="6">DELETE-TBL</option> <option value="7">ALL</option> </select></td> <td> </td> </tr> </tr> <tr> <td> </td> <td><input type="submit" name="Submit" value="Submit"></td> <td> </td> </tr> </table> </form></td> </tr> </table> <%@ page language="java" import="java.sql.*" contentType="text/html; charset=KSC5601" %> <% String strMode; strMode = request.getParameter("mode"); if (strMode != null) { Class.forName("org.gjt.mm.mysql.Driver"); //----- // Database Connect Info //----- String db_user = "test_raw"; String db_user = request.getParameter("dbuser"); String db_pw = "wkqqltf"; String db_pw = request.getParameter("dbpwd"); String db_conn = "jdbc:mysql://210.183.235.70:3306/ndb_test"; //----- int i; int nRecordCount = 0; String strTotalRow = request.getParameter("totalrow"); nRecordCount = Integer.parseInt(strTotalRow); //nRecordCount = 100; String TableName = "DBBMT_" + rt_uid; String Qry_CrtTable = "create table " + TableName + " (id int,field1 varchar(30),field2 varchar(30),field3 varchar(30), PRIMARY KEY (field1)) ENGINE=NDBCLUSTER"; String Qry_InsertTable; String Qry_SelectTable; </pre>
--	--	--

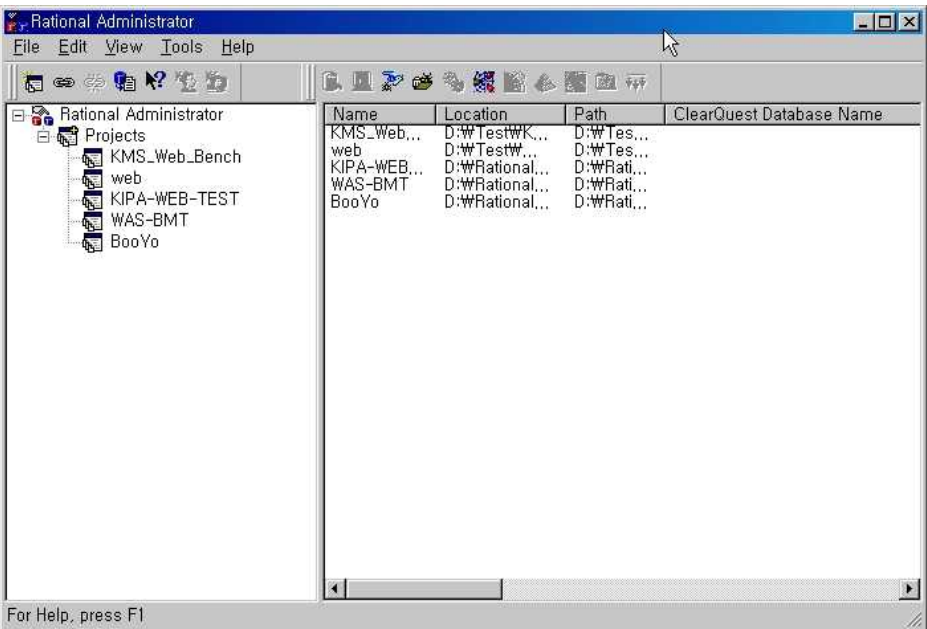
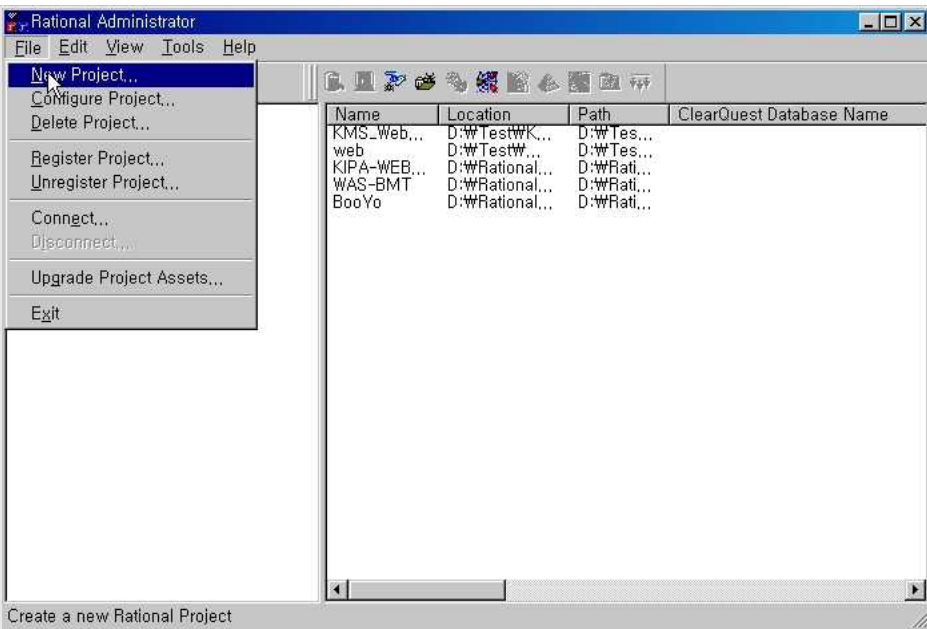
		<pre> String Qry_UpdateTable; String Qry_DeleteTable; String Qry_DropTable; String Field1 = "Field1-"; String Field2 = "Field2-"; String Field3 = "Field3-"; String strChoiceCmd = request.getParameter("choice_command"); int nChoiceCmd = Integer.parseInt(strChoiceCmd); out.println("CrtTable [" + Qry_CrtTable + "]"); Connection Conn = DriverManager.getConnection(db_conn,db_user,db_pw); //out.println("
"); //out.println("Result
"); //out.println("-----") //out.println("-----
"); switch(nChoiceCmd) { case 1 : Statement stmt1 = Conn.createStatement(); stmt1.executeUpdate(Qry_CrtTable); out.println("
 End Create Table " + TableName + "!!!"); stmt1.close(); break; case 2 : Statement stmt2 = Conn.createStatement(); for (i=1; i<nRecordCount+ 1; i++) { Qry_InsertTable = "insert into " + TableName + " values(" + i+ ", " + Field1 + i + ", " + Field2 + i + ", " + Field3 + i + ")"; stmt2.executeUpdate(Qry_InsertTable); if (i%10 == 0) { //out.println("[INSERT RECORD] " + TableName + " Count = " + i); //out.println("
"); } } out.println("
 End Insert Record " + TableName + "!!!"); stmt2.close(); break; case 3 : Statement stmt3 = Conn.createStatement(); Qry_SelectTable = "select sum(id) from " + TableName; ResultSet rs3 = stmt3.executeQuery(Qry_SelectTable); if (!rs3.next()) { //out.println("no data"); } } } </pre>
--	--	--

		<pre> } else { //out.println("[SELECT QUERY] " + TableName + " total record sum = " + rs3.getString(1)); } out.println("
 End Select Record " + TableName + "!!"); stmt3.close(); break; case 4 : Statement stmt4 = Conn.createStatement(); for (i=1; i<nRecordCount+ 1; i++) { Qry_UpdateTable = "update " + TableName + " set field1 = " + "updated" + Field1 + i + "" + " where field1 = " + Field1 + i + ""; stmt4.executeUpdate(Qry_UpdateTable); if (i%10 == 0) { //out.println("[UPDATE " + TableName + " RECORD Count] = " + i); //out.println("
"); } } out.println("
 End Update Record " + TableName + "!!"); stmt4.close(); break; case 5 : Statement stmt5 = Conn.createStatement(); Qry_DeleteTable = "delete from " + TableName; stmt5.executeUpdate(Qry_DeleteTable); //out.println("[DELETE QUERY] " + TableName + " is deleted !!"); out.println("
 End Delete Record " + TableName + "!!"); stmt5.close(); break; case 6 : Statement stmt6 = Conn.createStatement(); Qry_DropTable = "drop table " + TableName ; stmt6.executeUpdate(Qry_DropTable); //out.println("[Drop Table] " + TableName + " is dropped !!"); out.println("
 End Drop Class " + TableName + "!!"); stmt6.close(); break; case 7 : // Connection Statement stmt7 = Conn.createStatement(); // Create Table stmt7.executeUpdate(Qry_CrtTable); //out.println("
 Create Table " + TableName + "!!
"); </pre>
--	--	--

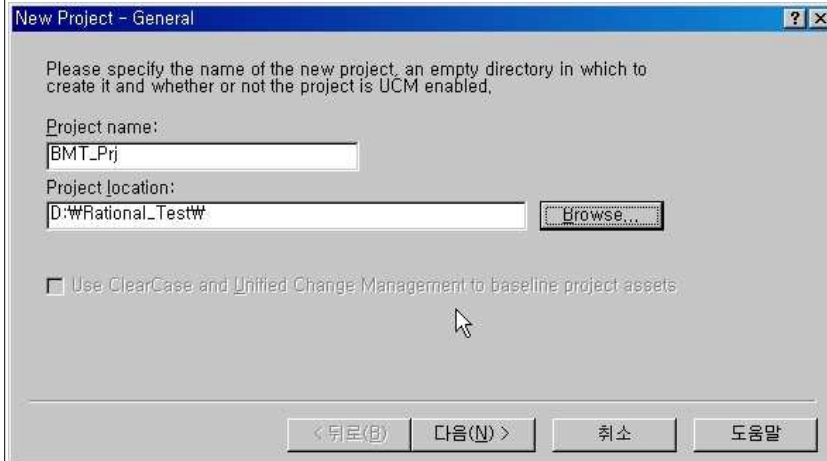
		<pre> // Insert for (i=1; i<nRecordCount+ 1; i++) { Qry_InsertTable = "insert into " + TableName + " values(" + i+ ", " + Field1 + i + ", " + Field2 + i + ", " + Field3 + i + ")"; stmt7.executeUpdate(Qry_InsertTable); if (i%10 == 0) { //out.println("[INSERT RECORD] " + TableName + " Count = " + i); //out.println("
"); } } // Select Qry_SelectTable = "select sum(id) from " + TableName; ResultSet rs9 = stmt7.executeQuery(Qry_SelectTable); if (!rs9.next()) { //out.println("no data"); } else { //out.println("[SELECT QUERY] " + TableName + " total record sum = " + rs9.getString(1)); //out.println("
"); } // Update for (i=1; i<nRecordCount+ 1; i++) { Qry_UpdateTable = "update " + TableName + " set field1 = " + "updated" + Field1 + i + "" + " where field1 = " + Field1 + i + ""; stmt7.executeUpdate(Qry_UpdateTable); if (i%10 == 0) { //out.println("[UPDATE " + TableName + " RECORD Count] = " + i); //out.println("
"); } } // Delete Qry_DeleteTable = "delete from " + TableName; stmt7.executeUpdate(Qry_DeleteTable); //out.println("[DELETE QUERY] " + TableName + " is deleted !!"); //out.println("
"); // Drop Qry_DropTable = "drop table " + TableName ; stmt7.executeUpdate(Qry_DropTable); //out.println("[Drop Class] " + TableName + " is dropped !!"); //out.println("
"); </pre>
--	--	--

		<pre> out.println("
 End ALL Routine " + TableName + "!!!"); stmt7.close(); break; } Conn.close(); } }%> </body> </html> </pre> 웹브라우저에서 http://ip:8090/dbbmt-mysql.jsp 확인 아래와 같은 화면을 볼수 있다. DataBase Benchmarking Page DB UserName root DB Password Total Row 1000 Select Command ===== ▼ Submit DB UserName : DB User DB Password : DB password Total Row : Insert, update, select시 레코드 갯수 Select Command : Create Table, Insert등 쿼리 구문
비	고	

나. Rational Robot 테스트

단계	항목/시험/결과	
	시험항목	Rational Robot 테스트 - Rational Administrator 를 실행하면 아래와 같은 화면이 출력됨 
1	시험절차	- 먼저 새로운 프로젝트를 생성한다. 

- 프로젝트 이름과 프로젝트를 위한 새로운 디렉토리를 생성후 지정한다



New Project - General [?] [X]

Please specify the name of the new project, an empty directory in which to create it and whether or not the project is UCM enabled.

Project name:

Project location:

☐ Use ClearCase and Unified Change Management to baseline project assets

< 뒤로(B) 다음(N) > 취소 도움말

- 패스워드를 지정한다 (디폴트 : 공백)



New Project - Security [?] [X]

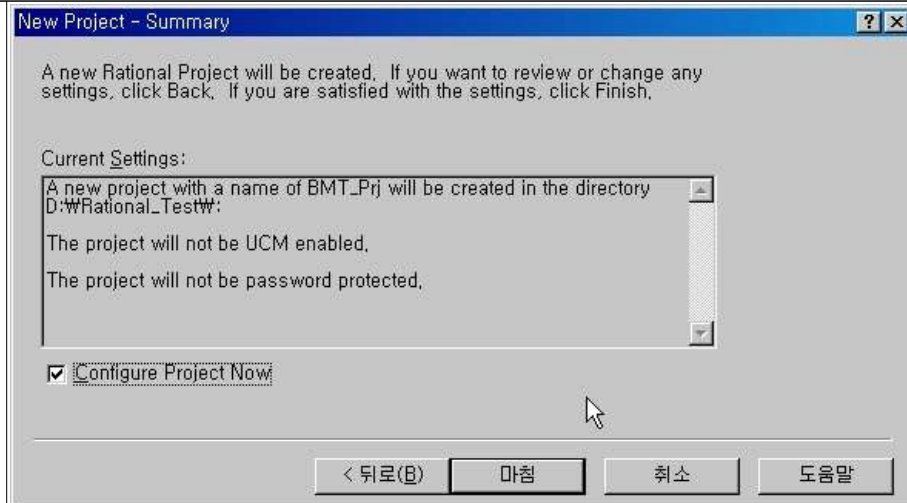
Each Rational Project may be associated with a password. This password is required to connect to, configure or delete the project.

Password:

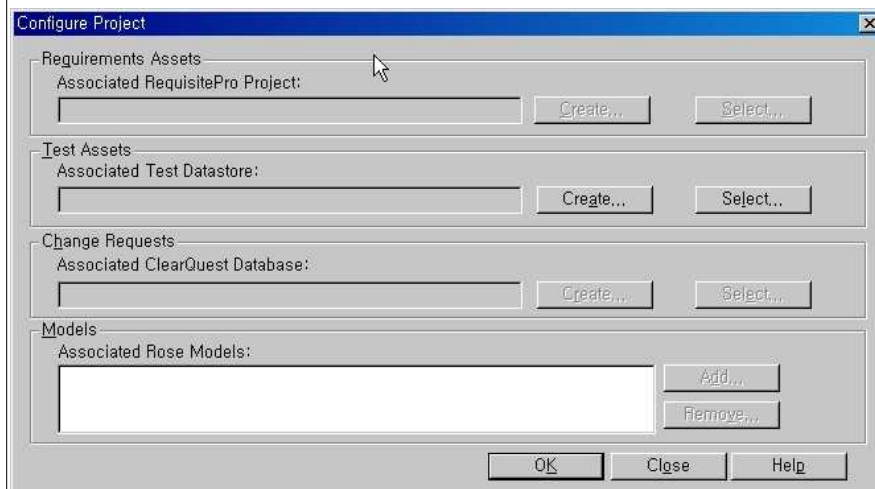
Confirm:

< 뒤로(B) 다음(N) > 취소 도움말

- 마침버튼 클릭



- Create 버튼을 클릭하여 Datastore 공간을 만든다.



- Microsoft Access로 생성을 한다.

- 다음버튼 클릭

Microsoft Access Settings (Page 2 of 4)

New test datastore path

D:\WRational_Test\WTestDatastore

Browse...

< 뒤로(B) 다음(N) > 취소

- 다음버튼 클릭

Initialize New Test Datasource from existing assets (Page 3 of 4)

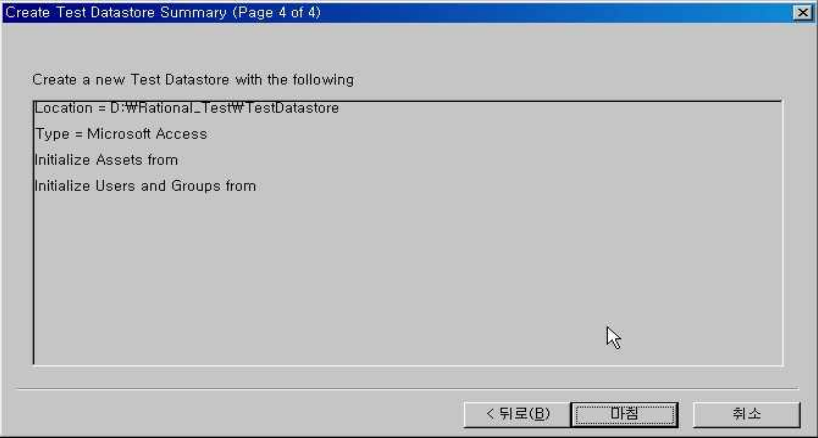
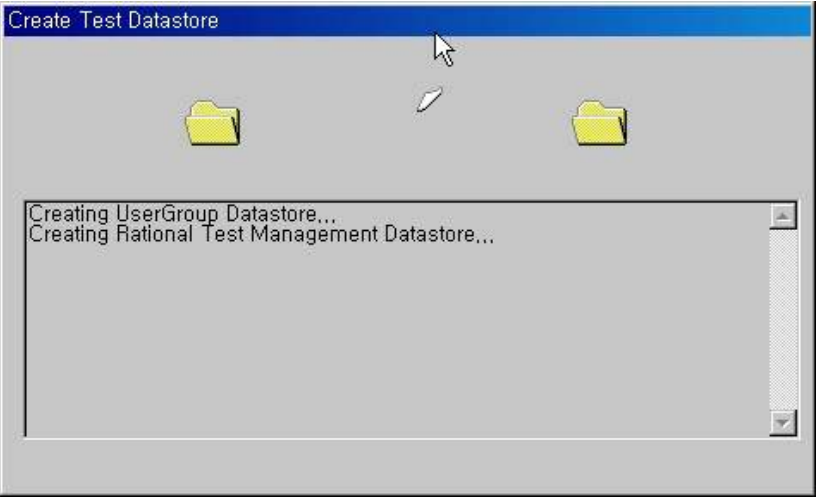
Initialization Options

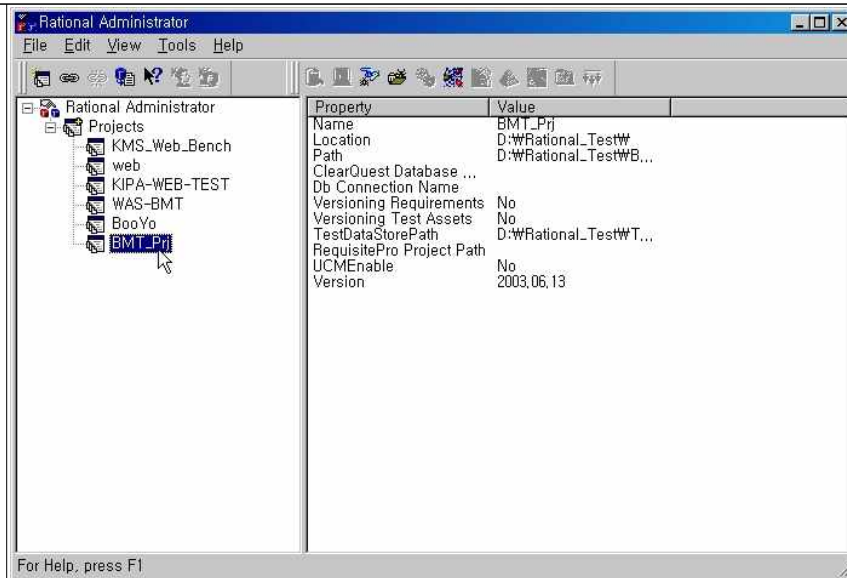
☐ Initialize Assets from the following test datasource:

☐ Initialize Test Users and Groups from the Rational project:

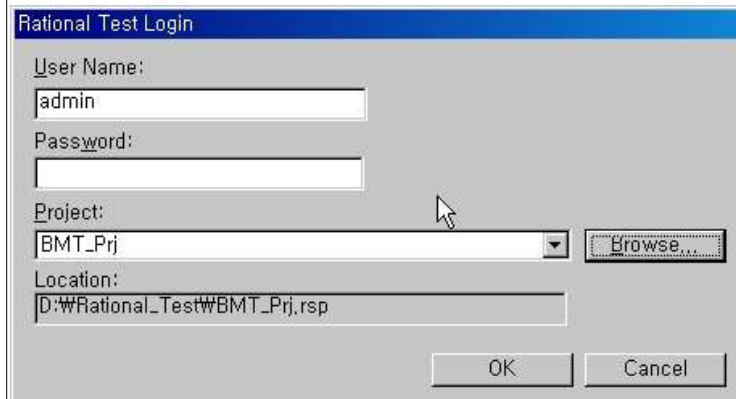
< 뒤로(B) 다음(N) > 취소

- 마침버튼 클릭

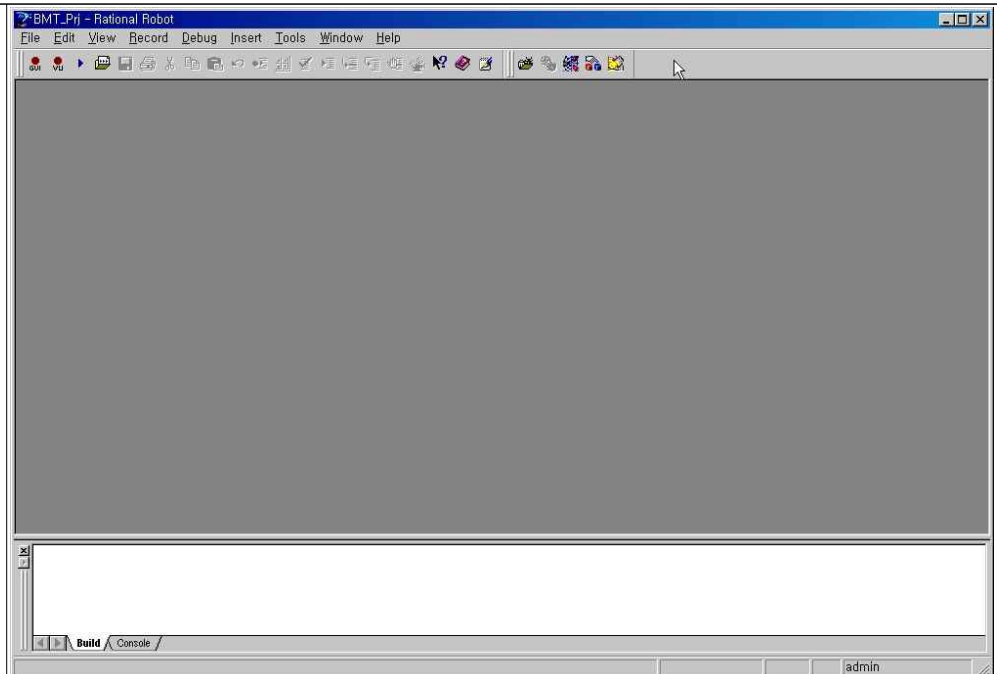
		
	- DB 생성진행	
	- 아래와 같이 새로운 프로젝트가 생성된 것을 확인할 수 있음 - 이제 새로운 프로젝트로 접속을 합니다. - 프로젝트에 마우스 클릭후 오른쪽버튼 클릭하여 connect 메뉴를 클릭합니다.	



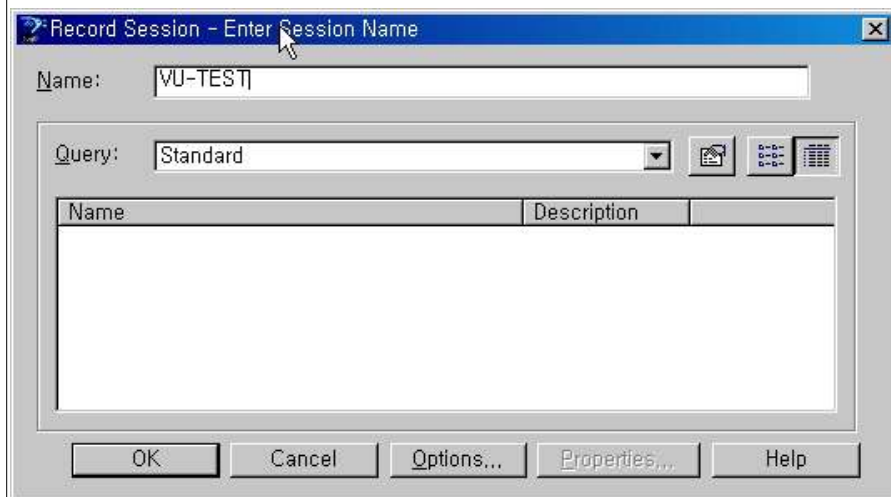
- 아래와 같이 로그인 화면이 나타납니다.
- Project란에 Browse버튼을 클릭한후 새로생성한 프로젝트 이름을 선택합니다.
- OK 버튼을 클릭하여 로그인합니다



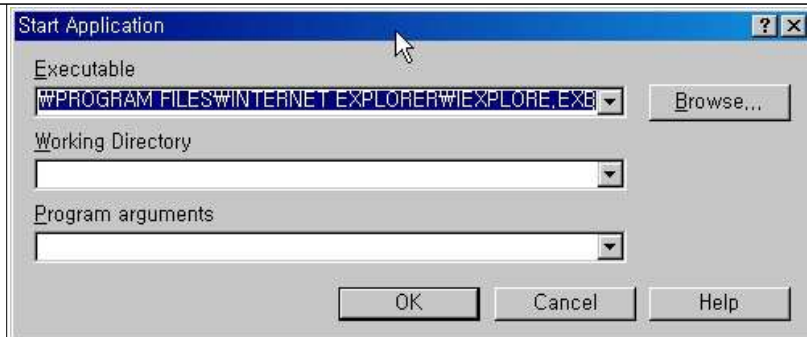
- 로그인시 아래와 같은 화면이 출력됩니다.
- 성능테스트를 하기 위해서 VU 버튼을 클릭합니다.



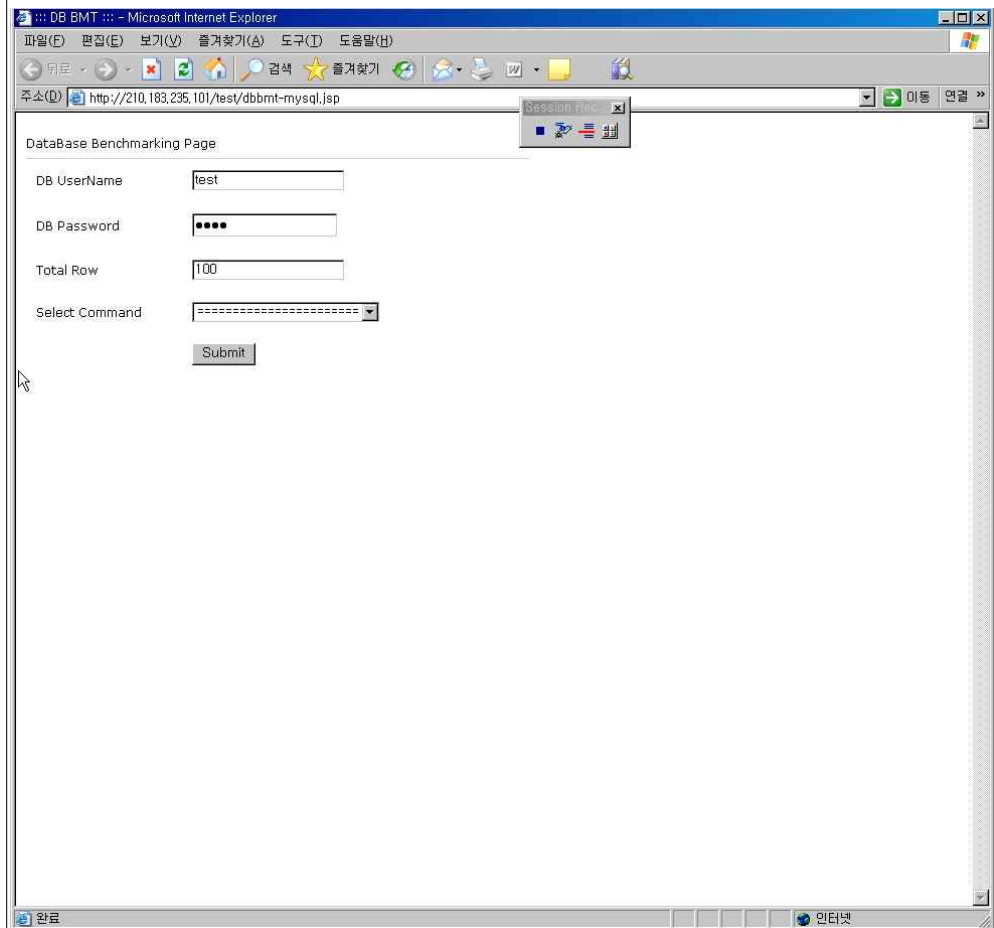
- 이름을 정의후 OK 버튼 클릭



- 웹브라우저 명령어(IEXPLORER.EXE)를 선택합니다.



- 성능테스트를 하고자 하는 URL을 입력합니다.






 세션종료버튼


 세션정의버튼

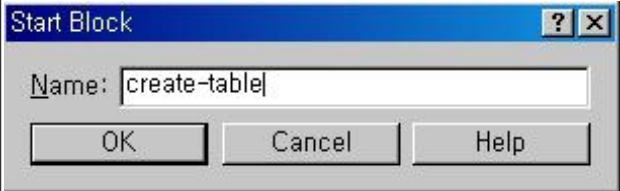



 START 버튼

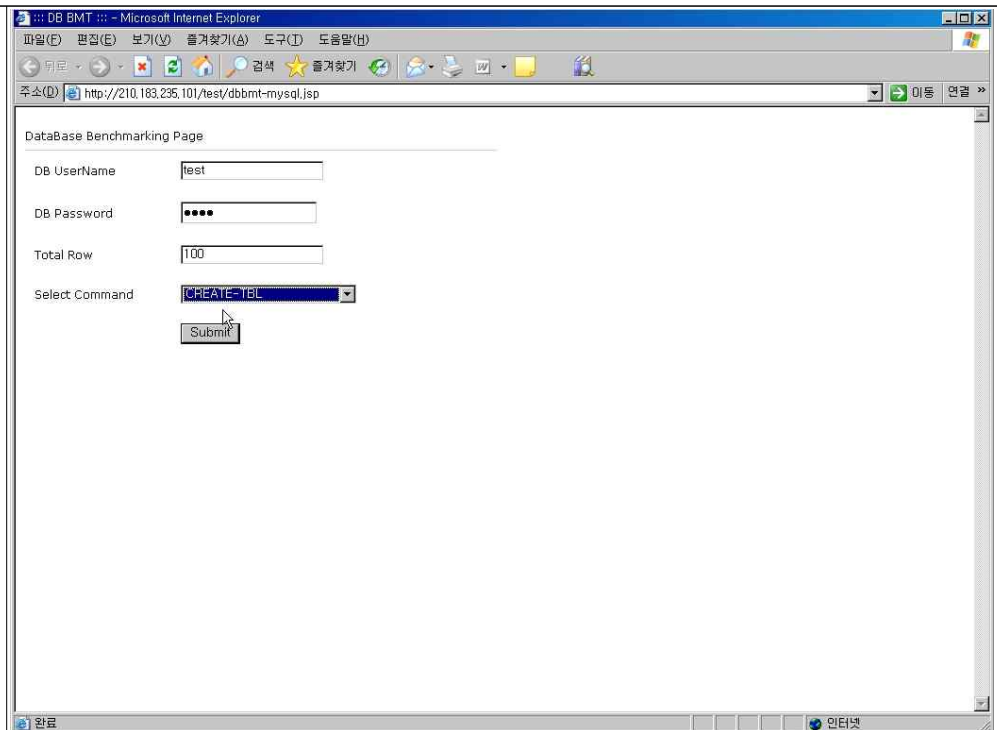

 STOP 버튼

DB의 테이블만드는 부분까지를 블록을 지정하기 위해

- START 버튼을 클릭하여 블록을 시작한다
- 블록명은 create-table이라고 임의로 지정후 OK버튼 클릭



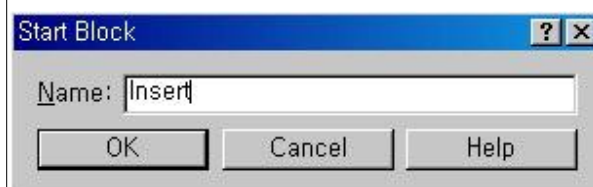
- 테이블 만드는 부분에 대해서 액션을 취한다
 (ex: select 콤보박스에서 CREATE-TABLE 선택후, submit 버튼클릭)



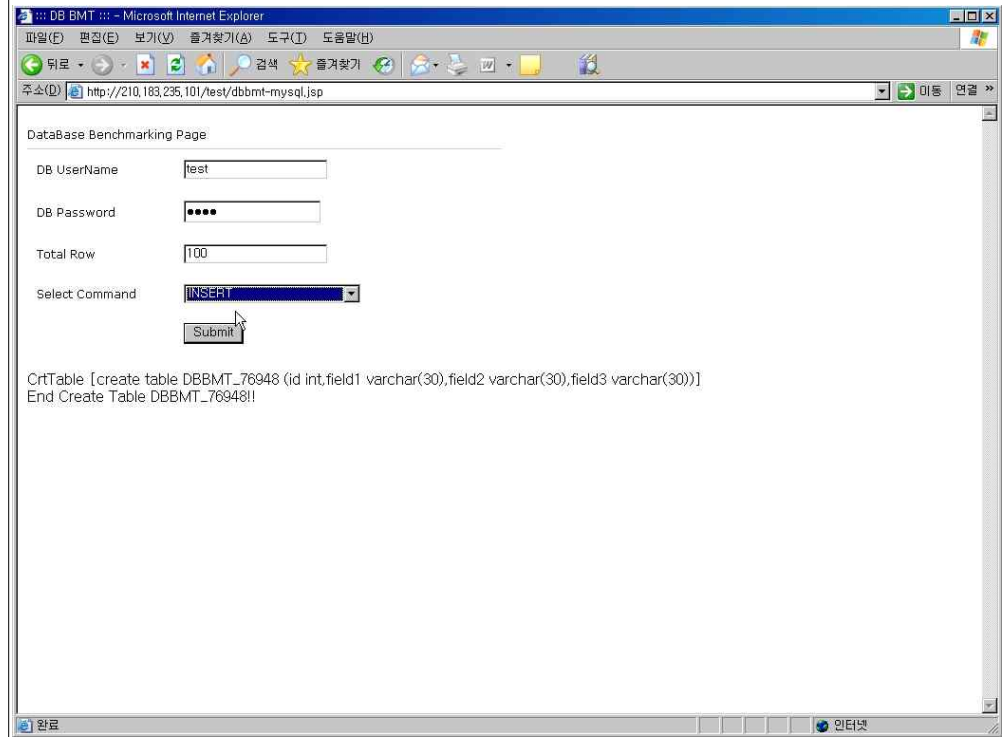
- 테이블이 생성되었으므로 위 아이콘의 맨우측아이콘 (STOP 버튼)을 클릭하여 테이블 만드는 블록을 종료한다



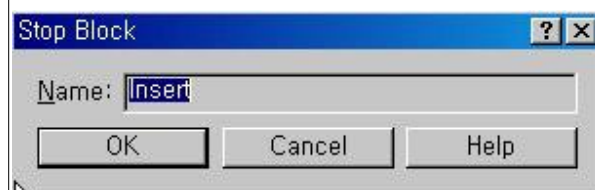
- OK 버튼클릭으로 종료
- 위와 똑같은 방법으로 이젠 테이블에 INSERT 하는 부분을 블록으로 지정해보겠습니다.
- START 블록 클릭하여 insert라는 이름 정의후 OK 버튼 클릭



- 웹브라우저에서 테이블에 INSERT하는 액션을 정의합니다.
(ex : select 콤보박스에서 INSERT 선택후, submit 버튼 클릭)



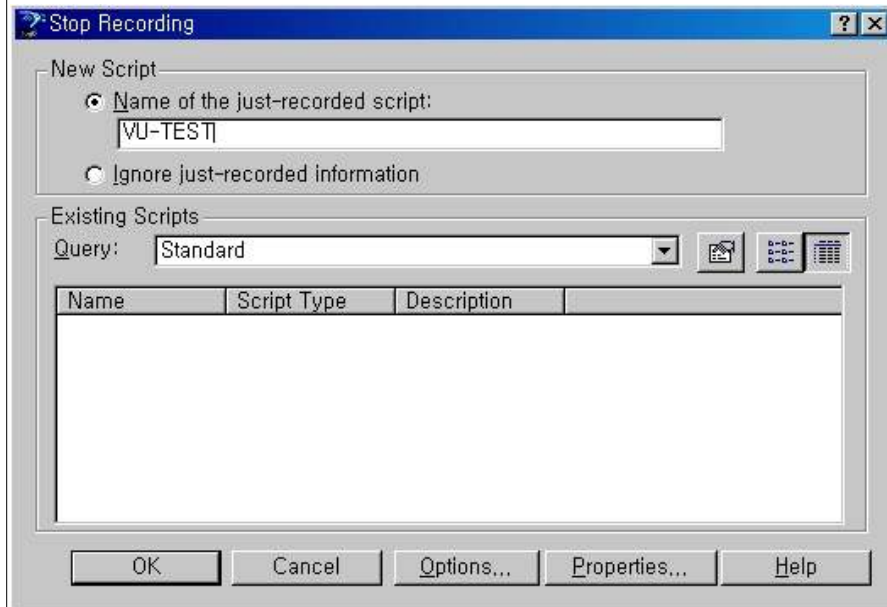
- 테이블에 INSERT 되었으므로 위 아이콘의 맨우측아이콘 (STOP 버튼)을 클릭하여 INSERT 블록을 종료한다



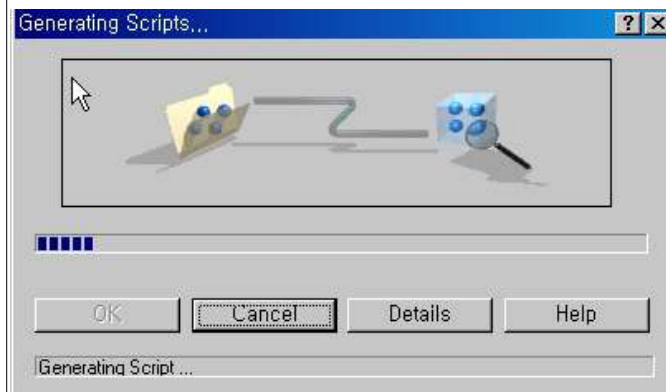
- OK 버튼클릭으로 종료
- 이젠 세션 STOP 버튼 클릭(아래그림 맨좌측아이콘)으로 모든 세션을 종료한다

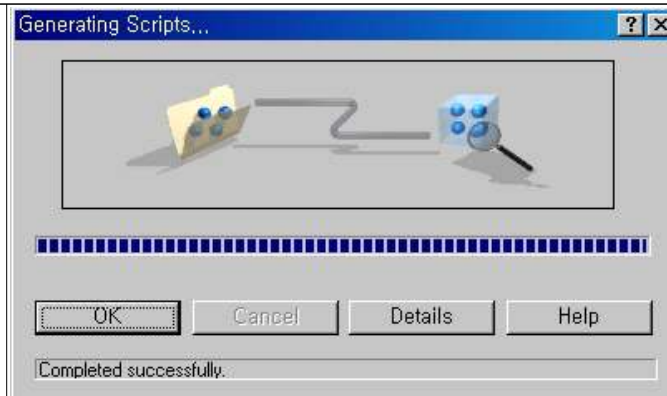


- 스크립트 이름을 지정후 OK 버튼 클릭

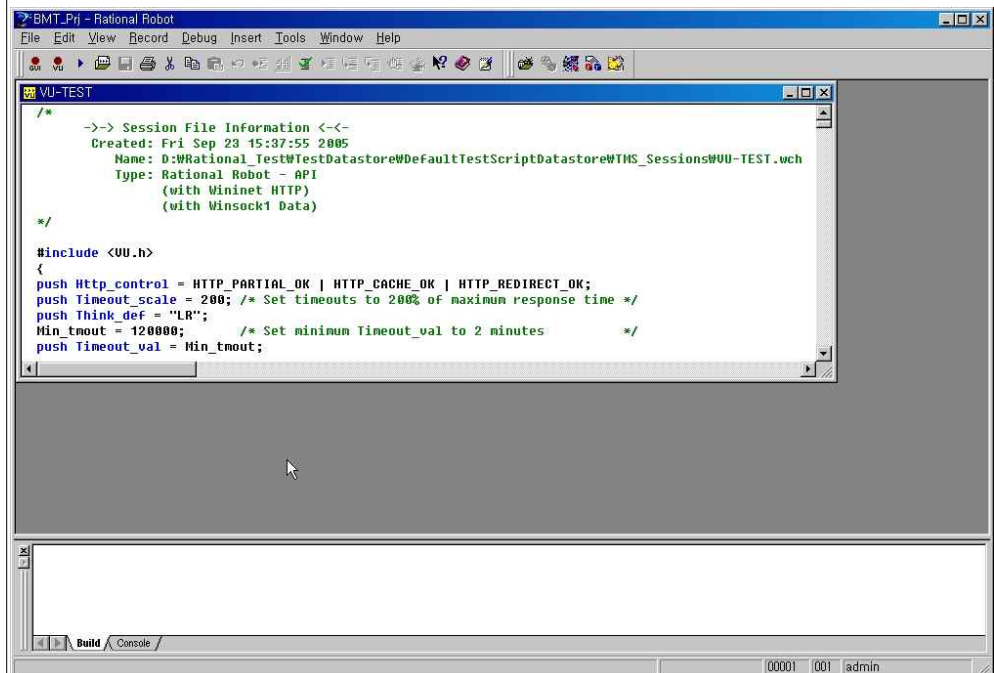


- 스크립트 생성하는 화면이 출력됨



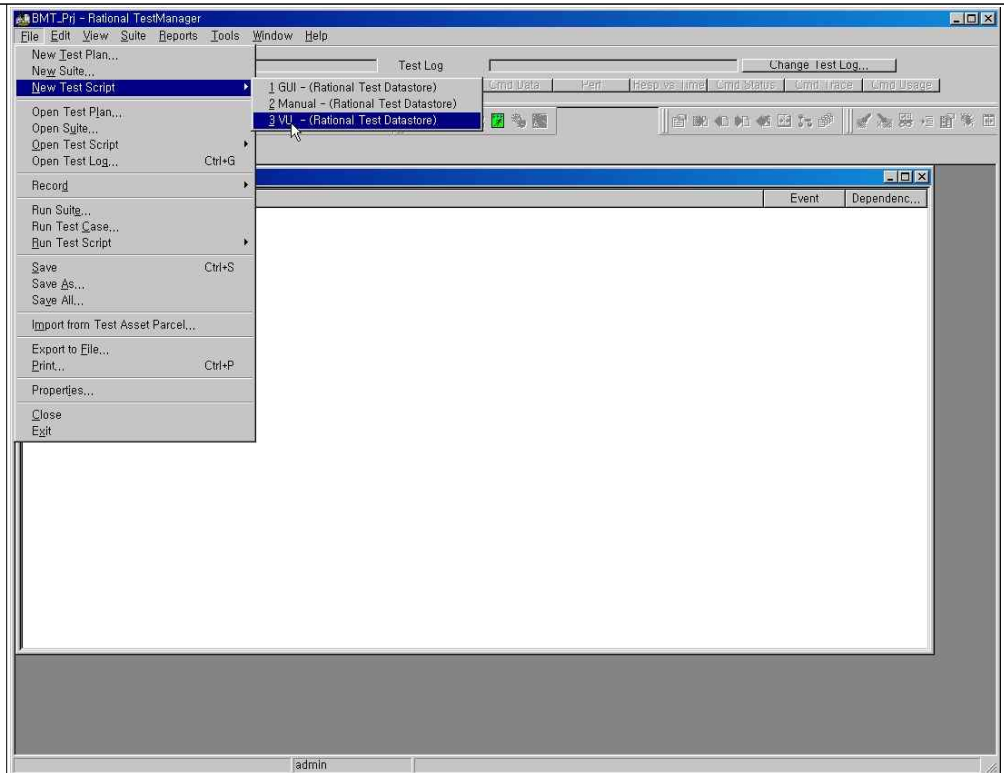


- 정상적으로 진행시 아래와 같은 스크립트가 생성됩니다.

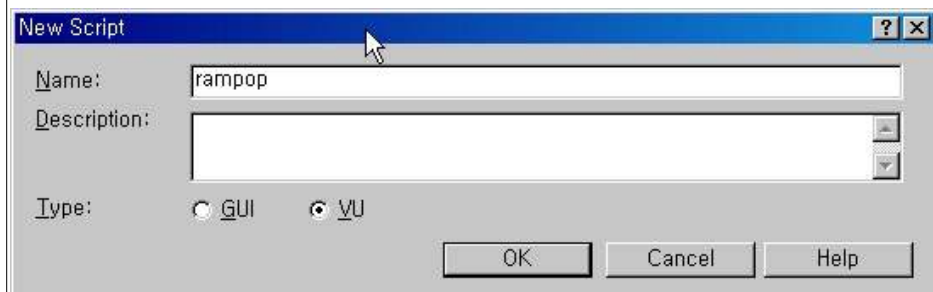


- rampop 스크립트를 생성합니다. (동시접속자에 대한 정확성을 높이기 위한 스크립트임)

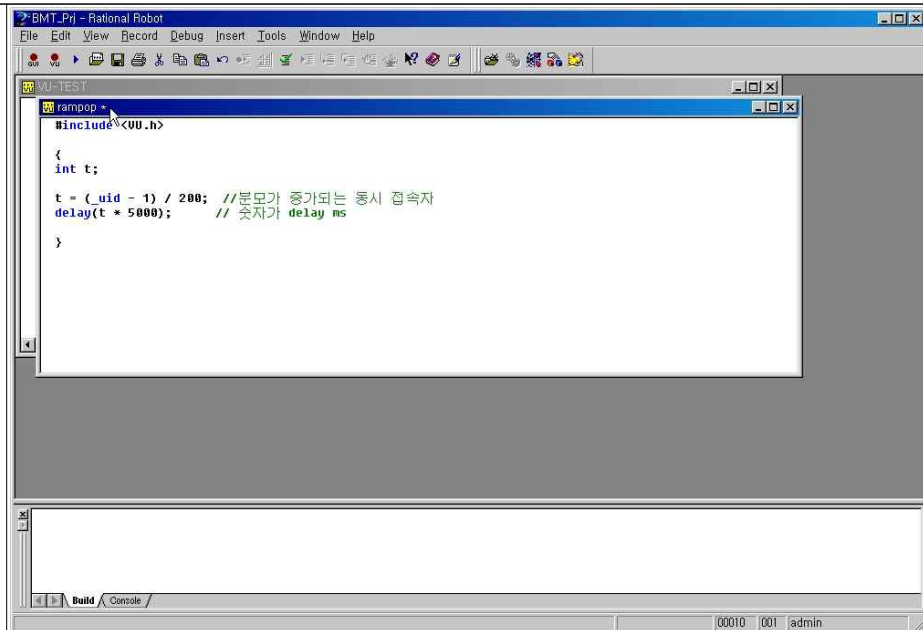
- File->New Test Script->VU 클릭



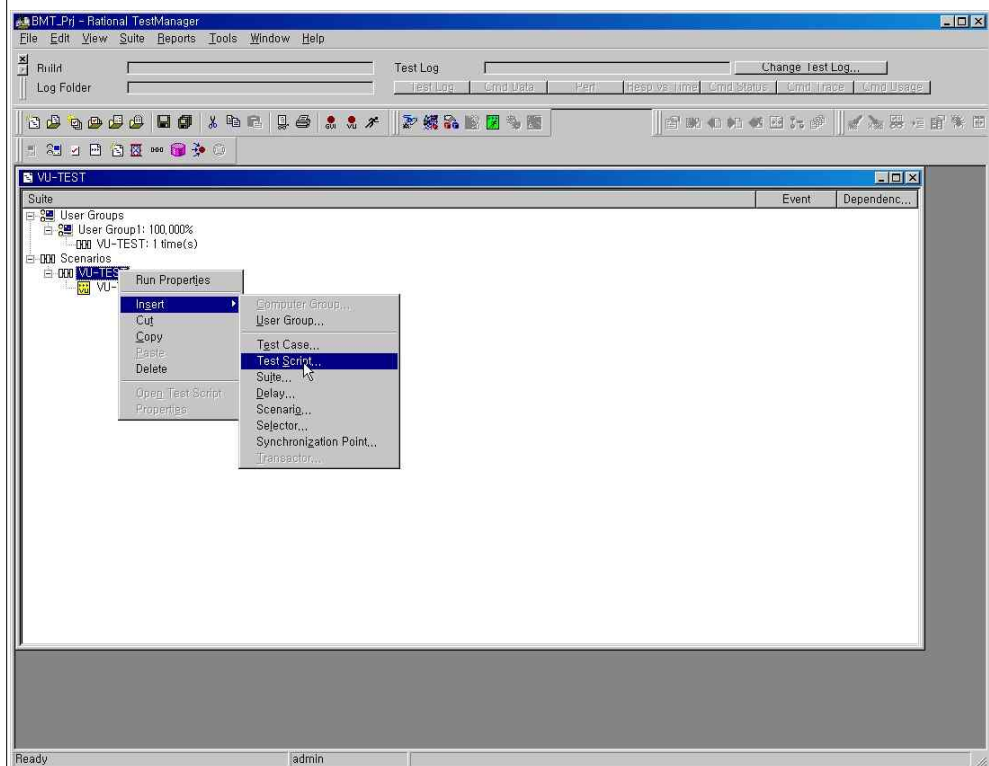
- 이름 정의후 OK 버튼클릭

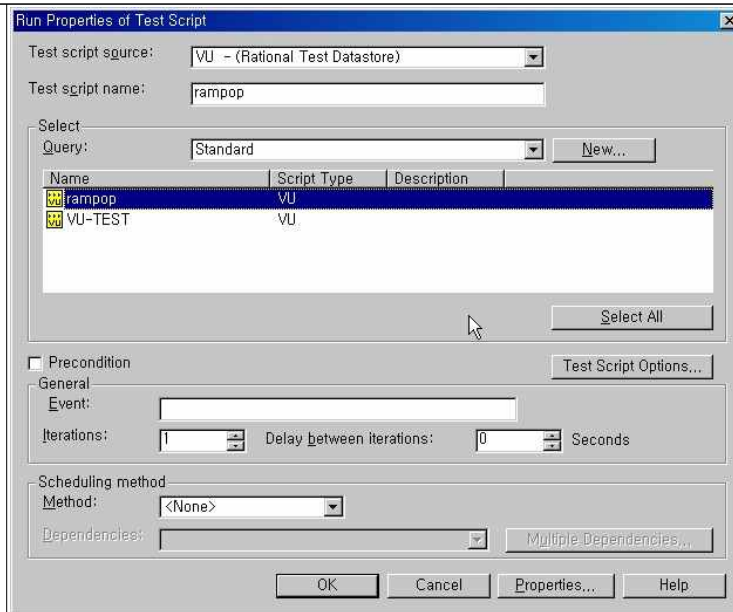


- 생성되면 아래의 코드를 삽입합니다.

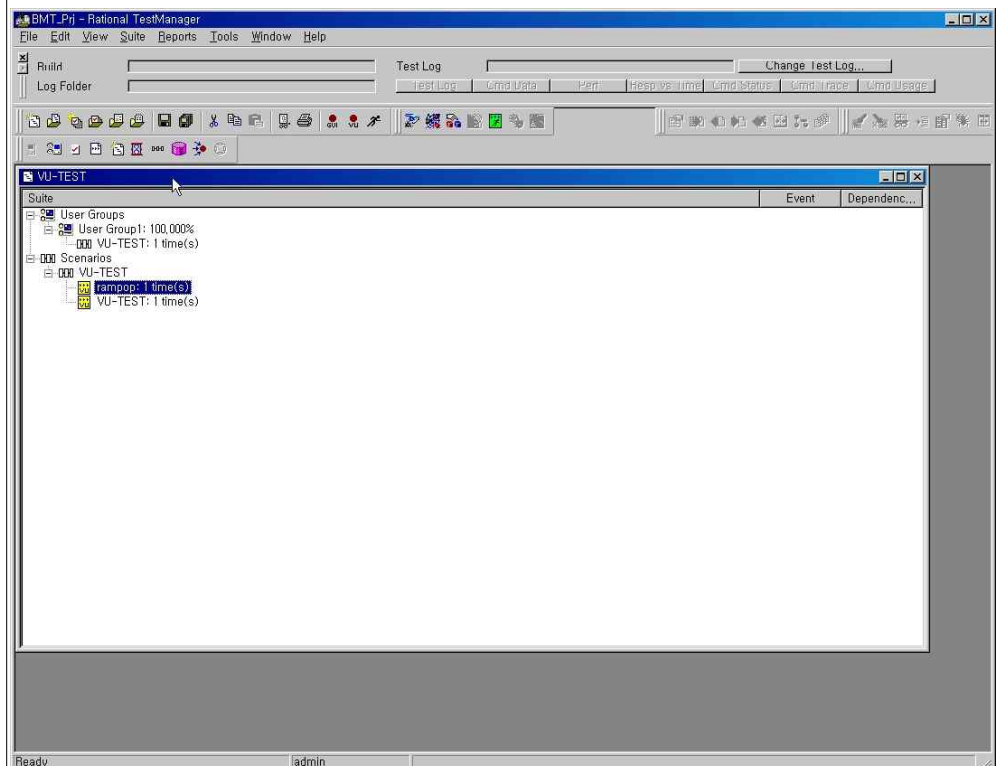


- 생성한 rampop 스크립트를 시나리오에 추가합니다.
- 시나리오에 프로젝트클릭->insert->Test Script..에서 rampop선택

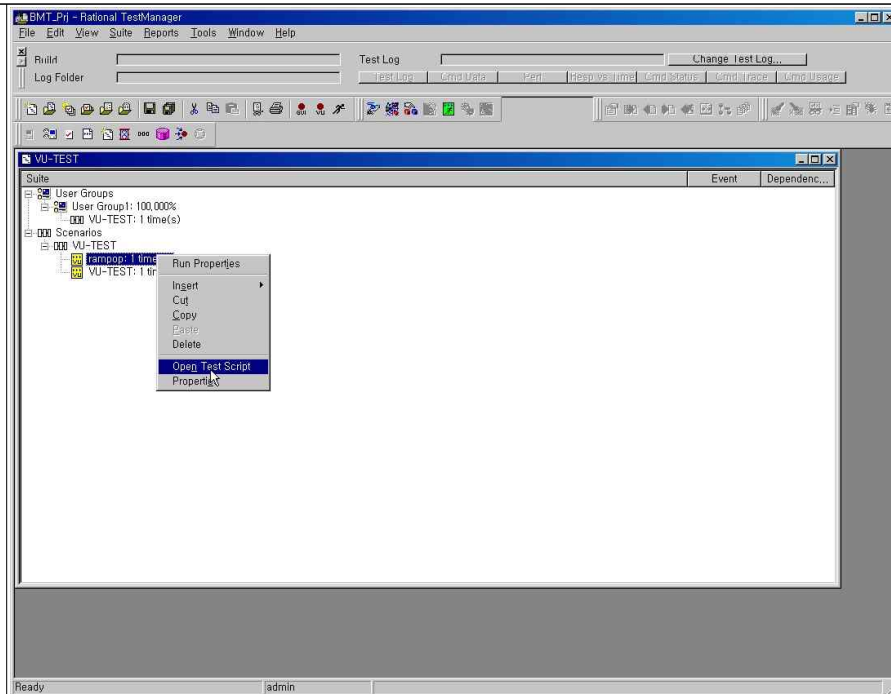




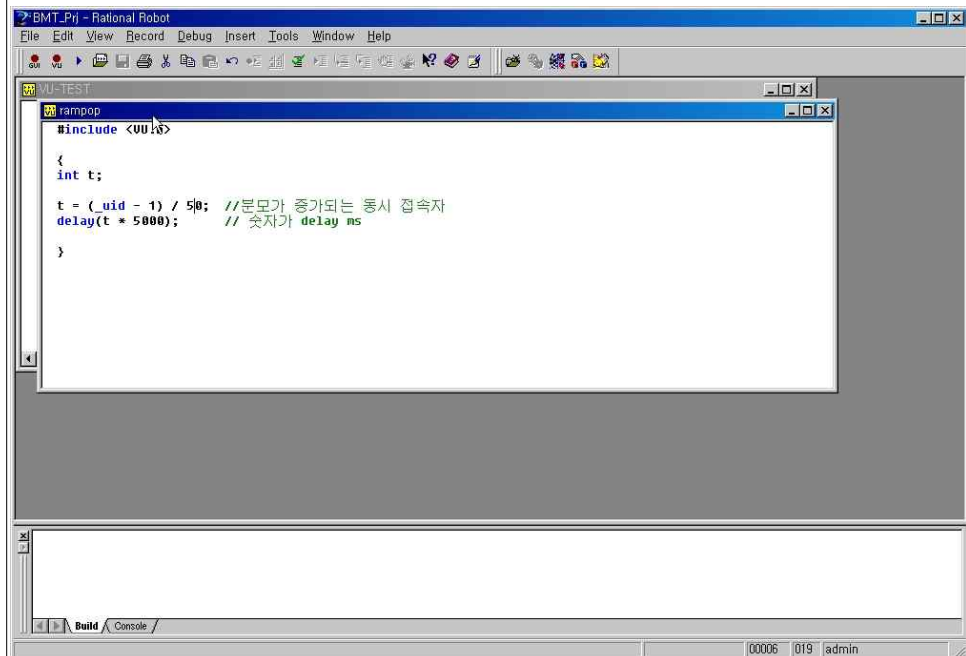
- 시나리오에 rampop 스크립트가 추가된 것을 확인할 수 있음



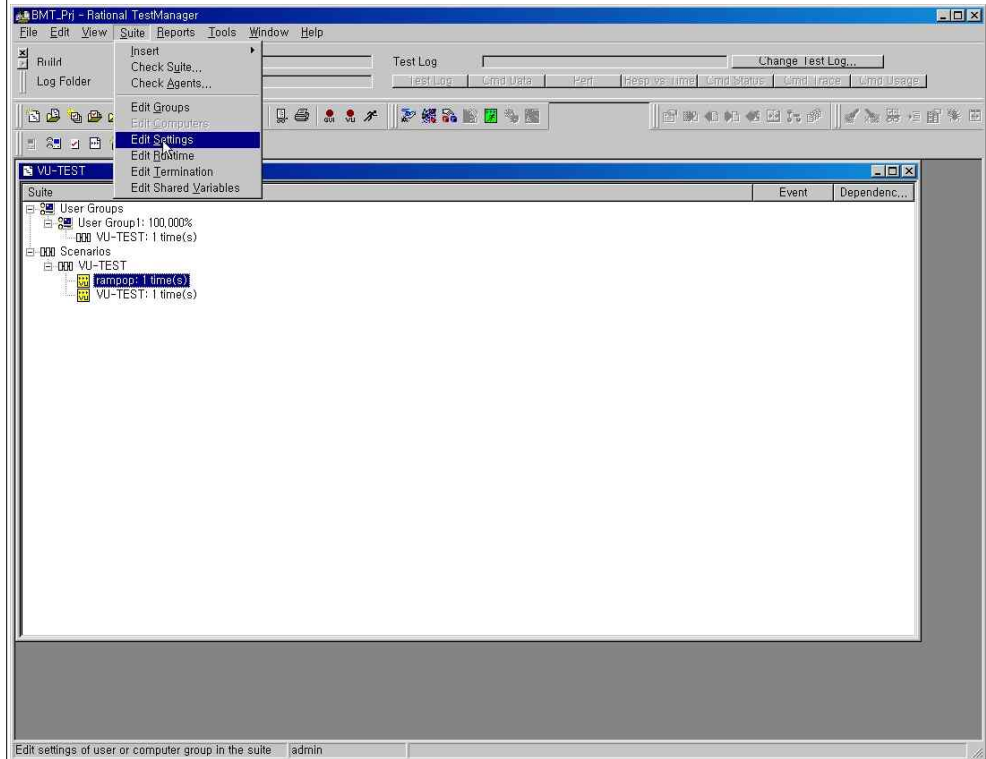
- rampop 스크립트를 수정합니다. (동시접속자 수를 지정후 저장)
- rampop 스크립트 -> Open...Script클릭



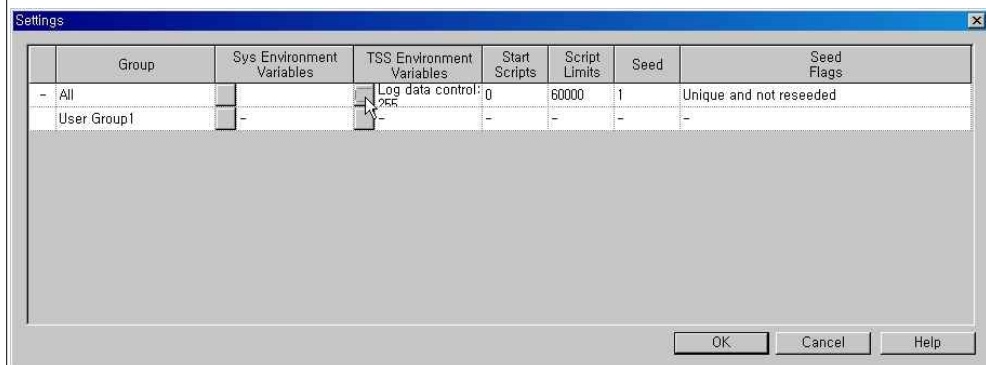
- 여기에선 50으로 수정(동시접속자를 50으로 정의)



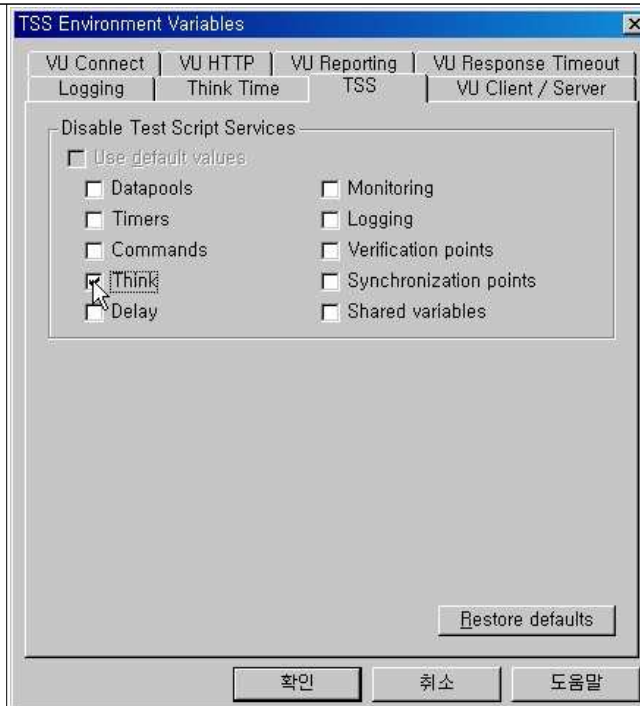
- think 옵션을 체크한다 (빠른 테스트를 위해 thinking 하는 시간을 뺌)
- Suite메뉴에서 Edit Scripts 클릭



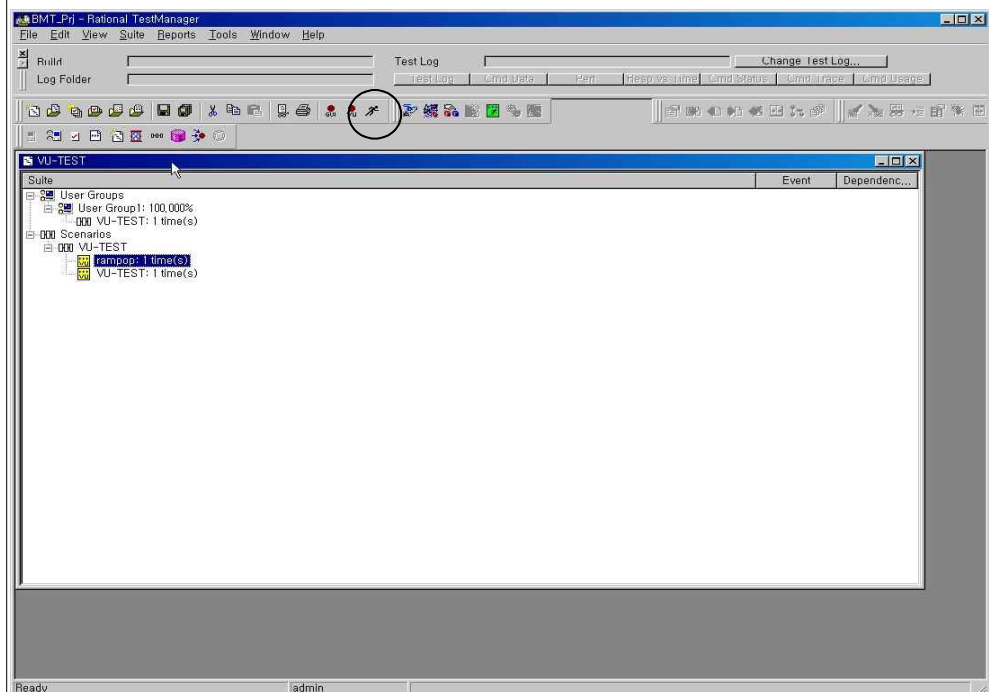
- TSS Env... 클릭



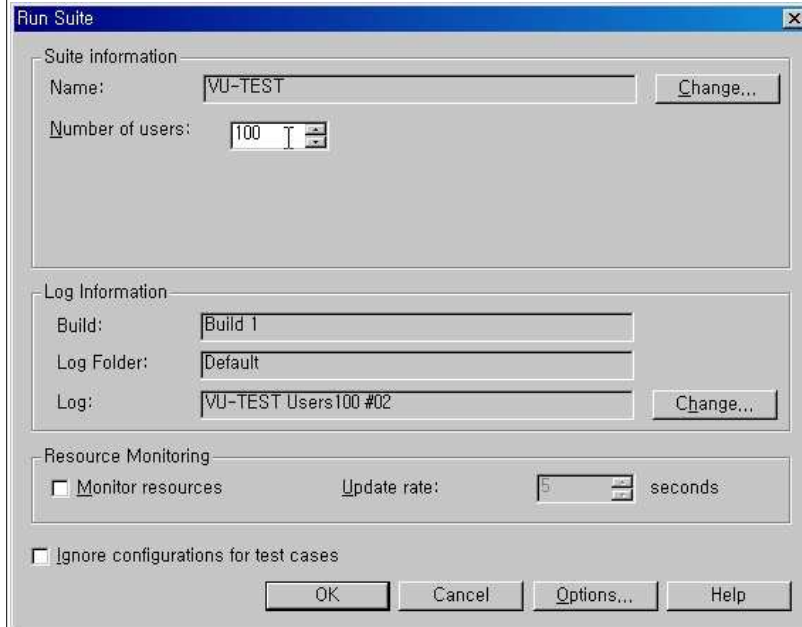
- Think 체크후 확인버튼 클릭



- 이제 시나리오에 대해서 실행
- 사람달리는 모습의 아이콘을 클릭



- 가상유저수를 지정후 OK 버튼 클릭 (여기선 100으로 지정)

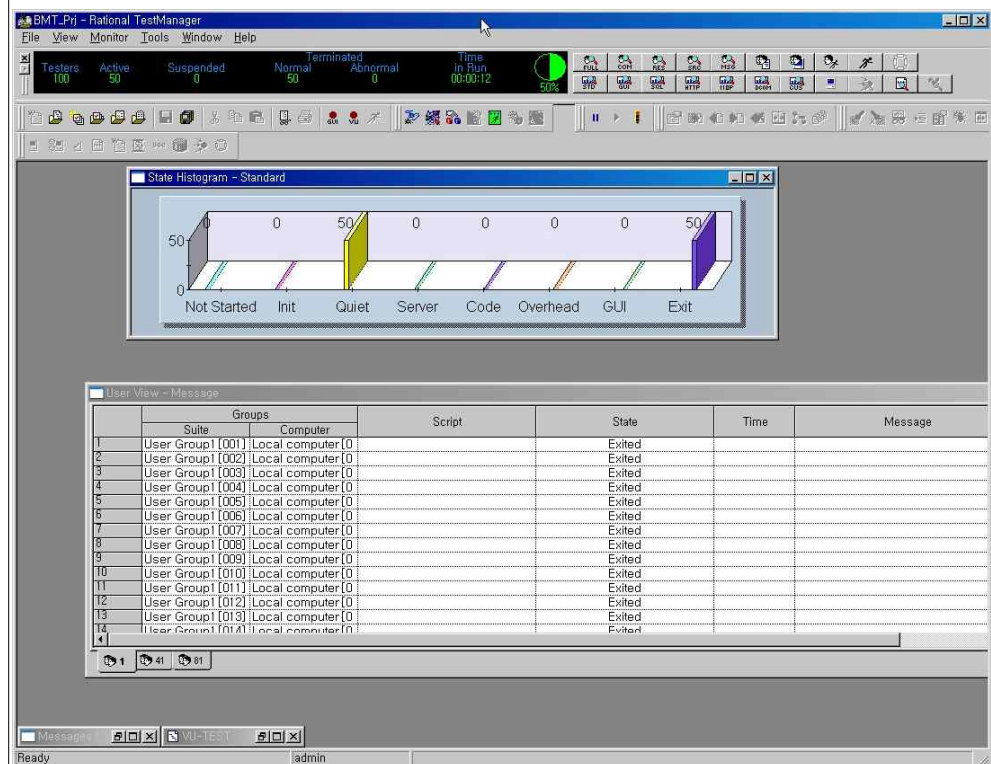


The 'Run Suite' dialog box is shown with the following settings:

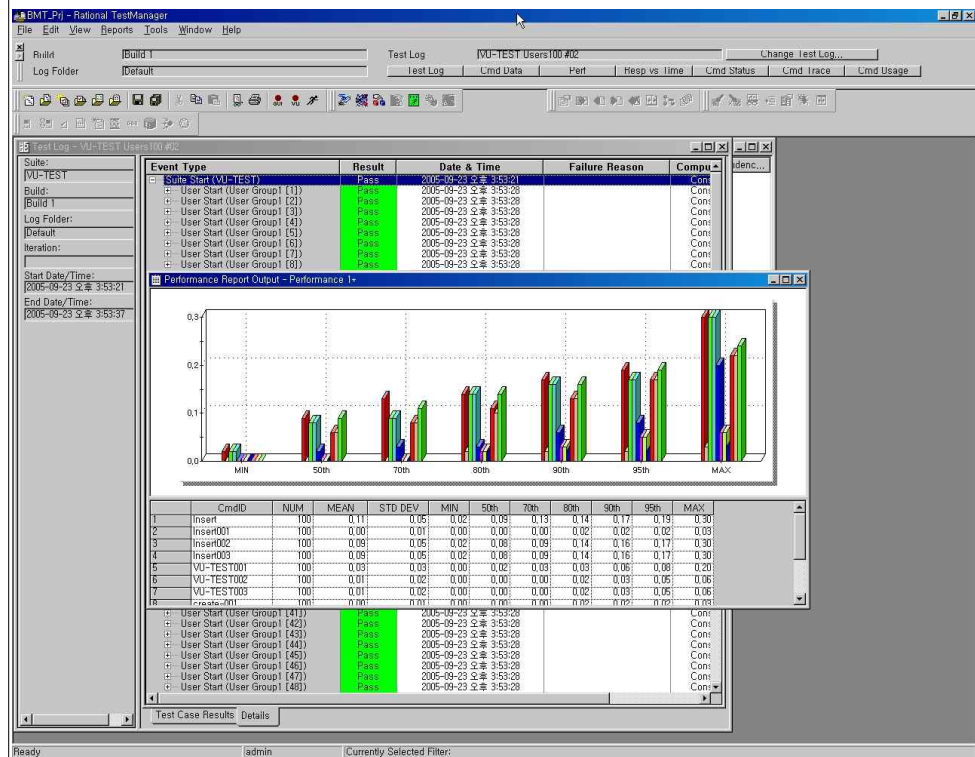
- Suite information:**
 - Name: VU-TEST
 - Number of users: 100
- Log Information:**
 - Build: Build 1
 - Log Folder: Default
 - Log: VU-TEST Users100 #02
- Resource Monitoring:**
 - ☐ Monitor resources
 - Update rate: 5 seconds
 - ☐ Ignore configurations for test cases

Buttons at the bottom: OK, Cancel, Options..., Help.

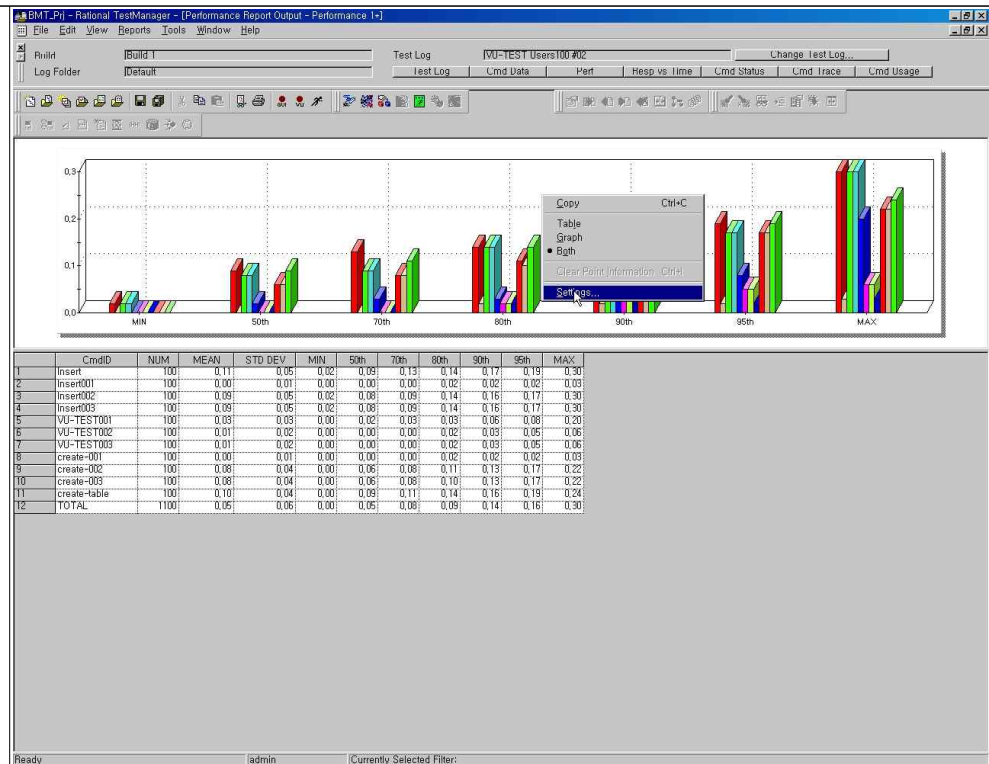
- 가상 100유저에 대해서 테이블 생성, INSERT 에 대해 성능테스트를 진행



- 아래와 같은 성능 그래프가 출력됨



- 위그래프는 스크립트내에 있는 모든 성능수치가 출력되므로..
create-table 및 insert 블록에 대해서만 성능을 보기위해 편집한다
- 그래프위에 마우스 클릭 setting 클릭



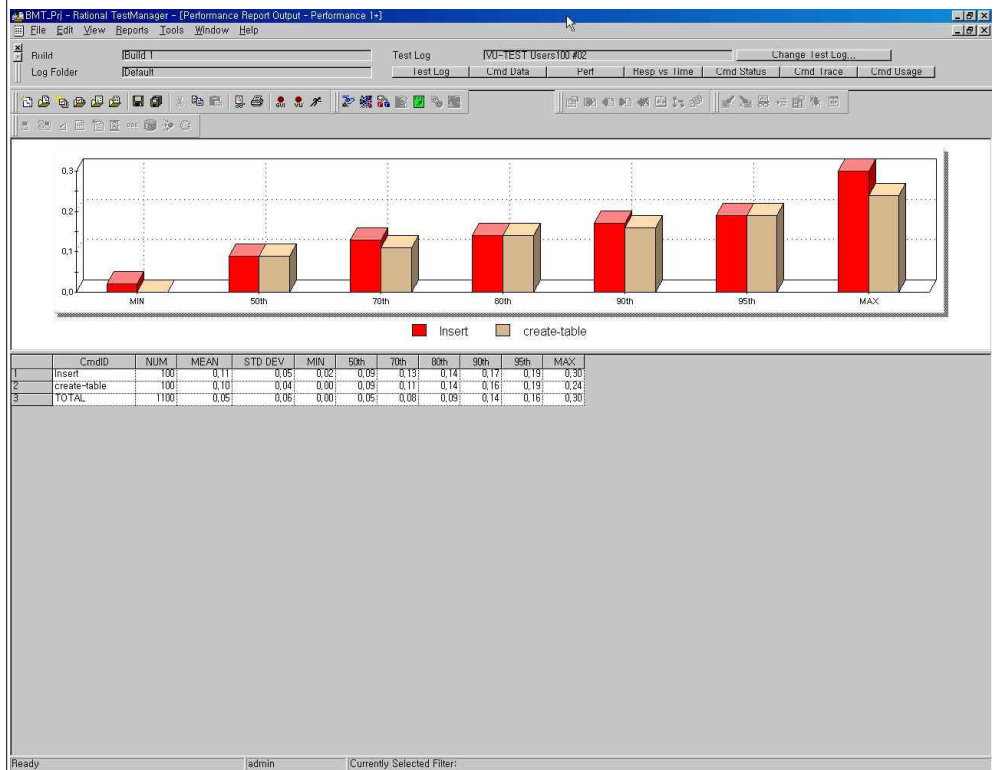
- 그래프 스타일 체크



- Select Command IDs에서 create-table 및 Insert만 선택한다



- create-table 및 Insert에 대해서만 성능결과가 출력됨



시험결과

비 고