

Upotreba IBM Rational Suite 7.0.1 alata za testiranje softvera

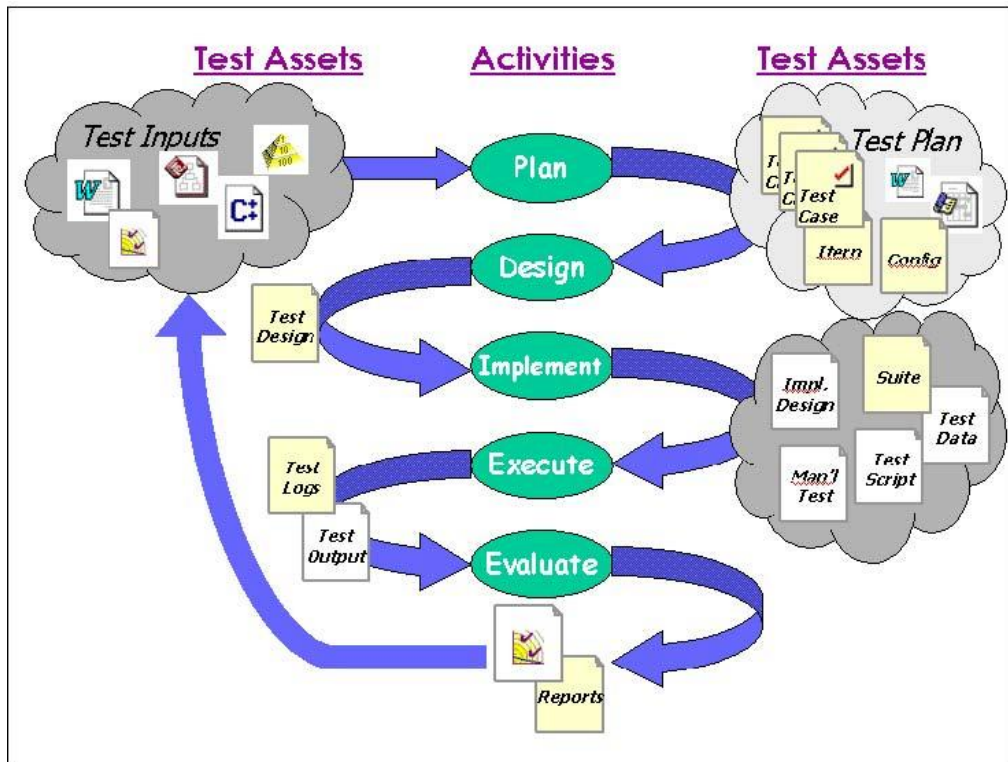
Sadržaj

1.	Uvod	3
1.1	Aktivnosti testiranja	3
1.1.1	Planiranje testiranja.....	3
1.1.2	Dizajniranje Testova.....	3
1.1.3	Implementacija testova	4
1.1.4	Izvršavanje testova	4
1.1.5	Evaluacija testova	4
1.2	Korišćenje alata u aktivnostima testiranja	4
2.	Rad sa alatom Rational Administrator.....	6
3.	Rad sa alatom Rational TestManager.....	8
3.1	Pokretanje alata	8
3.2	Glavni elementi radnog okruženja.....	8
3.3	Definisanje plana testiranja	9
3.4	Definisanje ulaza testiranja	9
3.5	Dizajniranje test primera (test cases).....	11
3.6	Implementacija test primera	14
3.7	Izvršavanje test primera	16
3.8	Izveštaji aktivnosti testiranja.....	18
4.	Rad sa alatom Rational Robot	19
4.1	Podешavanje za rad	19
4.2	Kreiranje test skriptova.....	19
4.3	Izvršavanje test skriptova	21
5.	Rad sa alatom Rational ManualTest.....	22
5.1	Kreiranje test skriptova.....	22
5.2	Izvršavanje test skriptova	22
6.	Rad sa alatom Rational SiteCheck	23
7.	Evaluacija testiranja.....	25

1. Uvod

1.1 Aktivnosti testiranja

Testiranje se sastoji od sledećih aktivnosti: Planiranje testiranja, dizajn testova, implementacija, izvršavanje i evaluacija testova. Svaka od ovih aktivnosti ima ulazne i izlazne artefakte. Sledeća slika ilustruje aktivnosti i artefakte testiranja.



1.1.1 Planiranje testiranja

U okviru ove aktivnosti cilj je da se odgovori na sledeća pitanja:

- Šta i zašto se testira? – Zahtevi, modeli i drugi ulazi testiranja govore o tome šta će se testirati i šta je cilj testiranja.
- Gde se testira? To jest, na kojoj hardverskoj i softverskoj platformi treba izvršavati testove?
- Kada se testira? – Planovi po iteracijama govore o tome u kojoj fazi razvoja određeni testovi su planirani za sprovođenje.
- Ko testira? – Ko sprovodi koju aktivnost testiranja.

1.1.2 Dizajniranje Testova

Cilj ove aktivnosti je da pruži odgovor na pitanje: kako će se sprovesti testiranje. Rezultat ove aktivnosti (artefakt) je test primer. Tokom dizajniranja test primera specificira se sledeće:

- Koraci (na višem nivou apstrakcije) koji su potrebni u interakciji sa aplikacijom i celim sistemom u svrhu sprovođenja testa.
- Način provere da li je test uspešno izvršen.
- Preduslovi i posledice (post-uslovi) za posmatrani test.

1.1.3 Implementacija testova

Ova aktivnost obuhvata pravljenje višestruko upotrebljivih test skriptova koji realizuju test primere, dakle svaki skript se povezuje sa nekim test primerom. Implementaciju je moguće izvršiti na različite načine, na primer za manuelno testiranje i automatizovano testiranje, i u zavisnosti od tehnologije na kojoj je realizovana aplikacija koja se testira.

1.1.4 Izvršavanje testova

Ova aktivnost podrazumeva izvršavanje implementacije testa da bi se proverilo da li sistem korektno realizuje određenu funkcionalnost (ili da bi se sagledale performanse i slično).

1.1.5 Evaluacija testova

Aktivnost procene testova uključuje:

- Određivanje validnosti izvršavanja testa. Da li je izvršen do kraja. Da li je otkazao jer nisu zadovoljeni preduslovi?
- Analiza izlaza testiranja. Na primer, u testiranju performansi, cilj je da se na osnovu podataka zaključi da li su performanse sistema prihvatljive.
- Pregled zbirnih rezultata da bi se proverila pokrivenost u odnosu na plan testiranja, ulaze testiranja, konfiguracije itd. Ova vrsta izveštaja može da posluži za ocenu toka testiranja i analizu trendova.
- Uticaj promene zahteva i drugih ulaza testa na plan testiranja. Na primer, ako se neki zahtev izmeni, koje test primere u planu treba proveriti i ažurirati u skladu sa izmenom zahteva.

1.2 Korišćenje alata u aktivnostima testiranja

Postoji višestruka korist od korišćenja alata u procesu testiranja:

- sistematika procesa: alati "vode" kroz određene aktivnosti i na sistematizovan način hijerarhijski organizuju i memorišu sve neophodne artefakte
- poboljšano upravljanje: u svakom trenutku moguće je proceniti u kom procentu je testiranje obavljeno i šta je još ostalo da se uradi
- produktivnost: automatizovano testiranje obezbeđuje da se veći broj testova sprovodi u jedinici vremena nego manuelno
- ponovljivost testa (podrška regresionom testiranju): kroz formalno implementiranje testova kroz test skripte obezbeđeno je da se test više puta može ponoviti pod identičnim uslovima

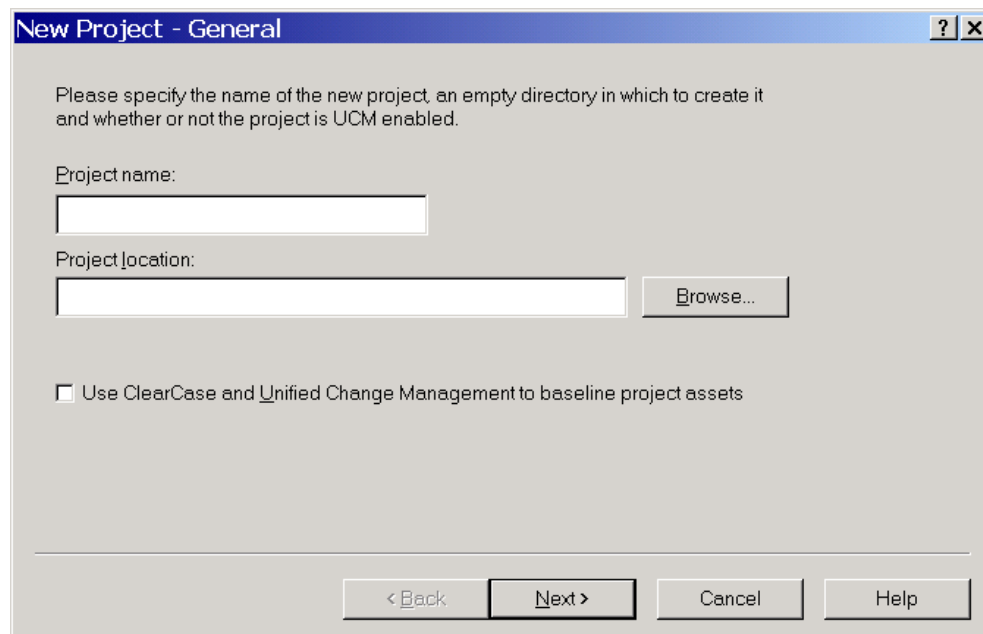
Firma Rational nudi vrlo razrađeno rešenje automatizacije procesa testiranja u vidu niza alata različitih namena, koji saraduju međusobno i sa ostalim razvojnim alatima za druge faze razvojnog ciklusa softvera:

- Rational Administrator – služi za administraciju repozitorijuma svih artefakata koji se odnose na projekat i aktivnost testiranja, pravljenje, ažuriranje projekata, dodela prava pristupa pojedinim članovima tima itd.
- Rational TestManager – prvenstveno se koristi za organizaciju i upravljanje procesom testiranja, pravljenje test planova, dizajn testova i evaluaciju rezultata testiranja
- Rational Robot – implementacija testa kroz automatizovano funkcionalno i performansno testiranje zasnovano na test skriptovima, sa funkcijom snimanja korisnikovih akcija nad aplikacijom i autogenerisanja skripta.
- Rational TestFactory – analiziranje strukture aplikacija (prvenstveno gui, java, c++, vb) rada podrške implementacije testa kroz automatizovano generisanje test skriptova
- Rational QualityArchitect – specijalizovan za testiranje komponentnih tehnologija srednjeg sloja, EJB i COM
- Rational ManualTest – alat za implementaciju manuelnog testa
- Rational SiteCheck – alat za ispitivanje i kontrolisanje strukture i funkcije web aplikacija
- Rational Purify – detektuje greške u izvršavanju i curenje memorije
- Rational Quantify – sakupljanje i analiza podataka o performansama
- Rational PureCoverage – podaci o pokrivenosti koda (white box testiranje)

U narednim poglavljima slede uputstva za korišćenje pojedinih alata (osnovni scenariji) za sprovođenje opisanih aktivnosti testiranja.

2. Rad sa alatom Rational Administrator

Na početku aktivnosti razvoja i testiranja potrebno je napraviti repozitorijum u kome se čuvaju svi artefakti. Startuje se Rational Administrator i navede ime projekta i lokacija (prazan direktorijum), a zatim opciona lozinka:



New Project - General

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.

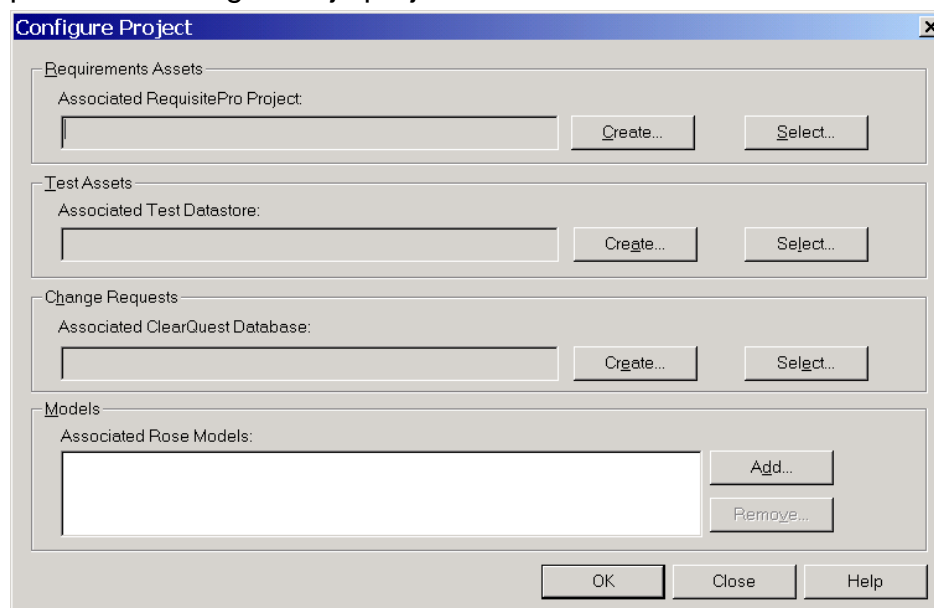
Project name:

Project location:

Use ClearCase and Unified Change Management to baseline project assets

< Back Next > Cancel Help

Klikom na finish, kreira se projekat će biti kreiran sa ekstenzijom .rsp i odmah prelazi na konfigurisanje projekta:



Configure Project

Requirements Assets
Associated RequisitePro Project:

Test Assets
Associated Test Datastore:

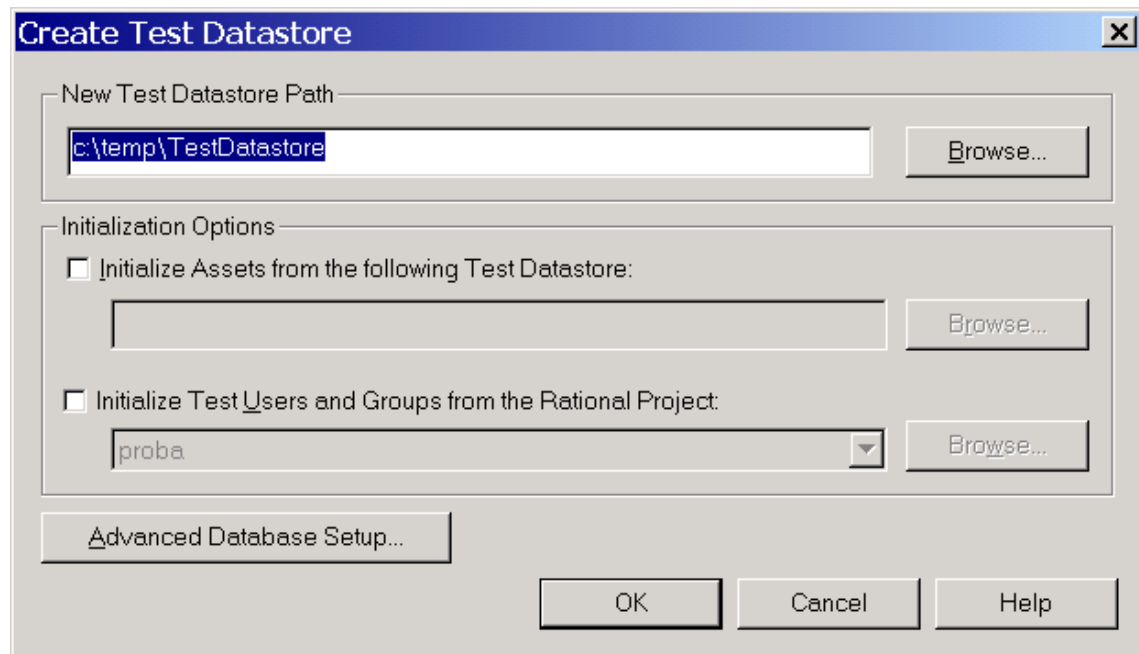
Change Requests
Associated ClearQuest Database:

Models
Associated Rose Models:

OK Close Help

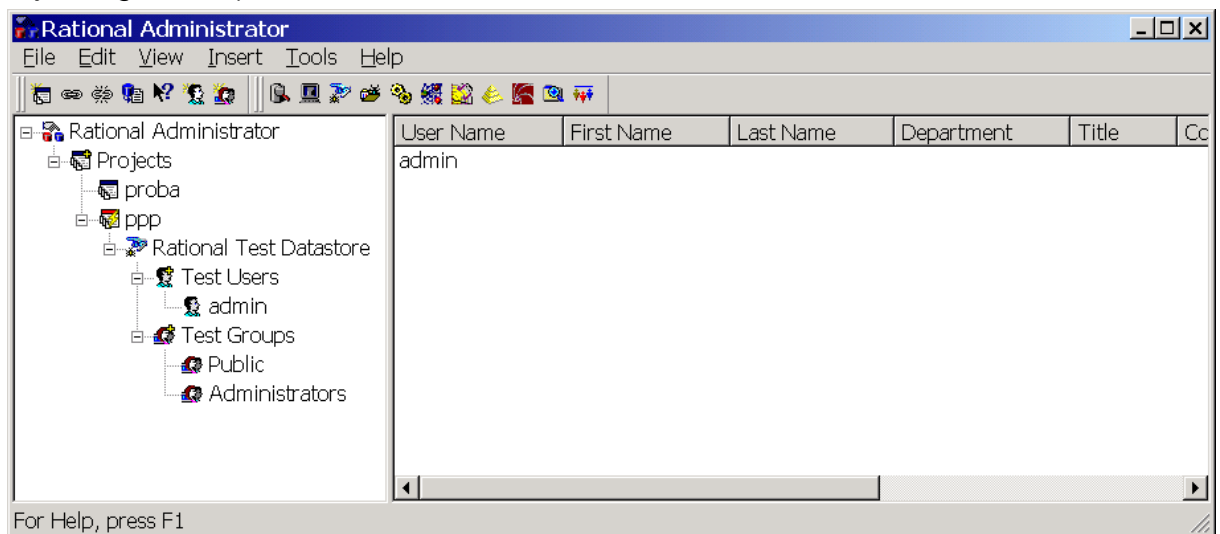
Projekat sadrži najviše opciono RequisitePro bazu (korisnički zahtevi), opciono ClearQuest bazu (zahtevi za izmenama i prijave bagova) i jedan Test Datastore (bazu test artefakata) i proizvoljan broj Rose modela.

Potrebno je samo kreirati Test Datastore klikom na odgovarajuće Create dugme, pri čemu se dobija dijalog:



Klikom na OK (pošto podrazumevane vrednosti, da se DataStore kreira u istom direktorijumu gde i projekat, da se ne importuju asseti i korisnici iz drugog projekta i da se koristi Access baza zadovoljavaju) inicira se kreiranje DataStore-a, čime je završen proces kreiranja repozitorijuma.

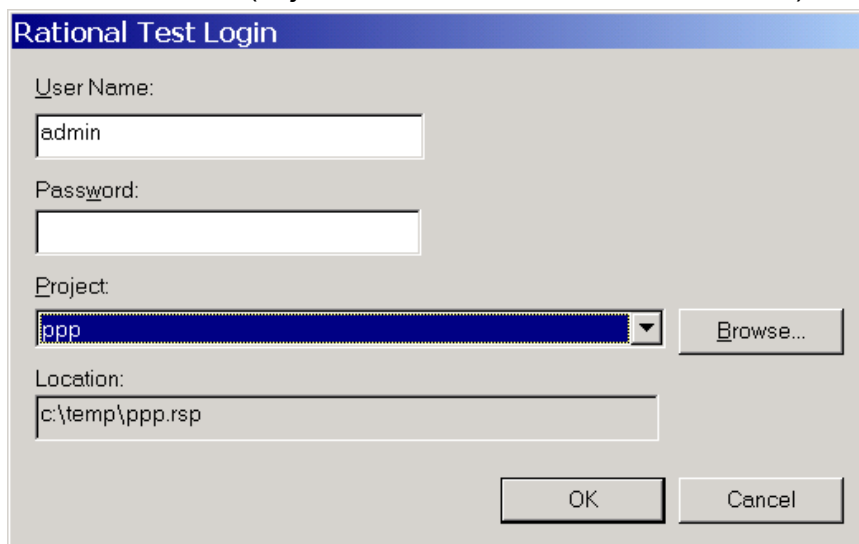
Dobija se osnovni ekran Administratora, u kome je moguće (desnoklikom na projekat), povezati se (Connect) na kreirani projekat i po potrebi formirati korisnike i zadavati prava pristupa (inicijalno postoji jedan test korisnik, admin, što je dovoljno za pokretanje drugih alata).



3. Rad sa alatom Rational TestManager

3.1 Pokretanje alata

Po startovanju ovoga alata, potrebno je izabrati projekat iz liste i zadati korisničko ime i lozinku (koji su definisani alatom Administrator):

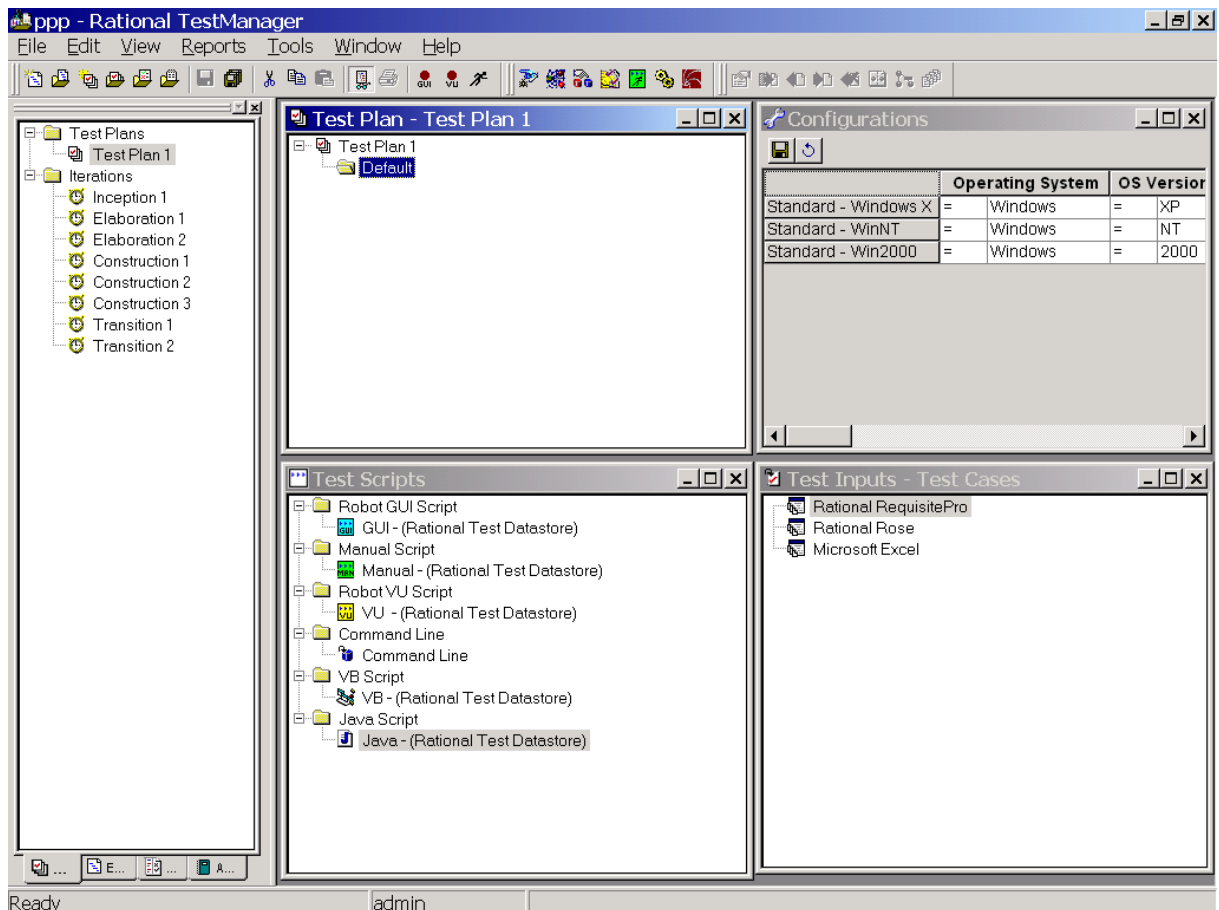


The screenshot shows a 'Rational Test Login' dialog box. It has a title bar with the text 'Rational Test Login'. Below the title bar, there are four input fields: 'User Name:' with the text 'admin' entered, 'Password:' (empty), 'Project:' with a dropdown menu showing 'ppp' and a 'Browse...' button to its right, and 'Location:' with the text 'c:\temp\ppp.rsp' entered. At the bottom of the dialog are 'OK' and 'Cancel' buttons.

3.2 Glavni elementi radnog okruženja

Radno okruženje prikazano na slici dobija se po startovanju alata, i posle zadavanja komandi Test Plan 1/Open (desnoklik na Test plan 1 i open), View/Test Inputs, View/Configurations i View/Scripts. Usidreni prozor sa leve strane je prozor radnog prostora (Test Assets Workspace).

Test plan predstavlja hijerarhijski uređenu (po folderima) kolekciju test primera (engl. test case). Test primeri su u relaciji sa određenim iteracijama razvojnog ciklusa, da bi se znalo u koje vreme se ti testovi moraju izvršavati. Konfiguracije definišu određene hardversko/softverske platforme na kojima se određeni test primeri moraju izvršavati. Test primeri, koji služe u fazi dizajna testova (opis na višem nivou apstrakcije) se konkretizuju, odnosno implementiraju pravljenjem odgovarajućih test skriptova. Ulazi testiranja (Test inputs) su relevantna dokumenta i modeli na osnovu kojih se razvijaju određeni test primeri. Radi se o traceability relaciji (relacija praćenja) koja služi za analizu uticaja (impact analysis) promena zahteva i funkcionalne specifikacije na test primere.



3.3 Definisane plana testiranja

U prozoru radnog prostora (Test Asset Workspace) desnoklikom na Test Plans, izborom stavke New Test Plan, kreira se novi plan testiranja (inicijalno već postoji Test Plan 1).

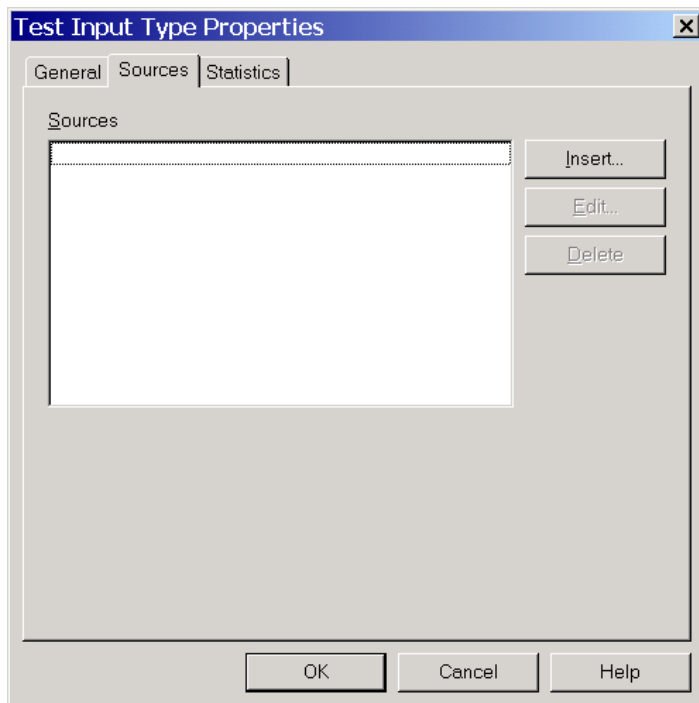
Pre ažuriranja test plana, potrebno je otvoriti njegov prozor. To se radi selektovanjem plana u prozoru radnog prostora i duplim klikom na njega.

Test plan se prikazuje kao hijerarhijska kolekcija Test Case Foldera (inicijalno postoji prazan default folder). Folderi služe za logičko grupisanje test primera, na primer, prema članovima tima koji su zaduženi za njih, ili prema podsistemima tj. funkcionalnim grupama aplikacije. Folderi se ubacuju desnoklikom na čvor u listi i izvorom stavke Insert Test Case folder.

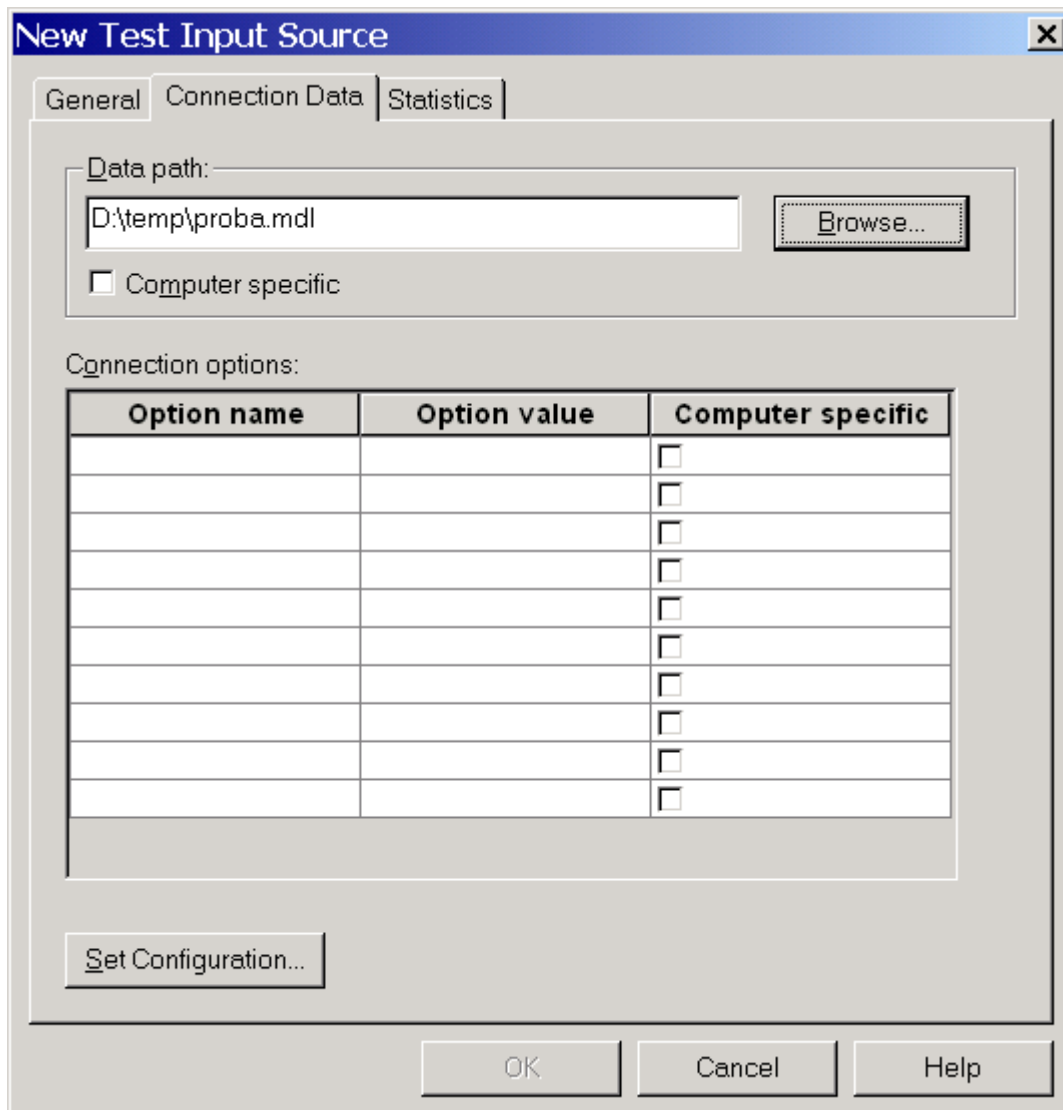
Iz prozora test plana takođe je moguće test plan povezati sa određenom iteracijom razvojnog ciklusa, ili sa određenom hw/sw konfiguracijom. Tada će svi test primeri u tom planu to naslediti iz plana. Alternativno, iteraciju i konfiguraciju je moguće dodeliti pojedinim test primerima.

3.4 Definisane ulaza testiranja

Ukoliko se, na primer, želi određeni Rose model definisati kao ulaz testiranja (čime se omogućava da se pojedini elementi modela, na primer, use case-ovi, vezuju za određene test primere) potrebno je sa View/Test inputs otvoriti prozor ulaza testiranja u radnoj oblasti, zatim desnoklikom na Rational Rose i izborom iz popup menija stavke Properties, otvoriti sledeći dijalog i izabrati tab sources:



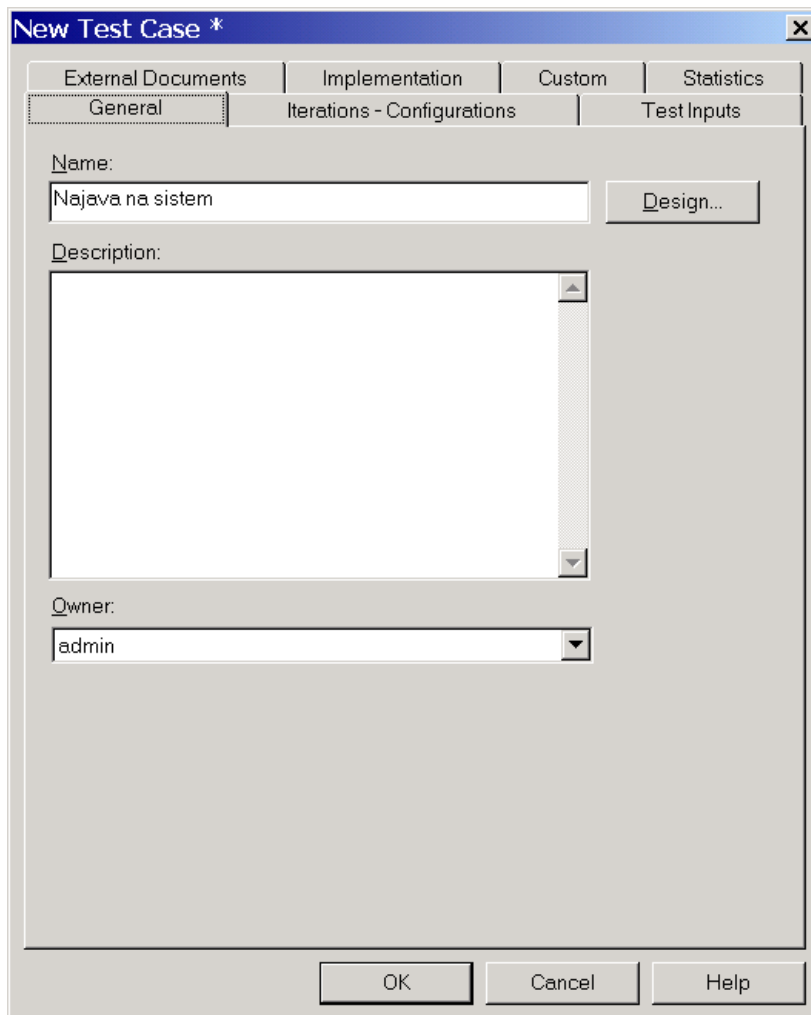
Zatim kliknuti na Insert, u dobijenom dijalogu (tab General) uneti ime ovog izvora (npr. ime Rose modela), zatim odabrati tab Connection data, kao na slici:



U polju Data path uneti ime Rose fajla sa modelom i kliknuti na dugme OK, čime se ovaj model dodaje u izvore testiranja.

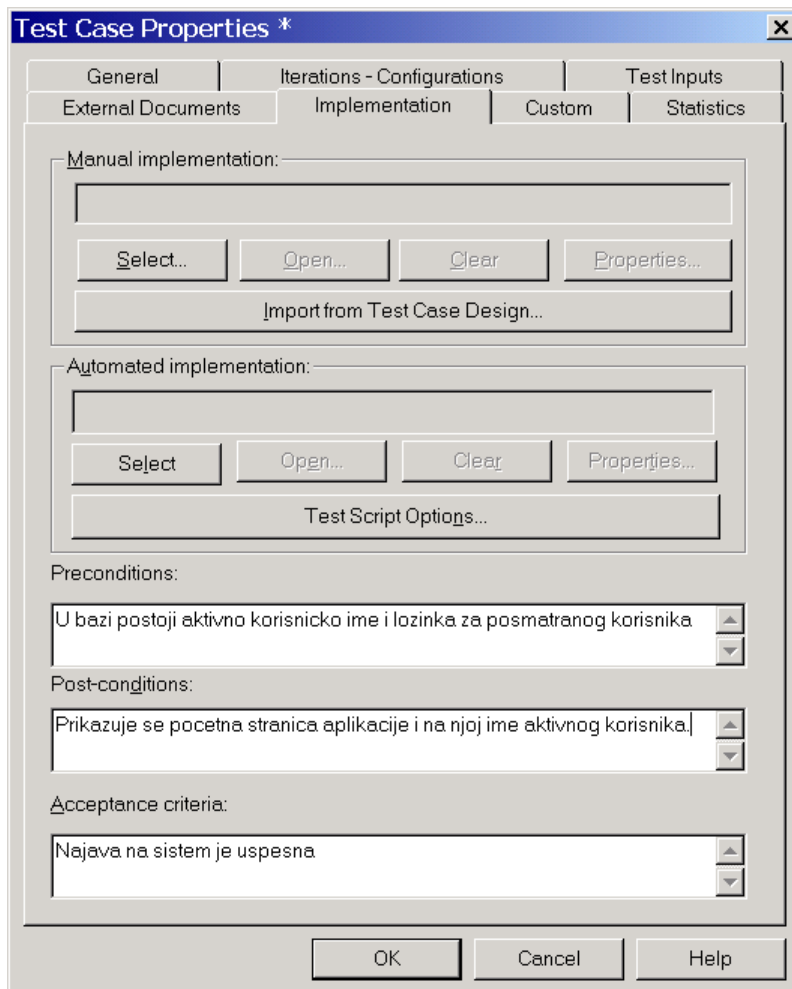
3.5 Dizajniranje test primera (test cases)

U prozoru test plana, klikom na čvor u listi i izborom stavke Insert Test Case, otvara se Property dijalog za definisanje novog test primera:



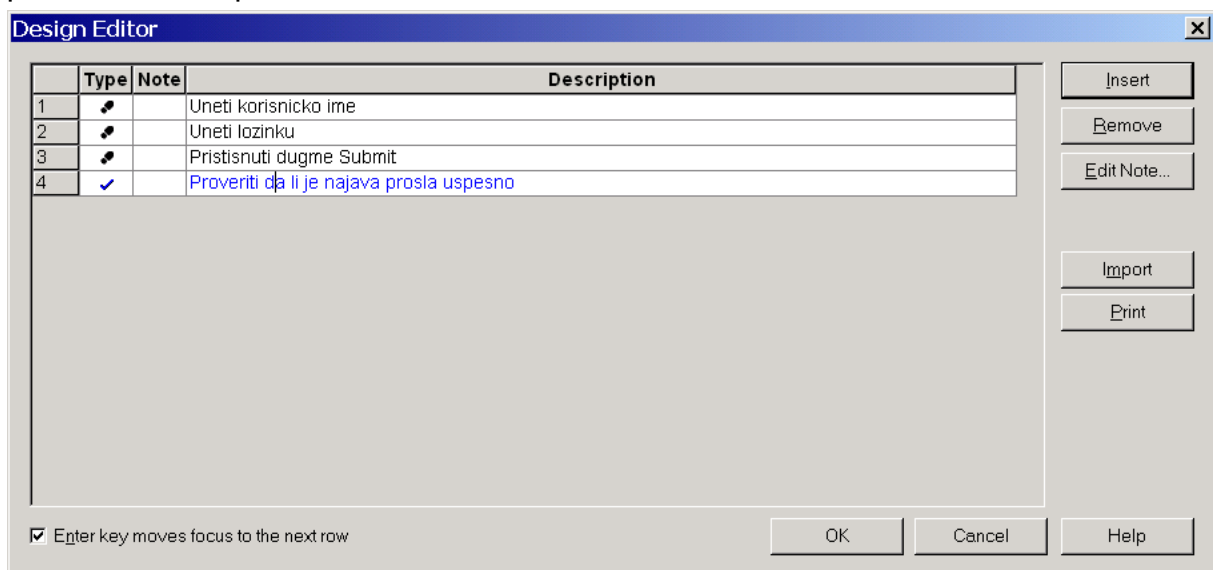
Potrebno mu je zadati ime, a moguće ga je vezati za određene iteracije, konfiguracije, ulaze testiranja (npr. određeni use case u modelu), izborom odgovarajućih tabova dijaloga. Kratak opis se može navesti, (npr. na koju funkcionalnost se odnosi i koji scenario se testira).

Tab implementation prikazan je na sledećoj slici:



U fazi dizajniranja test primera, potrebno je popuniti polja preconditions, post-conditions i acceptance criteria.

Najzad, u okviru taba General, izborom dugmeta Design otvara se Design Editor za posmatrani test primer:



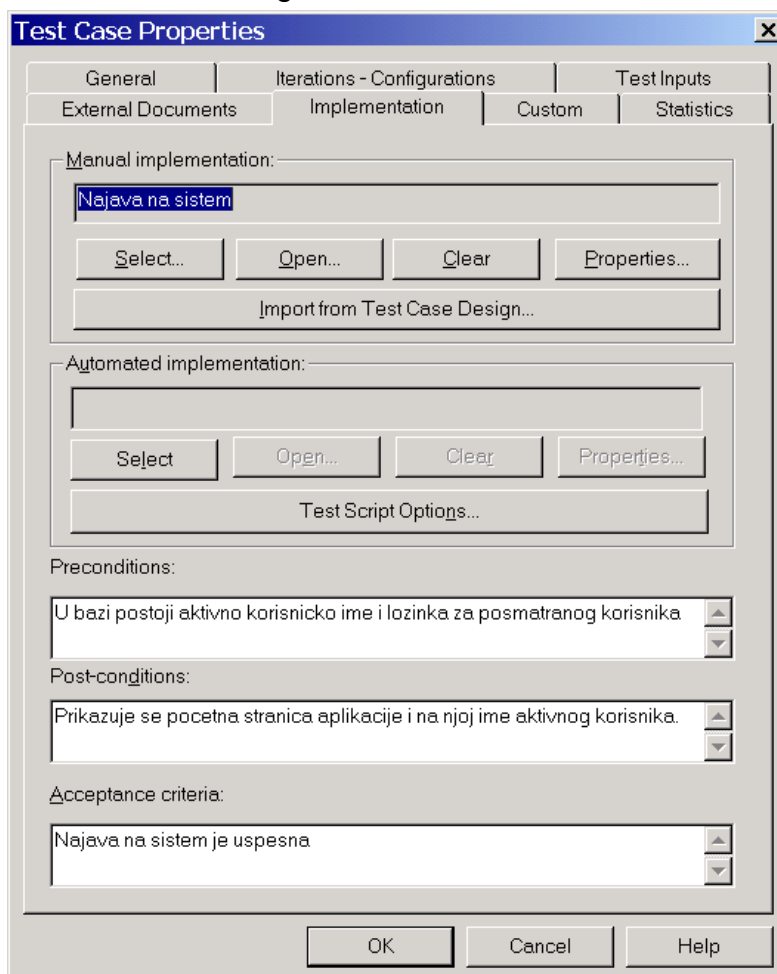
Za test primer definiše se niz koraka (izvršavanja) i verifikacionih tačaka koje su prikazane plavo (novi red se ubacuje pritiskom na enter). Verifikaciona tačka

specificira mesto na kome je potrebno izvršiti neku proveru stanja u posmatranoj aplikaciji.

3.6 Implementacija test primera

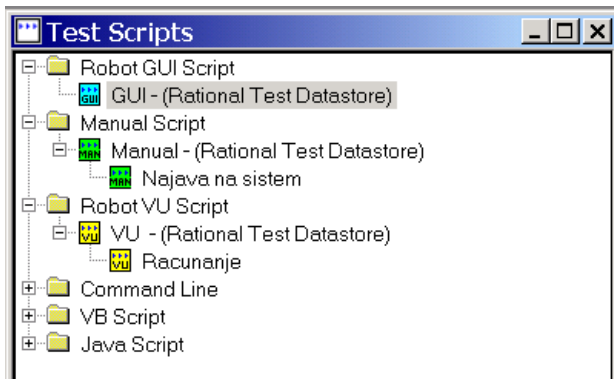
Implementacija test primera predstavlja njegovu konkretizaciju, sa ciljem da se on može izvršavati nad aplikacijom. Moguće je manuelno i automatizovano izvršavanje. U oba slučaja potrebno je definisati odgovarajući test skript (u programskom jeziku Rational alata, varijanti Ca ili Basica). U slučaju manuelnog izvršavanja, skript služi da prikazuje poruke čoveku koji testira aplikaciju i uzima od njega podatke. U slučaju automatizovanog izvršavanja, aplikacija se pod kontrolom test alata izvršava automatski i svi podaci i testiranja su automatizovani.

Skript manuelna implementacije može da se automatski napravi na osnovu dizajn informacija: selektuje se test primer, izabere properties, u dijalogu se izabere tab implementation i u sekciji dijaloga Manual implementation, izabere dugme Import from test case design.

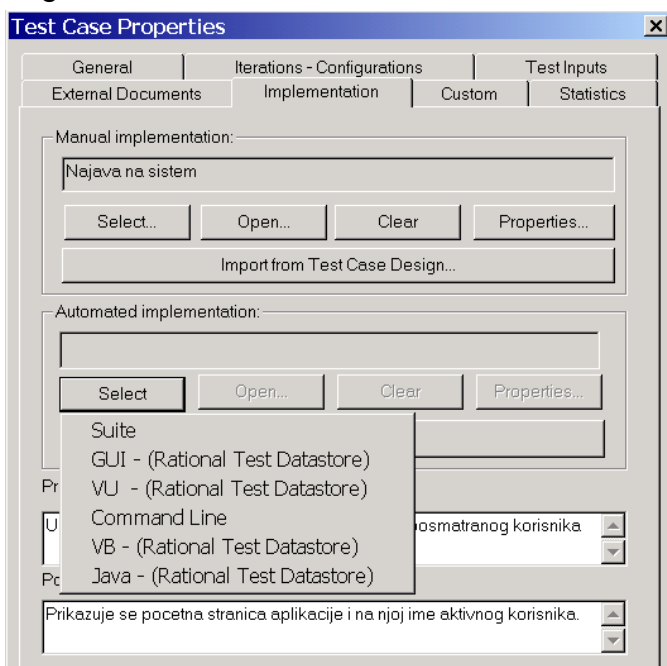


Time je kreiran manuelni skript (što se može proveriti i u Prozoru Test scripts radnog okruženja) i pridružen posmatranom test primeru.

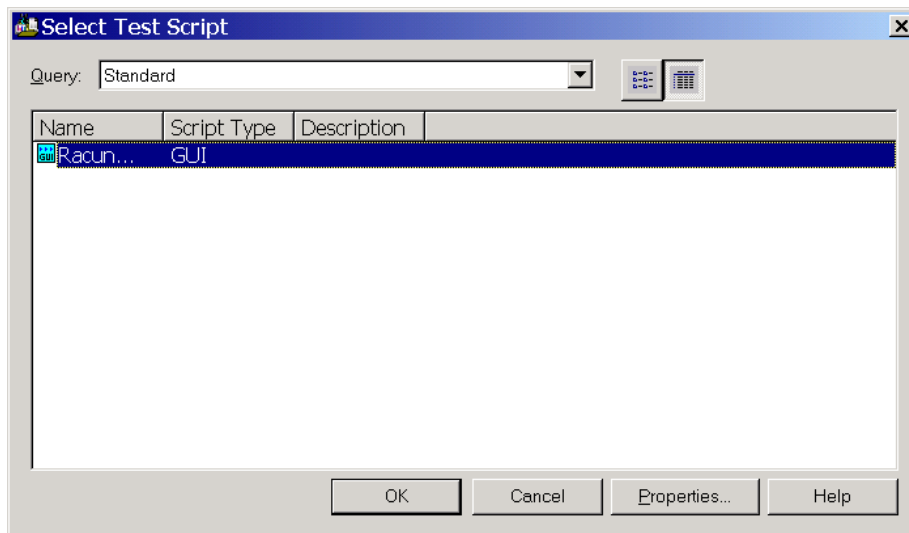
Automatizovani skript implementacije pravi se na sledeći način: iz menija se stavkom View/Test scripts otvori prozor sa skriptovima.



U prozoru se izabere odgovarajući Robot Datastore (GUI služi za testiranje grafičkog interfejsa, a VU za testiranje veze klijent-server i upita u bazu). Desnoklikom i izborom stavke New Test Script iz popup menija startuje se alat Rational Robot koji je opisan u sekciji 4. Kada se završi rad sa alatom Robot , treba otvoriti Test plan, izabrati Test Case kojem hoćemo da pridružimo napravljeni skript, izabere se stavka properties, tab Implementation. Zatim u delu Automated implementation treba izabrati dugme Select:



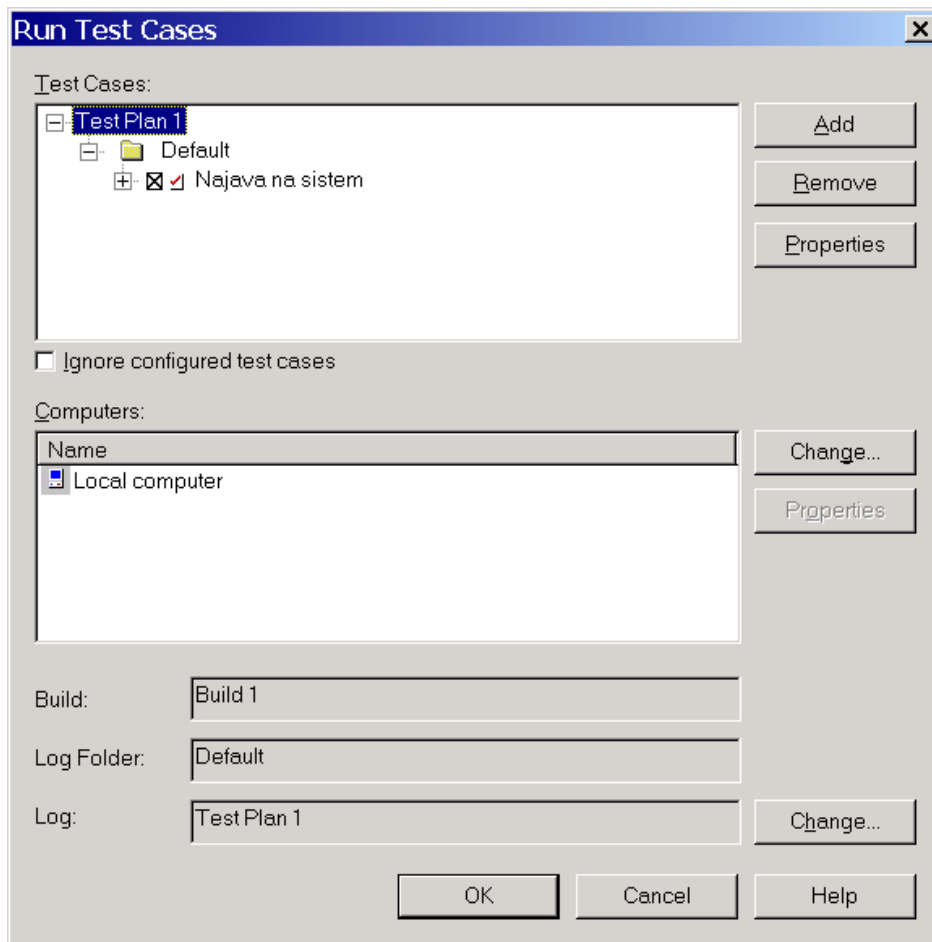
Iz menija odabrati GUI ili VU i u dijalogu koji se pojavi, izabrati novokreirani skript:



3.7 Izvršavanje test primera

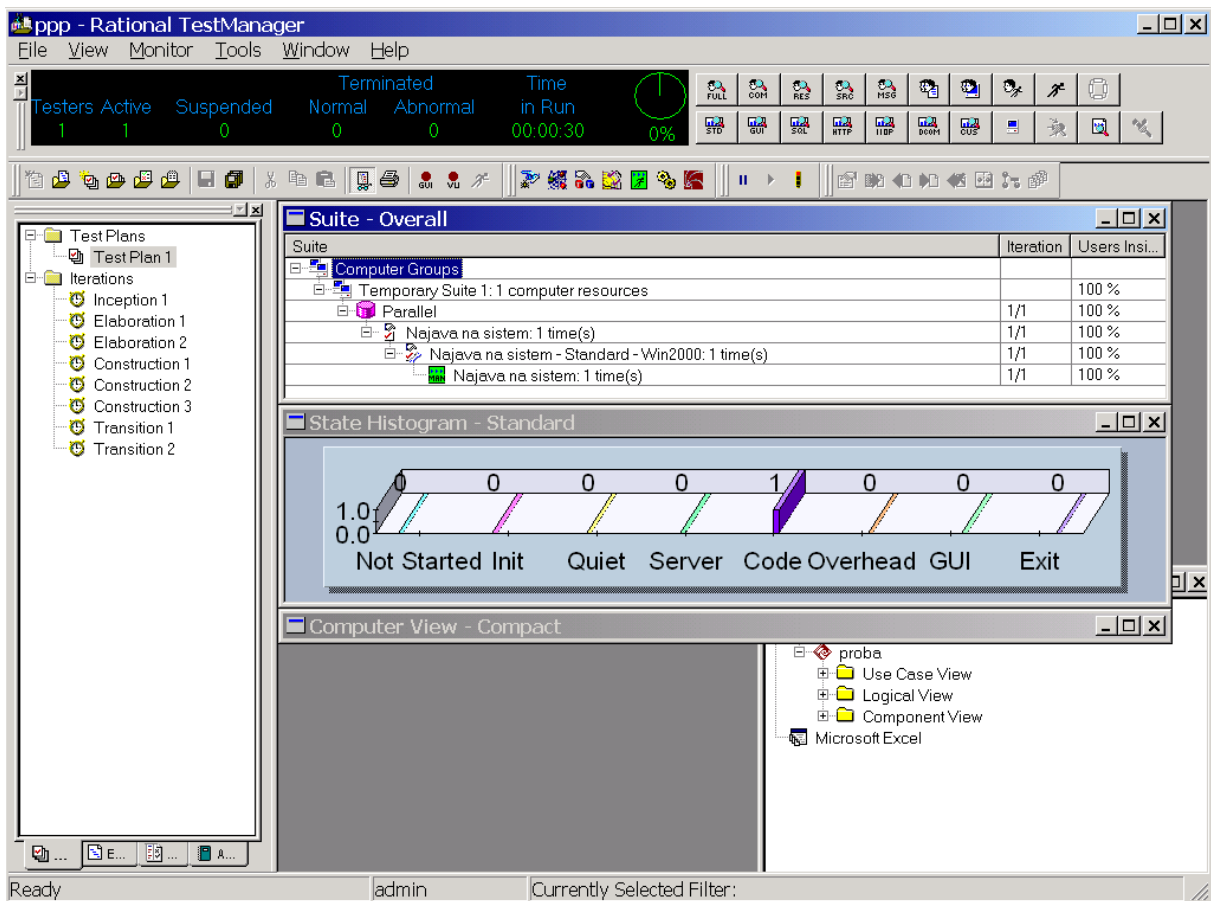
Izvršavanje se inicira ili tako što se u workspace prozoru desnoklikom na test plan izabere stavka Run, čime se biraju za izvršavanje svi test primeri u tom planu za određenu iteraciju, ili se može u prozoru test plana, desnoklikom na određeni test primer, pokrenuti samo njegovo izvršavanje. U oba slučaja dobija se sledeći dijalog, gde se prikazuje koji su test primeri odabrani i gde se može odabrati ili izbaciti neki od test primera za izvršavanje (checkbox levo od imena primera).

Može se odabrati neki od definisanih kompjutera na kome će se aplikacija izvršavati, koji build (verzija) te aplikacije će se izvršavati i u koji Log zapis će ići rezultati izvršavanja. Ovde u principu treba prihvatiti ponuđene inicijalne vrednosti i ne vršiti izmene.

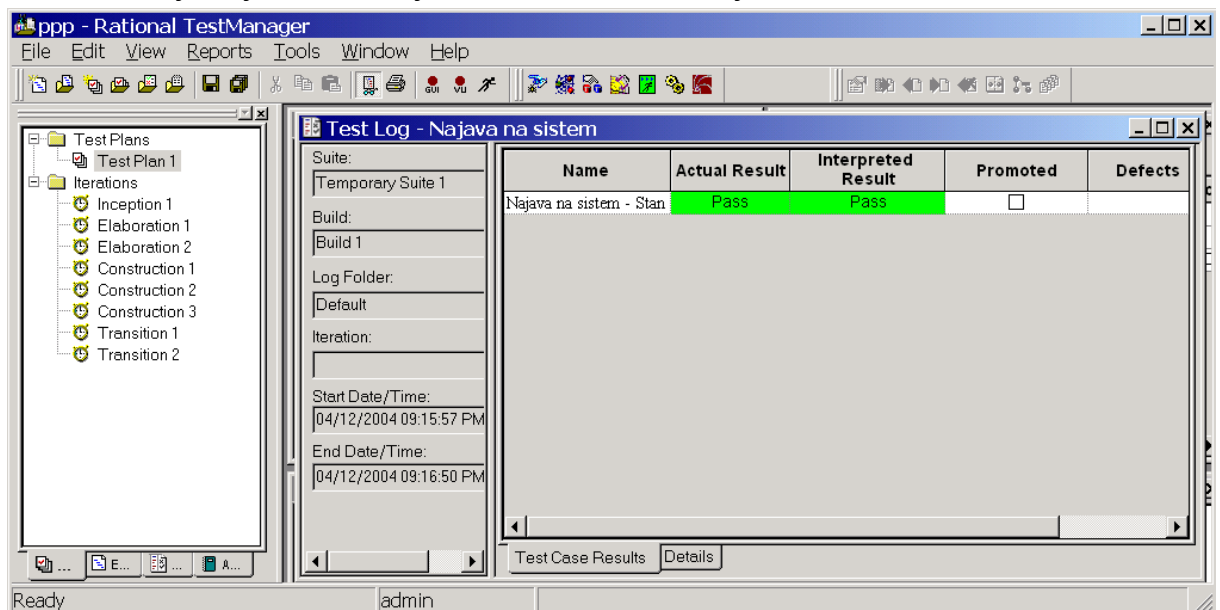


Pritiskom na OK, izvršavanje započinje. Kontrola se prenosi na jedan od alata Robot ili ManualTest, u zavisnosti od toga da li je skript manualan ili automatski. Rad sa ovim alatima opisan je u narednim odeljcima.

Za vreme izvršavanja test skriptova u TestManager-u prikazuju se podaci vezani za tekuću sesiju testiranja, kada je počelo, koliko ljudi testira, na kojim mašinama, histogram koji pokazuje u kojem stanju su testovi itd.



Posle završetka izvršavanja svih test primera, u TestManager-u prikazuje se Log prozor sa informacijama o izvršavanju testa. Treba checkirati u koloni promoted sve rezultate izvršavanja koji se žele trajno sačuvati za izveštaje.



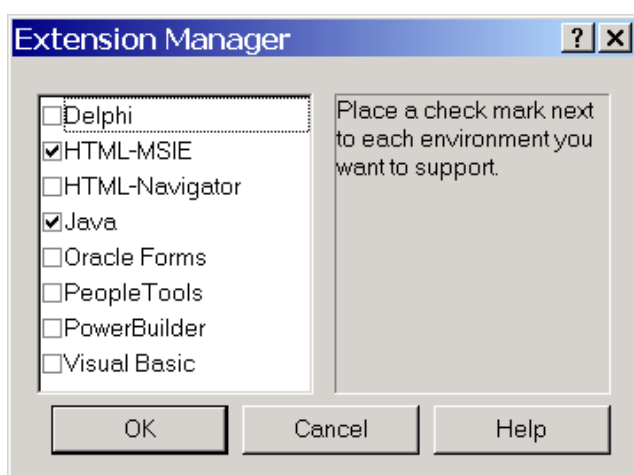
3.8 Izveštaji aktivnosti testiranja

U TestManageru moguće je dobiti veliki broj izveštaja koji se tiču procesa i artefakata testiranja. Najvažniji izveštaji opisani su u odeljku 7.

4. Rad sa alatom Rational Robot

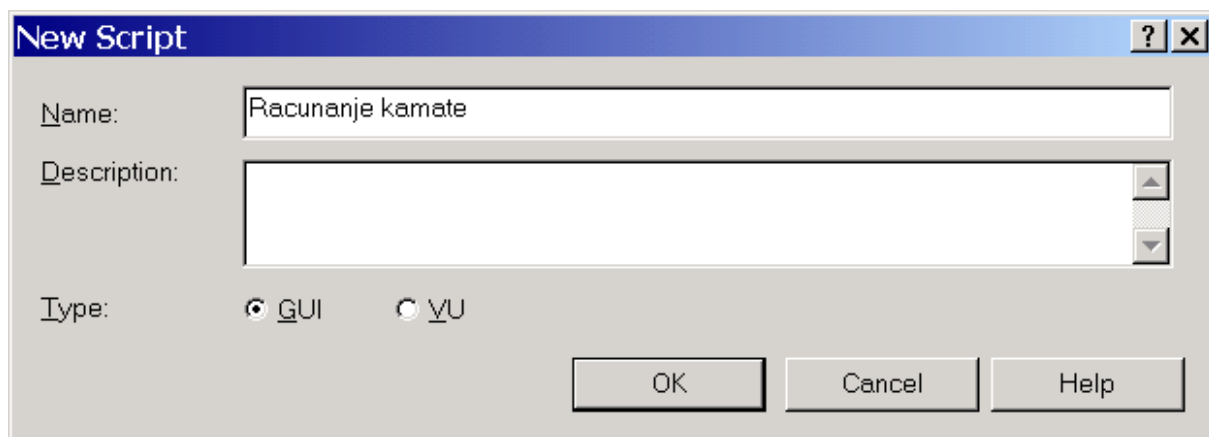
4.1 Podešavanje za rad

Pri prvom pokretanju alata Robot, pre kreiranja ili izvršavanja skriptova, potrebno je podesiti alat za odgovarajuću vrstu aplikacije. Potrebno je iz menija aktivirati Tools/Extension Manager dijalog i izabrati (za web aplikacije tankog klijenta), HTML-MSIE (ako će se kao klijent koristiti MS Internet Explorer) ili HTML-Navigator, ako se kao klijent koristi Netscape Navigator. Potrebno je ponovo pokrenuti alat da bi se ekstenzija aktivirala.



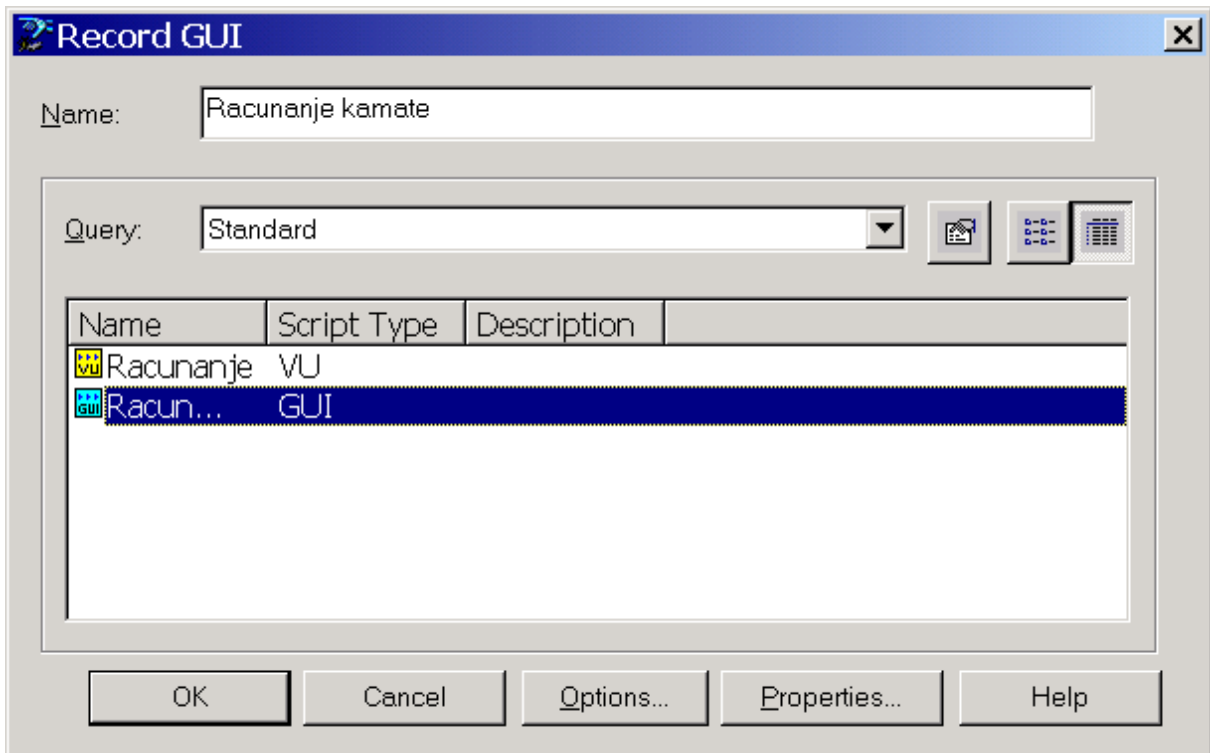
4.2 Kreiranje test skriptova

Kreiranje test skripta inicira se najčešće iz TestManagera (ili u samom alatu Robot, izborom stavke File/New Script). Potrebno je popuniti podatke za ime, opcioni opis i tip: GUI je za testiranje korisničkog interfejsa a VU za vezu klijent server i bazu podataka.

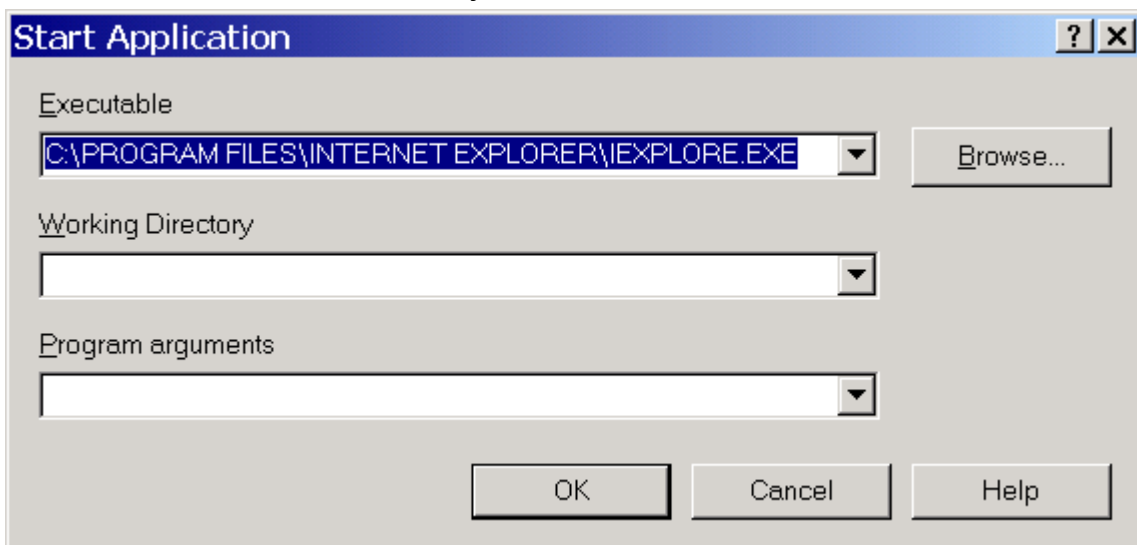


Posle ovoga otvara se editorski prozor sa inicijalnim skript kodom, gde je moguće manuelno editovati kod skripta. Alat Robot poseduje mogućnost automatskog generisanja skript koda, na osnovu zapamćene sesije manualnog rada korisnika sa aplikacijom (slično kao Record Macro u Microsoft Wordu). Ovim se postiže velika

produktivnost kreiranja skript koda. Ova funkcija se aktivira stavkom File/Record GUI za GUI skriptove, ili File/Record Session za VU skriptove. U dijalogu koji se pojavljuje potrebno je izabrati neki od postojećih skriptova, koji će time biti zamenjen, ili uneti ime novog skripta.



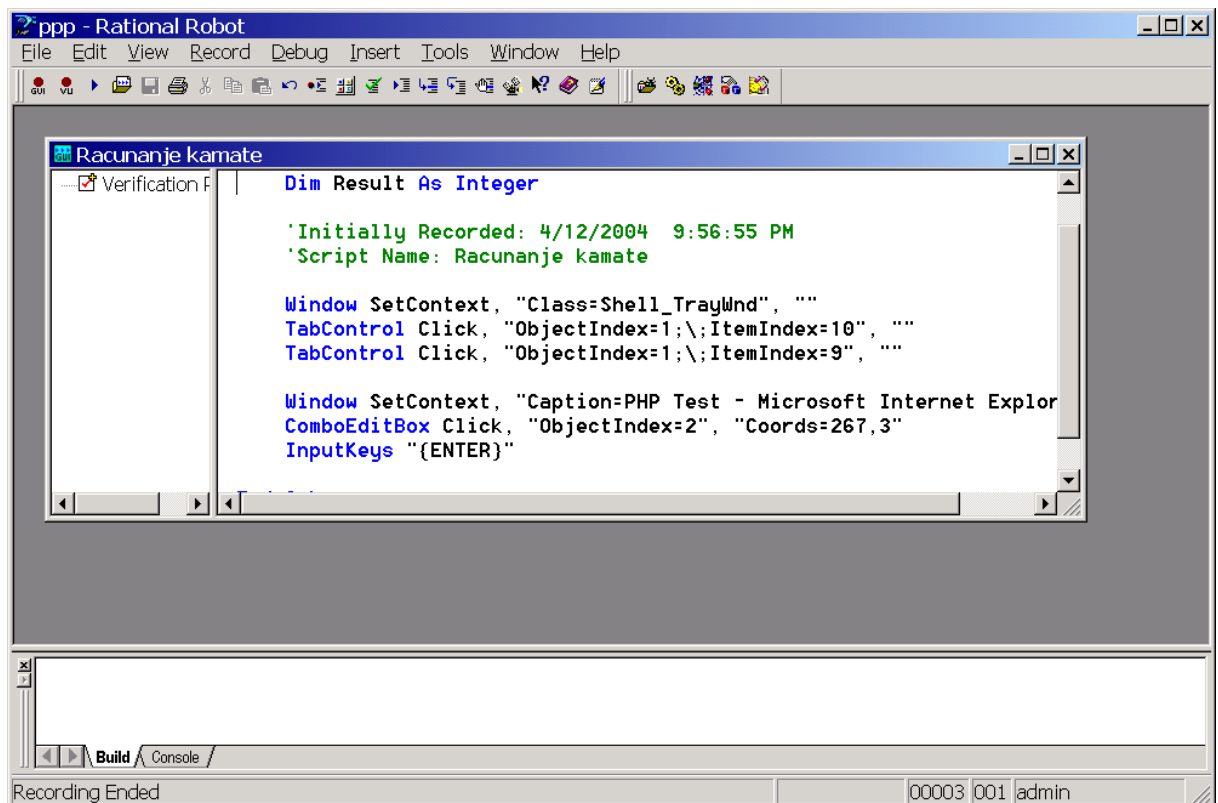
Posle pritiska na OK, aktiviraće se mod pamćenja akcija (prikazuje se mali toolbar na kome se može pritiskom na dugme okončati ovaj mod). U varijanti VU recording, biće potrebno definisati klijentsku aplikaciju, dok u GUI varijanti, svi prozori trenutno aktivni na test računaru se uzimaju u obzir.



Moguće je takođe aktivirati GUI insert toolbar za umetanje u rezultujući skript raznih vrsta verifikacionih tačaka:



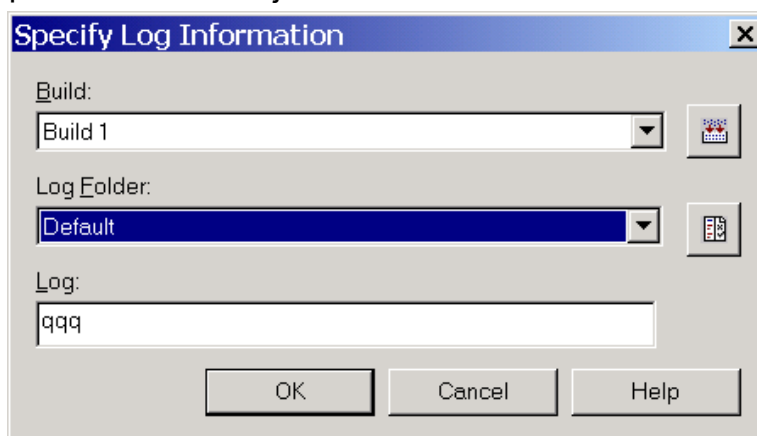
Pojedinosti definisanja verifikacionih tačaka opisane su u uputstvu za alat Robot. Posle završetka moda Record, automatski se dobija odgovarajući skript kod, koji se može editovati:



4.3 Izvršavanje test skriptova

Najčešće se izvršavanje inicira iz TestManagera, kada se ceo proces automatski izvršava (Robot prozor se i ne pojavljuje). Videti poglavlje 3.7.

Izvršavanje se može pokrenuti i neposredno iz alata Robot, komandom File/Playback, kada se pre izvršavanja otvara dijalog gde se navodi u koji Log se pamte podaci o izvršavanju:



Pregle Loga će naravno biti dostupan u TestManageru.

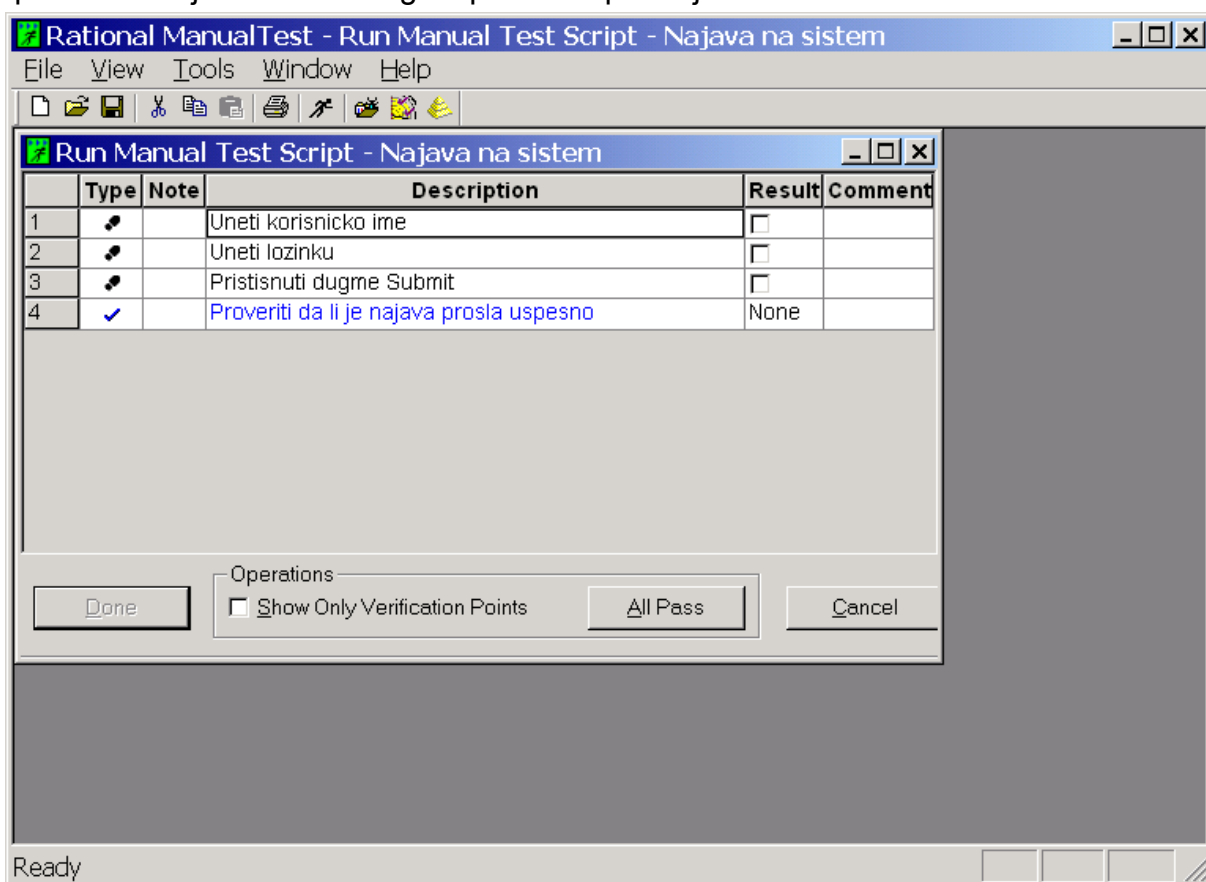
5. Rad sa alatom Rational ManualTest

5.1 Kreiranje test skriptova

Kreiranje manuelnih skriptova najčešće se radi na osnovu dizajna test primera u alatu TestManager, kako je opisano u odeljku 1.1.3. Moguće je naravno skript kreirati i iz alata ManualTest izborom stavke File/New pri čemu je postupak isti kao pri dizajniranju test primera u alatu TestManager.

5.2 Izvršavanje test skriptova

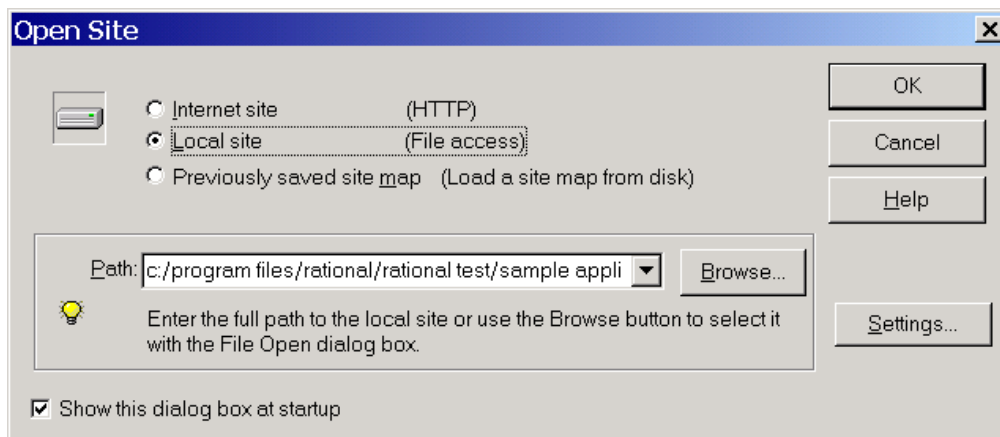
Izvršavanje Test skripta pokreće se najčešće iz alata TestManager kako je ranije opisano. Dobija se sledeći izgled prozora aplikacije:



Korisnik manuelno treba da izvršava korake kako su oni opisani u skriptu i čeka u koloni Result. Ako je u pitanju verifikacioni korak, korisnik treba da proveri da li su uslovi u aplikaciji zadovoljeni i iz liste izabere Pass ili Fail. Korisnik takođe uz bilo koji korak može da doda svoj komentar sa opažanjima vezanim za izvršavanje. Na kraju treba da pritisne dugme done, čime se kontrola vraća u TestManager i prikazuje TestLog prozor kako je ranije opisano.

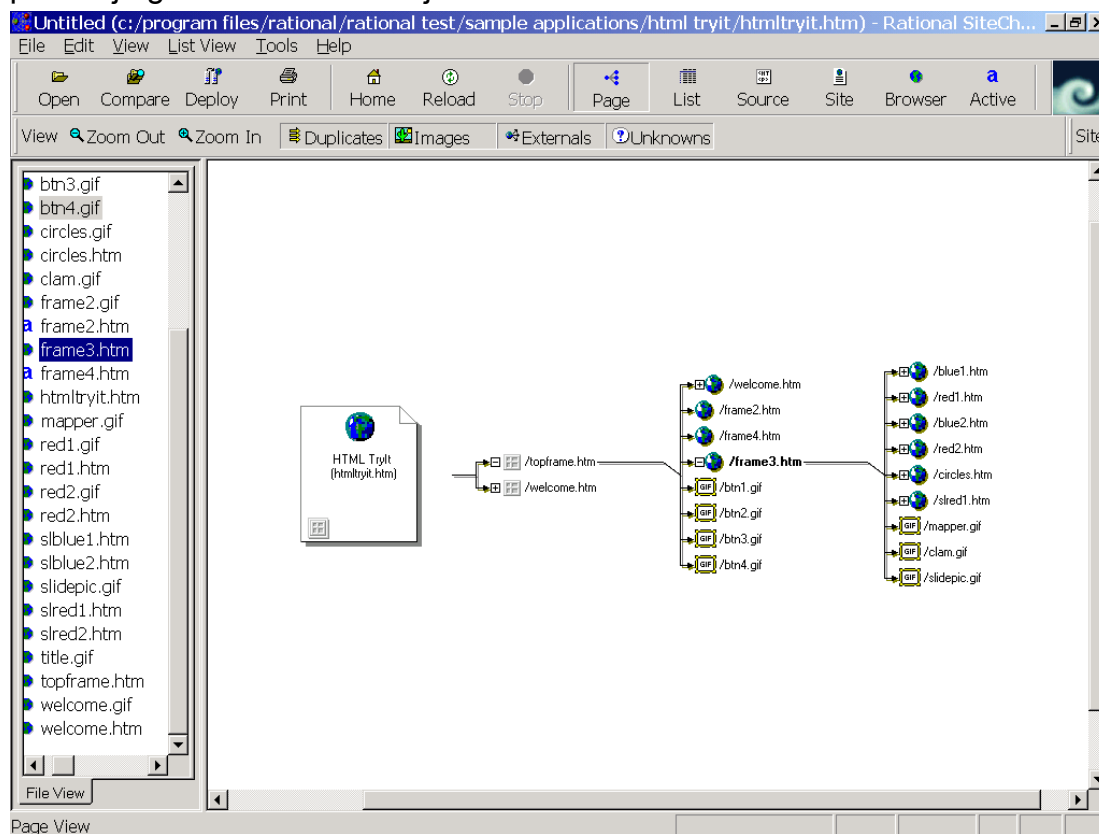
6. Rad sa alatom Rational SiteCheck

Pri pokretanju alata, potrebno je navesti početnu stranicu sajta koji će se izstraživati.

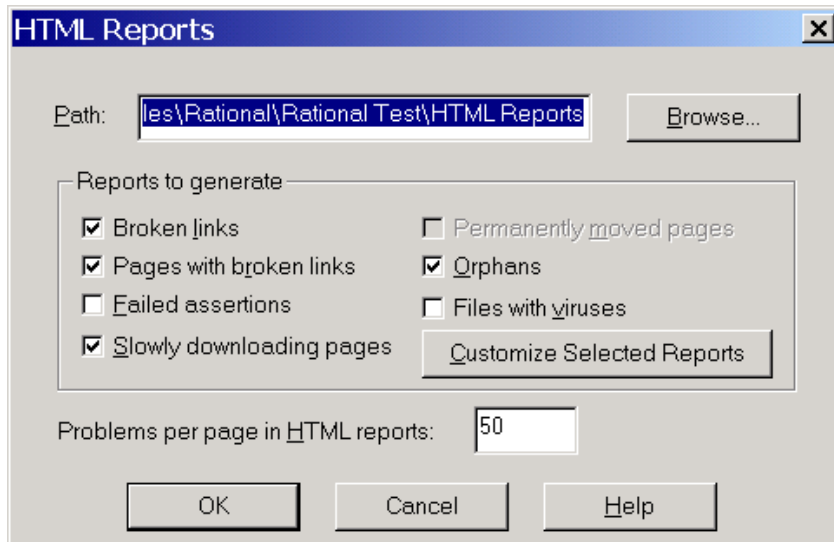


Postupak analize je automatizovan uz asistenciju korisnika oko aktivnog sadržaja (formi).

Kada se analiza završi, moguće je vršiti različite preglede sajta. Primer prikazuje grafički strukturu sajta:



i dobiti različite izveštaje o otkrivenim potencijalnim defektima:



7. Evaluacija testiranja

U alatu TestManager definisan je veliki broj izveštaja koji se odnose na artefakte i proces testiranja. Na sledećoj slici odabran je tab Analysis u prozoru radnog prostora, gde su vidljive vrste izveštaja po kategorijama (vezano za test primere i plan, za testiranje performansi i listinge raznih entiteta u repozitorijumu test podataka, kao što su konfiguracije, iteracije, korisnici itd). Izveštaji se otvaraju desnoklikom na stavku i izborom opcije Run.

Na slici su prikazana dva otvorena izveštaja:

- Test plan Development Coverage koji prikazuje koliko je ukupno i koji test primera u plan uneto, a koliko je implementirano
- Test plan Execution Coverage prikazuje koliko i kojih test primera je implementirano, koliko je izvršavano i koji su sa pozitivnim ishodom a koji sa negativnim.

The screenshot shows the Rational TestManager interface. On the left is a tree view of report categories. The main area displays two 'Test Plan' reports. The top report shows Development Coverage, and the bottom report shows Execution Coverage.

Test Plan 1	Planned Test Ca...	Implemented Te...
Test Plan 1	2	2
Default	2	2
Najava na sistem	2	2
Standard - Win2000	1	1

Test Plan 1	Implemented Te...	With Results	Passed Test Cas...	Failed Test Cases
Test Plan 1	2	1	0	1
Default	2	1	0	1
Najava na sistem	2	1	0	1
Standard - Win2000	1	1	0	1

Finalno, navedimo primer izveštaja o detaljima test plana:

Test Plan Detail Report **Test Plan 1**

Default

- Najava na sistem*

Owner: admin **Suspect:** No
Created By: admin **Custom 1:**
Last Modified By: admin **Custom 2:**
Creation Date: 4/12/2004 **Custom 3:**
Modification Date: 4/12/2004

Description:

Acceptance Criteria: Najava na sistem je uspesna

Pre-conditions: U bazi postoji aktivno korisnicko ime i lozinka za posmatranog korisnika

Post-conditions: Prikazuje se pocetna stranica aplikacije i na njoj ime aktivnog korisnika.

Auto Implementation: Racunanje kamate
Manual Implementation:

Test Case Design

#	Type	Text	Notes
1	Step	Uneti korisnicko ime	
2	Step	Uneti lozinku	
3	Step	Pristisnuti dugme Submit	
4	VP	Proveriti da li je najava prosla uspesno	