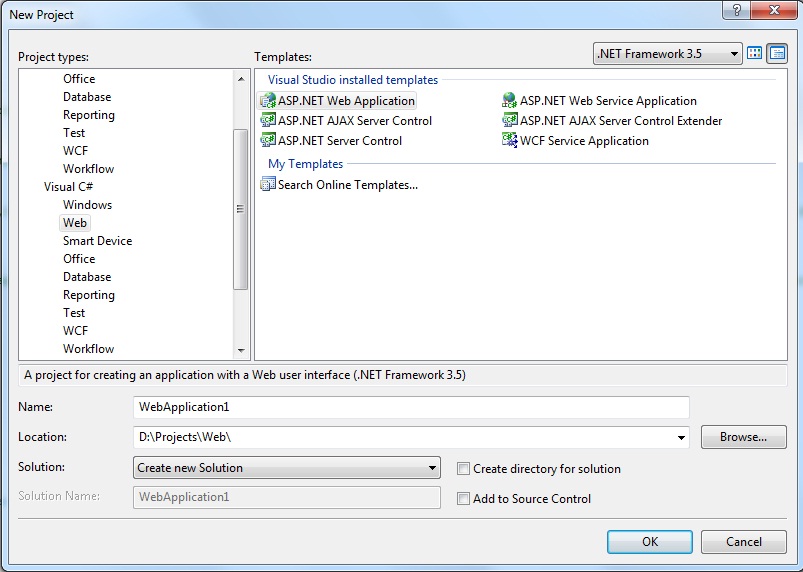
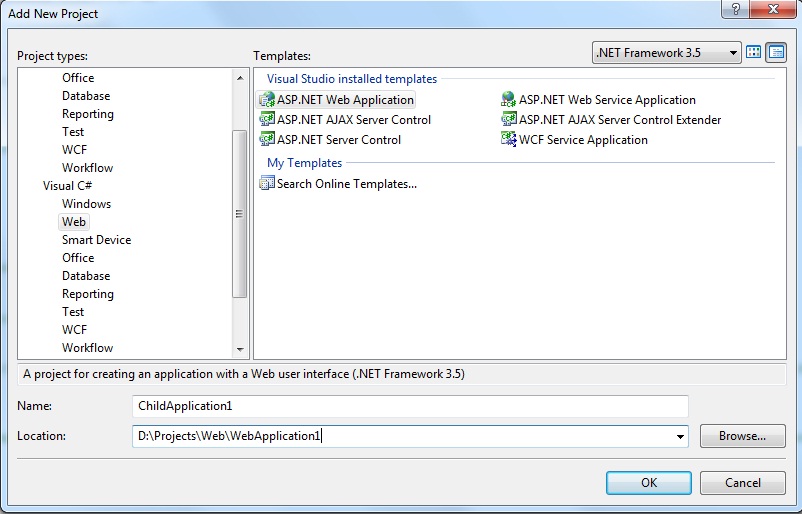
***Procitati VS2008WithIIS7DevelopmentTools.htm***

1. **Single Session Solution** – kao i u VS2003, glavni projekat se napravi u root folderu, a ostali projekti, bilo da su “obicni”, bilo da su web aplikacije, se prave u podfolderima root-a. Dodaju se kao reference glavnom projektu. (videti dokumente iz VSS u bazi ITMR4: Documents > general > KnowledgeBase.doc i oba dokumenta iz Documents > general > SingleSessionApplication)

**npr.** pravljenje projekta WebApplication1: File>New>Project> select ASP.NET Web Application. Unese se ime, a u Location se unese folder u kom ce se nalaziti root folder novog projekta. Odcekirala sam Create directory for solution. Klik na OK i napravljen je novi, glavni projekat u folderu D:\Projects\Web\WebApplication1. U tom folderu se nalazi i solution i file-ovi projekta.



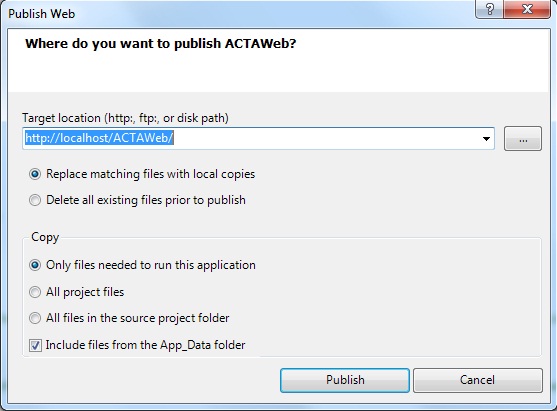
Novi projekat se dodaje desnim klikom na solution, Add>New Project.

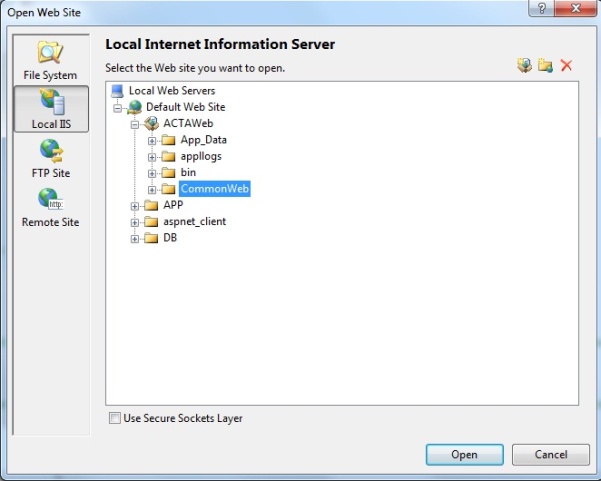


Izabere se vrsta projekta, unese se ime, a u Location se unosi root folder aplikacije, odnosno, folder u kom se nalazi glavni Web projekat. Klikom na OK, pravi se child projekat u folderu D:\Projects\Web\WebApplication1\ChildApplication1. Ako je child projekat Web projekat, iz njega se brisu Web.config i Global.asax file-ovi. Koristice te file-ove glavnog projekta. U reference glavnog Web projekta se dodaje referenca na ChildApplication1.

1. **ACTA-3.0** – U novoj verziji ACTA-e, Web i Windows aplikacija se nalaze u istom solution-u. Projekti Web aplikacije se nalaze u zasebnom folderu, ACTAWebApplication. Njihova struktura na disku je i dalje ista. U root folderu celog solution-a se nalazi projekti i odgovarajuci file-ovi root web projekta (ACTAWeb), a child projekti se nalaze u odgovarajucim subfolder-ima. Instalacija ostaje ista. Kao virtuelno ime root foldera u instalaciji ce se zadati ACTAWeb. Ako se dodaje novi child projekat, kao putanja se bira <…/ACTA-3.0> da bi struktura web projekata na disku ostala ista.
2. **Publikovanje** – Da bi se aplikacija uspesno publikovala i debug-ovala preko IIS-a, VS se mora pokrenuti sa administratorskim privilegijama (desni klik, Run as administrator). Pri publikovanju, glavni projekat se publikuje na root WebSite-u, a ostali se publikuju na podfoldere root WebSite-a, koji se prave kao virtuelni, a ne web. C:\inetpub\wwwroot – dodat je korisnik ASP.NET Machine Account i date su mu sve permisije nad tim folderom. To je bilo obavezno sa VS2003. Nisam isprobavala da li bez toga radi aplikacija pravljena u VS2008.

**npr.** Publikovanje WebApplication1: Desni klik na projekat pa Publish, ili Bild>Publish WebApplication1 (Opcija publish je dostupna za projekat cija stranica je trenutno otvorena i u fokusu u VS.)



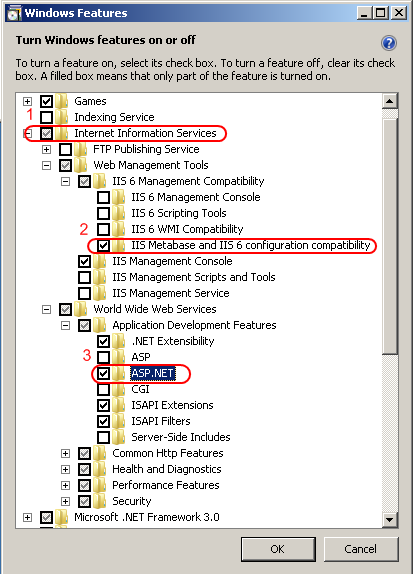


Target lokacija, klik na dugme … Izabere se LocalIIS. Pod Default Web Site se napravi nova Web aplikacija (prva ikonica gore desno), i izabere se novonapravljeni folder. Kao opcija kopiranja se izaberu samo neophodni file-ovi. Zatim se publikuju child web projekti. Isto kao i glavni, samo sto se pod IIS, u root web aplikaciji folderu pravi virtuelni folder (druga ikonica gore desno), i taj se izabere za publikovanje child projekta. Za virtualni folder se unosi alias (ime foldera) i fizicka putanja (C:\inetpub\wwwroot\WebApplication1\ChildApplication1)). Proveriti na IIS da li su folderi korektno napravljeni, root, odnosno WebApplication1 treba da je WebSite, a podfolder ChildApplication1 treba da je “obican”. Ako je i on greskom WebSite, desnim klikom na njega i izborom opcije Remove, postaje “obican”. Nakon uspesnog publikovanja, pokretanjem <http://localhost/WebApplication1> u browser-u, pokrece se aplikacija sa servera, ako se pokrece sa racunara na kom je instalirana, odnosno, sa http://<ip\_address or computer\_name>/WebApplication1, ukoliko se pokrece sa drugog racunara, odnosno, klijenta.

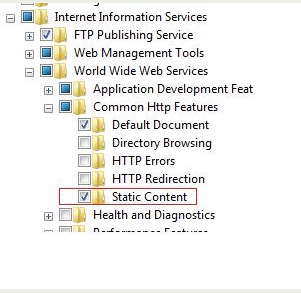
**napomena:** Adrese na koje se prosledjuje se navode punom putanjom, da bi sve radilo iz publikovane verzije. **npr.** na stranicu default.aspx iz WebApplication1 projekta se prosledjuje sa Response.Redirect(“/WebApplication1/default.aspx”), na stranicu default.aspx iz projekta ChildApplication1 se prosledjuje sa Response.Redirect(“/WebApplication1/ChildApplication1/default.aspx”). Da bi ovaj nacin prosledjivanja radio i pri pokretanju iz okruzenja preko default web servera, u Property strani glavnog projekta, izabere se Web tab i u Virtual path polje se unese /WebApplication1.

**Problem sa publikovanjem:** Ako se menjalo izmedju koriscenja IIS-a i default web servera za debugovanje (objasnjeno u debug-ovanju), da bi publikovanje bilo uspesno, polja Project URL, i Override application root URL, moraju biti prazna i odcekirana pri prelasku na default web server. (pogledati ErrorPublishingWebProject.htm)

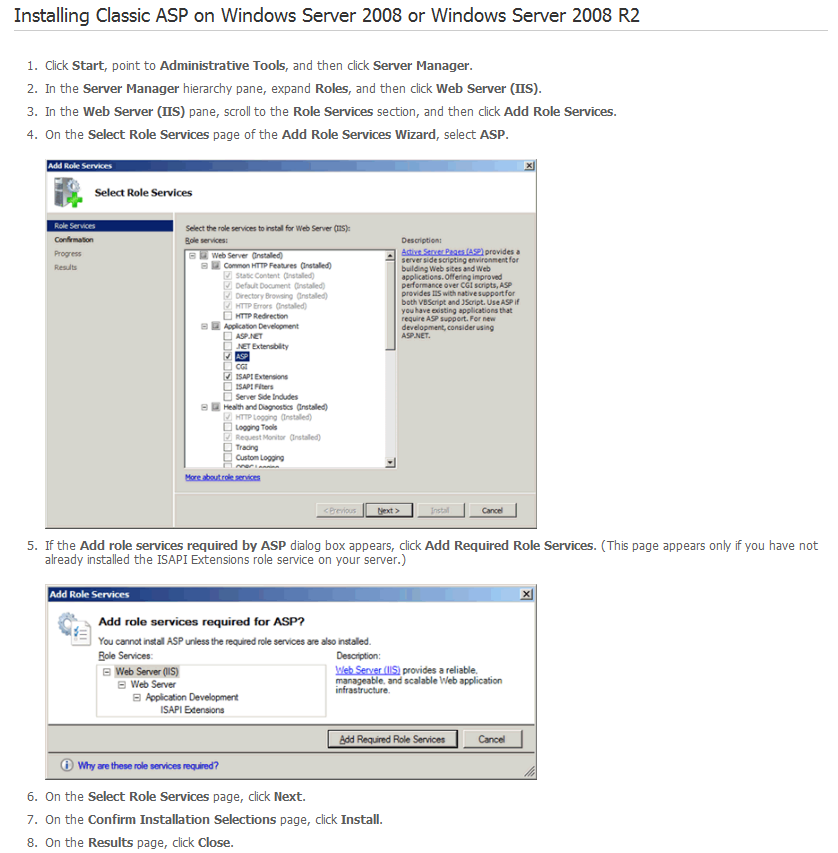
1. **Instalacija** – Da bi se aplikacija pokretala iz browser-a, ne mora se publikovati. Moze se napraviti Setup projekat. Slicno kao kod Windows aplikacija, doda se novi projekat u Solution, samo sto se ovde bira Web Setup Project. U File System-u se napravi odgovarajuca struktura foldera. U glavnom, Web Application Folder-u se dodaju sadrzaji iz parent projekta (u nasem slucaju ACTAWeb). Dodaje se Primary output, Content files, Localized resources svih projekata koji imaju Resource.resx file-ove. Pod tim folderom, za svaki child projekat, dodaje se Web folder, i u taj folder se dodaju Content files odgovarajuceg child projekta. Mogu se podesiti jos neke opcije setup-a. Naziv root folder-a aplikacije na IIS-u, odnosno folder-a u koji se smestaju sadrzaji parent projekta kao i folderi svih child projekata, se podesava na Properties Web Application Folder-a, opcija VirtualDirectory. Folder treba da se zove kao i parent projekat jer je taj naziv navodjen u aplikaciji, da bi sve funkcionisalao kako treba. Taj folder se moze podesiti i u toku same instalacije. U Properties samog setup projekta se mogu podesiti parametri vezani za proizvodjaca. Ako se zeli napraviti instalacija koja sadrzi i instalaciju .Net Framework-a, Crystal Reports-a i slicno, u Properties prozoru setup projekta se izabere Prerequisites, pa u tom prozoru se cekira sta se zeli jos instalirati po potrebi, i izabere se opcija Download prerequisites from same location as my application. Nakon rebild-a setup-a, u Debug folder-u setup projekta se pored .msi i setup.exe, nalaze i folderi sa izabranim instalacijama. Da bi sve radilo dobro, redosled instaliranja mora da bude sledeci. Najpre se instalira IIS. Instalacija je objasnjena u dokumentu navedenom na pocetku ovog dokumanta. Potrebno je cekirati opcije kao na slici (Za Windows Vista i Windows 7).

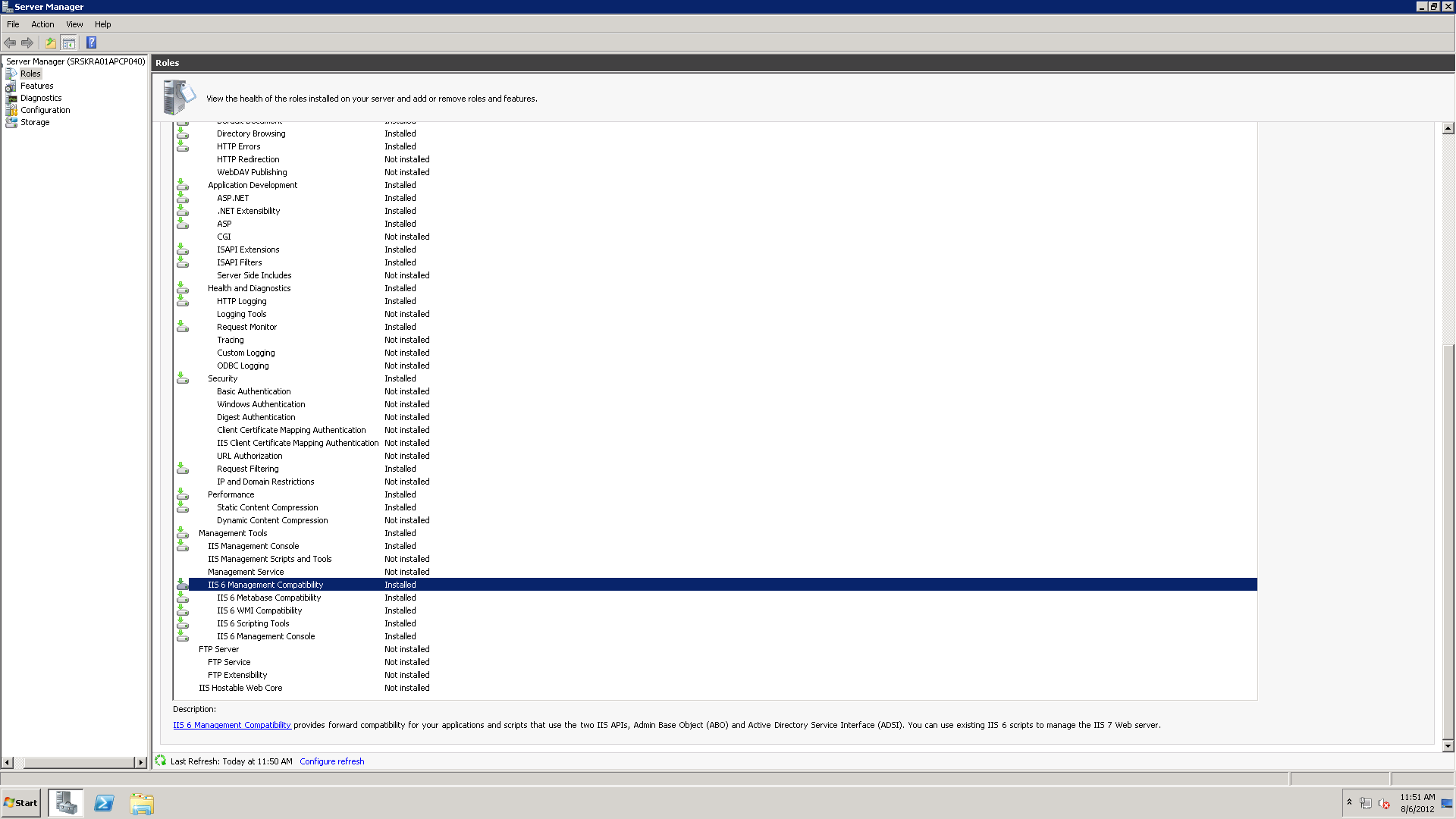
****

Ukoliko se ne prikazuju stilovi, ne ocitavaju slicice… proveriti da li je cekirana opcija Static content kao na slici:



**Na sledećoj slici je prikazano kako se čekiraju opcije za Windows Server 2008:**



Na isti način treba da se instalira IIS6 Management Compatibility:

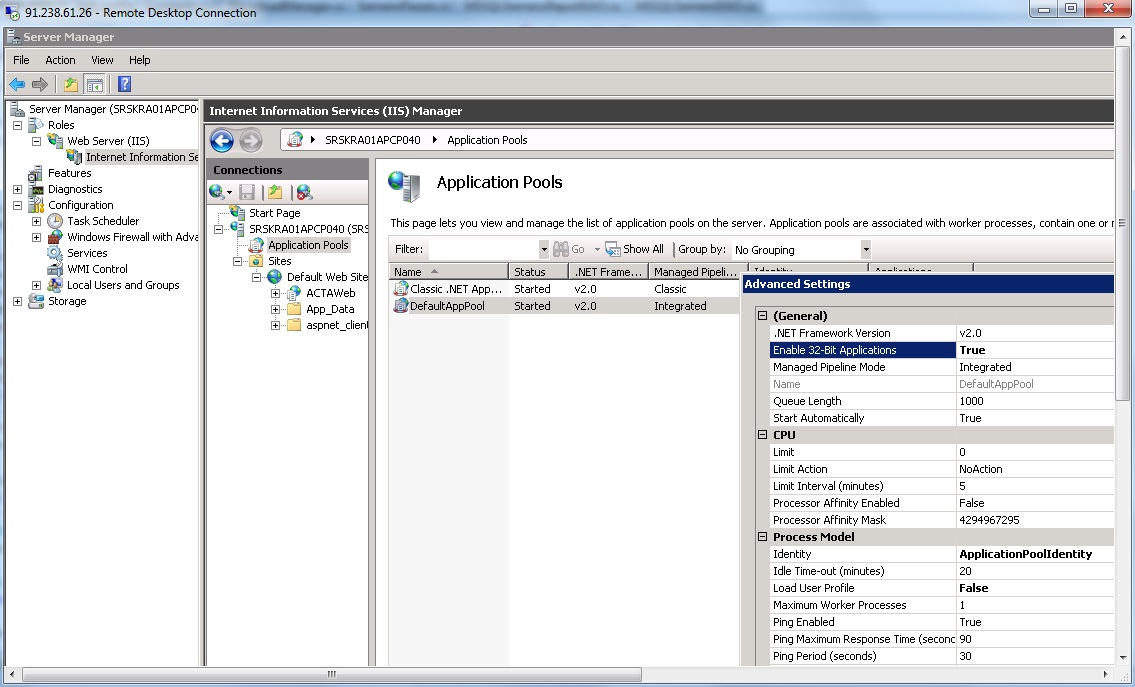
Zatim se instalira Crystal Report, bilo kroz instalaciju web aplikacije, bilo zasebno, da bi se odgovarajuci folder, naveden u prethodnom delu, popunio sadrzajima potrebnim za ispravan prikaz Crystal Report-a. Negde sam procitala da se ovom folderu moraju dati sve permisije i ASPNET user-u i Network service user-u. Mada, kada sam probala instalaciju na XP-u gde nisam davala te permisije, sve je radilo ispravno, tako da ne znam je li ovaj korak uopste potreban. Doduse, Windows 7 je malo rigorozniji po pitanju prava pa je mozda na njemu potrebno. Pozeljno je nakon instalacije Crystal Report-a proveriti odgovarajuci folder pod IIS-om, pa ako je prazan, dati privilegije, deinstalirati Crystal Report pa instalirati opet. Aplikacija se instalira na kraju. Pri instalaciji, mora se paziti sta je navedeno kao virtual directory. To mora biti naziv parent projekta. Nakon instalacije, pozeljno je proveriti strukturu foldera na IIS-u. Vrsni folder treba da je oznacen kao Web aplikacija, ostali folder-i treba da su obicni. Aplikacija se u browser-u pokrece sa [http://localhost/<root](http://localhost/%3croot) folder\_name> ako se pokrece sa racunara na kom je instalirana, odnosno sa servera, ili sa http://<ip\_address or computer\_name>/<root\_folder\_name>, ako se pokrece sa drugog racunara, odnosno klijenta.

Treba user-u IIS\_IUSRS dati Full Control nad folder-om **ACTAWeb\CommonWeb\files**.

Ukoloko se pri pokretanju javi greska

Could not load file or assembly 'Microsoft.SqlServer.BatchParser' or one of its dependencies. An attempt was made to load a program with an incorrect format.

Podesiti opciju Enable32-bitApplications DefaultAppPool kao na slici:



1. **Debug-ovanje** – po default-u se projekat iz okruzenja pokrece i debug-uje na default web serveru. Ako se to zeli promeniti, u Property strani projekta, na Web tab-u se izabere opcija Use IIS Web server, URL polja se popune i izmene se sacuvaju (nije mi uspelo). Da bi debug radio i da bi se stranice domasile i preko default web servera, potrebno je podesiti Virtual path (ranije opisano). Ako se sa Use IIS Web server, vraca na Use Visual Studio Development Server, mora se paziti da sva polja iz Use IIS Web server budu prazna i odcekirana da bi se projekti mogli publikovati. U VS2003 je postojao problem sa promenama u “obicnim” projektima koji su ukljuceni u glavni web projekat. Naime, promene su prihvacene i moze se debug-ovati promenjeni kod, tek nakon sto se menjani projekat izbilduje, iz glavnog web projekta se izbaci stara referenca na taj projekat i doda se nova (na taj nacin sam ja uspela da se promene vide i pri pokretanju glavnog, odnosno da se pri bildovanju glavnog projekta, “povuce” novi .dll i .pdb “obicnog” projekta). Pri debug-ovanju na default web serveru u VS2008, ovaj problem ne postoji. Mozda ce postojati ako uspem da debagujem na IIS.

Jedno resenje da se debag-uje preko IIS-a je da se Default Web Site physical path podesi na bin folder projekta koji se debaguje (Properties ili Advance settings). Onda je moguce izabrati opciju Use IIS Web server, kreirati virtuelni direktorijum i debug-ovati. Ali, onda se na IIS prevuku svi file-ovi projekta, odnosno, solution-a. I sta ako postoji vise Web aplikacija? Svaki put kad se zeli debug-ovati neka, menjati physical path Default Web Site-a? Zasad ostavljam pokretanje i debug-ovanje preko default web servera.

1. **Logout** – Da bi se uspesno izlogovalo i da bi se korisnik vratio na Login stranu, potrebno je napraviti posebnu logout stranu koja ce pozivati Session.Abandon() i zavrsiti trenutnu sesiju. Na klik menija “Izlaz”, poziva se Logout() javascript funkcija iz Logout.js u CommonWeb. Logout() otvara logout stranu koja zavrsava sesiju. Ako se izabere zatvaranje prozora, aplikacija se zavrsava i browser se zatvara. Ako se izabere da se prozor ne zatvori, korisnik treba da se vrati na login stranu i da ne moze da se sa back i forward krece kroz vec vidjene strane. Nakon zavrsetka sesije pokrece se nova. U Session\_Start() se redirektuje na login stranu. Da se ne bi odmah pozivao Session\_End i da se ne bi beskonacno mnogo puta islo kroz Session\_Start(), Page\_Load() login strane i Session\_End(), u Session\_Start() je potrebno nesto staviti u Session, **npr.** Session[“dummy”] = “dummy”. Negde sam citala da Session.Abandon() izaziva kraj sesije i prolazak kroz Session\_End() samo ako je nesto prethodno stavljeno u Session pa ovo stavljanje na pocetku resava i taj problem. Dodeljivanje javascript funkcije kliku dugmeta ili menija se moze raditi u Page\_Load() ili Render() metodi strane ako ne reaguje na dodelu onclick atributa u html-u strane.

**Izmena:** Ovaj nacin logout-a je preuzet iz ShipmentsWeb-a, ali nije uvek radio ispravno pa je logout promenjen, odnosno, radi se ista stvar ali se ne otvara nikakav popup. Na click Izlaz (Exit) se poziva metoda koja odradi Session.Abandon(), FormsAuthentication.SignOut() i redirektuje na Login stranu.

1. **Vracanje na prethodnu stranu** – u URL-u se u Back parametru prosledi strana sa koje se odlazi pa se sa Request[“Back”] pokupi ta adresa i redirektuje se na tu stranu.
2. **Lokalizacija** – Promenljive potrebne za lokalizaciju se moraju definisati svaki put pre koriscenja.
3. **Forms Authentication (preporucen vid autentifikacije)** – U Web.config se u <system.web> doda <authentication mode="Forms">. Strana za logovanje koja se ocekuje je Login.aspx u root folderu. Ako se zove drugacije, mora se navesti u <forms>. Nakon sto se unesu user name i password i nakon sto se uspesno validiraju mora se pozvati FormsAuthentication.RedirectFromLoginPage(), koji ce napraviti forms authentication ticket i forms authentication cookie, dodeliti cookie kolekciji Cookies i preusmeriti korisnika na trazenu stranu (pogledati FormsAuthentication.htm). Dok god se ne pozove ta metoda, korisnik ne moze videti nijednu stranu. Ako korisnik nije autentifikovan, svaki zahtev za odredjenom stranom vraca korisnika na Login stranu.
4. **Session i FormsAuthentication timeout** – U Web.config se postave vrednosti ova dva timeout-a. Potrebno je da je Session timeout <= 2\*FormsAuthentication timeout.

<authentication mode="Forms">

<forms loginUrl="~/Login.aspx" timeout="15" slidingExpiration="true"></forms>

</authentication>

U <forms> tagu se zadaje loginUrl, timeout i slidingExpiration se postavi na true, sto znaci da svaki put kada se u drugoj polovini timeout-a posalje neki zahtev serveru, ovaj timeout se produzava na taj trenutak + timeout.

<authorization>

<deny users="?"></deny>

</authorization>

Ovo znaci da za pristup bilo kojoj stranici, korsnik mora biti autentifikovan. Ukoliko nije, ili je timeout istekao, korisnik se redirektuje na loginUrl, odnosno, Login.aspx stranu. Na toj strani se vrsi zatvarenje postojece konekcije (ako session timeout jos nije istekao i konekcija postoji u Session-u i nije zatvorena), Logout (odnosno FormsAuthentication.SignOut() ako se doslo nakon isticanja session timeout-a, potrebno je izlogovati user-a) i praznjenje Session objekta (ako nije uradjen Session.Abandon() kroz izlazak iz aplikacije ili nije istekao session timeout).

<sessionState mode="InProc" cookieless="false" timeout="30"/>

Na ovaj nacin se zadaje session timeout. U Session\_Start() se proverava da li je zapoceta sesija zapoceta prvi put ili nakon istianja timeout-a. (videti metodu u Global.asax). U koliko je vec bila zapoceta i istekla, korisnik se preusmerava na Login.aspx sa porukom o isticanju sesije. Pri load-ovanju Login.aspx, zatvaraju se svi pop-up ovi i izlazi se iz svih iframe-ova. Odnosno, trazi se roditelj, glavna strana u kojoj se otvara forma za logovanje. Ovo odradi js funkcija closeWindows() na toj strani.

1. **Pokretanje aplikacije** – Nakon uspesnog logovanja korisnik se preusmerava na DefaultPage.aspx. Na toj strani postoji js funkcija koja otvara aplikaciju u novom prozoru koji je maksimizovan i bez dodatnih toolbar-ova, dugmadi za navigaciju, mogucnosti menjanja adrese i slicno.
2. **Prikaz rezultata (sortiranje, page-iranje, selektovanje)** – Rezultati pretraga se prikazuju u ResultPage.aspx strani. Na strani za pretragu (**npr.** PassesSearch.aspx) nalazi se filter, zatim iframe u kom ce se prikazati stranica sa rezultatima pretrage i eventualno deo za manipulaciju pronadjenih rekorda. Iframe je podesen da je autoscroll-abilan. Sirine i visine elemenata ResultPage-a su podeseni tako da staju u iframe bez scrolla, ali ako ne stanu, da za svaki slucaj postoji scroll. Pri inicijalizaciji te strane, u Session se postavljaju parametri koji su stalni za tu pretragu (headers – imena polja koja ce stajati u header tabeli, fields – polja iz select statement-a koja se prikazuju, fieldsFormating – vrednosti iz Constants.FormatTypes koje pokazuju da li odgovarajuce polje treba formatirati i na koji nacin ili ne u prikazu, tables – tabele iz kojih se vrsi pretraga, key – polje, odnosno, vrednost koja jedinstveno karakterise pronadjeni rekord). Ako je select slozeniji, odnosno, u from klauzuli se navodi vise tabela pa polja moraju biti navedena sa imenom tabele ispred, onda se polja navode sa aliasima pa ce se kasnije za ime polja koje se prikazuje u datagrid-u navesti alias.

**npr.** tabela passes p, polje event\_time se navodi ps.event\_time AS pass\_time, a kao vrednost polja pri kreiranju kolone za prikaz se prosledjuje pass\_time. Ne znam zasto, ali ako se za ime polja prosledi ps.event\_time, ne prikazuju se podaci u datagrid-u.

Da bi kasnije selektovanje radilo korektno i da bi svaki rekord mogao da ima jedinstveni identifikator, medju poljima mora biti navedeno i polje koje jedinstveno odredjuje taj rekord (id ili rec\_id; ako ga jedinstveno odredjuju vise kolona, mislim da postoji varijanta da se datagrid-u dodeli i vise polja kao jedinstveni identifikator, ako bude potrebe, istrazivacu.)

Kada se izaberu kriterijumi pretrage i pritisne dugme za pretragu, najpre se formiraju parametri za pretragu i smestaju se u Session. Od izabranih kriterijuma kreira se filter, postavlja se polje po kom se sortira i sort direction, a zatim se iframe preusmeri na ResultPage, odnosno kao adresa mu se dodeli ta strana. ResultPage kreira select za pronalazenje rezultata pretrage i kreira datagrid za prikaz rezultata pretrage. ResultPage se sastoji od panela na kom je smesten datagrid i dela za manipulaciju stranicama pretrage. Da datagrid ne bi rastao u beskonacnost u zavisnosti od broja rekorda koje prikazuje, smesten je na panel fiksne visine. Panel je autoscroll-abilan, pa ako broj rekorda predje vidljiv deo panela, pojavljuje se scroll za pregled svih rekorda. Datagridu se ne postavlja visina, jer se menja u zavisnosti od broja prikazanih rekorda.

**Inicijalizacija** – Pri load-ovanju strane, najpre se pokupe parametri iz Session-a. Izracuna se broj pretrazenih rekorda i popuni padajuca lista sa stranama. U svakom momentu se iz baze prevlaci i prikazuje samo jedna strana. Ako nije pronadjen nijedan rekord, dugmad za navigaciju medju stranama, kao i padajuca lista sa stranama ce biti disable-ovani. Zatim se inicijalizuje datagrid. Prvo se inicijalizuje tabela koja ce da bude header, a zatim i datagrid. Pokusavala sam da datagrid prikazuje svoj header, ali se taj header gubio prilikom scroll-ovanja. Datagrid se inicijalizuje i ako se na stranicu dolazi prvi put i ako ne, jer se vrednosti headera i datagrida (zapravo, sve sto je dinamicki napravljeno u kodu) iz nekih razloga, meni zasad nepoznatih, gube. U header tabeli se najpre pravi kolona sa check box-om za selektovanje, odnosno deselektovanje svih rekorda, a zatim i kolone ciji broj i nazivi su prosledjeni u parametru Session[“header”]. Od parametra prosledjenog kroz Session[“fields”] se kreiraju polja koja se prikazuju u kolonama i za svako polje se kreira kolona koja se dodaje datagridu. Datagridu je na samoj stranici dodeljena i ItemTemplate kolona u vidu check box-a, potrebna za selekciju rekorda. Parametar Session[“key”] se postavi kao DataKeyField datagrid-a (nesto kao Tag u ListView, jedinstveni identifikator reda).

**Formatiranje prikazanog polja** – pri slanju parametara za prikaz rezultata, u filedsFormating parametar se smesti Dictionary za polja koja treba formatirati. Key je index polja (0 je prva kolona iz baze koja se prikazuje) a value je tip formatiranja uzet iz Constants.FormatTypes. Pri kreiranju kolone, prosledi se i odgovarajuci tip formatiranja iz fieldsFormating prosledjenih kroz Session. U zavisnosti od vrednosti tipa formatiranja getFormat() metoda ResultPage-a vrati vrednost format string-a i ta vrednost se dodeli DataFormatString property-ju kolone koja se kreira (dodela se vrsi ako je prosledjen neki tip formatiranja razlicit od None).

**Formatiranje prikazane vrednosti** – kroz fieldsFormatedValues parametar Session se prosledi Dictionary ciji je key index polja (0 je check box, a vrednosti iz liste krecu od 1), a vrednost je Dictionary sa originalnom vrednoscu iz baze i vrednoscu koja treba da bude ispisana u listi pretrage. Kada se popunjava resultGrid u ItemBound se prolazi kroz sva polja odredjenog reda (izuzev 0-tog, odnosno, check box-a) i getFormatedValues() metoda vraca vrednost koja treba da pise u listi.

**Popunjavanje datagrid-a (prikaz rezultata pretrage)** – nakon inicijalizacije, popunjava se datagrid. Ako se stranica prikazuje nakon akcije sa druge strane pa je potrebno prikazati stranicu rezultata pretrage koja je bila prikazana pre akcije, parametar Session[“samePage”] ce biti postavljen na true i tada se prikazuje trenutna stranica. Ako je taj parametar null ili nije postavljen na true ili je trenutna stranica 0, prikazuje se prva stranica rezultata pretrage. Kreira se select od parametara prosledjenih kroz Session. Osnovni select izgleda ovako: SELECT Session[“fields”] FROM Session[“tables”] WHERE Session[“filter”] ORDER BY Session[“sortCol”] Session[“sortDir”]. Malo je modifikovan u zavisnosti da li je u pitanju SQL Server ili MySql, tako da se pored ovih parametara prosledjuje i od kog do kog rekorda se zeli prikaz (videti u MSSQLResultDAO.cs i MySQLResultDAO.cs metodama). Kao DataSource datagrid se postavlja DataTable da bi selekcija radila ispravno. Ako se postavi DataView, selekcija ne radi, bar ne ova kakvu sam nasla. U zavisnosti od broja strana i od broja prikazane strane, postavljaju se dugmad za navigaciju na enabled/disabled i bira se odgovarajuci rekord u padajucoj listi stranica.

**Sortiranje** – Za sortiranje sluze link button-i u header tabeli. Pri dinamickom pravljenju header link button-a, u ID sam stavljala naziv polja koja je prikazana u koloni ciji header je odgovarajuci button (nisam nasla neko polje kao Tag u Windows aplikacijama). Pri kliku na to dugme, menja se polje za sortiranje i sort order po potrebi i poziva se metoda za popunjavanje datagrid-a i to sa prvim i poslednjim rekordom prve stranice.

**Selektovanje** – Za selekciju rekorda sluze check box-ovi na pocetku svakog reda. U ItemBound metodi se svakom check box-u dodeljuje jedinstvaeni identifikator item-a koji predstavlja, odnosno, za ciju selekciju sluzi. Na njihov klik se okida javascript funkcija highlightRow() koja u zavisnosti od chekiranosti, postavlja izabran red na novu boju i dodaje vrednost reda u hidden polje ResultPage-a, odnosno, vraca red na staru boju i brise ga iz hidden polja koje sadrzi vrednosti (jedinstvene identifikatore) izabranih redova. Pri menanju boje reda na boju koja oznacava da je red izabran, staru boju reda sam sacuvala u forecolor property-ju da bih kad se red deselektuje znala na koju boju da vratim (boje Item-a i AlternatingItem-a se razlikuju). Za selekciju/deselekciju svih redova sluzi check box u headeru. Njegov klik okida selectAll() javascript funkciju koja pronalazi sve check box-ove i postavlja ih na stanje header check box-a (“okida” click check box-a koji odradi menjane boje i upis/brisanje iz hidden polja). Ako se nakon selekcije i neke akcije vraca na stranu pa je potrebno otvoriti stranu sa koje se otislo sa postojecom selekcijom, nakon akcije, u Session se smesta trenutna selekcija. Kada se ponovo otvora strana, pre bind-ovanja datagrid-a se proveri da li u Session postoji selekcija. Ako postoji, pokupi se, taj parametar Session-a se isprazni, u ItemBound metodi se selektovani rekordi chekiraju, menjaju boju i vrednost se smesta u hidden polje. Kada se na filter strani zeli izvrsiti neka akcija sa selektovanim rekordima ResultPage-a otvorenog u iframe-u filter strane, na klik dugmeta koje treba da pokrene zeljenu akciju se izvrsi javascript funkcija getSelectedKeys() kojoj se kao argumenti proslede naziv hidden polja filter strane u koje se zeli upisati selekcija, poruka koja se prikazuje ako nema selektovanih rekorda, i naziv frame-a koji sadrzi ResultPage. Funkcija prepisuje vrednost hidden polja ResultPage-a u hidden polje filter strane, tako da nakon klika dugmeta, hidden polje filter strane sadrzi selekciju ResultPage-a. Kako klik dugmeta automatski “okida” PostBack strane, u Page\_Load sam, ako hidden polje sadrzi neku vrednost, pozvala metodu koja bi trebalo da bude Click metoda i koja ce da pokupi selekciju iz odgovarajuceg hidden polja i izvrsi neku akciju. Nakon sto se selekcija pokupi, hidden polje se prazni. Nisam nasla pametniji nacin kupljenja selekcije sa druge strane i neku akciju sa istom. Hidden polje za smestanje selekcije na ResultPage-u se prazni pri svakom popunjavanju datagrid-a. Ako se vraca na stranu gde je potrebno prikazati prethodnu selekciju, ona se nalazi u Session pa ce se pri bind-ovanju datagrid-a, odgovarajuce polje za selekciju ponovo popuniti selektovanim vrednostima.

Ako se selekcija ne zeli prikazati, stranica se poziva sa parametrom showSelection=false. U tom slucaju se malo drugacije zadaju indeksi za formatiranje vrednosti jer check box kolona vise ne postoji i ne broji se.

**Prikaz podataka kojih nema u bazi** - I u tom slucaju se prosledjuju header i fields, formatting fields, formatting values, data table list i data table columns. Tables se postavlja na null, i key ako nema selekcije. U suprotnom se data table list i data table columns postavljaju na null pri inicijalizaciji SQL parametara. Ako se prikazuju ‘virtuelni’ podaci (ne radi se select iz baze), kroz fields se prosledjuju prava imena kolona grid-a bez aliasa. Kada se formiraju redovi za prikaz, formira se najpre List<DataColumns> kroz koju se proslede kolone koje ce imati resultTable na ResultPage, zatim se od podataka za prikaz formiraju redovi. Jedan red prikaza je List<object> a svi oni se pakuju u List<List<object>> pa se kroz data table columns i data table list prosledi lista kolona i lista podataka. Zatim se ResultPage pri inicijalizaciji parametara i popunjavanju datagrid-a gleda da li je session data table list == null, ako jeste radi kao i do sada, ako nije, broj redova uzima iz prosledjene liste a popunjavanje se vrsi tako sto se od prosledjenih kolona napravi resultTable a zatim se puni odgovarajucim listama objekata (od prve do poslednje strane za prikaz). Kod sortiranja se takodje ispituje da li je data table list null, pa ako nije, sortira se prosledjena lista (ubacena inner class za sortiranje liste, trenutno podrzava objekte tipa int, string, decimal, double, float, datetime, prosirivacemo ako se pojavi jos neki) i prikaze, ako jeste, sortira po starom. Sve u svemu, jako je bitno da se u inicijalizaciji SQL parametara pojave svi, pa koji treba da se setuju na neke vrednosti da se setuju, a koji nisu potrebni da se setuju na null, da se ne bi preneli sa neke druge strane. **npr.** TLLoadPage.aspx.

1. **ErrorPage** – Ukoliko pri izvrsavanju aplikacije dodje do greske, korisnik se preusmerava na ErrorPage gde se prikazuje greska. U headeru strane je meni za navigaciju. Ukoliko se u zahtevu za preusmeravanje na ErrorPage prosledi adresa za nazad kroz parametar Back, onda je i vidljivo Back dugme i klikom na njega se vraca na stranu gde se desila greska. Klikom na Meni dugme, vraca se na stranu sa glavnim menijem, a klikom na Exit, korisnik se izloguje. Ako u zahtevu za preusmeravanje postoji parametar Header i njegova vrednost je false, meni za navigaciju se ne prikazuje, nego samo deo sa porukom o gresci (**npr.** ako se greska desi u strani otvorenoj u iframe-u, kao ResultPage, nema svrhe u iframe-u otvarati prethodnu stranu ili meni ili se izlogovati sa iframe-a, nego se u njemu prikaze samo greska, a navigacija se vrsi sa glavne strane).
2. **Stilovi** – ACTAWebStylesheet.css dokument koji se nalazi u projektu CommonWeb u folderu css, definise stilove koriscene u aplikaciji. Pri definisanju elemenata na strani, navodi se ID, runat=“server”, CSSClass. Visine elemenata su definisane u stilovima, a sirine, ako je potrebno, navode se na strani jer se razlikuju od strane do strane. Kada se prave tabele, redovi, celije, elementi u celijama, pazi se da se sirine uklapaju. Ako se u celiji prikazuje neki element koji treba da je posebno poravnat, **npr.** centrirano dugme, mora se pri definiciji celije navesti i HorizontalAligment. Probala sam da u definiciji stila postavim vrednost text-align, ali se ona ipak ne odnosi na poravnanje samog elementa u okviru celije. Ako se zele postaviti jos neke vrednosti koje nisu obuhvacene opstim karakteristikama pri definisanju stilova (**npr.** AutoPostBack property i slicno), navode se u definiciji elementa. Stilovi definisu boju pozadine, border-e, poravnanje teksta, boju teksta, font.
3. **Menjanje boja (stila prikaza aplikacije)** – Kompletno menjanje boja aplikacije se vrsi tako sto se postave sve boje u ACTAStyleSheet.cs, par boja u Constants.cs u u js funkcijama na ResultPage.aspx i SmallResultPage.aspx. Potrebno je i napraviti nove slicice za tabove.
4. **Izvestaji** – Na klik dugmeta za generisanje izvestaja izvrsava se btnReport\_Click() metoda. Prema parametrima smestenim u Session prilikom pretrage, najpre se pronalazi broj rekorda koji zadovoljava te uslove. Ako ima pronadjenih rekorda za izvestaj, pronalaze se svi ti rekordi i smestaju u DataTable, a zatim se prolazeci kroz svaki red dobijene tabele, popunjavaju tabele dataset-a za izvestaj. U Session se smeste dataset kao i polja znacajna za prikaz izvestaja i preusmeri se na ReportPage stranu koja ce u svom iframe-u prikazati izvestaj cija se adresa prosledi kroz Report parametar adrese. Strana koja se prikazuje u iframe-u kreira odgovarajuci Crystal Report i prikaze ga u CrystalReportViewer. Kada se aplikacija pokrece iz okruzenja, izvestaji se prikazuju u potpunosti. Kada se publikuje, bilo je problema sa prikazom panela za navigaciju, odnosno, celog toolbar-a izvestaja. Problem je bio u tome sto je odgovarajuci folder C:\inetpub\wwwroot\aspnet\_client\system\_web\2\_0\_50727, koji treba da sadrzi potrebne slicice, script-ove, .css file-ove, bio prazan. Taj folder se popunjava potrebnim file-ovima pri instalaciji Crystal Report-a za VS2008, a ne pri publikovanju aplikacije. U sledecem delu o instalaciji bice detaljnije objasnjen redosled instalacije.
5. **Izveštaji – publikovanje:**
6. .rdlc files – property Build Action treba postaviti na Content umesto Embedded Resource.
7. U web.config. treba u sekciju <assebmlies> dodati linije:

<add assembly="Microsoft.ReportViewer.WebForms, Version=9.0.0.0, Culture=neutral, PublicKeyToken=B03F5F7F11D50A3A"/>

<add assembly="Microsoft.ReportViewer.Common, Version=9.0.0.0, Culture=neutral, PublicKeyToken=B03F5F7F11D50A3A"/>

1. U web.config treba u sekciju <httpHandlers> treba dodati liniju:

<add verb="\*" path="Reserved.ReportViewerWebControl.axd" type = "Microsoft.Reporting.WebForms.HttpHandler, Microsoft.ReportViewer.WebForms, Version=9.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a" />

d)Na serveru treba ručno dodati http hadler u IIS na način koji je opisan u file-u Post.aspx.htm, i treba ga dodati na nivou Default Web site-a, a ne na nivou aplikacije ACTAWeb koja je ispod.

1. **Izveštaji – eksport u Excel file –** treba koristiti liste umesto tabela u izveštajima u kojima ima subreport-a, dakle field subreport treba da bude član liste, a ne u ćeliji u tabeli.
2. **Meni** – U Menu kontroli se podesi orjentacija menija (horizontalno ili vertikalno) i u sekciji <Items> se dodaju MenuItem-i. Stil statickih i dinamickih item-a kao i hover stil se podesi u kontroli menija. Da bi se postiglo rastojanje medju padajucim menijima, u stilovima za menu item-e sam stavila minimalni border cija je boja ista kao boja pozadine. Pokusavala sam sa podesavanjem Horizontal i Vertical Offset i Item Spacing, ali ostaju bele praznine koje nikako ne mogu da postavim na boju pozadine. Ako se ne zeli rastojanje, stavi se border neke druge boje. Text se statickim menu item-ima dodeljuje tako sto se metodom FindItem(valuePath) menu kontrole pronadje odgovarajuci menu item po zadatoj vrednosti, a onda mu se dodeli tekst. Najpre sam i menu item-e iz padajucih menija, dodavala na samoj stranici, u sekciji <Items>, kao pod item-e odgovarajuceg menu item-a, ali je dodeljivanje teksta dinamickim menu item-ima, tj. item-ima iz padajucih menija sam bilo malo komplikovano. Pokusavala sam sa FindItem(valuePath) (**npr.** “Passes/Passes 1”) gde je “Passes” value statickog menija, a “Passes 1” value jednog od njegovih podmenija, ali se na taj nacin nista ne pronadje). Tako da sam za one menije koji imaju podmenije prolazila kroz sve ChildItems, i u switch-u zadavala tekst prema value vrednosti menija. Na ovaj nacin mora jako da se pazi kada se meniji dodaju, brisu, premestaju, menjaju putanje… mora se kompletno ispravljati kod za dodelu teksta. Onda sam na samoj stranici ostavila samo vrsne menu item-e, a u setLanguage() metodi koja se poziva u Page\_Load(), postavljala tekst vrsnim menijima i kreirala MenuItem-e i dodavala ih odgovarajucim vrsnim menu item-ima. I na ovaj nacin mora da se pazi kada se menja struktura menija, ali je malo preglednije, nema foreach, switch, mislim da je lakse izvrsiti potrebne izmene.
3. **Tabovi** – Tabovi na stranici se postizu koriscenjem menija i MultiView objekta. Izborom odredjenog menija, postavlja se kao aktivan odgovarajuci View objekat. Koriscenjem razlicitih CSS stilova za menu item-e i selektovane menu item-e, dodavanjem slicica za MenuItem.SeparatorImageUrl i MenuItem.ImageUrl, postize se karakteristicni izgled tabova na strani. Ako se tabovi zele u okviru nekog postojeceg formata stranice, na toj stranici se stavi iframe u kome ce se otvarati stranica sa tabovima.

**Dodavanje novog taba** – Ako se zeli dodati novi tab odgovarajucoj kategoriji user-a, mora se dodeliti odgovarajucoj strukturi u Constants.cs (CategoryTabs), zatim se dodaje novi MenuItem i novi View u TabPage.aspx, i u TabPage.cs se odgovarajucem MenuItem-u dodeli tekst, a odgovarajucem iframe-u se dodeli url zeljene stranice koja ce se prikazati u novom tab-u.

**Brisanje taba** – U TabPage.cs se izbrisu odgovarajuce dodele teksta i url-a, iz TabPage.aspx se brise odgovarajuci MenuItem i View, iz Constants.CategoryTabs se izbrise tab iz odgovarajuce kategorije user-a.

1. **Prikaz proizvoljno iscrtane grafike** – Jedna strana sluzi za iscrtavanje grafickog objekta. Pri load-ovanju te strane se kreira Bitmap objekat, a zatim i Graphics objekat. Iscrta se zeljena grafika a zatim se Bitmap sacuva na sledeci nacin:

// Draw graphics

Bitmap barBmp = drawBar(width, height, iopair);

MemoryStream memStream = new MemoryStream();

// Render BitMap Stream Back To Client

barBmp.Save(memStream, ImageFormat.Png);

memStream.WriteTo(Response.OutputStream);

Ako se preusmeri na ovu stranicu, prikaze se samo grafika. Ako se ova stranica stavi kao image url Image ili ImageButton objekta, dobije se zeljena grafika u okviru postojece aplikacije. **npr.** BarSegment.aspx i EmployeeWorkingDayView.aspx.

1. **Prenos podataka izmedju stranica** – Podaci medju stranicama prenose kroz Request string ili kroz Session. Preko Request stringa se prenose stringovi, a preko Session mogu i objekti razlicitih tipova.

**napomena:** Pri pozivu stranica koje ‘kupe’ podatke iz Session, mora se dobro paziti da su podaci prethodno ispraznjeni (ako je zaostalo nesto sa druge strane) i napunjeni odgovarajucim vrednostima.

1. **Cuvanje podataka na jednoj strani** – Ako je potrebno podatke cuvati i koristiti iste vrednosti na jednoj strani, NIKAKO ne koristiti staticke promenljive jer se ta vrednost deli medju svim korisnicima aplikacije, odnosno te strane. Ili podatke prenositi kao parametre metoda ili ih cuvati preko ViewState-a (na ovaj nacin mogu da se cuvaju prosti tipovi, ResultPage.aspx – primer modifikovanja get i set metoda i koriscenje ViewState-a) ili preko Session-a (cuva sve tipove objekata).
2. **Cuvanje podataka stranice pri navigaciji medju stranicama** – U CommonWeb.Misc postoje dve metode Dictionary<string, string> SaveState(Control ctrl, string key, Dictionary<string, string> filterState) i void LoadState(Control ctrl, string key, Dictionary<string, string> filterState). Kada se sa stranice navigira na sledecu (sa koje se po potrebi moze vratiti na prethodnu) i zeli se sacuvati stanje postojecih kontrola na stranici da bi se isto prikazalo pri povratku, poziva se metoda SaveState sa parametrima te stranice, naziva stranice i praznog dictionary-ja koja rekurzivno prolazi kroz sve kontrole te stranice i kada naidje na kontrole odredjenog tipa, smesta njihove ID-eve i vrednosti u dictionary filterState. Rezultat ove metode je dictionary ciji key su nazivi kontrola (<naziv stranice>.<id kontrole>) a vrednosti su vrednosti odgovarajuce kontrole. Taj dictionary se smesti u Session i na taj nacin je stanje stranice sacuvano. Nakon povratka na stranicu poziva se metoda LoadState sa parametrima te stranice, nazivom stranice i dictionary filterState koji sadrzi stanje kontrola stranice preuzeto iz Session. Ova metoda rekurzivno prolazi kroz sve kontrole na stranici i odgovarajuce kontrole postavlja na sacuvane vrednosti. Trenutno su u ovim metodama obradjene kontrole tipa TextBox, DropDownList, CheckBox i RadioButton. Kako se na stranicama pojavljuju kontrole drugih tipova cije stanje treba cuvati, tako treba prosirivati ove dve metode.
3. **Dinamicko dodavanja User control-a** – Kada se user kontrole dinamicki dodaju, **MORA** im se eksplicitno zadati ID, inace se nece ponasati na ocekivan nacin, **npr.** dodeljeni dogadjaji se ne ‘okidaju’ svaki put.
4. **Event Bubbling (prenos podataka/dogadjaja od kontrole prema stranici koja je sadrzi)** – U kontroli se definisu EventHandler i OnBubbleClick metoda:

public event ControlEventHandler BubbleClick;

protected void OnBubbleClick(ControlEventArgs e)

{

if (BubbleClick != null)

BubbleClick(this, e);

}

Kada se kontrola dodaje na parent strani, dodeli joj se event handler:

pairData.BubbleClick += new ControlEventHandler(BubbleControl\_BubbleClick);

void BubbleControl\_BubbleClick(object sender, ControlEventArgs e)

{

try

{

// action on parent page

}

catch (Exception ex){}

}

Kada se iz kontrole zeli izazvati neka akcija na parent strani, poziva se metoda OnBubbleClick, koja prosledjuje zahtev i izaziva izvrsenje akcije na roditeljskoj strani. Parametri i handler-i mogu biti i posebno definisane klase koje nasledjuju EventArgs, odnosno delegati. **npr.** ControlEventArgs.cs, IOPairDataUC.ascx i IOPairBarChange.aspx.

1. **Tooltips** – Na kontrole koje sadrze tooltip kao property, dodeljuje mu se zeljena vrednost i ta vrednost se cuva prilikom PostBack-a. Na ListViewItem-e, tooltip se dodaje kroz dodavanje atributa “title”. Ovaj atribut se ne cuva u pri PostBack-u pa da se tooltip-ovi ne bi gubili, potrebno je dodeliti ih u metodi koja se dodeli OnPreRender dogadjaju kontrole.
2. **Pisanje u log file** – IIS user-u treba dati prava nad odgovarajucim folder-om, ako vec nisu data. U <root directory> key u Web.config uneti putanju root folder-a (“C:\inetpub\wwwroot\ACTAWeb”). U suprotnom cesto dolazi do greske pri automatskom trazenju i odredjivanju root folder-a aplikacije.
3. **javasript funkcije** – U Functions.js file-u se nalaze mnoge js funkcije (ukidanje desnog klika, za postavljanje kursora na default, za selekciju/deselekciju svih item-a iz liste, za otvaranje pop-up prozora…). Da bi se nekom klijentskom dogadjaju dodelila odgovarajuca metoda, potrebno je ukljuciti ovaj file na .aspx delu strane, a zatim odgovarajucem atributu dodati ovu metodu. Nakon postavljanja default kursora na strani, mnoge kontrole ostaju bez karakteristicnog kursora ‘rucice’ kada se prelazi preko njih. Ovo se resava dodavanjem atributa onmouseover i onmouseout odgovarajucoj kontroli.

**js funkcije na serverskoj strani** – js se na serverskoj strani izvrsava na sledeci nacin:

ClientScript.RegisterStartupScript(GetType(), "close", "window.close();", true);

1. **Default button** – Ako se zeli definistai default button na strani, odnosno dugme koje reaguje na Enter, ID zeljenog dugmeta se dodeli defaultbutton atributu taga <forms> na strani.
2. **Password Text Box** – Pri menjanju lozinke potrebno je mode text box-a staviti na password. U tom slucaju se pri PostBack-u ne cuva sadrzaj text box-a. Umesto TextBox kontrole se koristi modifikovana kontrola PasswordTextBox koja nasledjuje TextBox kontrolu. Ova kontrola je u CommonWeb u namespace-u opp. Da bi mogla da se koristi na strani, potrebno je u <pages><controls> sekciju Web.config-a dodati liniju

<add tagPrefix="opp" assembly="PasswordTextBox" namespace="opp"/>

**npr.** ChangePassword.aspx.

1. **Konekcija sa bazom** – Ne sme se NIKAKO koristiti staticka konekcija kao u ACTAAdmin aplikaciji. Na taj nacin svi ulogovani korisnici koriste jednu konekciju i aplikacija vrlo cesto ‘puca’. Za svako pokretanje aplikacije, pravi se nova konekcija i smesta u Session, a zatim se svugde koristi ta konekcija. Pri pravljenju Common objekata, u konstruktoru im se OBAVEZNO prosledjuje konekcija iz Session-a. Ova konekcija se zatvara u Session\_End() u Global.asax ili nakon sto istekne Session timeout ili nakon regularnog izalska iz aplikacije. Ako se iz aplikacije izadje na neregularan nacin, SQL Server sam unisti konekciju nakon nekog vremena.
2. **Cache-iranje** – Desavalo se da na nekom IE (npr. IE9 na XP-u) vec otvarane .aspx strane od strane user control-e kesiraju negde u memoriji i ako kontrola sa istim id-em ponovo zahteva istu stranu, otvara se kesirana. Da se to ne bi desavalo, u Page\_Load metodi zahtevane strane se ‘iskljuci’ kesiranje na sledeci nacin:

Response.Cache.SetCacheability(System.Web.HttpCacheability.NoCache);

1. **Pop-up** – U Functions.js se nalaze metode za otvaranje pravih pop-up prozora (otvaraju se sa window.showModalDialog()). Ako se pre zatvranja prozora zeli izvrsiti neka akcija, zatvaranje se izvrsi js, ali preko registracije script-a.

ClientScript.RegisterStartupScript(GetType(), "close", "window.close();", true);

Ako se zatvaranjem prozora zeli izavati neka akcija roditeljskog prozora, mora se izazvati PostBack roditelja i na taj nacin ‘odraditi’ zeljenu akciju na roditeljskom prozoru. **npr.** WUUnitsPicker.aspx (slekcija cvora drveta) ili IOPairBarChange.aspx i TLTMDataPage.aspx.