

## **Specijalna poglavlja softwareskih sistema**

**Decembar 2003**

### **OPIS ARHITEKTURE I TIPIČNIH FUNKCIJA SCADA SOFTWARESKIH PAKETA SA MMI SOFTWAREOM**

#### ***Opis IWS softwareskog paketa***

Indusoft Web Studio ( IWS ) je softwareski proizvod namjenjen implementaciji MMI interfejsa i SCADA funkcija u industrijskom ambijentu primjene PC baziranih sistema nadzora, vođenja i upravljanja tehničkim i poslovnim sistemima.

Fleksibilnost IWS softwareskog paketa dozvoljava korisniku da dizajnira i implementira aplikacije za:

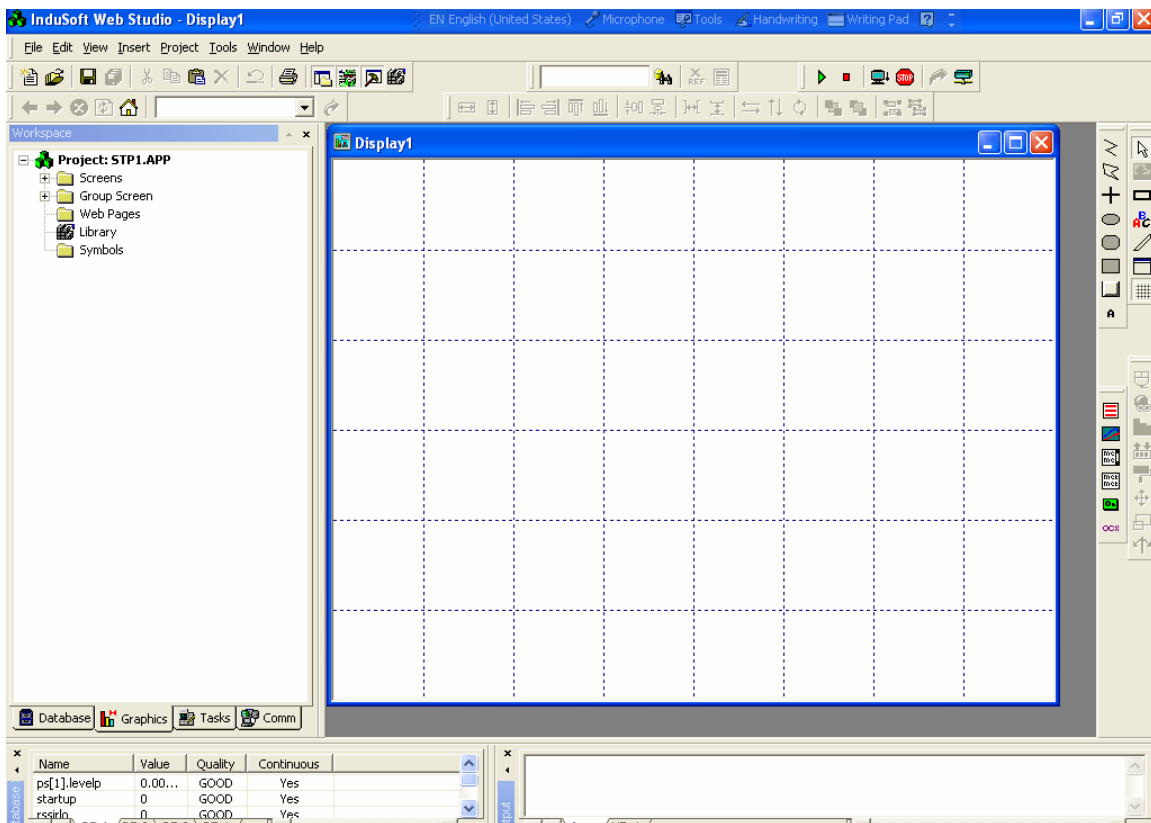
- prikupljanje podataka
- interfejs operator sistem ( man( human) machine interface MMI ,HMI)
- Koncentratori podataka u distribuiranim procesima
- Udaljene nadzorne stanice ( remote supervisory stations)
- Podsystemi u integrisanim sistemima vođenja tehničkih i poslovnih sistema ( MIS – management information systems, i EMS enterprise..)

IWS aplikacije se izvršavaju na PC baziranim konfiguracijama ili embeded sistemima ( sa WinCE OS ), u realnom vremenu povezanim sa procesom putem PLC-jeva, I/O uređaja ili drugih akvizicionih podsystema.

IWS je skup bogate i raznovrsne kolekcije softwareskih alata koja uključuje sve gradivne elemente neophodne da se razvijaju moderni MMI (HMI) i SCADA sistemi.

## IWS razvojni okružaj

Nakon starta programa , pojaviće se programski okružaj kao na slici :



IWS obezbjeđuje alate potrebne da korisnik kreira SCADA i MMI sistem sa mnogim inovativnim funkcionalnostima kao što su:

### ***Jednostavan i objektno orijentisani ekranski editor***

IWS ekranski editor dozvoljava korisniku da kreira razne vrste prozora i dijaloga, koji omogućavaju unos podataka selekcijom na ekranu i tastaturi, izbacivanje vrijednosti na proces, i automatsko ažuriranje ekrana na bazi ulaza sa procesa. Ostale mogućnosti ekranskog editora su:

- grupiranje objekata koje sačuvava korake njihove konstrukcije od individualnih objekata
- editiranje bez da se moraju degrupisati interne komponente i podgrupe uključene u grupu
- kompletna manipulacija bitmap objekata i bitmapa u backgroundu
- podrška za liniju statusa u aplikacionim prozorima i dijalozima

## Objektno orijentirana baza podataka

**Array tags** : Svaki tag ( varijabla) u bazi podataka može biti definiran kao polje od do 512 ulaza. Na bilo kojem mjestu u softveru gdje se koristi ime varijable, moguće je koristiti **Tag [ number ]** ili **Tag [ drugi tag ]**.

Polja pojednostavljaju mnoge konfiguracije i dozvoljavaju korištenje multipleksiranja na ekranima, recepturama, i komunikacionom interfejsu. Takodjer , ovo štedi i vrijeme kada se tagovi deklarišu.

**Indirect tags ( pointers)** : Korištenje *@Tag* konstrukta omogućava indirektno čitanje ili upisivanje. Naprimjer, ako **X** tag ima vrijednost "Setpoint" , i koristimo **@X** konstrukt, tada ćemo ustvariti čitati ili upisivati u vrijednost Setpointa ( zadate vrijednosti).

**Classes ( klase)**: Korisnik može definisati strukturu podataka kao što su : *ClassPID { PV, CV, SET, KP, KD}* i deklariše tagove ili čak i polja tipa *ClassPID* koji će imati grupu vrijednosti umjesto samo jedne vrijednosti.

Korisnik može čak kombinirati sve karakteristike tagova ( tj. kao polja, pointere i klase) u kombinacije oblika:

@Tag[drugi tag ] Tag[drugi tag ].SP.

## Matematske funkcije

IWS posjeduje interni programski jezik koji se koristi kod pisanja logičkih i matematskih relacija i matematskih kalkulacija koje mogu biti potrebne u različitim aplikacijama.

## Online konfigurisanje

Runtime taskovi odmah prihvataju nove konfiguracije i bez potrebe da se restarta ili rekompilira program. Korisnik može promjeniti bilo koji element konfiguracije, uključujući matematske kalkulacije, izvještaje, adrese u PLC uređaju, ili tip taga u bazi podataka. Sve ove promjene mogu se vršiti u letu ( on the fly) , bez zaustavljanja aplikacije ili procesa. Korisnik može takodjer izvršavati emulaciju aplikacije na razvojnom računaru i testirati je prije downloada na ciljnu runtime stanicu.

## Lako dodavanje simbola

Ponovo koristivi objekti ili grupe objekata koje pohranjujemo za ponovno korištenje se nazivaju simbolima. U vrlo kratkom vremenu možemo dodati novi simbol ili modificirati postojeći simbol, koji nam omogućava da ponovo koristimo simbole kako razvijamo našu aplikaciju.

## **Generator izvještaja**

IWS ima sve alate koji su potrebni korisniku da generiše i pohrani na disk izvještaje koji sadrže i tekst i grafiku, bez da se zahtjeva korištenje drugih softwareskih paketa kao što je naprimjer MS Excel.

## **Recepture u ASCII**

IWS baza podataka podržava direktni pristup fajlovima receptura napisanim u ASCII.

## **DDE, NetDDE, ODBC i OPC**

IWS obezbjedjuje DDE, NetDDE i OPC kao i ODBC interfejs za pristup relacionim bazama u WINDOWS runtime okruženju. Windows CE ( compact edition , Windows diskless OS za embedded konfiguracije), takodjer podržava OPC.

## **Sistem sigurnosti**

Interni sistem sigurnosti dozvoljava korisniku da doznači dozvole za individualne korisnike i korisničke grupe, i to svako sa svojom lozinkom. Korisnik može primjeniti ograničenja sigurnosti na aplikacije koje kreira sa IWS-om i može takodjer ograničiti pristup razvojnim alatima koji postoje u IWS-u samo na specifične korisnike. Sistem sigurnosti se takodjer može primjeniti i na udaljene klijente, spojene preko Intraneta ili Interneta.

## **Batch historijski fajlovi**

Mogućnosti skupljanja podataka u IWS-u dozvoljavaju korisničkim aplikacijama da pohrane i vade historijske podatke iz fajlova, koristeći arhivske fajlove na bazi imena fajla po vremenu ili imenima koje definiše korisnik, što je vrlo važna karakteristika za batch sisteme.

## **Alarmi**

IWS aplikacije mogu uključiti slijedeće karakteristike alarma:

- slobodno formatirane alarmne poruke
- korištenje sekundarnog ključa za pretraživanje
- pristup alarmima preko grupa ili tagova

## **Razvojna podrška**

IWS sadrži mnoge alate da pomogne korisniku da razvije svoju aplikaciju, kao naprimjer lagani korisnički interfejs, registar poruka ( message register), i kodovi događaja ( event codes) koji se koriste za vrijeme izvršenja programa. Razvojni alati takodjer dozvoljavaju direktni pristup bazi radi kreiranja i verifikacije varijabli.

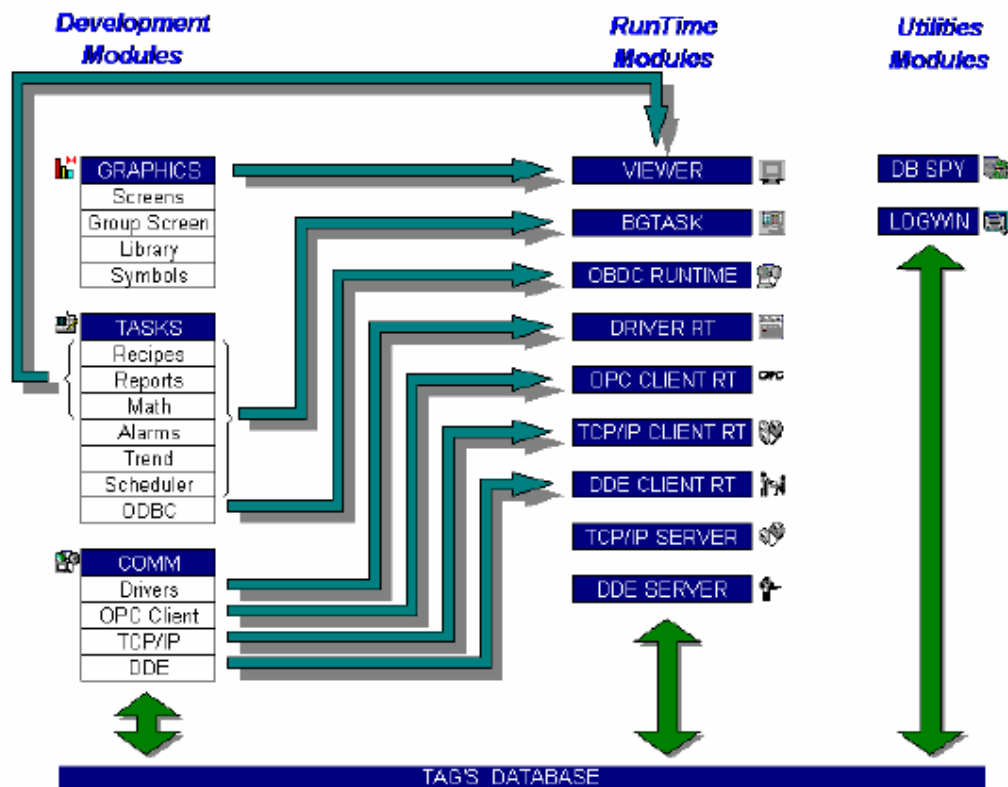
## **IWS aplikacioni programski interfejs ( API)**

Svi IWS softwareski moduli su razvijeni koristeći otvorene API-je. Ovo omogućava lagani razvoj i proširenje IWS-a pošto su razvojni "kernel" i aplikacioni taskovi odvojeni. API takodjer dozvoljavaju korisniku da kreira nove softwareske module, u bilo kojem jeziku koji obezbjedjuje .DLL podršku.

## **Dinamička podrška za različite jezike korisnika**

Da bi se krirala aplikacija koja podržava različite jezike operatora, treba kreirati aplikaciju u primarnom jeziku, a zatim kreirati tabelu koja pokazuje tekst u drugim jezicima. Nakon toga sistem može koristiti ovu tabelu za prevodjenje. Ova mogućnost dozvoljava kreiranje aplikacija koje dinamički mjenjaju interfejsni jezik za vrijeme izvršenja.

## IWS interna struktura



### Tag baza podataka

Tag baza podataka je srce IWS softwarea. U IWS-u, korisnik koristi ista imena tagova i u worksheetovima i na displejima, a IWS koordinira sa vrijednostima tagova izmedju modula. Svi moduli dijele informacije putem aplikacione baze podataka. Vrijednosti aplikacionih tagova i IWS internih tagova se pohranjuju u ovoj bazi za vrijeme izvršenja sistema. Aplikaciona baza je medij koji se koristi od strane svih modula da bi čitali ili upisivali vrijednosti.

Konfigurisanje aplikacije se sastoji od definiranja koji tagovi će se koristiti od strane svakog modula. Ovo znači da razvoj aplikacije slijedi istu logičku sekvencu, bez obzira na broj tagova koji je uključen u specifičnu aplikaciju.

### Razvoj modula

#### Grafika

Osnovna funkcija koju izvršava IWS je da obezbjedi **prozor u proces**. Ova mogućnost da prikaže status procesa, putem interakcije sa instrumentacijom ili

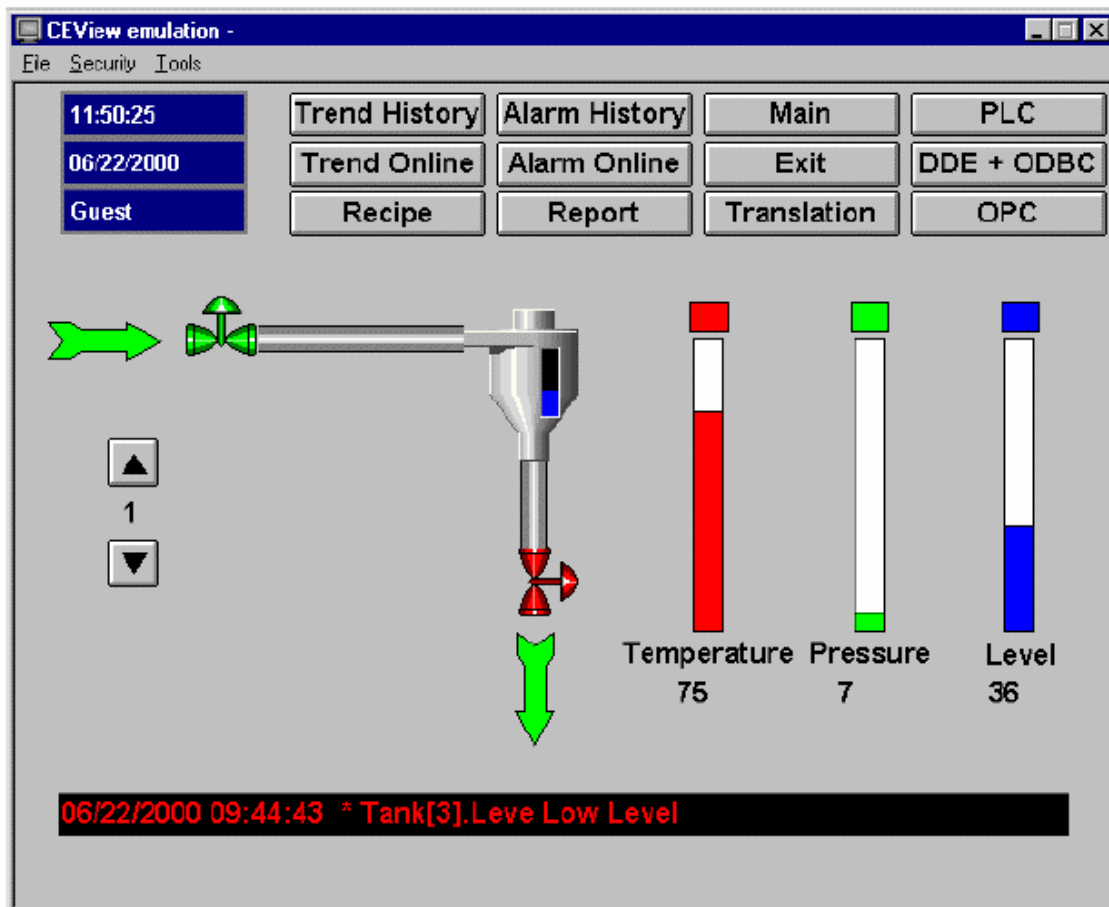
računarima, je opisana kao interfejs čovjek – sistem ( ljudsko biće – sistem ) = MMI (HMI ).

Aplikacije koje kreira IWS mogu nadzirati proces koristeći ekrane visoke rezolucije. IWS grafički alati se sastoje iz dva modula: **editora worksheetsa**, u okviru IWS desktopa i aplikacije u runtime **Viewer-u** .

Worksheet editor je ono što korisnik koristi da bi kreirao ili importovao grafiku. Grafički objekti ili simboli mogu biti dinamički objekti koristeći animacione linklove. Animacioni linkovi mogu učiniti da objekat ili simbol mjenja izgled da bi odrazio promjene u vrijednosti taga ili izraza. Svaki ekranski prikaz je skup statičkih i dinamičkih objekata.

Ekрани mogu imati opcionu bitmap sliku koja djeluje kao pozadina ( background) u objektnom prozoru. Naprimjer, statičke slike na slijedećem ekranu mogu biti dio bitmape pozadine objekta, a objekti sa animacijom u sloju dinamičkih objekata mogu odražavati promjene u procesu.

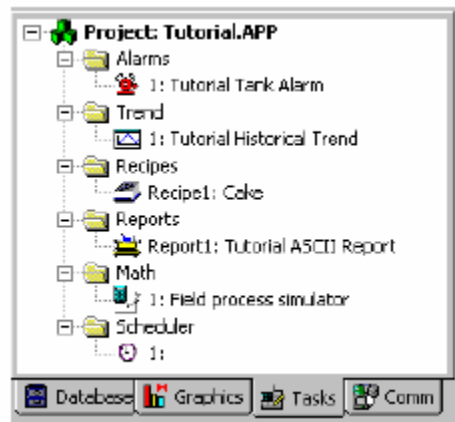
Na taj način korisnik može imati iluziju da je slika na ekranu trodimenzionalna:



Program Viewer-a omogućuje korisniku da vidi sliku emuliranu u runtime prozoru.

Svi IWS konfiguracioni taskovi zahtjevaju Windows selekcionu uredjaj tipa miša ili touch pada ili ekrana ( touch screen). Moguće je izvršavati aplikaciju u Vieweru bez miša ili touch ekrana ako konfiguriramo tasterske kombinacije za sve komande.

## IWS taskovi



IWS taskovi su okruženje gdje konfiguriramo worksheetove specifične za aplikaciju, od kojih se svaki sastoji od zaglavlja ( headera), gdje se definira globalna informacija u listu, te od tijela, gdje se koriste tagovi i izrazi u svakom tasku koji se konfigurirše.

### Alarmne grupe

Ovdje korisnik definira alarmnu grupu, njene karakteristike, i poruke koje se izvještavaju o alarmnim uslovima. Glavna namjena ovih alarma je da informiraju operatore o problemima ili promjenama stanja u procesu , tako da oni mogu poduzeti korektivne akcije.

Da bi se pokazale alarmne poruke na ekranu, korisnik mora kreirati alarmni objekat na ekranu.

### Trend grupe

Trend grupe vode evidenciju o ponašanju procesnih varijabli u vremenu. Korisnik može pohraniti uzorke u historijski fajl i pokazati i historijske i online ( real time ) uzorke u okviru trend grafova na ekranima.

### Recepture

Ovaj modul dozvoljava da očitavamo i upisujemo ASCII fajlove iz i na hard disk, on prenosi vrijednosti izmedju fajlova i memorije realnog vremena. Koristi se tipično kod pohranjivanja procesnih receptura, ali ovi fajlovi mogu pohraniti bilo



koji tip informacije kao što su operatorski logovi, lozinke, itd. Omogućava korisniku da pohranjuje podatke u XML formatu.

### **Izveštaji**

Ovaj modul se koristi da se konfiguriraju izvještaji sa sistemskim podacima, u bilo ASCII ili RTF formatu. Glavna namjena ovog modula je da olakša kriteranje izvještaja.

### **Matematski worksheet**

Ovaj modul dozvoljava implementaciju dodatnih rutina koje će se izvršavati sa osnovnim funkcijama IWS modula. Matematski worksheet je grupa programskih linija koje se izvršavaju kao jedan od pozadinskih taskova. Korisnik može konfigurirati matematske izraze u blokovima u raznim worksheetima.

Ovaj worksheet obezbjeđuje slobodni okružaj za logičke rutine i matematske kalkulacije koje projekat može trebati. Skript jezik koji se koristi za ove namjene je jednostavan i lagan za korištenje.

### **Šeduler ( scheduler)**

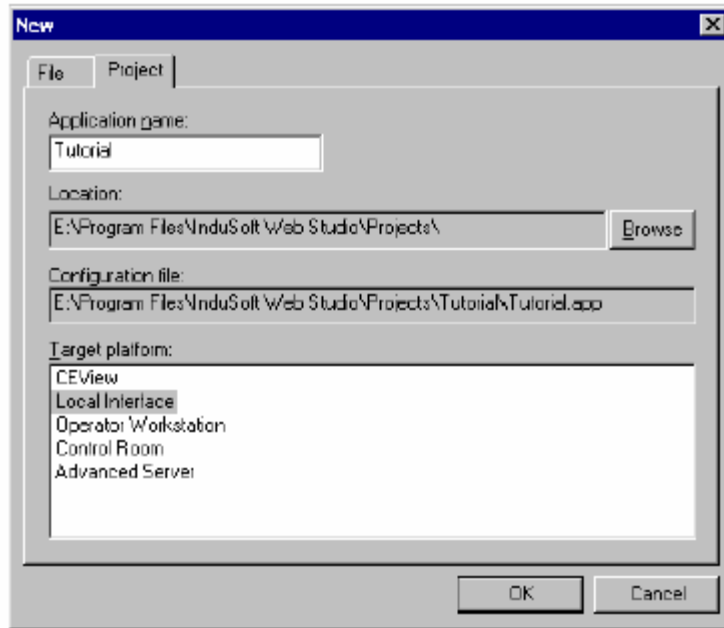
Ovaj modul generira vremenske baze koje se koriste u aplikaciji i koje su u stanju da trigeruju događaje.

### **ODBC konfiguracija ( nije moguća kod WinCE aplikacija )**

ODBC interfejs dozvoljava IWS aplikacijama da pristupaju bilo kojoj bazi kompatibilnoj sa ODBC protokolom, kao što je Access , Oracle, SQL Server itd.

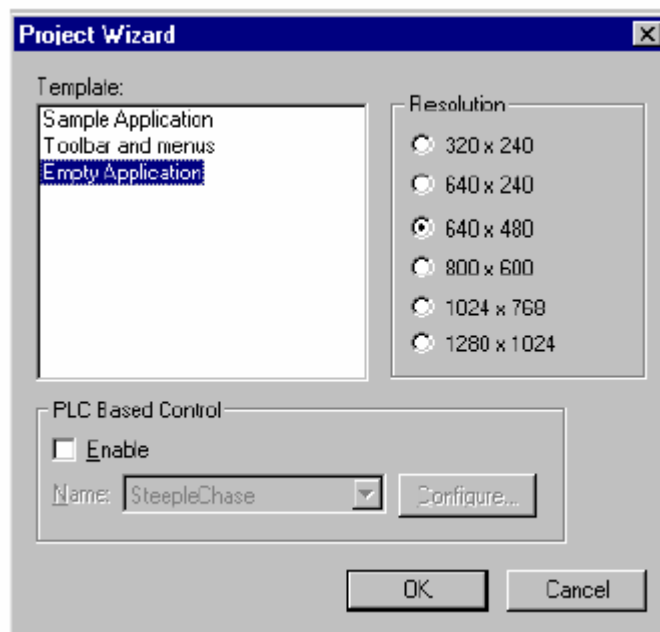
### **Kreiranje novog projekta**

Da bi pokazali rad sa IWS pročit ćemo kroz jedan demonstracioni primjer. Nakon startanja programa, pojaviće se ekran sa kojeg ćemo izabrati **File>>New** . Unesimo ime projekta *Tutorial* .



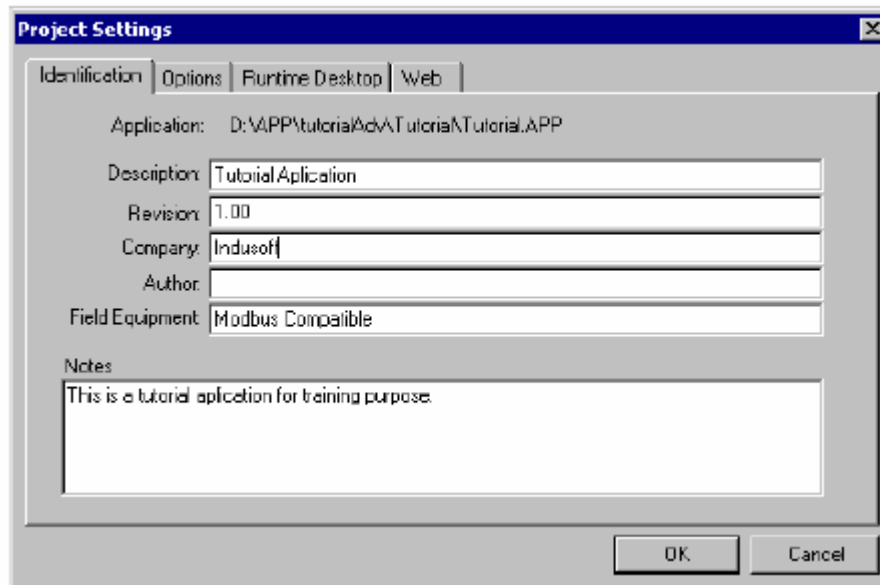
Za ciljnu platformu gdje će se aplikacija izvršavati u runtime-u možemo ostaviti default selekciju *Local Interface* .

Slijedeći prozor koji će se otvoriti je je Wizard za Project . Izabraćemo **Empty Application** opciju i 640x 480 , radio button za rezoluciju.



## Konfigurisanje setinga za projekat

U glavnom meniju treba selektirati **Project>>Settings** .. da se otvori prozor za projektne postavne vrijednosti (setinge). **Identification** tab je rezerviran za dokumentaciju o projektu. Ova polja su opcionalna.

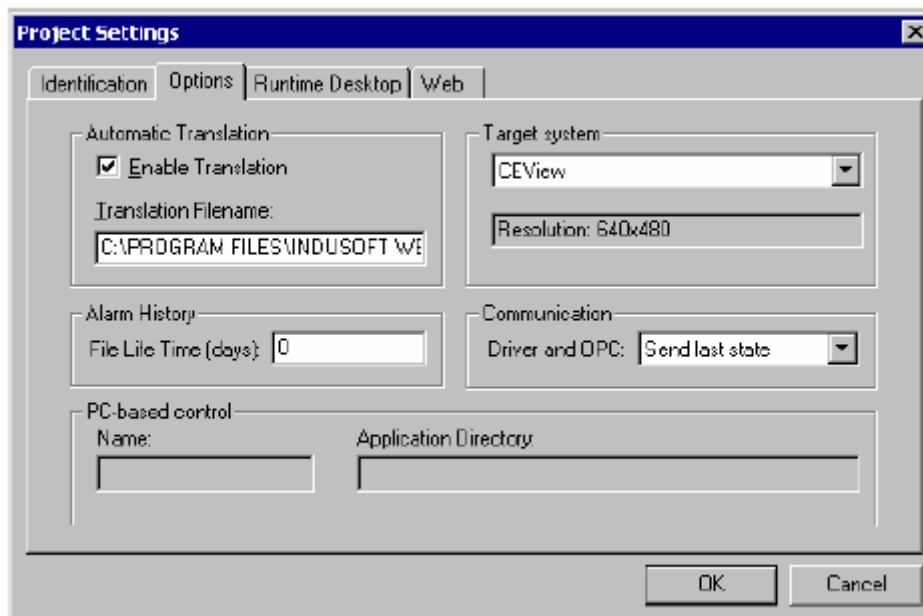


The screenshot shows the 'Project Settings' dialog box with the 'Identification' tab selected. The dialog has four tabs: 'Identification', 'Options', 'Runtime Desktop', and 'Web'. The 'Identification' tab contains the following fields:

- Application: D:\MPP\tutorial\Adv\Tutorial\Tutorial.APP
- Description: Tutorial Application
- Revision: 1.00
- Company: Indusoft
- Author: (empty)
- Field Equipment: Modbus Compatible
- Notes: This is a tutorial application for training purpose.

At the bottom right, there are 'OK' and 'Cancel' buttons.

Options tab sadrži setinge za jezik za prevodjenje , ciljni sistem, PC bazirano upravljanje, baferovanje za komande za drajver, i opšte informacije o aplikaciji.

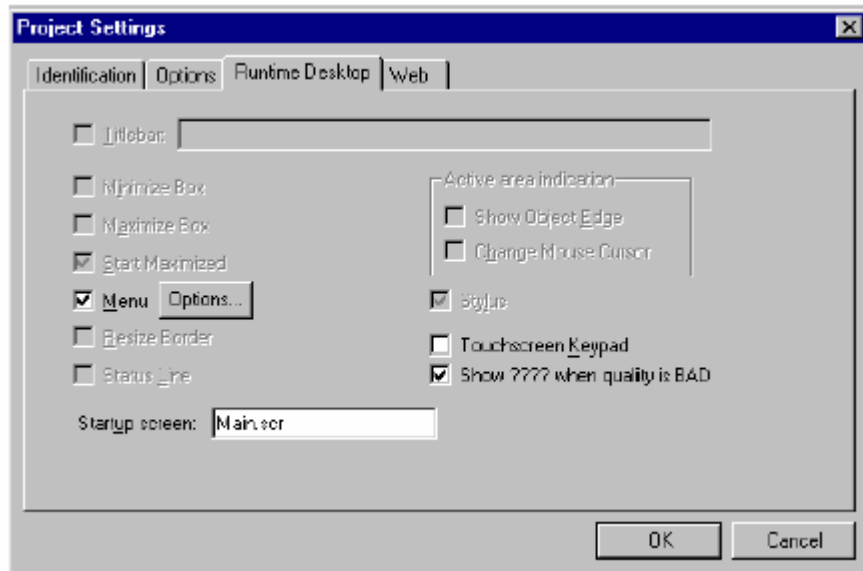


The screenshot shows the 'Project Settings' dialog box with the 'Options' tab selected. The dialog has four tabs: 'Identification', 'Options', 'Runtime Desktop', and 'Web'. The 'Options' tab contains the following fields:

- Automatic Translation:
  - Enable Translation
  - Translation Filename: C:\PROGRAM FILES\INDUSOFT WE
- Target system:
  - CEView (dropdown)
  - Resolution: 640x480
- Alarm History:
  - File Life Time (days): 0
- Communication:
  - Driver and OPC: Send last state (dropdown)
- PC-based control:
  - Name: (empty)
  - Application Directory: (empty)

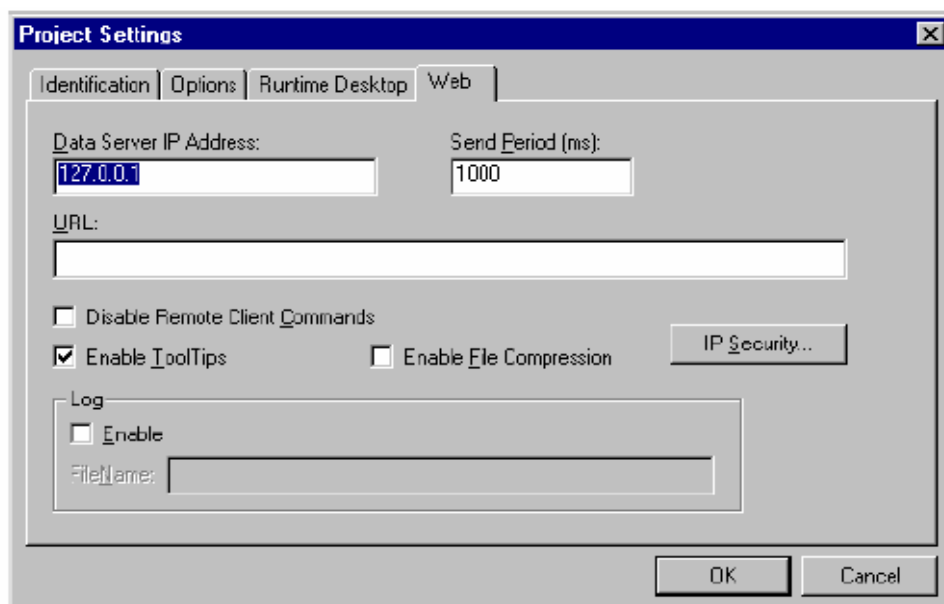
At the bottom right, there are 'OK' and 'Cancel' buttons.

**Runtime Desktop** tab sadrži globalne setinge za aplikaciju, koji određuju kako će se ona izvršavati na runtime radnoj stanici i koje opcije menija će biti raspoložive. Korisnik treba uvijek da specificira prvi ekran koji će se otvoriti u aplikaciji kada se izvršava u emulacionim modu ili na runtime radnoj stanici. Za ovaj tutorial ćemo unjeti Main.scr u polje **Startup screen** .



**Web** tab sadržava globalne setinge za udaljene thin klijente , koji će pristupati aplikaciji putem Web browsera kao što je Internet Explorer.

Ovi setinzi mogu biti konfigurisani bilo kada u toku razvoja projekta, ali poslije svake izmjene parametara nužno je izvršiti komandu **Verify Application** iz **Tools** menija , da bi se ažurirali HTML fajlovi sa novim setinzima.

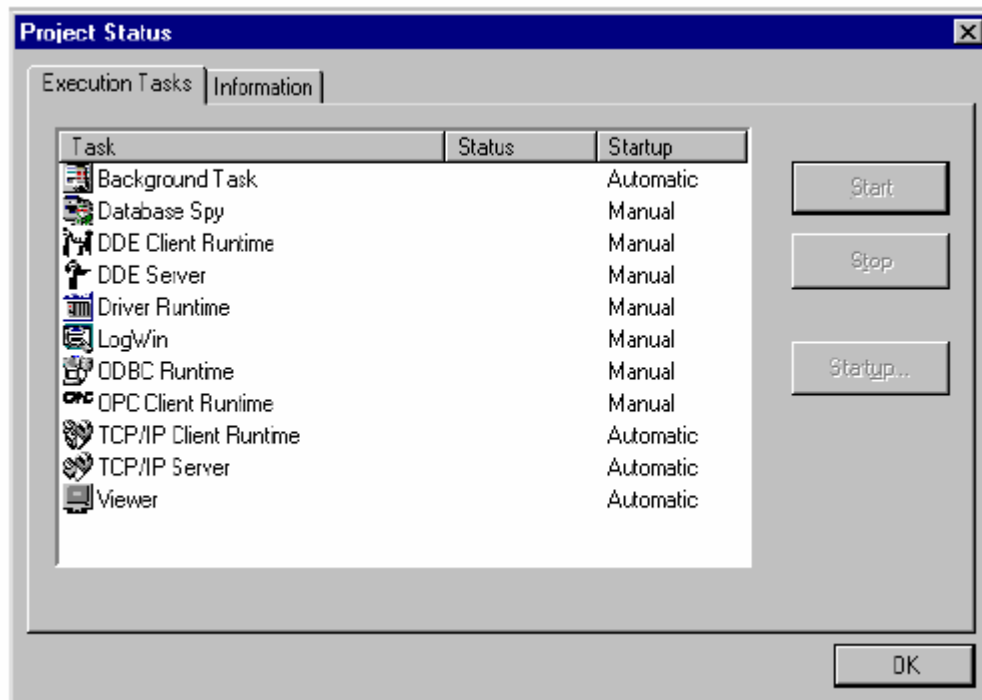


## Konfigurisanje statusa projekta

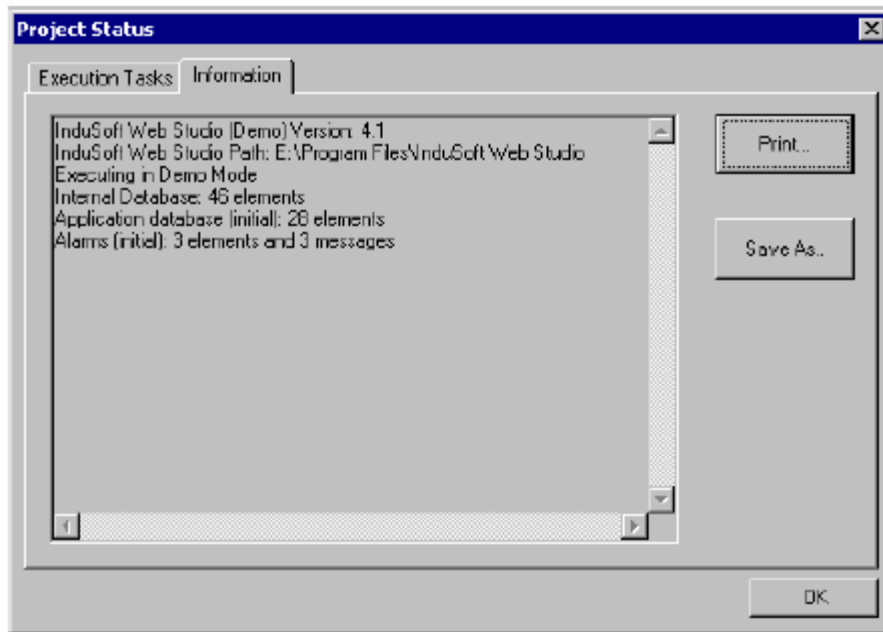
U glavnom meniju, izabрати **Project>>Status** da se otvori prozor Project Statusa. Ima dva taba.

**Runtime tasks** tab dozvoljava korisniku da nadzire i upravlja izvršenjem svakog runtime taska putem startanja i zaustavljanja taskova koristeći **Start** i **Stop** tastere.

**Startup** taster se koristi da konfigurira da li runtime task se starta sa *Automatic* ili *Manual* metodama. Ne koristi se u aplikacijama za lokalni interfejs pošto se svi potrebni runtime taskovi startaju automatski u ciljnom sistemu.



**System information** tab obezbeđuje neke opšte informacije o razvojnom sistemu i o aplikaciji :



## Konfigurisanje tagova baze podataka

Tagovi su varijable koje se koriste na ekranima i u worksheetovima taskova. Tagovi mogu biti komunikacione tačke u uređajima sa kojima PC je povezan, rezultati kalkulacija, alarmne tačke, itd. Tagovi koje kreira korisnik se nazivaju **aplikacioni tagovi**, tagovi koji su definisani od IWS programa se nazivaju **interni tagovi**.

Korisnik može koristiti obadva tipa tagova u svom programu sa bilo kojim IWS modulom, jedina razlika je u tome što interni tagovi imaju svoje unaprijed definirane funkcije.

Vrijednosti tagova se pohranjuju u aplikacionu bazu podataka.

Pravila sintakse za tagove su:

- može biti sastavljen od slova, brojeva, i znaka za podcrtavanje ( underscore \_ )
- mora početi sa slovom
- maksimalna dužina imena taga je 32 karaktera, a 16 karaktera za člana klase.
- Ime taga mora biti različito od imena internih tagova i matematskih funkcija.
- Imena tagova nisu case sensitivna ( velika i mala slova nemaju značenja )

Primjeri imena za tagove su : **temperatura, pritisak1, broj**

Kako je već rečeno IWS ne pravi razliku između malih i velikih slova mada se mogu koristiti radi jasnijeg pisanja tagova ( anpr. TankLevel umjesto tanklevel )

## Tipovi tagova

Vrijednost taga može biti jedna od četiri standardne vrijednosti :

**Boolean** ( 4 bajta ) : Bulova ili digitalna varijabla ( 0 ili 1 )

**Integer** ( 4 bajta ) : Cjeli broj ( pozitivni , negativni ili nula ). Ekvivalentan je long integeru u C programskom jeziku. ( sa opsegom – 2147483647 do 2147483647 )

**Real** ( pokretni zarez , 8 bajta ) : Realni brojevi interno pohranjeni kao dvostruka riječ . Ekvivalentan je tipu double kod C.

**String** ( ASCII , 256 bajta ) : Karakter string varijabla do 255 karaktera ( 0 do 254 ) , koja sadrži slova, brojeve, ili specijalne karaktere.

Primjer stringa : ***Recipe product x123 01-01-02 \*\*\* on \*\*\*\****

Svi tagovi se deklariraju u modulu aplikacione baze taba baze podataka. Pored ova četiri tipa koja su navedena, korisnik može definisati i novi tip koji se naziva **classes** ( klase ).

## Klase, polja i pointeri tagova

IWS tagovi mogu sadržavati jednostruku vrijednost ili polje vrijednosti. Tag polje je skup tagova sa istim imenom, ono je identificirano sa indeksima ( tj. to je matrica sa n linija i jednom kolonom ). Maksimalna veličina polja ( array ) je 512 ( pozicija od 0 do 511 ).

Naprimjer : ***tank [1] , tank [2], tank [3], tank [500]***

Treba koristiti tagove polja kada je to god moguće, jer to pojednostavljuje zadatak konfigurisanja. Predpostavimo da želimo da imamo prikaz koji nadzire svaki tank. Koristeći tagove polja , možemo konfigurirati jedan ekran koji sadrži tagove linkovane na bilo koji tank :

***Pritisak [tk] , temperatura [tk], temperatura [tk + 1 ]***

Tag tk je indeks koji sadrži broj željenog tanka. Indeks polja ( array ) može također biti tag , ili numerička vrijednost, ili izraz plus tag .

Da bi referencirali polje koje ima indeks sa aritmetičkim operatorom + u indeksu, moramo koristiti sljedeću sintaksu:

**< ime taga > [ < tag > + N ]**, gdje je N numerička konstanta

Primjer : **temperatura [tk + 2], temperatura [tk + 6]**

Koristeći tagove polja može nam uštedjeti mnogo vremena u razvoju aplikacije. Predpostavimo da su nam potrebni tagovi koji se odnose na temperaturu četiri tanka. Konvencionalni metod konfigurisanja bi bio :

Temperatura1 visoka temperatura na tanku 1  
Temperatura2 visoka temperatura na tanku 2  
Temperatura3 visoka temperatura na tanku 3  
Temperatura4 visoka temperatura na tanku 4

Koristeći polje taga ovo se pojednostavljuje na :

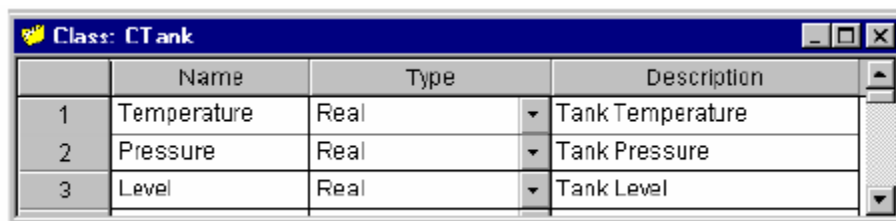
**Temperatura[j] visoka temperatura na tanku [j]**

Kada korisnik kreira polje taga sa četiri pozicije, sistem kreira ustvari 5 pozicija ( od 0 do 4 ). Dakle Tagprimjer [15] će imati 16 elemenata.

## Klase

Pored ova standardna četiri tipa tagova, korisnik može takodjer definisati nove tipove tagova, koji se zovu **klase ( classes)**. Kada se kreira klasa tip taga, onda on ne sadrži samo jednu vrijednost nego cijeli set vrijednosti. Korisnik može kreirati klasu tip tagova grupiranjem do 32 jednostavna taga, koji se zovu **elementi ( elements )**.

Primjer za klasu je dat na narednoj slici:



|   | Name        | Type | Description      |
|---|-------------|------|------------------|
| 1 | Temperature | Real | Tank Temperature |
| 2 | Pressure    | Real | Tank Pressure    |
| 3 | Level       | Real | Tank Level       |

Članovi klase mogu sadržavati standardne vrijednosti kao što je prethodno opisano. Ako kreiramo novi tag **Tank tipa Ctank**, mi u stvari kreiramo tag sa svim osobinama klase **Ctank**.

*Da bi* pristupili članovima klase taga, trebamo koristiti separator tipa tačke (.

Primjer : **Tank.Level , Tank.Temperature**



Ako je tag *Tank* polje, sintaksa bi bila : ***Tank[1]. Level , Tank[n].Temperature***

### Indirektni tagovi – reference i pointeri

IWS podržava indirektni pristup tagovima u bazi podataka. Naprimjer, posmatrajmo tag **X** tipa stringa. Ovaj tag može držati ime bilo kojeg drugog taga u bazi podataka ( to jest, može obezbjediti pointer na bilo koji drugi tip taga, uključujući tip class ). Sintaksa je :

**@ < name of indirect tag >**

Naprimjer, predpostavimo da tag sa imenom **X** sadrži string **Temp**. Čitajući iz ili upisujući u **@ X** , obezbjeduje pristup vrijednosti taga **Temp**. Da bi se referencirali na tag tipa klase, moramo takodjer koristiti Database Manager da definiramo tag tipa string , koji pointira na ovaj tag.

Možemo definirati ovaj tag direktno jednostavno deklarirajući ga , naprimjer

**@ X Class**

u koloni **Tag Name** , kao član klase . Na ovaj način, mi činimo da Xclass je referenca na drugi tag. Da bi pristupili tagu indirektno klase, treba koristiti slijedeću sintaksu:

**@<name of indirect tag> . <member>**

Primjer : **@ Xclass.Level**

U ovom primjeru, mi pristupamo članu *Level* taga na koji Xclass pointira. Kada kreiramo tagove za indirektno korištenje, treba postaviti **@ <name>** u kolonu za tag, a ne da ih kreiramo kao stringove. Za tip , treba upisati tip taga za koji se kreira referenca. Slijedeći Xclass primjer , biće :

|   | Name | Array Size | Type    | Description |
|---|------|------------|---------|-------------|
| 1 | @Z   | 0          | Integer | -           |
| 2 | @X   | 0          | Boolean | ▼           |

Bilo koji string tag je potencijalni indirektni tag ( pointer ).

### Parametri u poljima taga

*Polja ( fields )* su skup parametara koji se odnose na svaki tag u bazi. Aplikacija može pristupiti ovim poljima za vrijeme izvršenja ( runtime) ili za vrijeme razvoja aplikacije koristeći slijedeću sintaksu:

tagname-><field name>

Examples: level->Max Temp->Unit pv101->HiHiLimit

Možemo pristupiti slijedećim poljima kod runtima:

*Max* – maksimalna vrijednost taga

*Min* - minimalna vrijednost taga

**Opaska** : Ako pokušamo da upišemo vrijednost van ovog opsega specificiranog sa *Max* i *Min* na tag , vrijednost neće biti prihvaćena i poruka upozorenja će biti generisana u LogWin fajlu. Ako ne želimo da koristimo ove vrijednosti, treba unjeti vrijednost 0 u obadva polja.

*Unit* – je string do 8 karaktera koji se koristi da referencira inženjerske jedinice.

\**HiHiLimit* – je numerički prag za High High alarm

Primjer :           TP->HiHiLimit=70

\**LoLimit* - numerička vrijednost praga za Low alarm.

\**LoLoLimit* – numerička vrijednost praga za low low alarm

\**DevLimit* - numerička vrijednost praga za alarm devijacije ( odstupanja)

\**RateLimit* – numerička vrijednost praga za Rate limit ( brzine promjene)

\**DevSetpoint* - referentni setpoint za alarm devijacije

*Description* - opisno polje za dokumentaciju

\**AlrDisable* - onemogućava provjeru alarma prema slijedećem :1 onemogućava alarm , 0 omogućava alarm

Primjer :

TMP->AlrDisable=1

*Size* - veličina taga polja, default veličina je 0 , za tag skalar

*HiHi* - ako je različito od 0, High High alarm je prisutan

*Hi* - ako je nenula , high alarm je prisutan

*Lo* - ako je nenula, Low alarm je prisutan

*LoLo* - ako je nenula, Low Low alarm je prisutan

*Rate* - ako je nenula, Rate alarm je prisutan

*Dev* – ako je nenula , alarm devijacije je prisutan

*TimeStamp* – datum i vrijeme posljednje promjene vrijednosti taga.

*bo do b31* - dozvoljava pristup svakom bitu iz integer taga.

**Opaska** : Za vrijeme izvršenja ( runtime) , korisnik može modificirati polja koja se odnose na granice alarma ( indicirana sa \* u gornjoj tabeli )

### **Interni, aplikacioni i dijeljeni tagovi**

Interni tagovi su unaprijed definirani i izvršavaju specifične funkcije unutar IWS supervisorskih taskova. Većina ovih internih tagova su read-only. Da bi se promjenilo vrijeme , naprimjer, koristiti odgovarajuću matematsku relaciju radije nego upisujući direktno u interni tag za vrijeme.

**Primjer.** *Date* sadrži tekući datum u string formatu , a *Time* sadrži tekuće vrijeme u string formatu.

### **Aplikacioni tagovi**

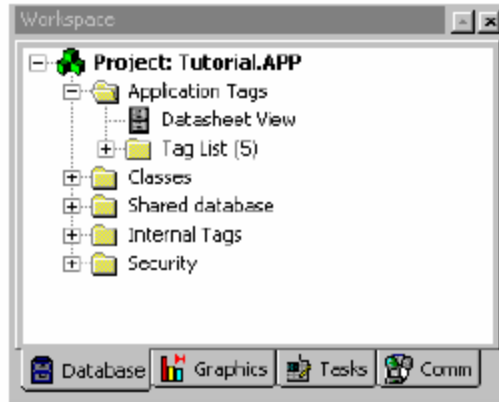
Aplikacioni tagovi se kreiraju od strane korisnika za vrijeme razvoja aplikacije, ( naprimjer, displeji, tagovi koji čitaju sa ili upisuju na procesnu opremu, tagovi koji se koriste u upravljanju, pomoćni tagovi da izvršavaju matematske operacije, itd. )

### **Dijeljeni tagovi**

Ovo su tagovi koji se očitavaju sa drugih softwareskih paketa ( najčešće sa nekog PC baziranog paketa za upravljanje procesom ). Ovi tagovi se ne mogu editirati u okviru IWS-a.

### **Primjer rada sa bazom podataka**

U prozoru radnog prostora ( workspace), izabрати **Database** tab. Kliknuti na **Application Tags** folder da ekspandira i zatim dvaput kliknuti na **Datasheet View** liniju da se otvori worksheet baze podataka:



Baza podataka se može ažurirati za vrijeme razvoja , tako da novi tagovi se mogu pojaviti kako se kreiraju. Mi možemo takodjer definirati tagove koje već od ranije znamo da ćemo trebati na početku naše aplikacije.

Prvi tagovi u našem tutorijalu sadrže vrijednosti za stanja ventila koji pune ili prazne tank ( valve\_Fill\_State i Valve\_Empty\_State ). Svaki ventil ima samo dva moguća stanja ( open, close ) , tako da tag koji će se pridružiti ovim ventilima treba biti Boolean tipa.

Biće tri tanka u aplikaciji , svaki konfigurisan na sličan način, i ima dva tipa ventila, jedan koji puni i jedan koji prazni tank.

Možemo koristiti polje ( array ) da brzo konfiguriramo tagove pridružene sa sva tri tanka .

Treba da konfiguriramo **Application Datasheet** kako slijedi :

|   | Name              | Array Size | Type    | Description                    | Web Data |
|---|-------------------|------------|---------|--------------------------------|----------|
| 1 | Valve_Fill_State  | 3          | Boolean | Fill valve state (openclosed)  | Server   |
| 2 | Valve_Empty_State | 3          | Boolean | Empty valve state (openclosed) | Server   |

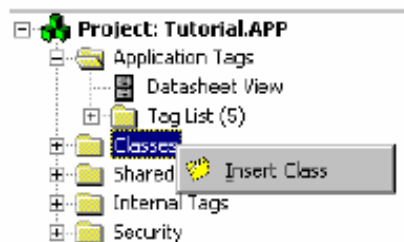
**Opaska** : Mi koristimo samo dvije linije u bazi da konfiguriramo 6 tagova. Polja ( arrays) reduciraju vrijeme potrebno za konfigurisanje baze. Takodjer, polja dozvoljavaju da konfiguriramo funkcije i skripte koje mogu biti optimizirane za ukupnu aplikaciju.

Mi smo konfigurisali tagove koji će primiti stanje svakog ventila. Sada, treba da konfiguriramo tagove koji će biti korišteni da šalju komande na host kontroler. Ovi tagovi imaju isti broj stanja i druge karakteristike kao i prethodno konfigurisani tagovi:

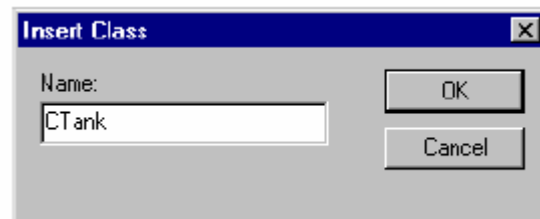
|   | Name                | Array Size | Type    | Description                      | Web Data |
|---|---------------------|------------|---------|----------------------------------|----------|
| 1 | Valve_Fill_State    | 3          | Boolean | Fill valve state (openclosed)    | Server   |
| 2 | Valve_Empty_State   | 3          | Boolean | Empty valve state (openclosed)   | Server   |
| 3 | Valve_Fill_Command  | 3          | Boolean | Fill Valve command (open/close)  | Server   |
| 4 | Valve_Empty_Command | 3          | Boolean | Empty Valve command (open/close) | Server   |

Konačno mi treba da kreiramo tagove da pohranimo osobine pridružene sa tankovima, temperature, nivoe, i pritiske. Ovi atributi su pridruženi sa jednim elementom opreme , tj. sa tankom.

Da bi kreirali novu klasu, selektiraćemo folder **Classes** ( u **Database** tabu) , kliknuti desnim tasterom na njega , i izabrati **Insert Class** :

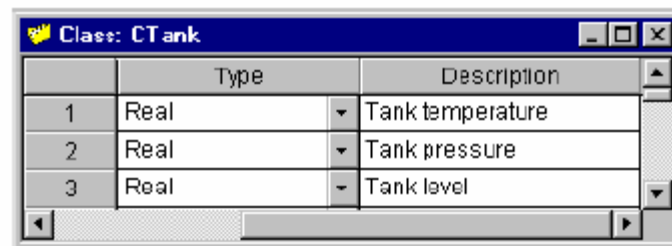


U dijalog boksu treba unjeti ime nove klase , **Ctank** i kliknuti na OK.



U Application Tags worksheetu baze, mi ćemo referencirati osobine tanka koristeći tagove tipa *Class: Ctank*.

Osobina svakog tanka je definirana kao član klase *Ctank* i svaka je definirana slično tagu sa tipom parametar : *Boolean, Integer, Real, ili String*. Sve osobine članova klase su analogne, zato ćemo ih deklarirati kao real.

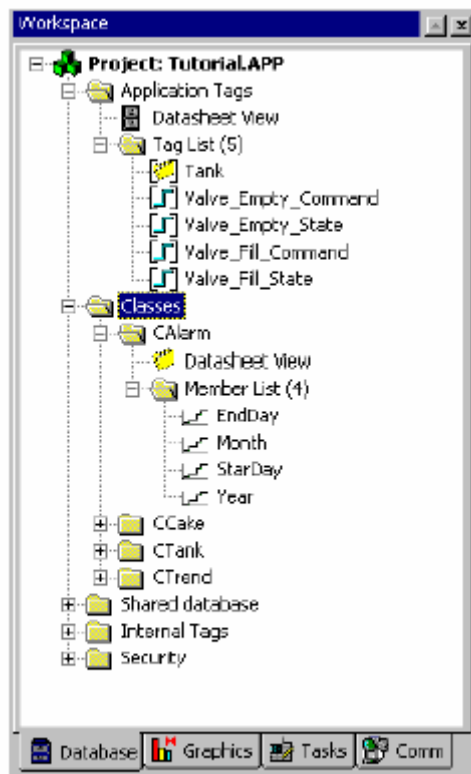


**Opaska:** Možemo ekspandirati Classes folder na subfoldere da vidimo strukturu podataka.

Zatvoriti **Class: Ctank** radni list i kreirati tag pridružen sa klasom. Da bi se kreirao ovaj tag , otvoriti **Application Datasheet** i umetnuti tag ( **Tank** ).U koloni **Type** izabrati opciju **Class:Ctank**. Pošto imamo tri tanka , konfigurisati **Array size** da bude 3.

|   | Name                | Array Size | Type         | Description                      | Web Data |
|---|---------------------|------------|--------------|----------------------------------|----------|
| 1 | Valve_Fill_State    | 3          | Boolean      | Fill valve state (open/closed)   | Server   |
| 2 | Valve_Empty_State   | 3          | Boolean      | Empty valve state (open/closed)  | Server   |
| 3 | Valve_Fill_Command  | 3          | Boolean      | Fill Valve command (open/close)  | Server   |
| 4 | Valve_Empty_Command | 3          | Boolean      | Empty Valve command (open/close) | Server   |
| 5 | Tank                | 3          | Class: CTank | Tank Data                        | Server   |

Sa ovim smo kompletirali kreiranje inicijalne baze tagova za aplikaciju, sada ćemo ekspandirati Application tag i Classes foldere u Database tabu da vidimo bazu podataka koju smo kreirali:



**Opaska:** Mi možemo krirati nove aplikacione tagove i nove klase, klikujući desnim tasterom na njihove foldere i izabirući **Insert** opciju.

Nadalje, mi možemo takodjer modificirati osobine tagova , pomoću desnog klika na njihovu ikonu i izabirući **Properties** opciju. Za vrijeme razvoja aplikacije, kada unesemo ime taga koji ne postoji u bazi, IWS će nas promptirati ako želimo da kreiramo ovaj novi tag. Ako prihvatimo, kreiraće prozor, koji dozvoljava da kreiramo ovaj tag na jedan lagan i brz način.

## Kreiranje novih ekrana

Prije nego što kreiramo bilo koji novi ekran, treba da razmislimo o strukturi aplikacionih ekrana. Moguće je otvoriti više od jednog ekrana istovremeno kod desktop OS, ali kod WinCE možemo kreirati default ekran sa zaglavljem (headerom) i dnom ekrana (footerom) kao template, i unositi objekte u prozor, a onda pohranjivati pod različitim imenima.

U headeru su obično objekti koji daju standardnu informaciju kao vrijeme i datum. U footeru su objekti na dnu ekrana , najčešće alarmni objekat koji pokazuje posljednji najsvježiji alarm.

Regularni dio ekrana je prostor između headera i footera. Pokazuje informacije o procesu, alarmnom ekranu, trendu, itd.

Prednosti korištenja ove strukture su da razvija ekrane u aplikacijama koji omogućavaju da:

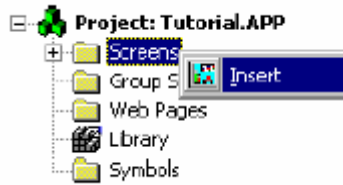
- Povezani ekrani su u skladu sa njihovom namjenom u aplikaciji
- Konfiguriramo linkove i dinamiku koja je zajednička za više ekrana samo jedanput.
- Daje aplikaciji default format na koji se Operator brzo navikne
- Omogućava nam da gradimo modularne ekrane koji mogu biti korišteni i u drugim projektima

Sa ovom preporučenom strukturom, možemo početi graditi ekrane za aplikaciju:

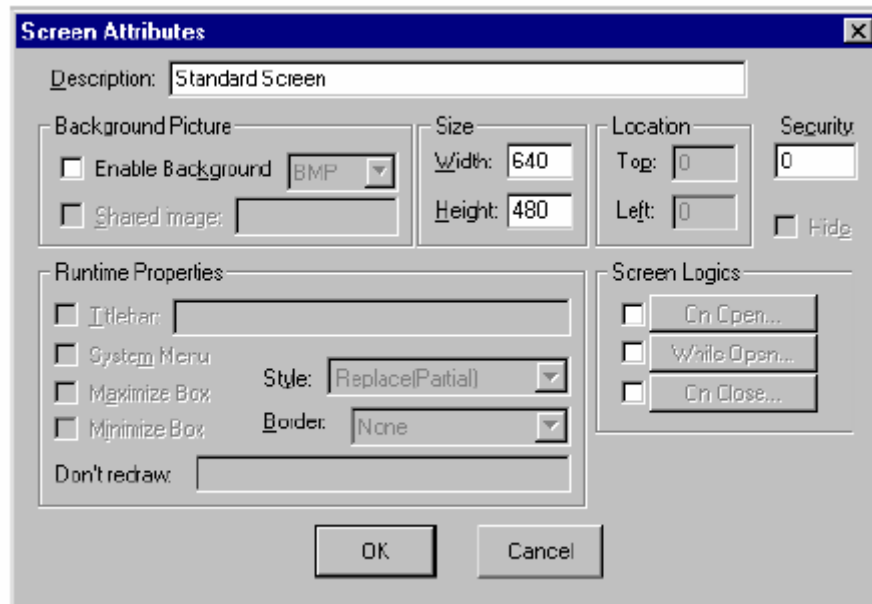
## Kreiranje standardnog ekrana

Standardni ekran će se koristiti kao template za sve druge ekrane . Ostali ekrani će se kreirati koristeći ovaj standardni i pohranjujući ih pod drugim imenima.

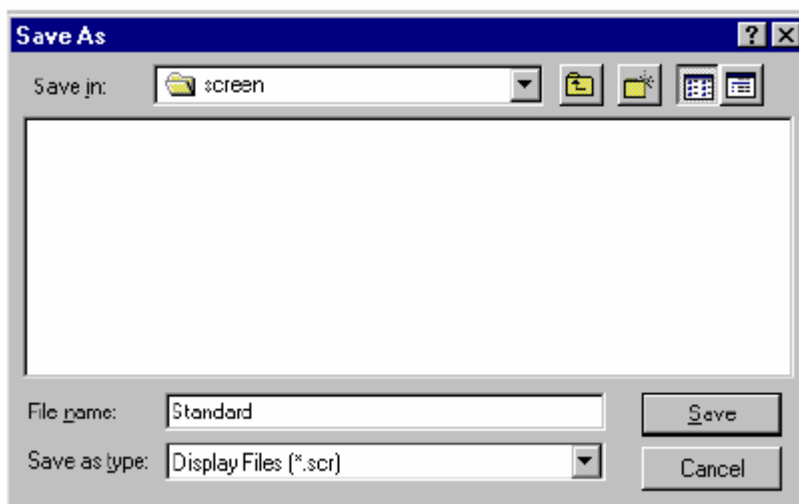
Izaberimo prozor radnog prostora ( workspace window) sa **Graphics** tabom, kliknuvši desnim tasterom na Screens folder , i izabirući **Insert** .



U prozoru ekranskih atributa , moramo konfigurirati opšte informacije o ekranu. Prvi ekran koji ćemo kreirati biće standardni ekran.

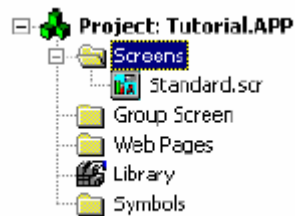


Kliknuti na OK taster da se pohrani sa default atributima i imenom fajla Standard.scr.

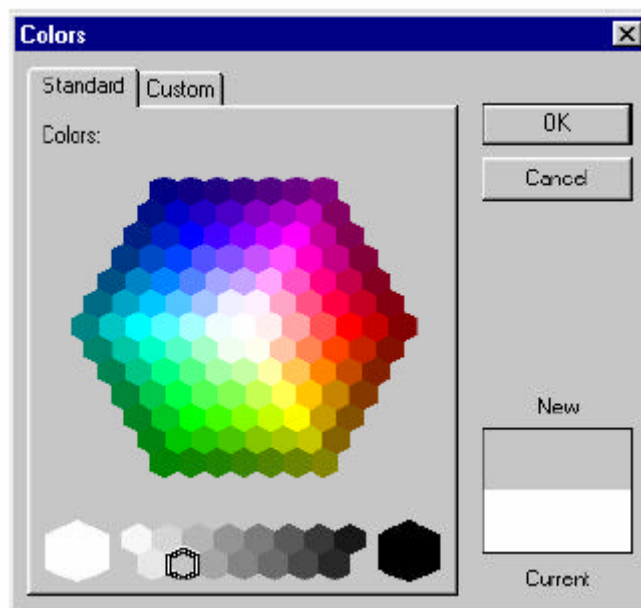




Ekspandirajući foldere u Graphics tabu, možemo vidjeti pohranjen ekran.



Nakon kreiranja ekrana, promjeniti boju pozadine. Izabrati sivu pozadinu koristeći **Background Color** taster na Tools da se otvori prozor boja, ili desno kliknuti na ekranu i izabrati **Background Color** opciju:



Izabrati sivu boju i onda **Ok**, ili sa dvostrukim klikanjem.

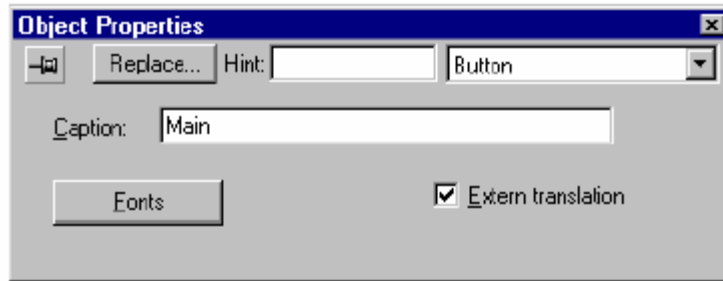
### **Crtanje Header objekata**

### **Crtanje tastera**

Nacrtati taster na ekranu selektirajući ga iz desne palete alata.




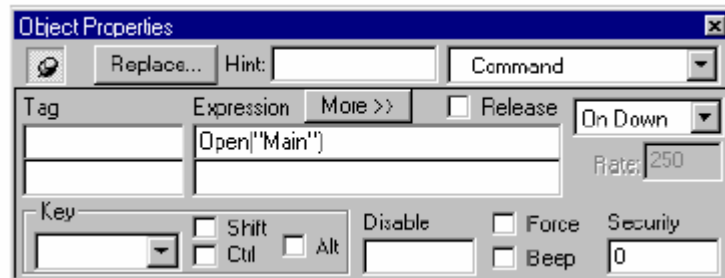
nakon toga, desno klikujući na taster izabrati Properties iz menija. U Caption polju unjeti Main, kao na narednoj slici:



## Kreiranje linkova za taster

Svi tasteri izuzev **Exit** se koriste za otvaranje ekrana. Funkcija **Open** ( "<ime ekrana>" ) se koristi za ovo. Mada ekrani još nisu kreirani mi možemo konfigurirati taster.

Da bi dodali dinamičku osobinu objektu, treba selektirati objekat, a zatim  **Command** na desnom izborniku alata, i otvoriti prozor osobina objekta, u kojem treba izabrati Command , u gornjem desnom uglu. ( ako nije , izabrati iz menija ). Izabrati Main taster , i dodati **Command** osobinu u unjeti **Open()** funkciju u polje **Expression** kao na slijedećoj slici:



Nakon ovoga kreirati još 11 tastera i labelirati ih :

**Trend Online, Trend History, Recipe, report, Alarm Online, Alarm History, LogOn, Exit, Translation, PLC i OPC.**

**Opaska:** Koristiti alate za kopiranje i pestiranje kao i alate za poravnjavanje objekata da se krieraju ovo dodatni tasteri:


Nakon njihovog kreiranja gornji dio ekrana će izgledati kao na slijedećoj slici:

|        |              |               |      |
|--------|--------------|---------------|------|
| Main   | Alarm OnLine | Alarm History | PLC  |
| Recipe | Trend OnLine | Trend History | OPC  |
| Report | Translation  | LogOn         | Exit |

Nakon ovoga izabrati tastere pojedinačno i konfigurirati ih sa odgovarajućim Open() funkcijama za otvaranje prozora sa labelom koja je na njima.


Kod tastera **LogOn** konfigurirati komandu **LogOn()** a kod tastera **Exit** konfigurirati izraz **ShutDown()**.

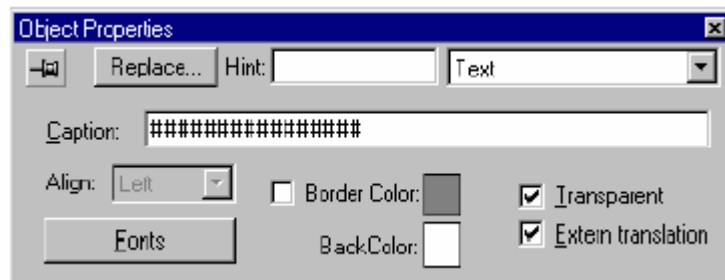
### Crtanje objekata – pravougaonik

Nacrtati tri pravougaonika sa  **Rectangle** alatom. Oni će izgledati kao na slijedećoj slici:


|        |              |               |      |  |
|--------|--------------|---------------|------|--|
| Main   | Alarm OnLine | Alarm History | PLC  |  |
| Recipe | Trend OnLine | Trend History | OPC  |  |
| Report | Translation  | LogOn         | Exit |  |

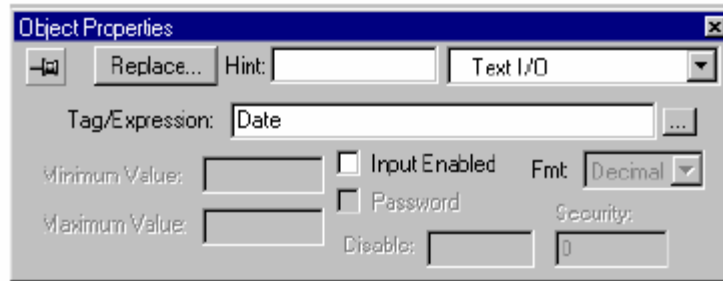
Ispuniti ove pravougaonike crnom bojom koristeći odgovarajuće alate.

Koristeći alat tekst na desnoj strani toolbara  **Text**, dodati tekst u svaki pravougaonik, koji se sastoji od 10 brojeva u gornjem pravougaoniku, kao na slici:



### Dodavanje dinamičkih osobina na legendu

Da bi pridružili tagove ili izraze sa tekst objektima, koristiti  **Text I/O** dugme iz toolbara alata sa desne strane. Selektirati tekst sa znakovima za brojeve ( # hash ), i onda selektirati **Text I/O** dugme. Otvoriti prozor za osobine objekta. Unjeti **Date** u polje **Tag/Expression**, kao na slici :



na isti način u srednji pravougaonik unjeti **Time** a u donji **UserName**.

Da bi brzo istestirali ove funkcije koje smo dodali, možemo izabrati taster



**Test Display**

na toolbaru za kontrolu izvršenja. Pojaviće se slijedeća slika:

|        |              |               |      |            |
|--------|--------------|---------------|------|------------|
| Main   | Alarm OnLine | Alarm History | PLC  | 09/22/2000 |
| Recipe | Trend OnLine | Trend History | OPC  | 14:56:29   |
| Report | Translation  | LogOn         | Exit | Guest      |

### Kreiranje Footer objekta

Oblast na dnu ekrana ( footer ) će uključivati objekat sa jednom alarmnom linijom koja će prikazati najsvježiji alarm. Da bi kreirali alarmni objekat, treba



**Alarm**

izabrati taster iz toolbara i izvući pravougaonik koji će pokazati liniju sa maskiranim slovima, kao na slijedećoj slici:

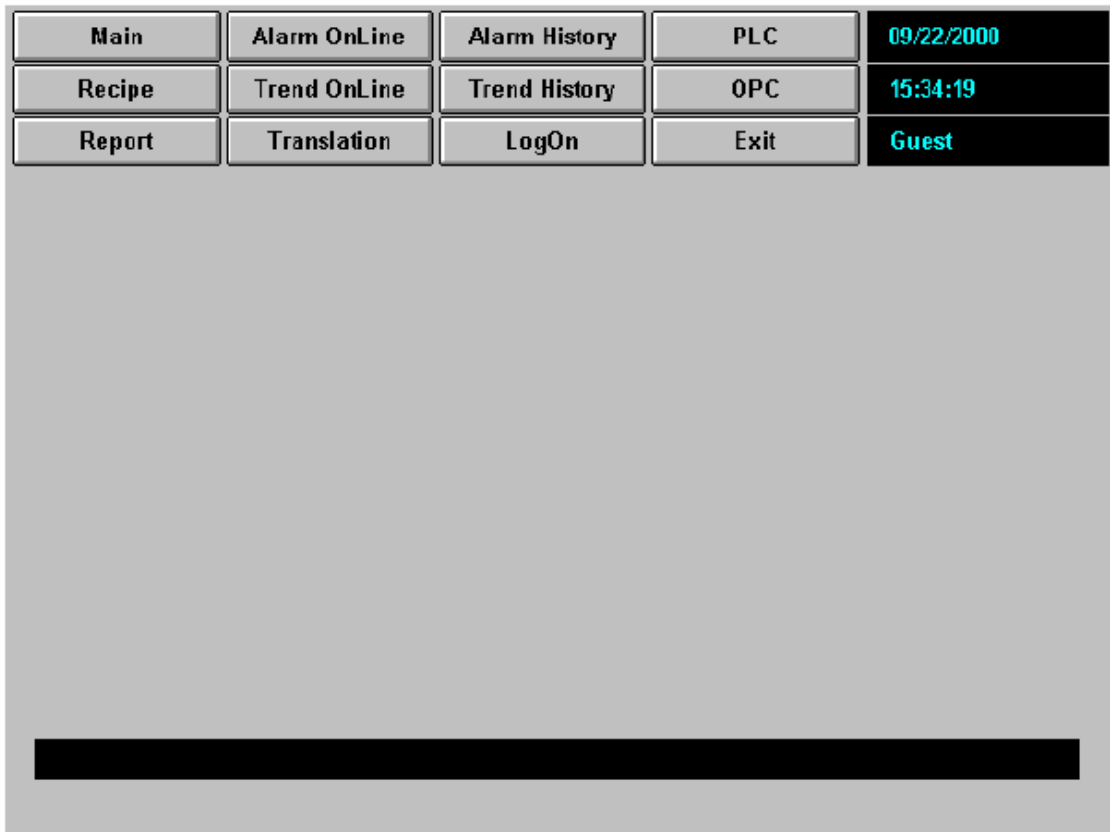


Otvoriti prozor osobina objekta i konfigurirati ga kao što je pokazano na narednoj slici:



Promjeniti boju slova ( fonts ) u bijelu.

Kada završimo kreiranje ovog prozora on treba na kraju da izgleda kao:



### Kreiranje glavnog ekrana

Ovaj prozor će se uvijek otvoriti kada aplikacija starta. Prvo ćemo pohraniti standardni ekran kao Main.scr. Standardni ekran će biti korišten kao template za sve druge ekrane koji će se generisati.

U ovom glavnom ekranu mi ćemo:

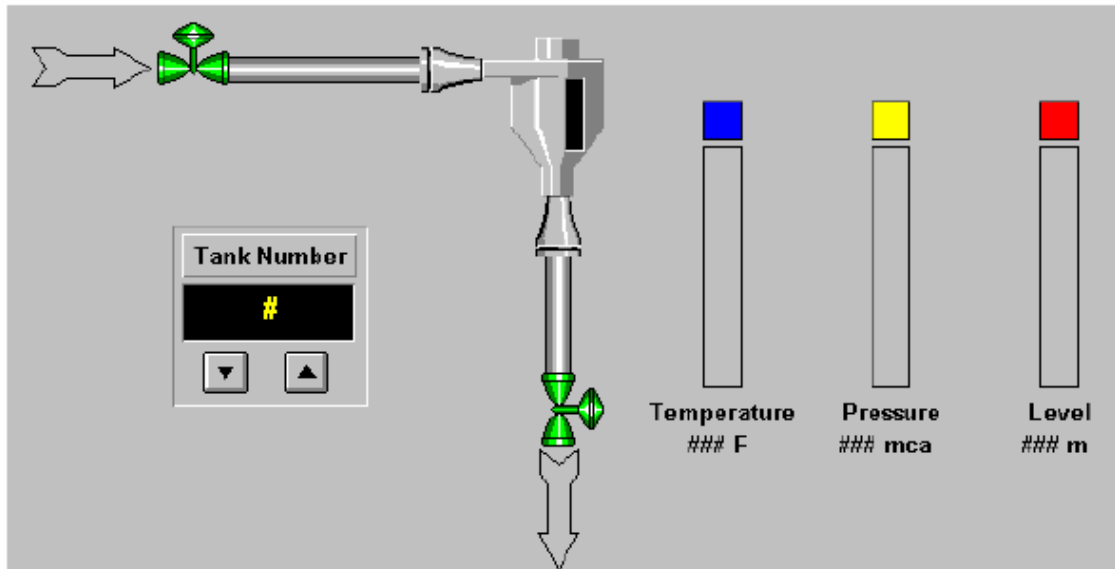
- prikazati osobine tankova ( temperaturu, pritisak, nivo ) grafički i numerički.
- Prikazati stanje ventila koristeći boju ( crvena = zatvoren ventil – closed, zelena = otvoren – open )
- Izdavati komande da otvorimo i zatvorimo svaki ventil pojedinačno.

**Opaska:** Pošto sva tri tanka imaju identične karakteristike, mi ćemo izgraditi samo jedan ekran koji je generički za bilo koji tank u našoj aplikaciji. Da bi se prebacili na neki drugi tank, jednostavno ćemo promijeniti indeks tagova u polju koji se koristi u konfiguraciji.

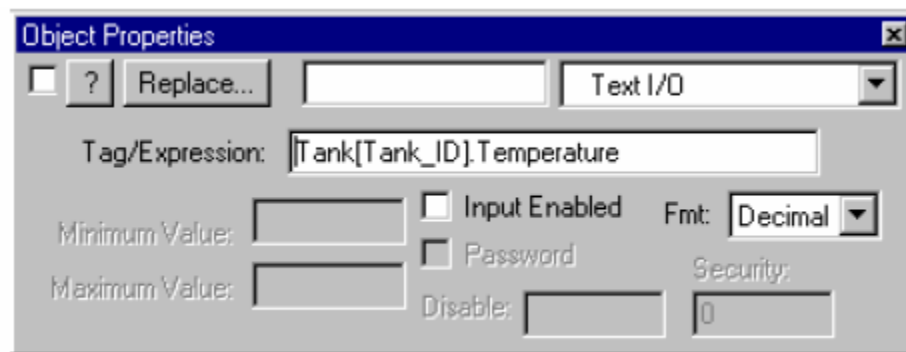
Individualni objekti od kojih se sastoji tanka sa pripadnim elementima ( strijelice, ventili, cjevovod, tank) će se uzeti iz ranije konfigurisanih objekata u biblioteci. Da bi pristupili ovoj biblioteci, treba desno kliknuti na **Library** folder u **Graphics** tabu

u prozoru radnog prostora i izabrati **Open**, ili izabrati  **Open Library** taster na standardnom toolbaru.

Kada završimo selekciju i unošenje i aranžiranje objekata na main prozoru, on će izgledati kao na slijedećoj slici:



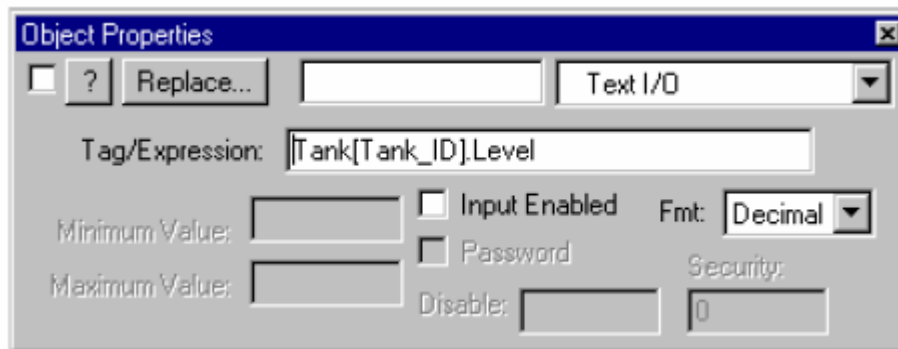
Za svaki tekst objekat, udružićemo prethodno kreirane tagove u **Tag Database**. Da bi primjenili ovu asocijaciju, korišćićemo **Text Input/Output** link kao i prije. Selektiraćemo tekst **####** ispod Temperature i kliknuti na **Text Input/Output** ikonu . Konfigurisati kao što je pokazano na slijedećoj slici:





Tako uraditi i za naredne tekstove za pritisak :

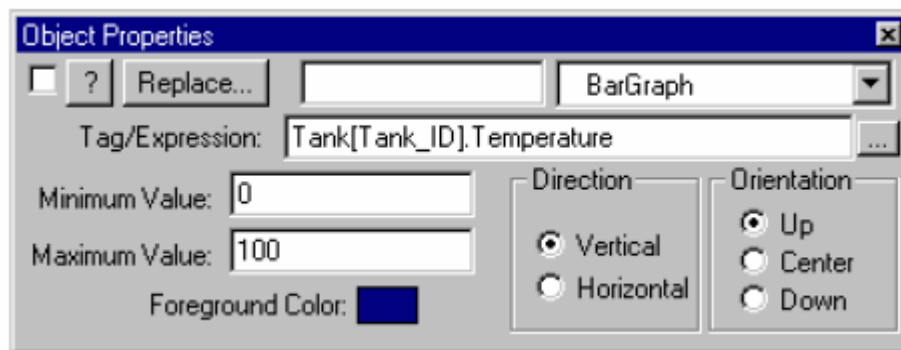


i za nivo:



Da pokažemo vrijednosti temperature, pritiska i nivoa u grafičkom formatu, koristićemo  bargraph link.

Kreiraćemo pravougaonik iznad teksta Temperatura i pridružiti mu tag na taj način što ćemo kliknuti na  "Bargraph". Konfigurisaćemo pravougaonik kao na slijedećoj slici:



isto uraditi iscrtavajući pravougaonik iznad "Pritisak" , konfigurirajući ga kao:

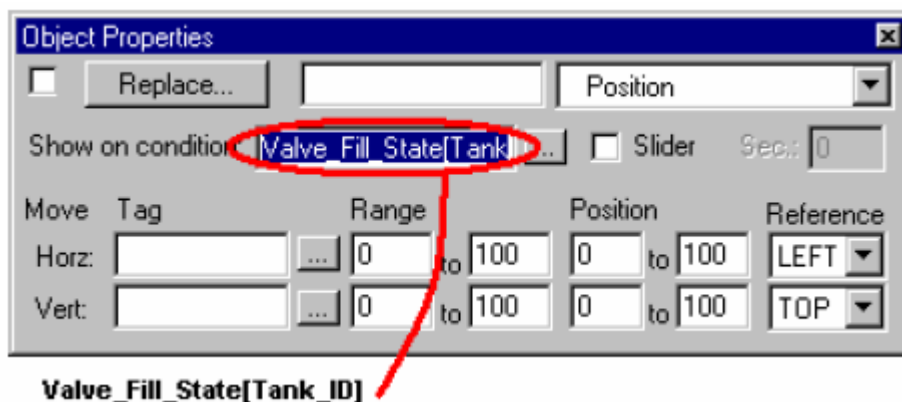


i konačno za bargraf iznad nivoa:



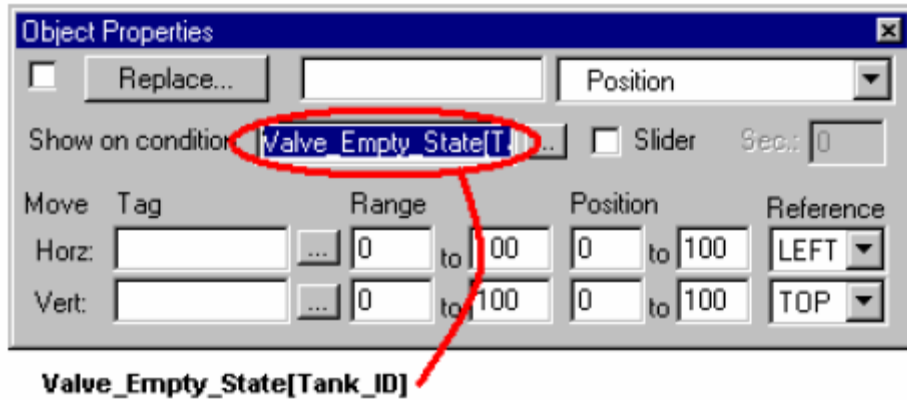
Za ventile "Valve\_Fill" i "Valve\_Empty" , konfigurirati dva linka: "Color" da pokažemo status ventila i "Command" da omogućimo autorizovanom korisniku da može izdavati komande na ventile.

Izabrati "Valve\_Fill" i izabrati osobinu "position". Konfigurirati na slijedeći način.

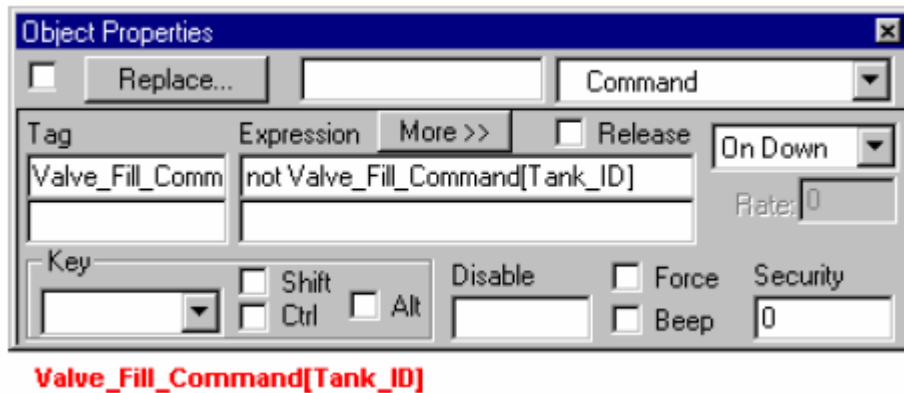


Uradićemo isto i na ventilu za pražnjenje tanka tj. na izlaznom ventilu:

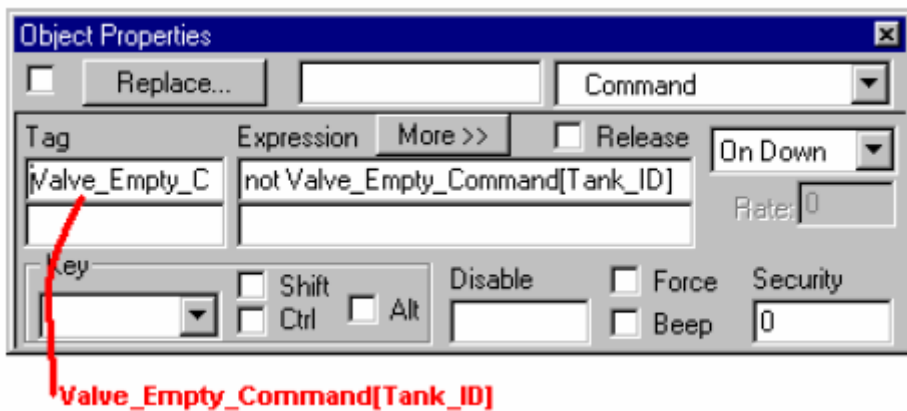






Konfiguriramo sada ulazni ventil ( **Valve\_Fill** ) za osobinu "Command" :

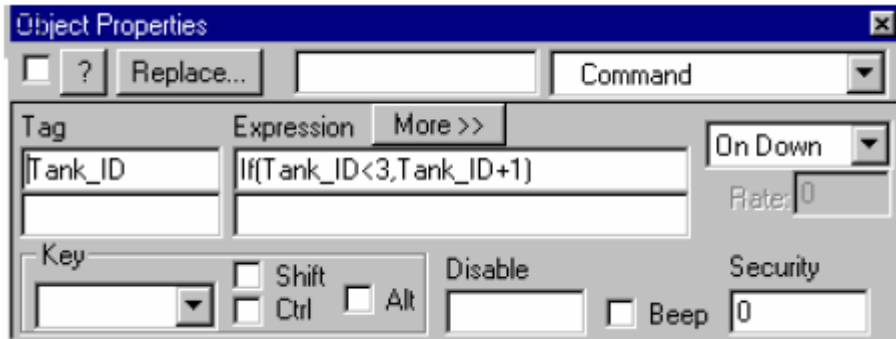


a za izlazni ventil za pražnjenje tanka će biti :

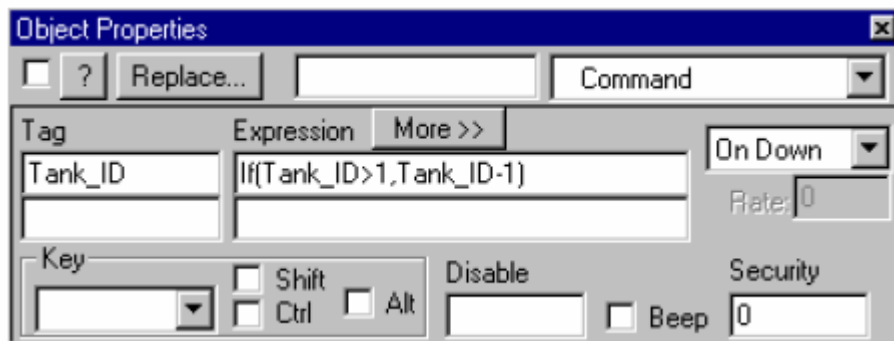


Sada trebamo konfigurirati tastere za povećanje i smanjenje broja tanka.

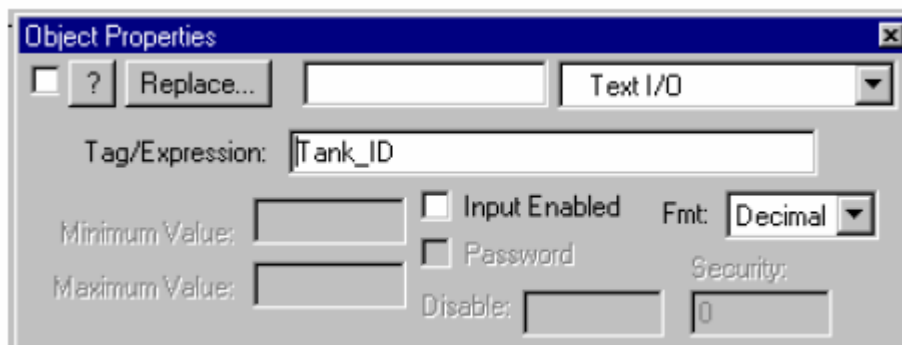
Selektiraćemo taster za povećanje ( increase )  i zatim kliknuti na ikonu  "Command". Konfigurisaćemo osobine kako je pokazano na narednoj slici:



Isto to uraditi i na tasteru za sniženje , i konfigurirati ga kao:




Da bi vidjeli broj tanka koji se nadzire, mi ćemo pridružiti tekst stringu Tank # , slijedeću **Text Input/Output** funkciju.



Za objekat tanka, selektiraćemo njegovu "BarGraph" dinamiku, kao na narednoj slici:



Sada možemo da pohranimo ovaj ekran i da vidimo kako će izgledati kod izvršenja ( runtime) , klikajući na ikonu *Run Application*  . U prozoru izvršenja možemo koristiti DBSpy da testiramo aplikaciju.

No prije nego što nastavimo dalje sa testiranjem i debugiranjem, pogledajmo kako se u IWS-u koristi mogućnost prevodjenja na različite jezike za Operatora, i kako možemo kreirati simulaciju procesa.

## Izrazi , funkcije i skript jezik

U ovom poglavlju pogledaćemo sintaksu skript jezika i funkcije u IWS-u. Ovaj skript jezik se koristi na mnogim mjestima kao,

- dinamičke osobine objekta u graditelju aplikacije ( Application builder-AB)
- ekranska logika u AB ( Application Builderu)
- radni list skedulera ( scheduler worksheet )
- matematski worksheet

Matematski izrazi imaju dvije kolone : **Tag** i **Expression** .

- **Tag** : Ime taga koji prima rezultat izraza u Expression koloni.
- **Expression** : bilo koji matematski izraz definiran u IWS-u.

Primjer :

|   | Tag Name | Expression |
|---|----------|------------|
| 1 | a        | 10 * c - 5 |

Varijabla a prima rezultat izraza :

$$(10c)-5.$$

**Važne napomene:** Ne čine se nikakva doznačavanja u koloni **Expression**. Ako napišemo A=2 , to znači da poredimo A sa brojem 2. Integer kao rezultat ovog izraza ( Boolean vrijednost 0 ako je lažan ili 1 ako je tačan ), bit će upisan u kolonu imena Taga.

Sistem nije osjetljiv na velika i mala slova ( tj. on je case insensitive). Da bi dodali komentare u liniju izraza , treba koristiti "//" karaktere.

### Tipovi podataka

- **Cjelobrojni brojevi** : 1 23 45 123
- **Brojevi sa pokretnim zarezom** : 1.234 , -774.233
- **Heksadecimalni cjeli brojevi**: 0x5 , 0xA0, 0xBC4
- **Stringovi** : "demo" , "novi demo"

Cjelobrojni brojevi su 32 bitni. Brojevi sa pokretnim zarezom su 8 bajtni a stringovi mogu imati do 255 karaktera.

### Primjeri pristupa aplikacionoj bazi podataka

Da bi čitali vrijednosti u bazi, treba direktno koristiti ime taga .

#### Primjer 1:

U slijedećoj liniji skripta, X tag će primiti sumu dva taga, **level** i **temp**:

|   | Tag Name | Expression   |
|---|----------|--------------|
| 1 | X        | Level + temp |

#### Primjer 2:

IWS dozvoljava da čitamo iz i upisujemo u tagove koristeći reference ili pointere. Tag koji se koristi kao pointer na drugi tag može biti deklarisan na dva načina: kao string ( pointer na nedefinisani tip ), ili kao pointer specifične vrste ( ovo se preporučuje):

|   | Name                | Size | Type    | Description                   |
|---|---------------------|------|---------|-------------------------------|
| 1 | Valve_Fill_State    | 0    | Integer |                               |
| 2 | @pointer_to_integer | 0    | Integer | // Pointer to a integer value |

Na prethodnoj slici, pointer\_default je varijabla tipa string koja je pointer. Varijabla @ pointer\_to\_integer je pointer na integer vrijednosti.

**Opaska** : Sinatksa @ tag dozvoljava tagu da pristupi drugom putem referenciranja.

Bilo koji tag koji je deklarisan kao string može biti korišten kao indirektni tag ( pointer ) .

## **Operatori u matematskim izrazima**

IWS podržava slijedeće vrste operatora:

### **Aritmetički Operatori**

- + sabiranje
- oduzimanje
- \* množenje
- / djeljenje
- > više od
- < manje od
- = jednako
- >= veće ili jednako
- <= manje ili jednako
- <> različito od ( nejednako )

### **Logički operatori**

- AND , logičko I
- NOT , logičko NE
- OR , logičko ILI
- XOR , ekskluzivno ILI
- & , I na nivou bita
- | , ILI na nivou bita
- ~ , NOT na nivou bita
- ^ , XOR na nivou bita
- >>n , rotirati desno n bita
- <<n , rotirati lijevo n bita

### **Funkcionalna lista**

IWS uključuje više od 100 funkcija koje se mogu lako i neposredno koristiti

### **SEND MESSAGE TO THE LOGWIN**

TRACE ( strOutputMessage)

## **Aritmetičke funkcije**

ABS(numValue)  
DIV(numDivisor, numDividend)  
FORMAT(strFormatFlag, numValue)  
GETBIT(strTagName, strBitNumber)  
MOD(numDivisor, numDividend)  
POW(numBase, numExponent)  
RESETBIT(strTagName, strBitNumber)  
ROUND(numValue)  
SETBIT(strTagName, strBitNumber)  
SQRT(numValue)  
SWAP16(strTagName)  
SWAP32(strTagName)  
TRUNC(numValue)

## **Statističke funkcije**

AVG(numValue1, numValue2, &ldots; , numValueN)  
MAX(numValue1, numValue2, &ldots; , numValueN)  
MIN(numValue1, numValue2, &ldots; , numValueN)  
RAND()

## **Logaritamske funkcije**

EXP(numExponent)  
LOG(numLogArg)  
LOG10(numLogArg)

## **Logičke funkcije**

IF(numCondition, numThen, numElse)  
TRUE(numExpression)  
FALSE(numExpression)

## **Funkcije za manipulaciju sa stringovima**

ASC2STR(strChar1, strChar2, &ldots; , strCharN)  
CHARTOVALUE("strTagName", "numArray")  
CHARTOVALUEW("strTagName", "numArray")  
NCOPY(strSource, numStartChar, numQtdChar)  
NUM(strValue)  
STR(numValue)  
STR2ASC(strChar)  
STRLEFT(strSource , numQtdChars)  
STRLEN(strSource)  
STRLOWER(strSource)  
STRRCHR(strSource, strCharSequence)  
STRRIGHT(strSource, numQdeChars)  
STRSTR(strSource, strSequence)  
STRSTRPOS(strSource, strCharSequence)  
STRTRIM(strReference, numOptionalFlag)  
STRUPPER(strValue)  
VALUETOCHAR("numArray", numQdeChar)  
VALUEWTOCHAR("numArray", numQdeChar)

## **Date and Time Manipulation**

CLOCKGETDATE(numSeconds)  
CLOCKGETDAYOFWEEK(numSeconds)  
CLOCKGETTIME(numSeconds )  
DATETIME2CLOCK(strDate, strTime)  
GETCLOCK()  
HOUR2CLOCK(strTime)  
SETSYSTEMDATE(strDate)  
SETSYSTEMTIME(strTime)

## **Trigonometrijske funkcije**

ACOS(numValue)  
ASIN(numValue)  
ATAN(numValue)  
COS(numAngle)  
COT(numAngle)  
PI()  
SIN(numAngle)  
TAN(numAngle)

## **Funkcije za otvaranje i zatvaranje prozora**

OPEN(strScrFile, numOptionalX1, numOptionalY1, numOptionalX2, numOptionalY2)  
CLOSE(strScrFile)

## **Sistem sigurnosti**

CREATEUSER(strUserName, strGroup, strPassw)  
REMOVEUSER(strUserName )



## **Funkcije za aktivaciju modula**

SHUTDOWN()  
APPACTIVATE(strAppTitle, numOptionalActiv)  
APPISRUNNING(strAppTitle )  
APPPOSTMESSAGE(strAppTitle, numwParam, numlParam)  
APPSSENDKEYS(strKeys1, strKeys2, &ldots; , strKeysN)  
CLEANREADQUEUE()  
CLOSESPASHWINDOW()  
DISABLEMATH()  
ENABLEMATH()  
EXITWINDOWS(numExitCode)  
ISSCREENOPEN(strScrName)  
ISVIEWERINFOCUS()  
LOGOFF()  
LOGON(strOptionalUser, numOptionalPassw)  
MATH(numMathWorksheet)  
NOINPUTTIME()  
RECIPE(strOperation&File)  
REPORT(strOperation&File)  
SETAPPPATH(strDirPath)  
SETVIEWERINFOCUS()  
VIEWERPOSTMESSAGE(strScrTitle, numwParam, numlParam)  
WAIT(numMilliseconds) <This function can cause errors if not used carefully>  
WINEXEC(StrFilePath, numOptionalState)

## **Funkcije za manipulaciju sa fajlovima**

FILECOPY(strSourceFile , strTargetFile)  
FILEDELETE(strFilePath)

FILELENGTH(strFileName)

FILERENAME(strOldName , strNewName)

FINDFILE(strFileMask)

PRINT(strFilePath)

RDFILEN(strSelectedFile, strSearchPath, strMask, numChangeDir)

### **Funkcije za printanje grafičkih ekrana**

PRINTWINDOW(strScrName)

### **Funkcije za prevodjenje teksta**

EXT(strText)

SETTRANSLATIONFILE(strFileName)

### **Multimedijalne funkcije**

PLAY(strWavFile)

## **Sistemske informacije**

DBVERSION()  
GETAPPHORIZONTALRESOLUTION()  
GETAPPVERTICALRESOLUTION()  
GETCOMPUTERNAME()  
GETHARDKEYMODEL()  
GETHARDKEYSN()  
GETPRODUCTPATH()  
GETOS()  
GETPRIVATEPROFILESTRING(str\_Section, str\_Name, str\_Default, str\_FileName)  
GETTICKCOUNT()  
INFOAPPALRDIR()  
INFOAPPDIR()  
INFOAPPHSTDIR()  
INFODISKFREE(strDiskUnit)  
INFORESOURCES(numResourceCode)  
NOINPUTTIME()  
PRODUCTVERSION()  
SETAPPALARMPATH(strPath)  
SETAPPHSTPATH(strPath)  
SETDATEFORMAT(strSeparator, strDateFormat)

## **Funkcije za pristup bazama podataka**

CHANGED(strTagName)  
FORCETAGCHANGE(strTagName, numValue)

## Konture ( loops )

FOR(numInitialValue, numFinalValue, numStep)

NEXT

## Interni tagovi

GOTO

LABEL

## Mail

CNFEEMAIL( strSMTP , strFrom, strPOP3, strUser, strPassword, numTimeout )

SENDEMAIL( strSubject, strMessage, strTO )

## Specijalne funkcije

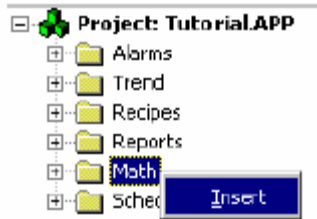
**Opaska :** Korisnik može koristiti Database Spy modul da izvrši svaki matematski izraz ( *math expression* ), treba da upiše izraz u polje imena Taga ( *Tag Name field*) i klikne mišom iznad Toggle tastera. Povratna vrijednost izraza biće prikazana u polju vrijednosti ( *Value field*) .

## Konfigurisanje radnih lista ( worksheets )

### Konfigurisanje matematskog worksheetsa ( simulacija procesa )

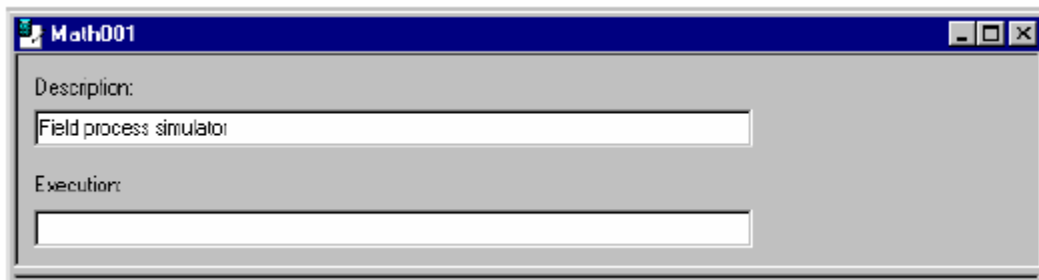
Prije nego što nastavimo razvoj aplikacije, treba da kreiramo skript da simuliramo neke varijable. U aplikacijama u realnom vremenu i procesu, ove varijable će dolaziti sa uređaja u procesu ( PLCs., I/O moduli , itd), Ova simulacija će biti uradjena u matematskom worksheetu , koji treba da se neprekidno izvršava.

- desno kliknimo na **Math** folder lociran u **Tasks** tabu, i izaberimo opciju **Insert** da kreiramo novi matematski worksheet,



Polje **Execution** je ono koje kontroliše izvršenje matematskih izraza. Ovdje možemo unjeti puni izraz , jednostavni uslov, ime taga , vrijednost , i kada je ovaj uslov TRUE, matematski worksheet će se izvršiti.

Mi ćemo u zaglavlju ( header) , konfigurirati ovo polje **Execution** sa vrijednošću 1. Ovo će omogućiti neprekidno izvršenje ovog matematskog sheeta, pošto je 1 uvijek TRUE uslov.



U tijelu worksheeta, konfigurisaćemo funkcije da simuliramo:

- status ventila, u skladu sa izdatim komandama
- nivo, temperaturu i pritisak u svakom od tri tanka

a, Da simuliramo status svakog ventila, mi ćemo prenjeti vrijednost od komandnih tagova na statusne tagove.

b. Da simuliramo osobine za temperaturu i pritisak za svaki tank, razmatrajmo ove osobine koristeći trigonometrijske funkcije ( sinusa i kosinusa)

c. Da simuliramo osobine nivoa u svakom tanku, podsjetimo se da obadva ventila i za punjenje ( fill ) i pražnjenje ( empty) , utiču na ovaj nivo.

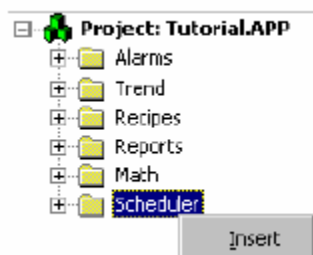
Sada sa ovom analizom , ispunimo tijelo matematskog worksheeta sa slijedećim izrazima :

| Description:            |                      |  |
|-------------------------|----------------------|--|
| Field process simulator |                      |  |
| Execution:              |                      |  |
| 1                       |                      |  |
|                         | Tag Name             | Expression   |
| 1                       | Valve_Empty_State[1] | Valve_Empty_Command[1]   |
| 2                       | Valve_Empty_State[2] | Valve_Empty_Command[2]   |
| 3                       | Valve_Empty_State[3] | Valve_Empty_Command[3]   |
| 4                       | Valve_Fill_State[1]  | Valve_Fill_Command[1]  |
| 5                       | Valve_Fill_State[2]  | Valve_Fill_Command[2]  |
| 6                       | Valve_Fill_State[3]  | Valve_Fill_Command[3]  |
| 7                       | Tank[1].Temperature  | $(\sin((\text{Second}/30)*\pi)+1)*50$  |
| 8                       | Tank[2].Temperature  | $(\sin((\text{Second}/20)*\pi)+1)*50$  |
| 9                       | Tank[3].Temperature  | $(\sin((\text{Second}/10)*\pi)+1)*50$  |
| 10                      | Tank[1].Pressure     | $(\cos((\text{Second}/30)*\pi)+1)*50$  |
| 11                      | Tank[2].Pressure     | $(\cos((\text{Second}/20)*\pi)+1)*50$  |
| 12                      | Tank[3].Pressure     | $(\cos((\text{Second}/10)*\pi)+1)*50$  |
| 13                      | J                    | For(1,3,1)   |
| 14                      | Tank[J].Level        | If(Valve_Empty_State[J]>Valve_Fill_State[J] and Tank[J].Level>0,Tank[J].Level-1)   |
| 15                      | Tank[J].Level        | If(Valve_Empty_State[J]<Valve_Fill_State[J] and Tank[J].Level<100,Tank[J].Level+1) |
| 16                      | Next                 |  |

**Opaska :** U našem primjeru ovaj worksheet br 001 se neprekidno izvršava. U aplikacijama u realnosti, potrebno je pažljivo razmotriti izvršenje svakog matematskog radnog lista da bi se poboljšala performansa sistema.

### Konfigurisanje radnog lista Skedulera

Da bi se kreirao novi radni list skedulera, desno kliknuti na **Scheduler** folder u **Task** tabu, iz **Workspace** prozora. Izabrati opciju **Insert** da se kreira novi list skedulera.



Konfigurisati radni list skedulera kako slijedi :

| Sched001.sch                        |          |         |          |            |             |                     |         |
|-------------------------------------|----------|---------|----------|------------|-------------|---------------------|---------|
| Description:<br>Training Schedulers |          |         |          |            |             |                     |         |
|                                     | Event    | Trigger | Time     | Date       | Tag         | Expression          | Disable |
| 1                                   | Clock    |         | 00:00:30 |            | TrendUpdate | notTrendUpdate      |         |
| 2                                   | Change   | Hour    |          |            |             | Open("Registering") |         |
| 3                                   | Calendar |         | 16:00:00 |            |             | LogOn()             |         |
| 4                                   | Calendar |         |          | 01/01/2000 |             | Open("LetsGo")      |         |
| 5                                   |          |         |          |            |             |                     |         |
| 6                                   |          |         |          |            |             |                     |         |

### Opaske :

**Clock** događaj se koristi da trigeruje akcije bazirane na regularnim vremenskim intervalima kao što su tajmeri i brojači. U koloni **Time**, korisnik može konfigurirati osnovno vrijeme ( minimalno 100 ms ), U koloni **Tag**, moramo konfigurirati tag koji će primiti rezultat iz izraza konfigurisanog u koloni **Expression**. Konačno, polje **Disable** se može koristiti da se spriječi da se izvrši izraz u liniji. Rezultati izraza u **Disable** polju će uvijek biti TRUE.

Događaj **Calendar** se koristi da trigeruje akcije u skedulirano vrijeme. Također, moguće je specificirati fiksni datum za događaj u **Date** koloni. Kolone **Tag**, **Expression** i **Disable** se koriste na isti način u sve tri skedulirane funkcije.

Događaj **Change** se koristi da trigeruje akciju nakon promjene u vrijednosti taga. U Trigger koloni, mi moramo konfigurirati tag koji će biti korišten da trigeruje događaj kada i se desila promjena u vrijednosti. Kolone **Tag**, **Expression** i **Disable** se koriste na isti način u sve tri skedulirane funkcije.

## Recepture i izvještaji

### Kreiranje receptura

IWS modul za recepture omogućava korisniku da kreira, napuni i izbriše recepte. Recepture su, u ovoj definiciji, grupe tagova čije se vrijednosti pohranjuju i vade kao i kod baza podataka.

Da bi pripremili recepturu, mi treba da kreiramo radni list ( worksheet ) recepture. Ovaj radni list će reći sistemu koje tagove korisnik želi da pohrani na disk radi kasnijeg vadenja, i gdje želi da pohrani podatke. Kada pohranimo recepturu, kreira se ASCII fajl sa vrijednostima Tagova i imenom fajla recepture. Za vadenje ovih imena tagova, sistem će ih naći u ovom ASCII fajlu u standardnom formatu ili XML formatu.

## Kreiranje radnog lista recepture

Prvo, treba kreirati klasu CLASS:CCake sa integer članovima : Suger, Fruit, Milk, Flour, i Yeast ( prašak za pecivo ):

|   | Name  | Type    |   |
|---|-------|---------|---|
| 1 | Sugar | Integer | ▼ |
| 2 | Fruit | Integer | ▼ |
| 3 | Milk  | Integer | ▼ |
| 4 | Flour | Integer | ▼ |
| 5 | yeast | Integer | ▼ |
| 6 |       |         | ▼ |

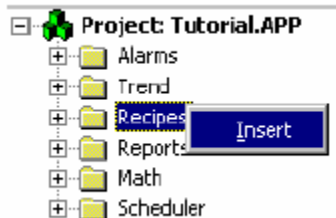
Nakon toga, treba kreirati Tag **Cake** tipa **class: CCake**.

**Opaska:** Podsjetimo se da sintaksa za pristup vrijednosti iz taga klase je:

`<tag_name>.<member_name>`. (e.g. **Cake.Sugar**, **Cake.Fruit**, etc...).

Nadalje, kreiraćemo Tag tipa STRING **RecipeName** ( koji nije tip klase) koji će se koristiti da pohranimo ime ulaznog fajla koji će se koristiti u ovom primjeru.

Sada ćemo izabrati **Tasks** tab u **Workspace** prozoru i kreiraćemo novi list za recepturu:



Ispunićemo polja kao na narednoj slici i pohraniti sa default imenom **Recipe1.rcp**.



| Description: |            | <input type="text" value="Cake"/>  | <input type="checkbox"/> Save As XML     |
|--------------|------------|--|--|
| File Name:   |            | <input style="border: 1px solid black;" type="text" value="{RecipeName}"/> | Register Number:<br><input type="text"/> |
|              | Tag Name   | Number of Elements   |  |
| 1            | Cake.Flour |  |  |
| 2            | Cake.Fruit |  |  |
| 3            | Cake.Milk  |  |  |
| 4            | Cake.Milk  |  |  |
| 5            | Cake.Yeast |  |  |

Polje "**File Name**" je polje gdje ćemo pohraniti vrijednosti tagova recepture. Ako unesemo ime taga u vitičastim ( curly { } ) zagradama ( kao u ovom primjeru ) , fajl će koristiti vrijednosti tagova da komponira ime fajla. Naprimjer, možemo imati "File Name" u obliku :

*c:\AppName\Recipe\{RecipeName}.*

U ovom slučaju, vrijednost unutar taga RecipeName će dati ime fajla, kod

*c:\AppName\Recipe\ directory.*

### **Polje broja registra ( register number field )**

Je tag koji definira broj registra koji treba biti očitao ili upisan u DBF fajl. Više se ne koristi.

**Broj elemenata ( Number of Elements ) :** Setuje broj pozicija tagova polja koja se koriste. Tako, ako želimo da imamo polje tagova veličine 120 u Recepturi, nema potrebe da unosimo imena tagova i indeks za svih 120 pozicija, tj. Tag [0 ], Tag [1 ], Tag [2 ], .....

Sve ono što je potrebno da uradimo je da unesemo ime Taga i u koloni broja da unesemo koliko ima pozicija.

### **Kreiranje ekrana recepture**

Treba otvoriti standardni ekran, nacrtati objekte pokazane na narednoj slici i pohraniti kao fajl "Recipe.scr".

Recipe Name: [#####]

Flour: [#####]

Fruit: [#####]

Milk: [#####]

Sugar: [#####]

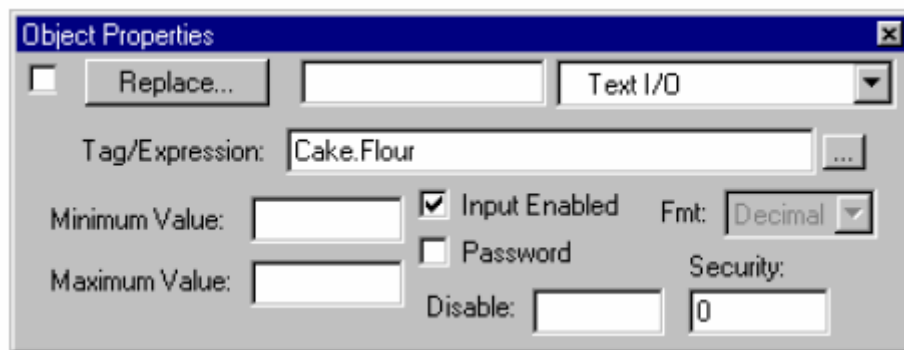
Yeast: [#####]

[Load]

[Save]

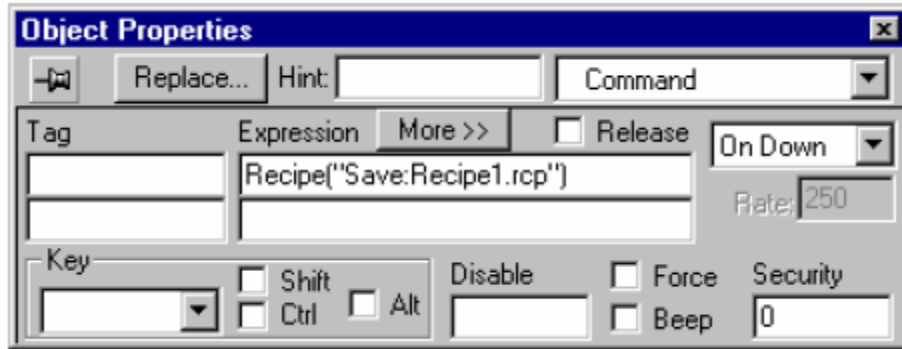
Sada je potrebno da pridružimo Text I/O funkciju ##### tekstu , i ispunimo svaki od polja Tag/Expression sa imenima tagova i članovima **Cake** taga. Ostavimo "Input Enabled" check boks čekiranim.

Primjer: Za ##### tekst ispred "Flour" teksta, unjeti `Cake.Flour` u Tag/Expression polje kao na slici:

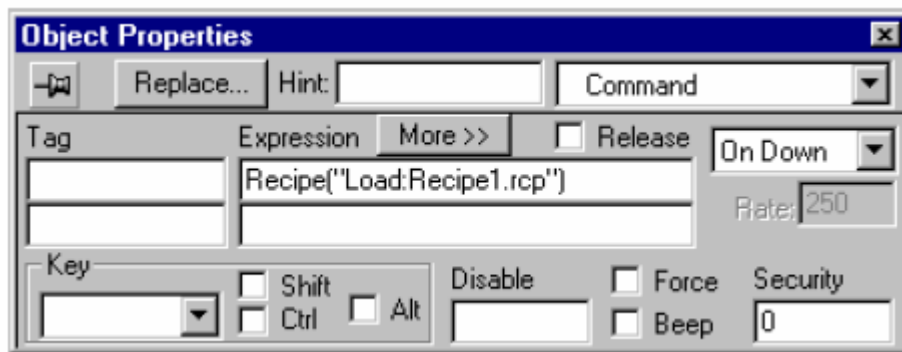


Za polje Recipe Name , treba pridružiti tag string **RecipeName**.

Unjeti "Command" osobinu u "Save" i "Load" tastere. U "Save" taster, treba unjeti slijedeću komandu :



a u "Load" taster :



Radi testiranja , unjeti neko ime za Recipe i vrijednosti za sastavne komponente. Pohraniti unose. Zatim treba unjeti neko drugo ime i vrijednosti za komponente i ponovo pohraniti.

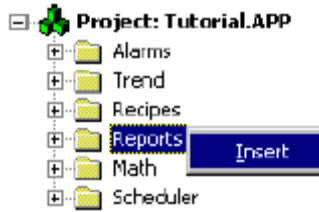
Sada treba unjeti ime prve recepture i loadovati je. Vidjeti kako će se pojaviti na ekranu.

### Krieranje izvještaja

IWS alat za uzvještaje dozvoljava vrlo lagano krieranje izvještaja , bez da je potreban bilo kakav drugi programski alat, kao VB, VBA, Excel, itd. Sve ono što korisnik treba da uradi je da pripremi masku izvještaja u ASCII formatu ili da koristi Report Writer tool iz IWS-a ( koji kreira RTF fajlove ), **stavljajući imena tagova koja želi da dobije isprintane vrijednosti u vitičaste zagrade.**

### Kreiranje ASCII izvještaja

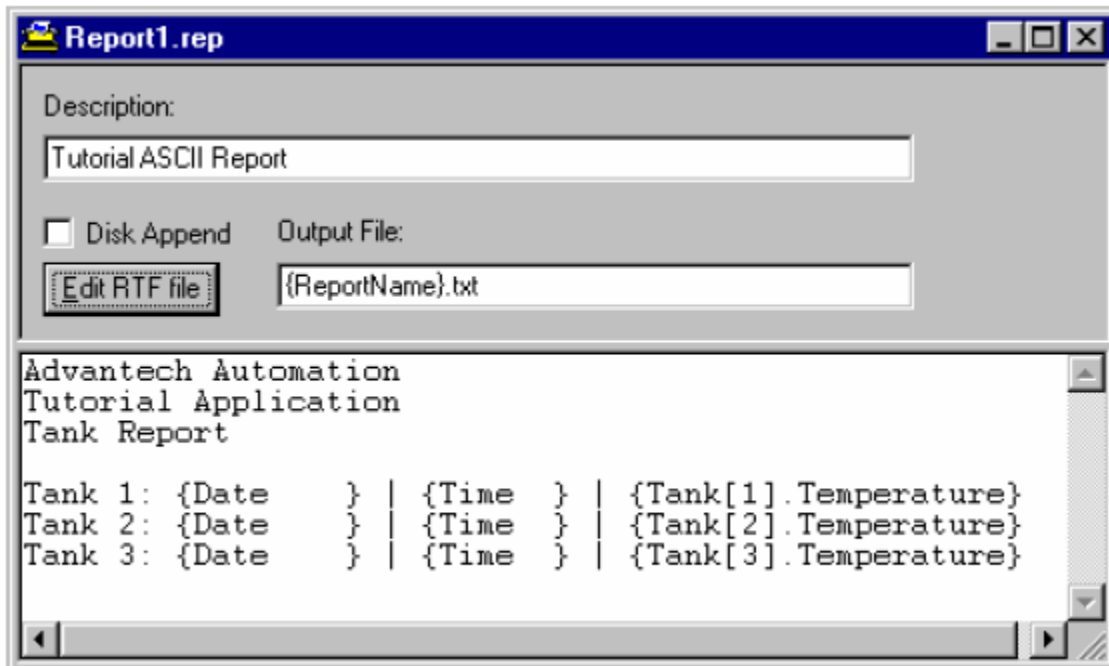
Prvo, kreirajmo radni list izvještaja koji će biti korišten da se kreira tijelo report fajla. Desno kliknimo na **Reports** folder u **Task** tabu u Workspace prozoru. Izabrati opciju **Insert** da kreiramo novi radni list izvještaja:



Sada treba konfigurirati radni list izvještaja kako slijedi:

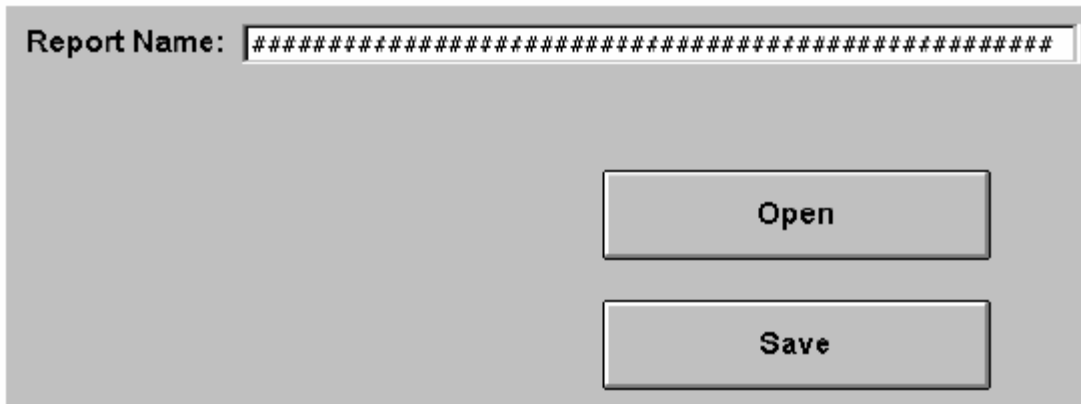
Kao i kod recepture, "Output File" je gdje će izvještaj biti kreiran uključujući i njegovo ime. Tag izmedju vitičastih zagrada se može koristiti i ovdje da se da ime izvještaju.

Check box "Disk Append" bit će objašnjen u slijedećem primjeru.

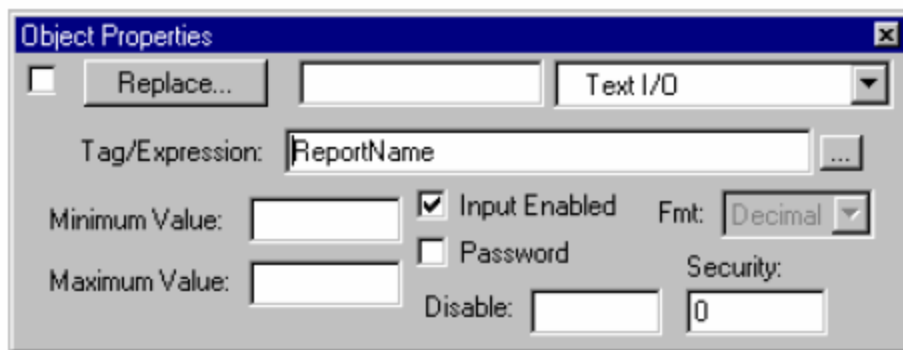


Pohraniti radni list izvještaja sa default imenom **Report1.rep**.

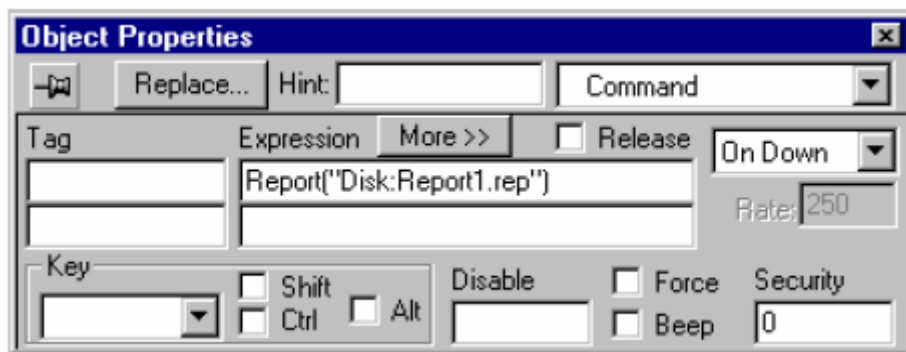
Kreirati "Report" ekran kako slijedi:



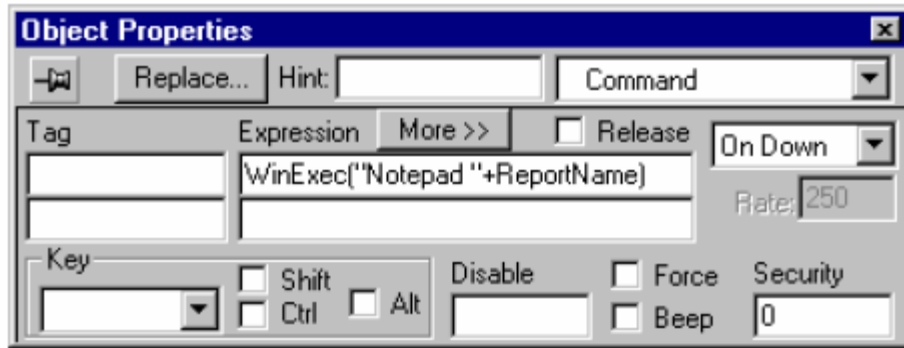
Sa tekstem ##### pridružiti Text I/O osobinu i konfigurirati ga kao na slijedećoj slici:



Uz "Save" taster dodati slijedeću komandu :



Uz "Open" taster dodati slijedeću komandu:



Ova komanda će koristiti Windows Notepad program da pokaže ASCII izvještaj koji je bio upravo kreiran.  
Slijedeći korak da se da ime izvještaju i testiraju komande "Save" i "Open".

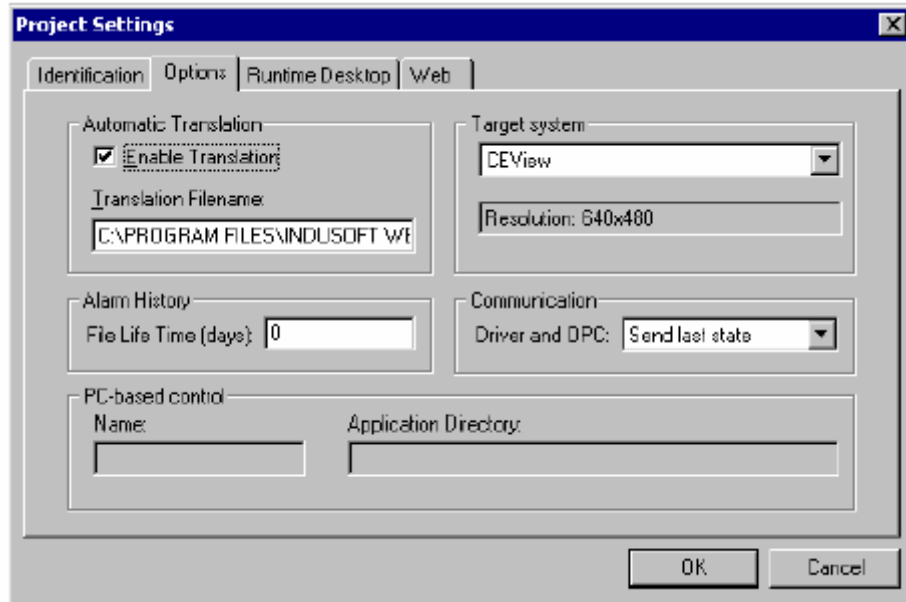
## Prevodjenje jezika

Alat za prevodjenje omogućava korisniku da promjeni tekstove na tasterima ili bilo kojem drugom polju teksta. Sve što je potrebno za ovu funkciju je da kreiramo translacione liste i koristimo funkcije prevodjenja.

Kada koristimo prevodilačku funkciju, ona traži sve tekstove koji su uneseni u kolonu "Original" i zamjenjuje ga sa tekstom koji je unesen u koloni "Translation". Ako naša aplikacija ima druge različite tekstove koji nisu uneseni u listu za prevodjenje, ovi tekstovi će ostati u originalnom obliku.

## Omogućavanje translacije

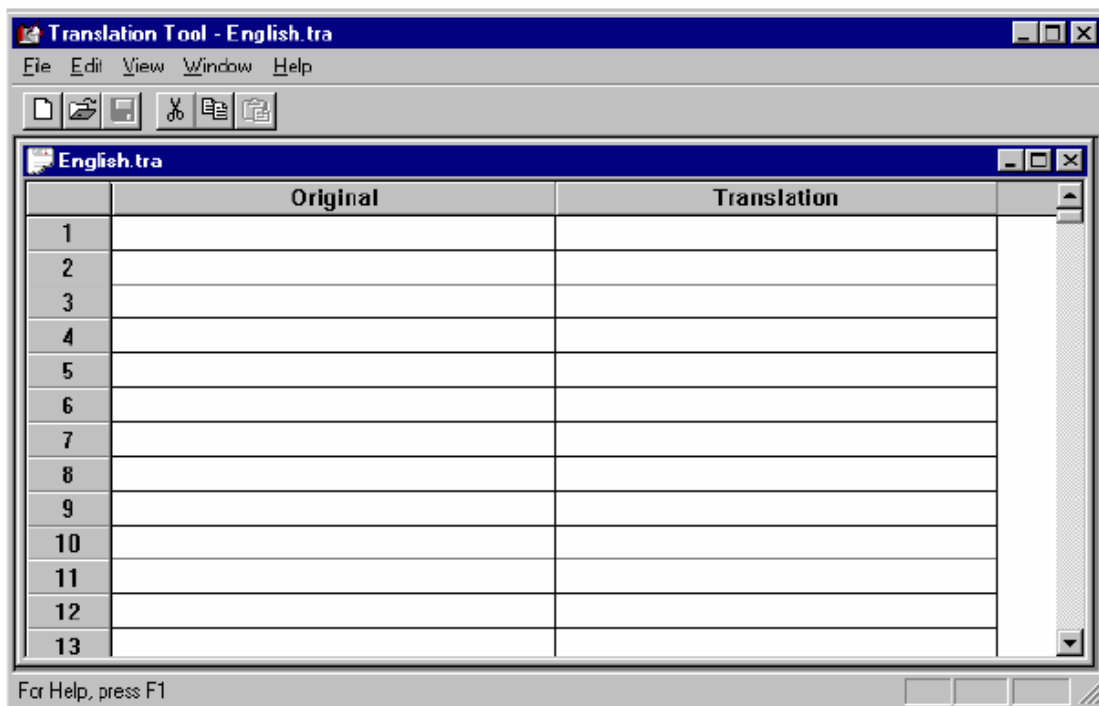
Da bi imali uspješno prevodjenje, prvi korak je da otvorimo prozor "Project Settings", u "Options" tabu. Naći ćemo na check box "Enable translation" kojeg treba čekirati.



## Kreiranje translacionih lista

Potrebno je kreirati prevodilački list za svaki jezik koji želimo da konfiguriramo u našoj aplikaciji uključujući i originalni jezik ( Engleski ).

Potrebno je kreirati za default jezik ( engleski ) nepopunjeni ( blank ) list i pohraniti ga sa imenom **English.tra** kao na slici :

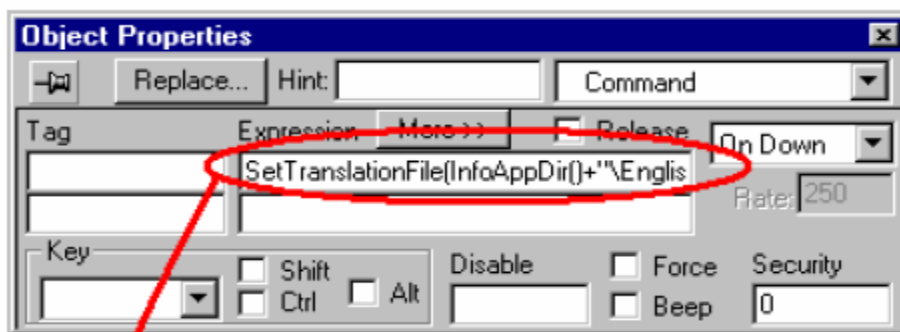


Sada je potrebno kreirati novi prevodilački list za svaki jezik koji hoćemo da imamo u prevodjenju.

Naprimjer za prevodjenje na bosanski kreiraćemo list sa desnom kolonom prevodjenja originalnih engleskih riječi i stringova teksta na bosanski i pohraniti ga kao **bosanski.tra**.

### Kreiranje translacionog ekrana

Potrebno je kreirati na ekranu sa kojeg želimo da korisnik izabere jezik postaviti tastere za izbor jezika. Na engleskom tasteru pridružimo komandnu funkciju , i ispuniti polje za **Objects Propreties** sa:

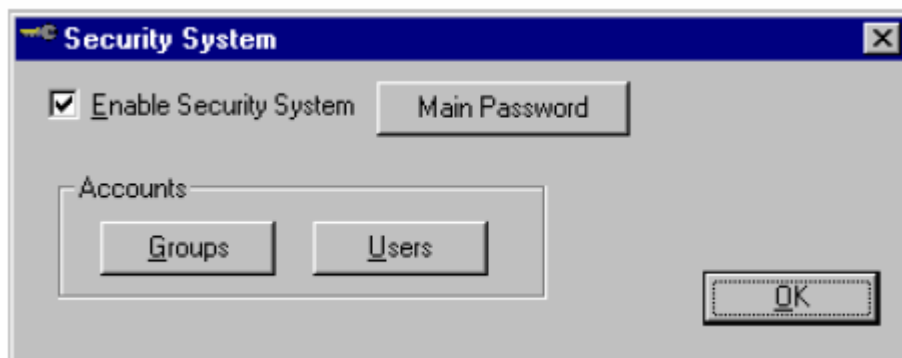


**SetTranslationFile(InfoAppDir()+\"English.tra\")**

U komandnom polju za taster Bosanski, unjeti izraz ; SetTranslationFile(InfoAppDir()+\"Bosanski.tra\").

### Konfigurisanje sistema sigurnosti

Folder **Security** nam omogućava da definišemo grupe i korisnike kao i njihova prava pristupa aplikaciji i IWS alatima. Preko taba **Database**, možemo selektirati i kreirati nove grupe i korisnike. Da bi pristupili **Security System** prozoru, desno kliknuti na **Security** folder i izabrati "Settings":





**Enable Security System Check box** – omogućuje IWS sistem sigurnosti.  
**Main Password Button** – otvara prozor Security System Main Password

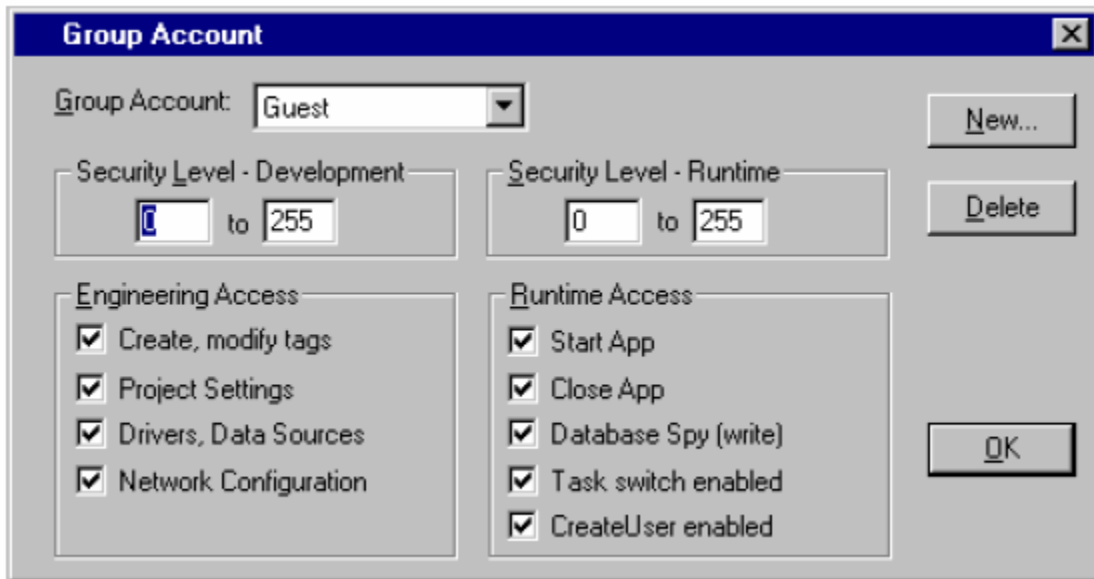
### Account Group Box

**Groups Button** – otvara **Groups** prozor

**User Button** – otvara **Users** prozor

### Groups ( Grupe )

**Groups Account Button** , na prozoru **Security System** , otvara prozor za račune grupa u kojem možemo kreirati i održavati grupe.



**Group Account Drop list** - omogućava izbor grupe kojoj korisnik pripada.

**Security Level Development Group Box** – definira nivo sigurnosti za svaku grupu ( 0 do 255 ). Svaki objekat za unos podataka u ekranu displeja ( kao što su ulazne komande , klizači –sliders, ili ekrani ), ima polje **Security Level**. Ako nivo objekta nije u grupi sigurnosti koja je trenutno logovana, tada je objekat onemogućen. Nivo 0 ( nula ) znači da je objekat uvijek omogućen.

**Security Level Runtime Group Box** - definira nivo sigurnosti za svaku grupu ( 0 do 255). Svaki objekat za unos podataka na displejiranom ekranu ( kao što su ulazne komande, klizači, ekrani ), ima polje **Security Level**. Ako nivo objekta nije u grupi sigurnosti koja je trenutno logirana, objekat će biti onemogućen. Nivo 0 znači da je objekat uvijek omogućen.

**Engineering Access Group Box** - Izlistava inženjerske ( razvojne ) taskove kojima korisnik može pristupiti kada je logiran u sistem. Uključuje i check

boksove za **Create, modify tags, Project Settings, Drivers, Data Sources , Network Configurations.**

**Runtime Access Group Box** – runtime moduli kojima dati korisnik može pristupiti. Uključuje check boksove, za **Start App, Close Appl, Database Spy, Task switch enabled , CreateUser enabled.**

**New Button** – otvara prozor **New Group Account** , u kojem možemo kreirati novu grupu.

**Delete Button** – briše selektiranu grupu korisnika.

### **Nivoi pristupa sigurnosti**

U prozoru **Group Account**, moguće je postaviti opseg vrijednosti pristupa u **Security Level- Development** boksu za grupu. Svakoj grupi se može doznačiti njen opseg vrijednosti. Kada se otvori bilo koji IWS radni list ( Alarm, Math, Recipe, Report, Scheduler, TCP Client, Trend, kao i oni koji nisu raspoloživi u WinCE kao: DDE Client, OPC Client, i ODBC ), moguće je postaviti opseg pristupa tom listu.

Treba kliknuti na bilo koji dio tijela radnog lista da se aktivira **Access Level** opcija pod **Edit** na glavnom baru menija. Kada se izabere **Access Level**, otvori se prozor u kojem se može doznačiti broj nivoa pristupa ( Access level ). Ovo znači sa bi se editirao list, biće potrebno da radni list ima **Access Level** unutar **Security Level – Development** opsega grupe korisnika koji je ulogovan u sistem.

Naprimjer, KorisnikA iz GrupeA ima **Security Access Level** opseg 0-10, KorisnikB iz GrupaB ima **Security Access Level** opseg od 5-15. Da nastavimo ovaj primjer , biće :

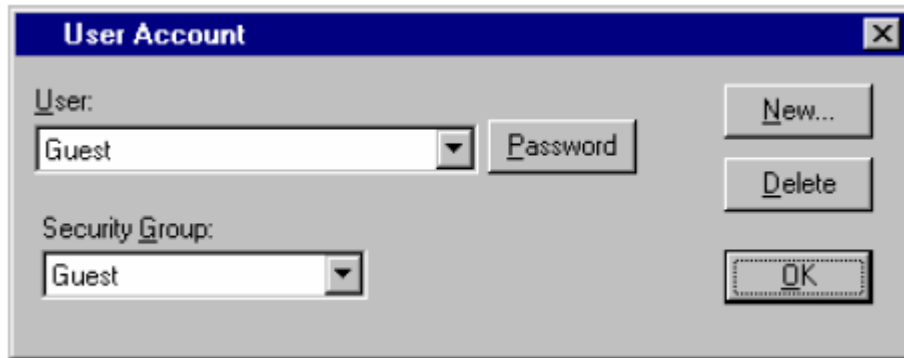
Math radni list 001 će imati Access Level = 1  
Math radni list 002 će imati Access Level = 7  
Math radni list 003 će imati Access Level =12  
Math radni list 004 će imati Access Level =20

U ovakvoj situaciji, samo KorisnikA može pristupiti Math radni list 001, obadvojica i KorisnikA i korisnikB mogu pristupiti Math radnom listu 002, samo KorisnikB može pristupiti Math radnom listu 003, i niko od KorisnikA i KorisnikB ne može pristupiti Math radni list 004.

### **Korisnici**

Taster **User Account Button** na prozoru **Security System** , otvara **User Account** prozor u kojem možemo kreirati i održavati račune za aplikacione korisnike. Definisaćemo aplikacione korisnike koji će biti u svakoj grupi u listi

**Group Account.** Korisnicima se takodjer može pristupiti otvarajući **User** folder unutar **Security** foldera ili sa selekcijom **User** opcije pod **Insert** u glavnom meniju.



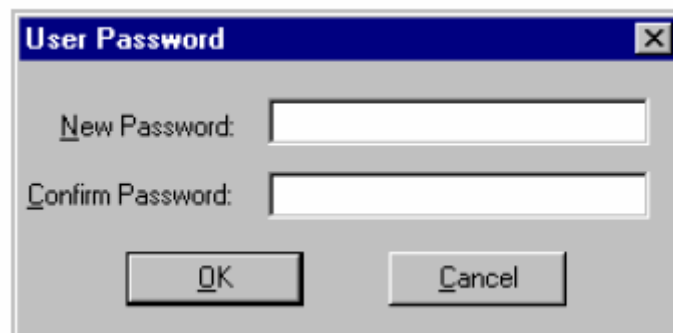
**User Drop list** – izlistava aplikacione korisnike u drop listi

**Security Group** - lista aplikacione grupe

**New Button** - otvara prozor **New User Account** da kreira novog korisnika

**Delete Button** – briše selektiranog korisnika

**Password Button** – otvara prozor **User Password**, u kojem se može definisati lozinka za korisnika.



### Kreiranje alarmnih grupa

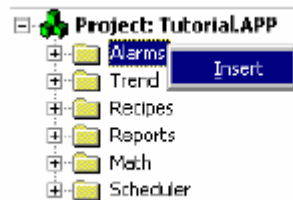
Prije kreiranja ekrana sa alarmnim objektima, treba da kreiramo alarmne grupe. Kreiraćemo Alarm\_Settings tag tipa "Class:Calarm", kao što se vidi nadalje:

| Class: CAlarm |          |         |
|---------------|----------|---------|
|               | Name     | Type    |
| 1             | StartDay | Integer |
| 2             | EndDay   | Integer |
| 3             | Month    | Integer |
| 4             | Year     | Integer |

Nakon kreiranja ove klase, kreiraćemo tagove. Prvi je tag kojeg ćemo nazvati **Alarm\_Settings**, tipa "CAlarm".

Kreirajmo takodjer string TAG kojeg ćemo koristiti kao alarmni filter. Ovaj tag će biti imenovan **Alarm\_Sel**, tipa string kao i tag tipa integer kojeg ćemo nazvati **View**, te dva Boolova taga **PGUp** i **PGDown**. Svi ovi će biti polja dimenzije 2.

Da bi kreirali novu alarmnu grupu, kliknimo desno na **Alarm** folder ( u radnom prostoru **Tasks** ) i izaberimo opciju **Insert**:



Konfigurisati list alarmne grupe kao na slici:

|   | Tag Name      | Type | Limit     | Message           | Priority | Selection |
|---|---------------|------|-----------|-------------------|----------|-----------|
| 1 | Tank[1].Level | Hi   | 80.000000 | High Level Tank 1 | 1        | A         |
| 2 | Tank[1].Level | Lo   | 20.000000 | Low Level Tank 1  | 1        | A         |
| 3 | Tank[2].Level | Hi   | 80.000000 | High Level Tank 2 | 2        | B         |
| 4 | Tank[2].Level | Lo   | 20.000000 | Low Level Tank 2  | 2        | B         |
| 5 | Tank[3].Level | Hi   | 80.000000 | High Level Tank 3 | 3        | C         |
| 6 | Tank[3].Level | Lo   | 20.000000 | Low Level Tank 3  | 3        | C         |

Kada izaberemo **Save to Disk** boks, sistem će zapisati sve alarme u ASCII fajl u alarm subdirektorij od direktorija aplikacije, sa ekstenzijom *hst*.  
Objašnjenje značenja pojedinih polja je:

**Group Name Field** : ime koje se koristi da razlikuje alarmne grupe

**Description Field** – unosi primjedbe i komentare radi dokumentovanja

**Disable field** – onemogućava sve alarme u grupi. Ovo polje treba biti popunjeno sa tagom. Ako je vrijednost taga iznad nule, grupa je onemogućena, i alarmne poruke se ne generišu. Ako polje ostaje nepopunjeno, grupa će biti uvijek omogućena.

**Remote Ack Field** - tag za potvrdu alarma. Potvrđivanje će se desiti kada postoji promjena u vrijednosti ovoga taga.

**Total Active Field** - sadrži ukupan broj aktivnih alarma u grupi. Sistem uvijek ažurira ovu vrijednost kada neki od alarma promjeni svoje alarmno stanje.

**Total Active or Unack Field** – sadrži ukupan broj aktivnih ili nepotvrđenih alarma u grupi. Sistem uvijek ažurira ovu vrijednost kada jedan od tagova promjeni svoje alarmno stanje.

### Group Box

**Summary Check-Box** – kada se izabere, šalje alarmne poruke ka alarmnom objektu na ekranu.

**Opaska :** Ako nismo izabrali **Summary** opciju, alarmi iz ove grupe se neće pojaviti u alarmnim objektima na ekranu ili printeru.

**Ack Check-Box** – Zahtjeva potvrđivanje alarmnih poruka. Raspoloživ samo ako su **Summary** polja omogućena.

**Beep Check-Box** – aktivira zvučni beep alarm sve dok se alarm ne potvrdi. Biće raspoloživ samo u slučajevima kada su polja **Ack** i **Summary** omogućena.

**Printer Check-box** – šalje alarmne poruke iz ove grupe na štampač. Ova opcija može biti korištena samo sa printerima koji štampaju liniju po liniju ( kao što je naprimjer dot matrix printer )

**Disk Check-box** – šalje alarmne poruke iz ove grupe na fajl na hard disku.

**Generate Ack Messages Check-box** - Generira poruke kada su alarmi iz ove grupe potvrđeni. Raspoloživa je samo kada su polja **Disk** ili **Printer** omogućena.


**Generate Norm Message Check-box.** Generira poruke kada god alarmi iz ove grupe se vraćaju u njihovo normalno stanje. Raspoloživa je samo kada su polja **Disk** ili **Printer** omogućena.

### Colors Group Box.

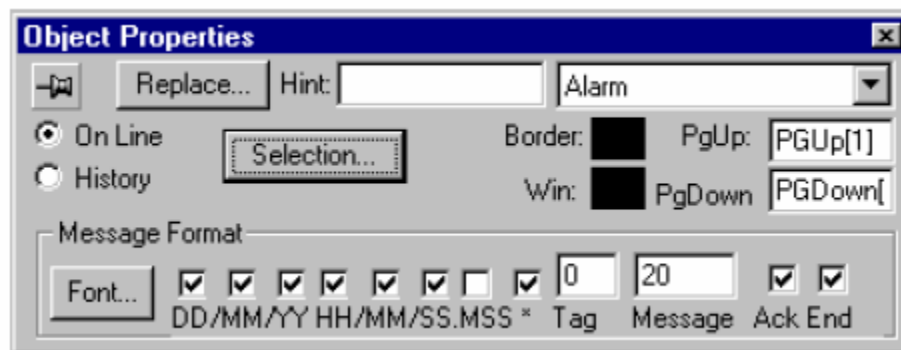
Kada se čekira **Enable** boks, korisnik može izabrati boje alarmnih poruka. Inače, koristiće se default boje.

### Kreiranje on-line alarmnih ekrana

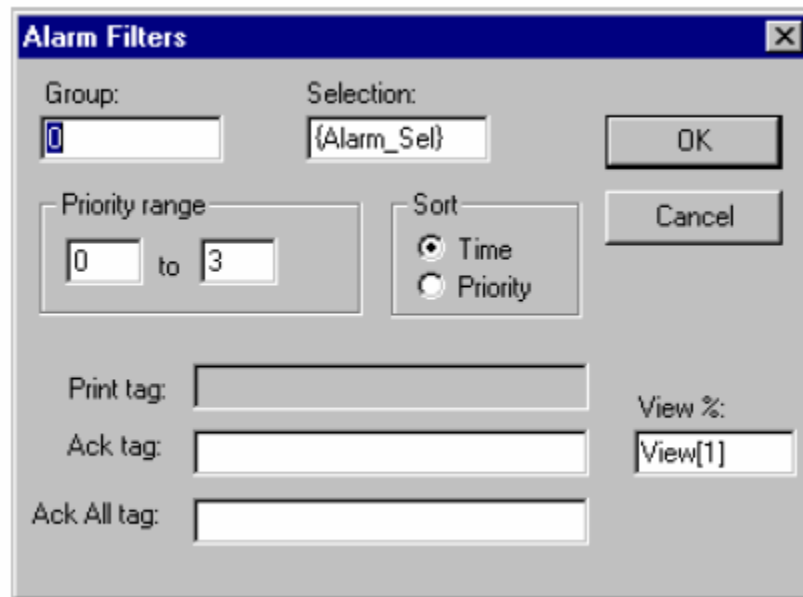
Otvorimo standardni ekran i pohranimo ga pod nazivom "AlarmOnline".

Kreirati  "Alarm" objekat na ekranu .

Dvostruko kliknuti na alarmni objekat da iskoči prozor "Object Alarm Properties".



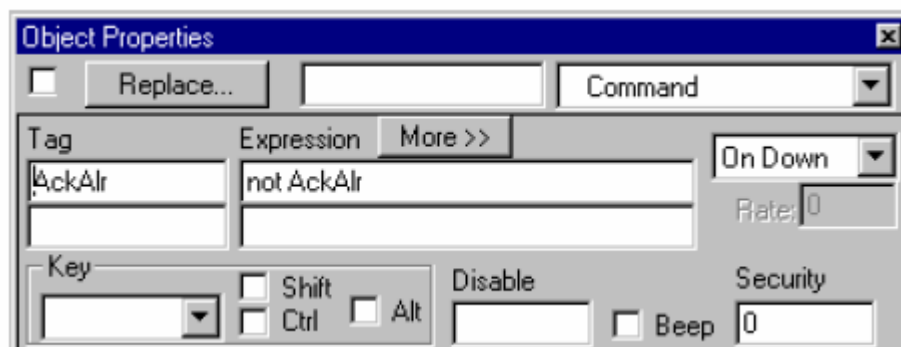
Pritisnuti Selection taster da konfiguriramo filter pokazan na sljedećoj slici:



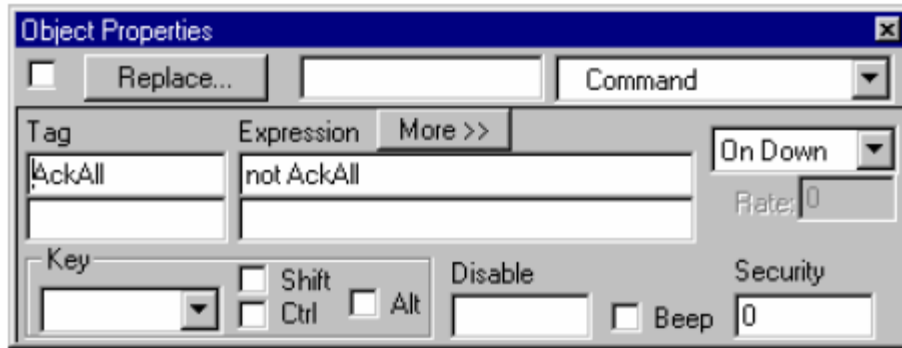
Kreirajmo sada dva tastera na ekranu da potvrđujemo ove alarme:



Prvi taster ćemo konfigurirati sa poljem "**Ack Last**" koristeći interni tag **AckAlr**, koji mora promijeniti svoju vrijednost da bi potvrdio posljednji alarm koji se pojavio. Konfiguriramo prvi taster kao što je pokazano na slici:



Drugi taster treba biti konfigurisan sa osobinama kao na sljedećoj slici:



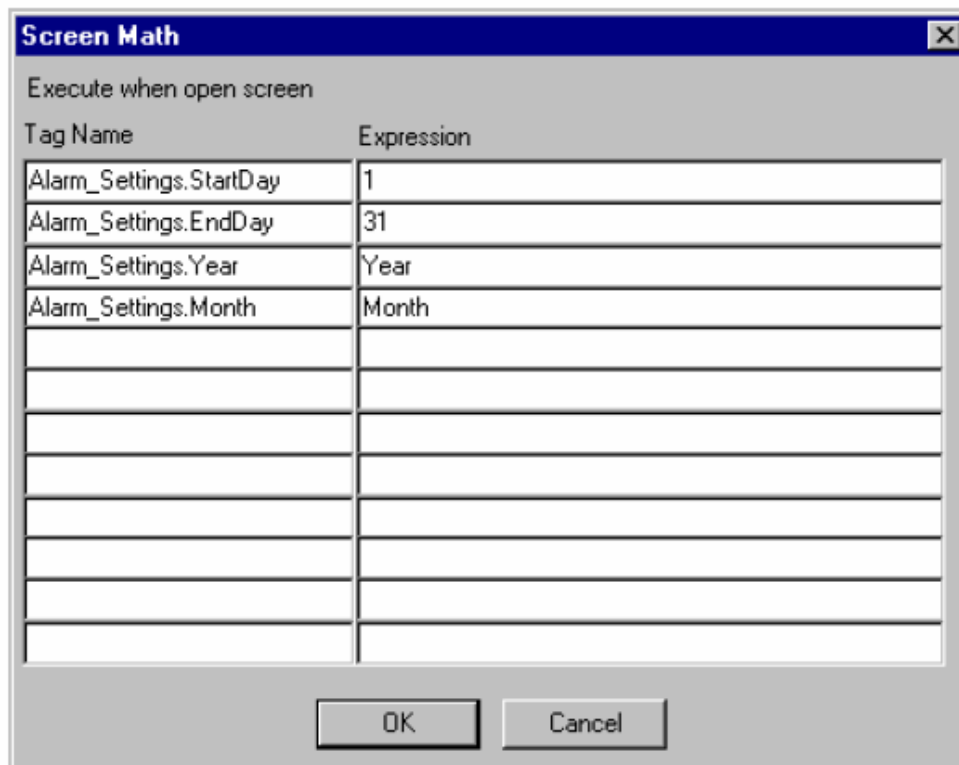
Nacrtati zatim tekst i objekte pokazane na narednoj slici i konfigurirati ih sa **Alarm\_Set tagom** da možemo sortirati alarmne poruke :

Selection:  A  B  C  All

### Kreiranje ekrana historijskog alarma

Otvoriti "AlarmonLine" ekran i zatim ga pohraniti kao "AlarmHistory".

U "Screen Attributes" dijalogu prozoru, izabrati u odjelu "Screen Logic", "On Open" check polje, i u prozoru koji će se otvoriti, unjeti slijedeću konfiguraciju:



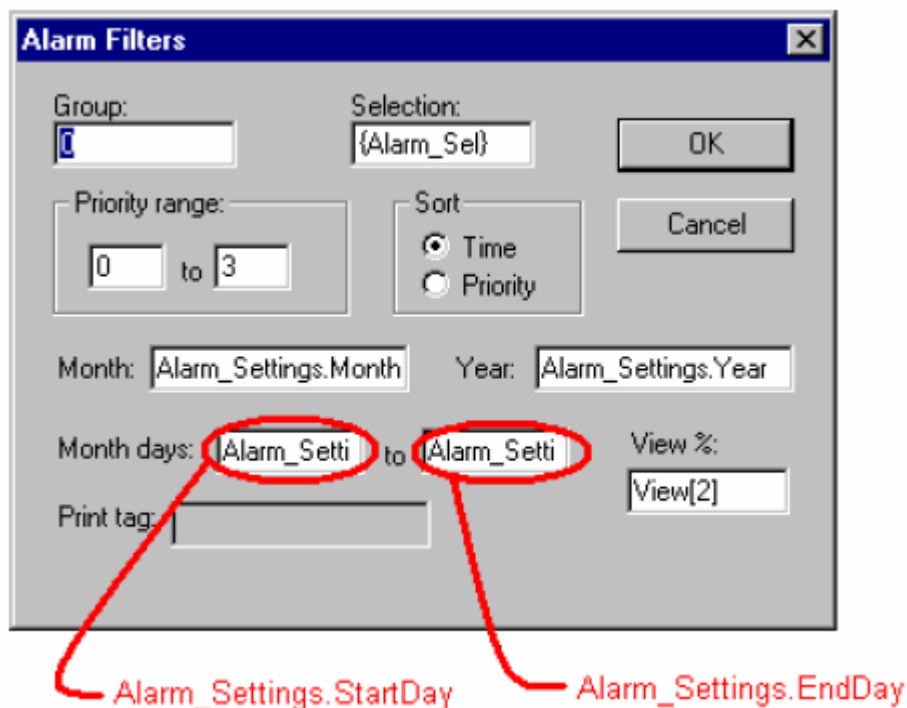


Kreirati "Alarm" objekat na ekranu , klikajući na "Alarm List" ikonu i selektirajući oblast na ekranu da prikaže alarme.

U prozoru "Object Alarm properties" konfigurirati kao na slijedećem ekranu:



pritisnuti **Selection** taster i konfigurirati filter kao na slijedećem prikazu :

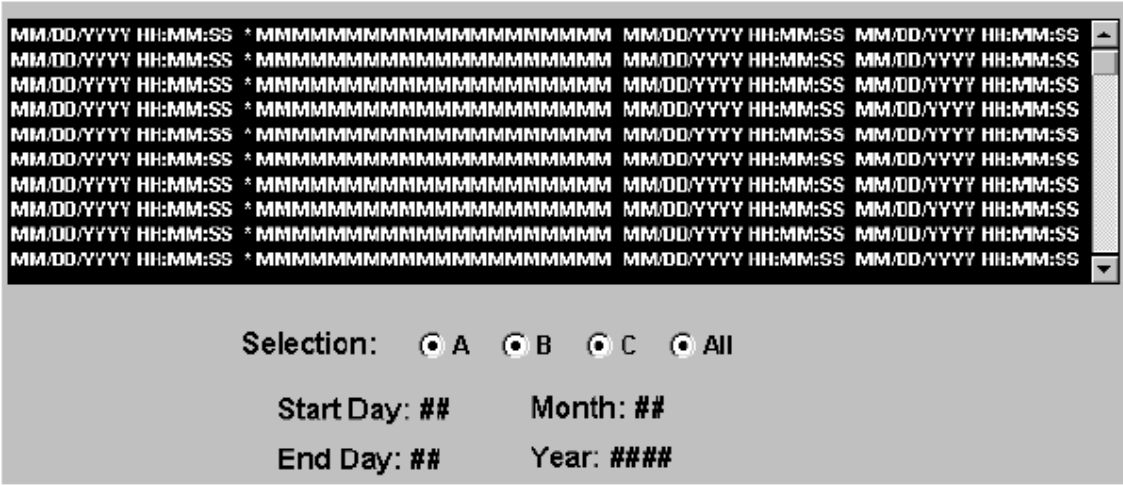


Na vertikalnom slajderu za alarmni prozor , zamjeniti ranije konfigurisane tagove PGUp[1], PGDown[1] i View[1] sa indeksima 2.


Da bi izabrali mjesec, godinu i dane na alarmnom objektu, kreirati Text I/O objekte i pridružiti im tagove **Alarm\_Settings.StartDay**, **Alarm\_Setrtings.EndDay**, **Alarm\_Settings.Month** i **Alarm\_Settings.Year** .

|               |            |
|---------------|------------|
| Start Day: ## | Month: ##  |
| End Day: ##   | Year: #### |

Konačno , završeni ekran historijskog alarma treba da izgleda kao na slijedećoj slici:



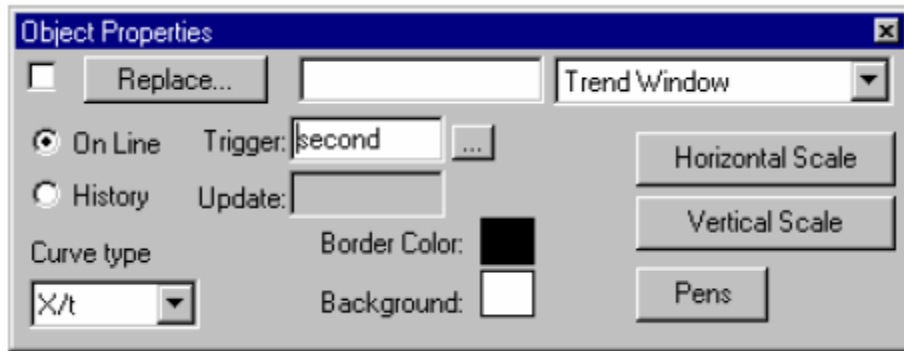
### Trendiranje

Trend task vodi evidenciju o ponašanju procesnih varijabli. Mi možemo pohraniti uzorke u historijski fajl i pokazati i historijske i online ( realno vrijeme ) uzorke u ekranima trend grafova. Da bi pokazali trend graf na ekranu, moramo kreirati trend objekat sa Trend  ikonom, na toolbaru alata za editiranje. a za pohranjivanje historijskih varijabli mi moramo kreirati radni list trenda.

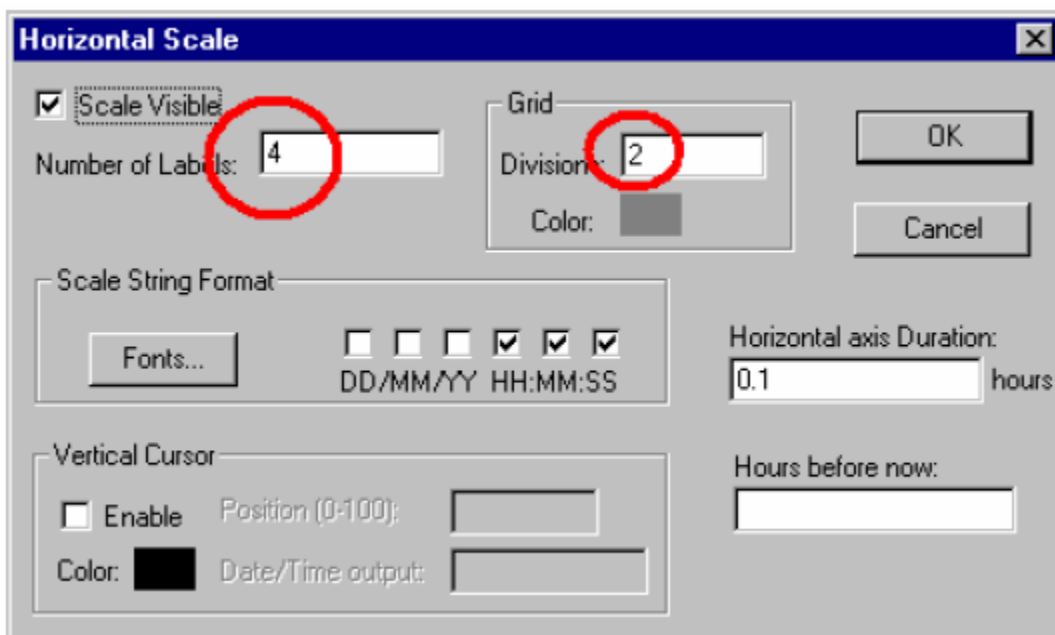
### Online ( real time ) trendiranje

Pokazaćemo u online trendu ponašanje temperatura u realnom vremenu, ažurirano svake sekunde. Otvorićemo standardni ekran i pohraniti ga kao "TrendonLine".

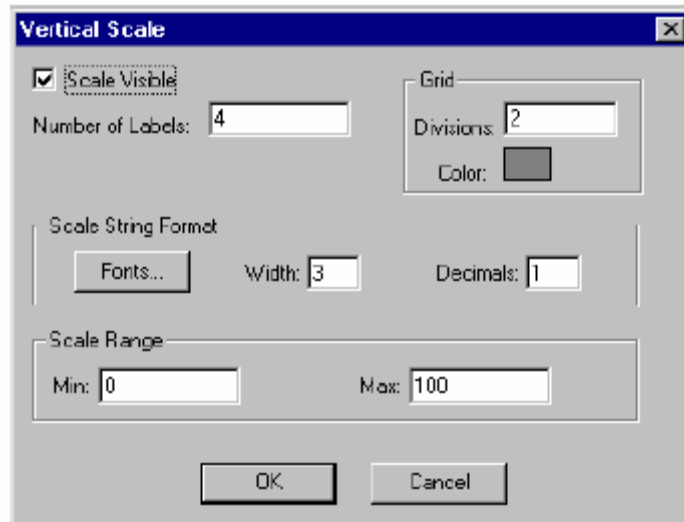
Konfigurisati insertirani trend objekat ka ona slici:



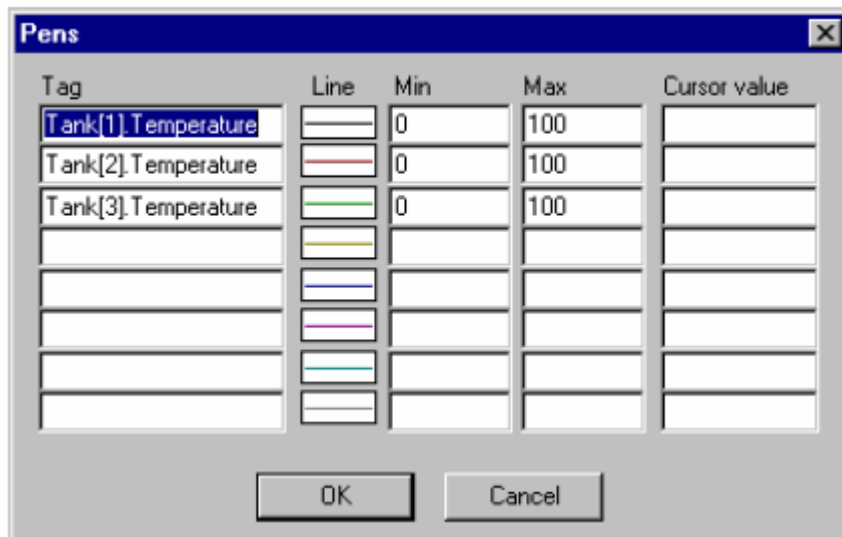
Kliknuti na "**Horizontal Scale**" i editirati kao na sljedećoj slici:



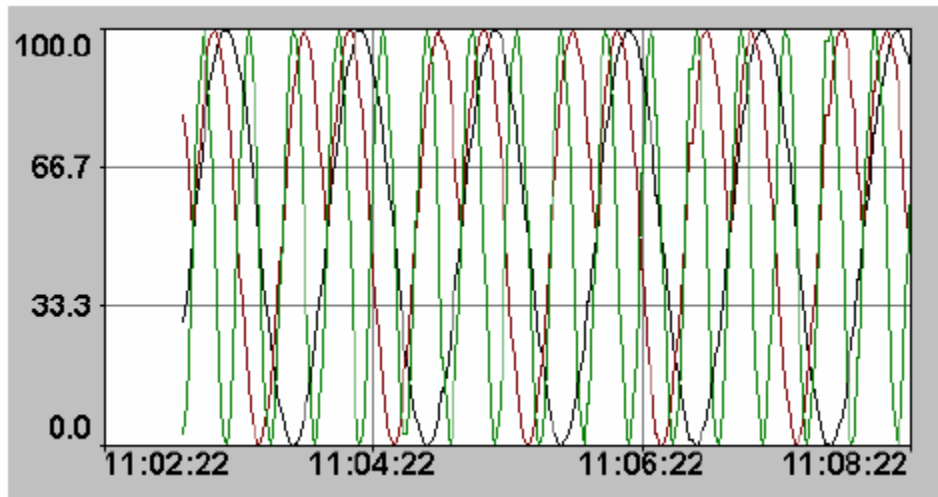
Kliknuti na "**Vertical scale**" i editirati kako slijedi :



Kliknuti na "**Pens**" i editirati kako slijedi:



Izvršiti ekran u run timu. Nakon nekoliko minuta , ekran će izgledati kao na slijedećoj slici:



### Kreiranje historijskog trenda

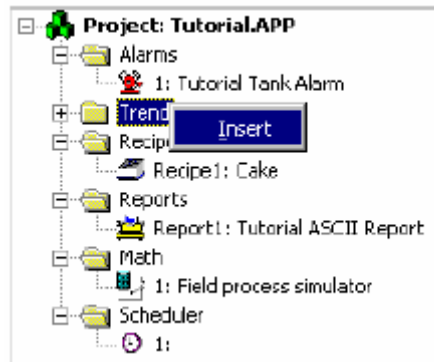
Pri kreiranju historijskog trenda pokazaćemo neke dodatne resurse koji su raspoloživi u okviru IWS paketa. Naravno, većina ovih resursa je na raspolaganju i kod kreiranja real time trendova.

### Kreiranje trend grupe

Prije kreiranja ekrana sa trend objektima, kreiraćemo grupu tagova koji će biti pohranjeni na disku da bi pokazali historijski trend u budućnosti. Jedan od tagova bit će nazvan **Trend** i pripadaće klasi **Ctrend**, kao što je pokazano na slijedećoj slici:

| Class: Ctrend |                |         |  |
|---------------|----------------|---------|--|
|               | Name           | Type    | Description                            |
| 1             | HiLim          | Integer | Trend Hi Limit                         |
| 2             | LowLim         | Integer | Trend Low Limit                        |
| 3             | Duration       | Real    | Trend Duration                         |
| 4             | StartDate      | String  | Trend Start Date                       |
| 5             | StartTime      | String  | Trend Start time                       |
| 6             | CursorOutput   | String  | Trend Cursor Date/Time Output          |
| 7             | CursorPosition | Real    | Trend Cursor Position                  |
| 8             | CursorPen1     | String  | Trend Intersection cursor/Pen 1 output |
| 9             | CursorPen2     | String  | Trend Intersection cursor/Pen 2 output |
| 10            | CursorPen3     | String  | Trend Intersection cursor/Pen 3 output |
| 11            | Update         | Boolean | Trend update trigger (Scheduler)       |

Kreirati novu trend grupu desno kliknuvši na **Trend** folder ( u radnom prostoru pod **Task** tabom ), i izabrati opciju **Insert**:

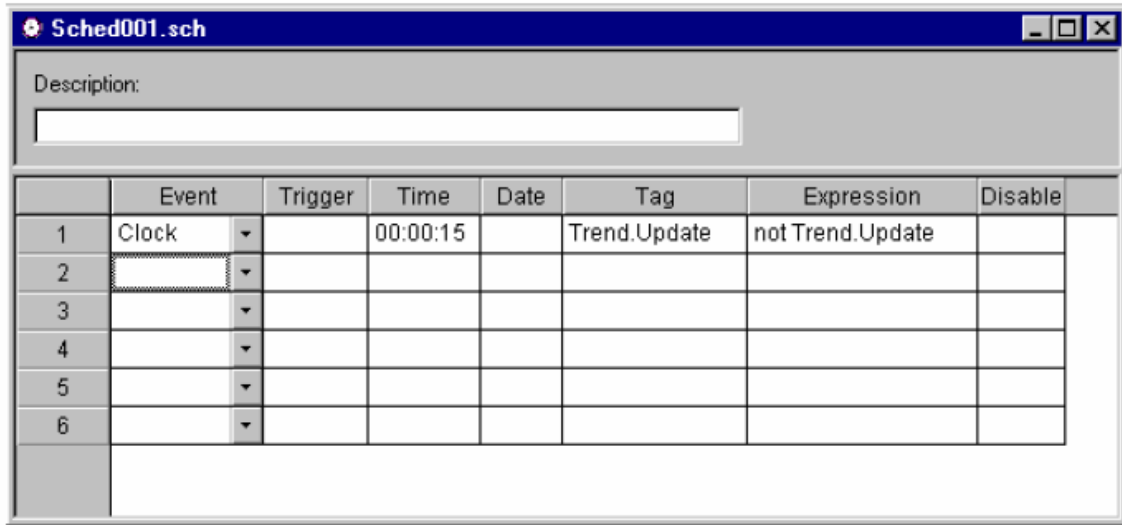


Konfigurirati radni list trend grupe kao na slici:

|   | Tag Name            | Dead Band |
|---|---------------------|-----------|
| 1 | Tank[1].Temperature |           |
| 2 | Tank[2].Temperature |           |
| 3 | Tank[3].Temperature |           |
| 4 |                     |           |
| 5 |                     |           |
| 6 |                     |           |

Sa ovim worksheetom mi konfiguriramo sistem da pohranjuje vrijednosti tagova **Tank[1].Temperature**, **Tank[2].Temperature**, **Tank[3].Temperature**. Check boks "Save on Trigger" i polje pored njega definiše brzinu pohranjivanja.

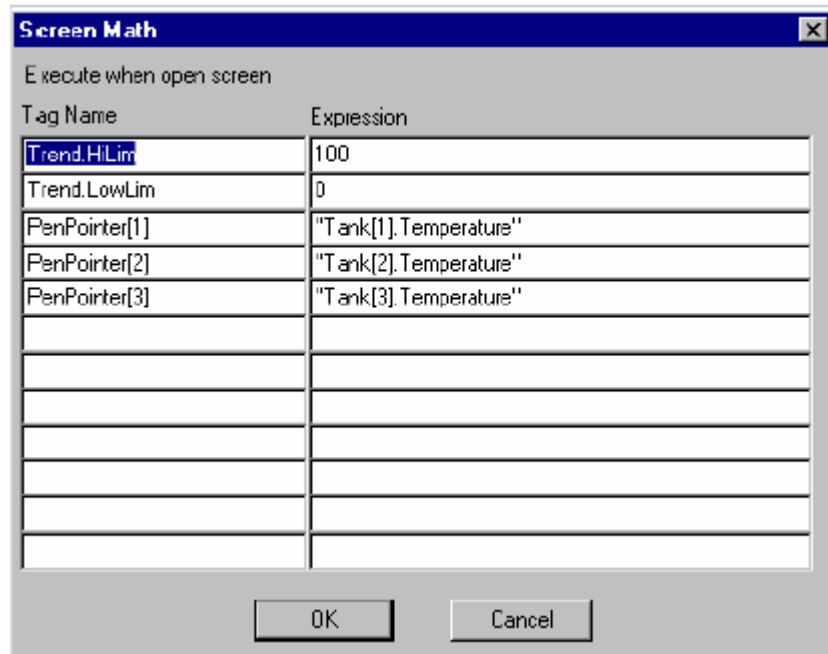
Da bi definirali brzinu pohranjivanja, kreiramo liniju u skeduleru koja će togirati (prebacivati on/off) **Trend.update** svakih 15 sekundi.



### Kreiranje ekrana za historijski trend

Otvorimo ekran "**TrendOnLine**" i pohranimo ga kao "**TrendHistory**".

Pošto ćemo koristiti mnogo varijabli, najbolje je da njihovu inicijalizaciju uradimo sa "**Screen Open**" logikom. Pritisnuti taster na **On Open** da konfiguriramo skript kao na narednoj slici:

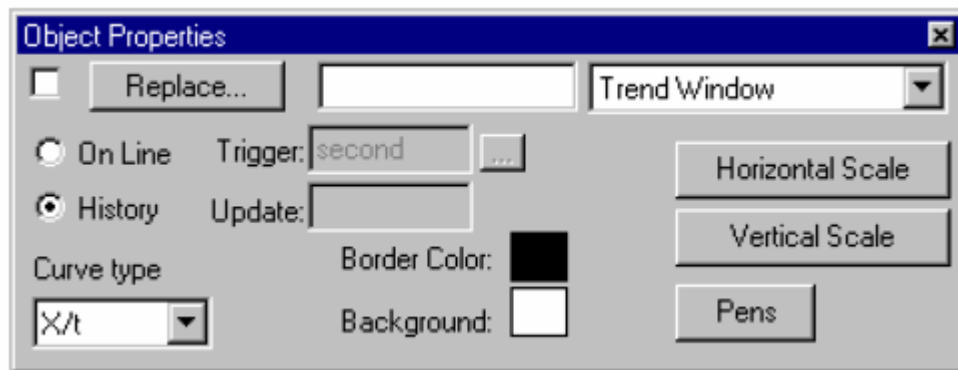


Ponekad je potrebno **podesiti Date format**. IWS default format je **MM/DD/YYYY**. Ako želimo drugačiji format , treba izvršiti funkciju **SetDateFormat ( "/" , "DMY")**.

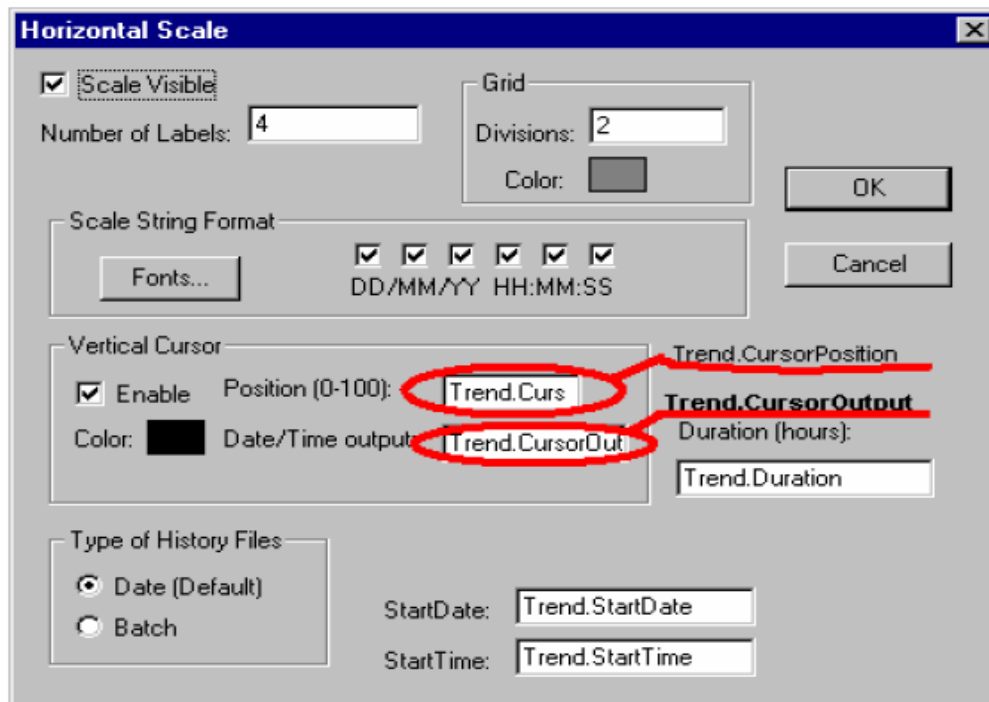
Ova funkcija definiše separator podataka ( u našem primjeru smo koristili uobičajeni "/" ali možemo ga zamjeniti sa "." , "-" , itd. ) , a drugi dio je redosljed ( mi smo uzeli "DMY" , a može biti i "YMD" ili neki drugi koji želimo).

Skript koji smo unjeli definiše imena pera i puni ih u tagove pointera i setuje granice trenda.

Kliknimo na trend objekat i popunimo **"Object Trend Properties"** kao na slici koja slijedi:



Zatim kliknuti na Horizontal Scale taster i konfigurisati kako slijedi:

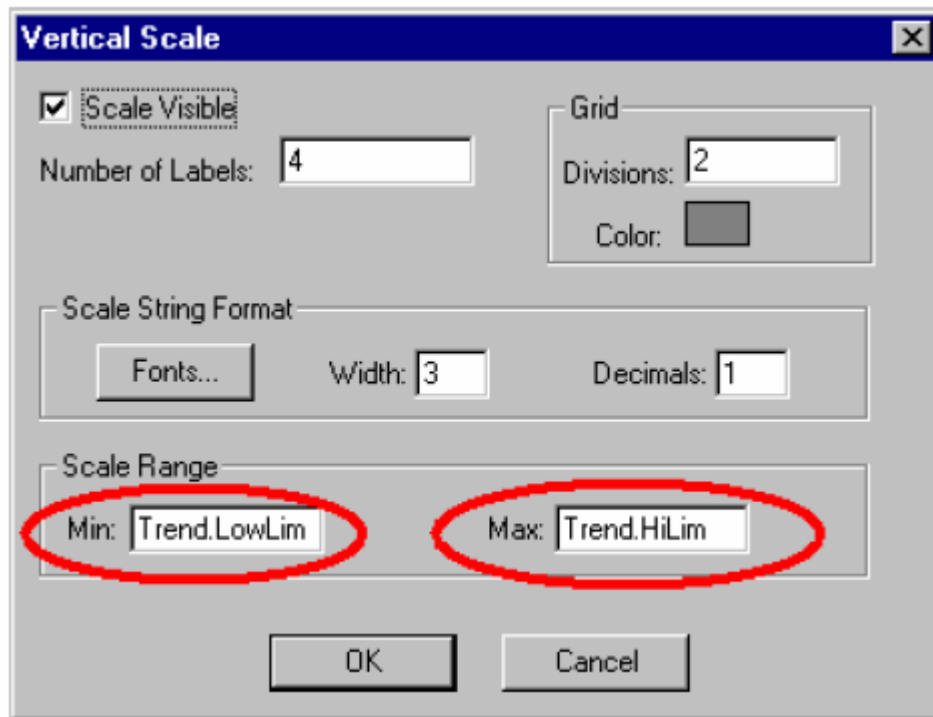




Kurzor nam omogućava da znamo vrijednosti varijabli u različitim pozicijama charta. Polje "**Position**" će biti korišten kod **slajd objekta i** tag unutar "Date/Time Output" polja prima datum i/ili vrijeme pozicije kurzora.

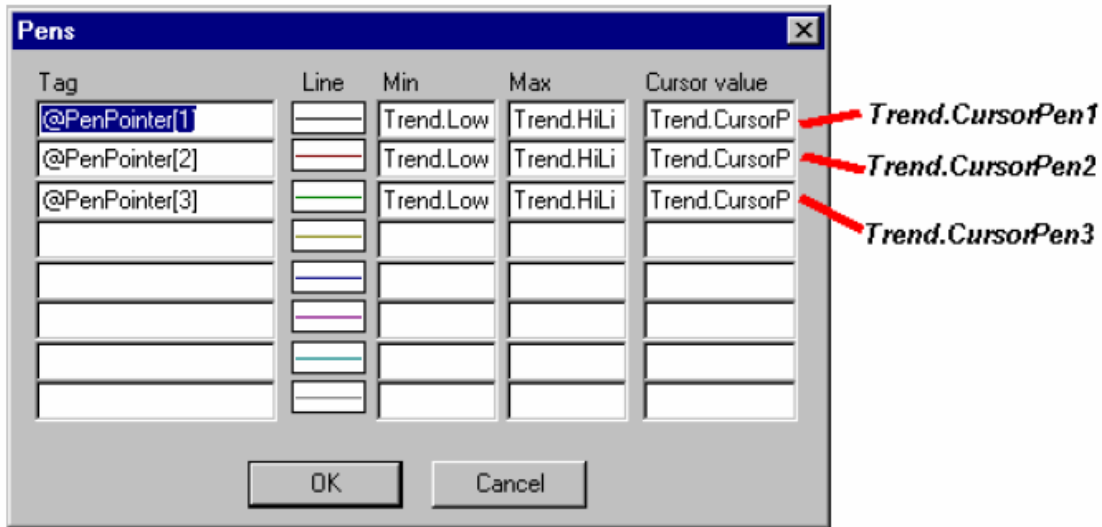
Primjetimo da polje "Duration" kao i StartDate i StartTime polja su popunjena sa Tagovima. Mi ćemo kreirati tekstove sa "Text I/O" i omogućenim ulazom da dozvolimo unošenje dinamičkih ulaza.

Pritisnuti **Vertical Scale** taster i konfigurirati ga kao na slici:



Sada se polja Min i Max u Scale Range pune sa tagovima. Ovi tagovi će imati Tekst I/O sa ulazima da se omogući konfigurisanje u runtimeu.

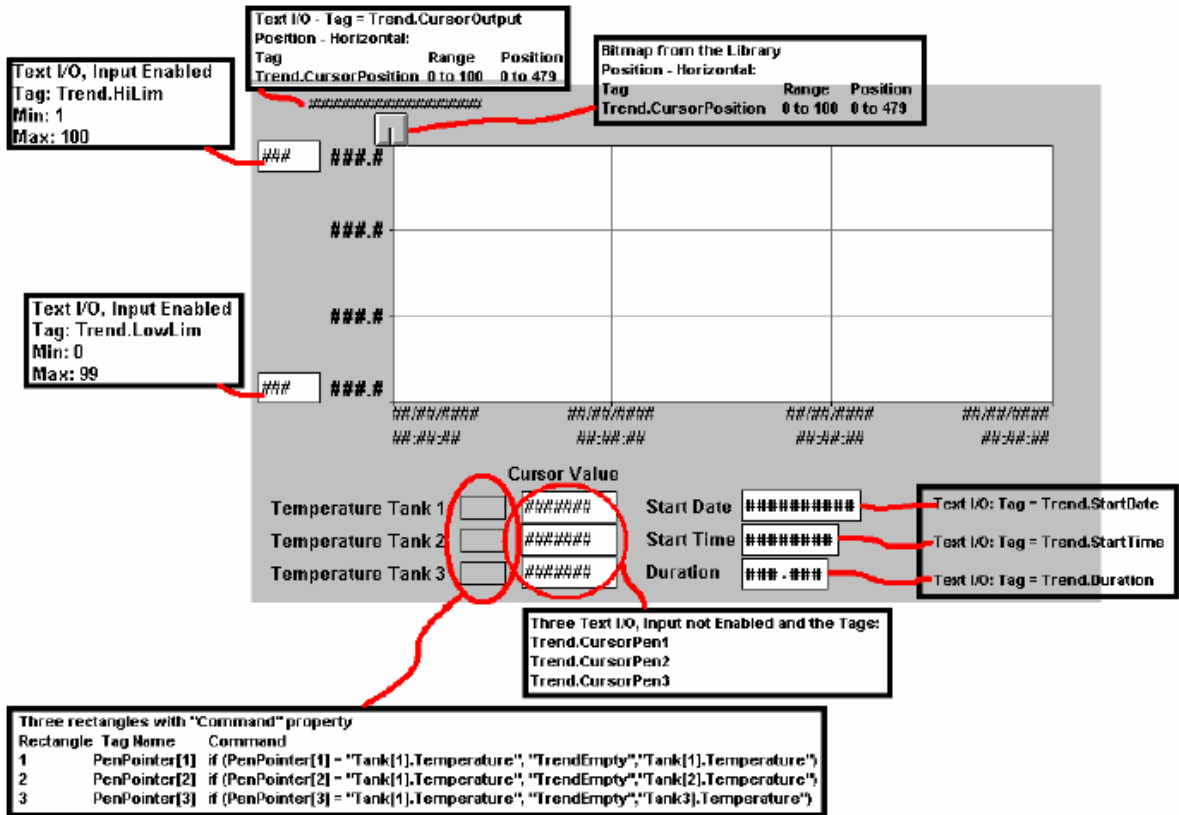
Pritisnuti na **Pens** taster i konfigurirati kako slijedi:



Sa ovom konfiguracijom mi smo u stanju da promjenimo granice Charta (**Trend.Low i Trend.High**) i tagovi **Trend.CursorPen1,2 i 3** će primiti presjeke između krive pera i traga kursora.

Tag Pointer unutar Tag polja će pointirati na **Tank.Temperature** tagove ili na prazni pomoćni tag koji ćemo zvati EmptyTag. Ovaj resurs nam omogućava da izaberemo da sakrijemo pera, kao što ćemo vidjeti kod runtimea.

Preostaje nam još da nacrtamo preostale objekte na ekranu, kao na narednoj slici i konfigurirati ih kako je naznačeno:



## ODBC

ODBC task je sposoban da organizuje razmjenu podataka izmedju IWS aplikacije i bilo koje relacione baze podataka koja podrzava ovaj interfejs. ODBC interfejs se izvršava u mrežnom okruženju i takodjer koristi standardnu konfiguraciju Windows ODBC.

### Instalacija

Pored IWS ODBC programa , treba takodjer konfigurisati Windows ODBC standardni drajver. Studio komunicira sa **User DNS** , čija konfiguracija se vrši pomoću **Control panela**.

### ODBC radni list

Desno kliknuti na ODBC folder da se ubaci novi radni list. Novi radni list će se pojaviti koji izgleda kao na slici:

|   | Tag Name | Column |
|---|----------|--------|
| 1 |          |        |
| 2 |          |        |
| 3 |          |        |
| 4 |          |        |
| 5 |          |        |

Kao i ostali radni listovi, i ovaj je podjeljen u dva dijela: zaglavlje ( header ) sa informacijama za cijelu grupu i tijelom ( body) koji sadrži tagove i reference ka vrijednostima koje će se čitati iz ili upisivati u bazu podataka.

### Zaglavlje ODBC radnog lista

Zaglavlje nam omogućava da definiramo tagove da bi startali iščitavanje i upisivanje tagova, postavljali vrijednosti koje će se vraćati, manipulirali sa parametrima pristupa bazi podataka, itd.

**Description field** - ovdje unosimo komentare radi dokumentovanja projekta

**Data Source name Field** – Isto **Data Source Name** ime koje smo konfigurirali u Windows Control Panel koje sadrži informaciju o pristupu specificiranoj bazi.

**User Field** – Ime korisnika koji ima pristup bazi podataka

**Password Field** - polje za unos lozinke korisnika

**Table Field** - ime tabele u bazi podataka

**Condition Field** - uslov traženja ili filter

**Status Field** - povratna vrijednost ( popunjena sa imenom taga )

0 - uspjeh

Bilo koja druga vrijednost - kod greške

**Transaction Field** - Popunjen sa tagom koji će imati promjenu vrijednosti kada se transakcija izvrši.

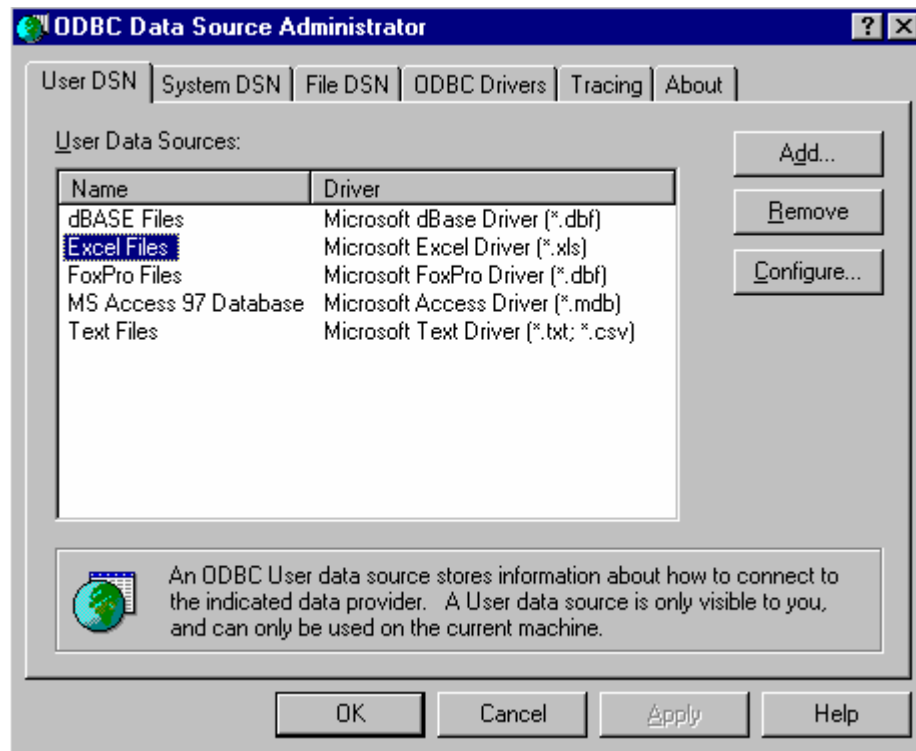
**Select, Next, Insert, Delete , ili Update trigger fields** – popunjen sa tagom koji djeluje kao triger. Svaka promjena ove vrijednosti će prouzrokovati da sistem izvrši komandu. Najmanje jedan od polja trigeru se zahtjeva.

### Tijelo ODBC lista

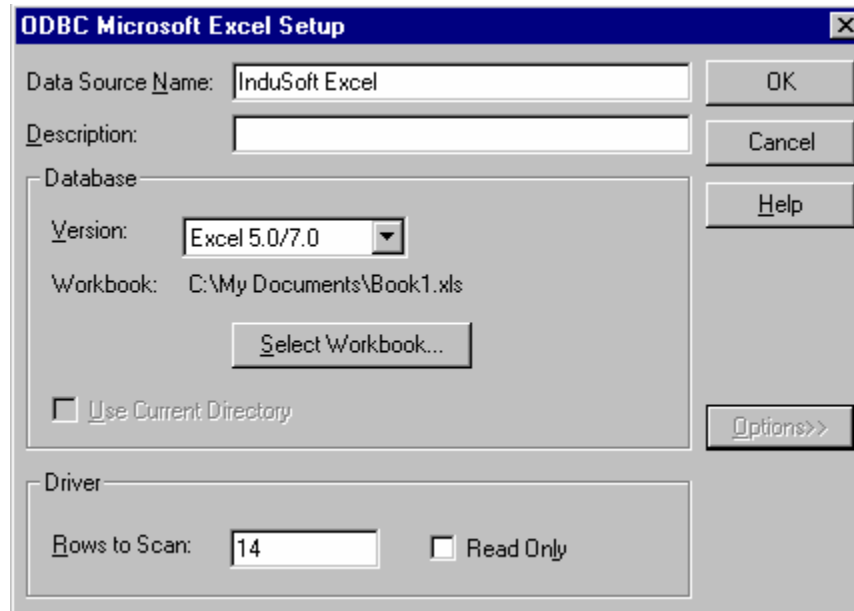
U ODBC radnom listu, uspostavljamo relaciju izmedju polja u tekućem registru iz tabele baze podataka.

### Postavljanje ODBC interfejsa sa Excel fajlovima

U ODBC Data source Administratoru , izabrati Excel files:

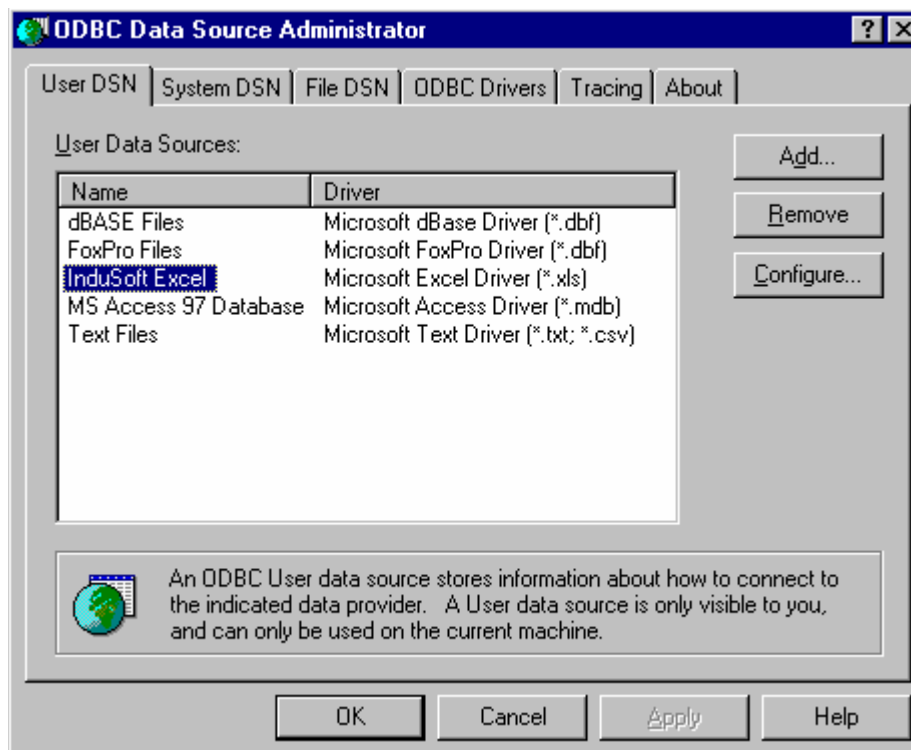


Kliknuti na **Configure** taster. Pojaviće se ODBC Microsoft Excel Setup prozor kao na slijedećoj slici:



U polje **Data Source name** unjeti konfiguraciono ime koje će se koristiti i u ODBC radnom listu u polju **DSN**.

Kliknuti na **Select Workbook** taster da se konfigurira Excel file koji ćemo koristiti. Vratimo se na **ODBC Data Source Administrator** prozor, i tada će na prozoru **User DSN** pojaviti to ime kao na slici :



## ODBC radni listovi u IWS

Nakon što smo konfigurisali **ODBC Windows** interfejs, moramo konfigurisati radne liste **IWS Studio ODBC**-a. Preko **Task** taba , unjeti novi ODBC radni list. Provjeriti da je ODBC Runtime setovan da starta automatski na **RunTime Tasks** tabu u **Project Status**-u . Nakon toga je samo potrebno startati projekat. Aplikacija će manipulirati sa Select, Next, Insert, Delete, i Update triggerima da omogući izmjenu između redova u Excelu i tagova konfigurisanih u radnom listu.

### Kodovi grešaka koje se mogu pojaviti

#### Select komanda

1. Greška u ODBCPREPARE funkciji
2. Greška u ODBCBINDCOL funkciji
3. Greška u ODBCEXECUTE funkciji
4. Greška u ODBCSETCH funkciji

#### Next komanda

5. Greška u ODBCSETCH funkciji

#### Insert komanda

6. Greška u ODBCPREPARE funkciji
7. Greška u ODBCEXECUTE funkciji
8. Greška u ODBCCOMMITE funkciji

#### Update komanda

9. Greška u ODBCPREPARE funkciji
10. Greška u ODBCEXECUTE funkciji
11. Greška u ODBCCOMMITE funkciji

#### Delete komanda

12. Greška u ODBCPREPARE funkciji
13. Greška u ODBCEXECUTE funkciji
14. Greška u ODBCCOMMITE funkciji

## DDE i NetDDE

Dynamic Data Exchange ( DDE) je protokol za dinamičku razmjenu između Windows aplikacija kao Excel i bilo kojeg drugog Windows programa koji podržava ovaj interfejs. DDE konverzacija je interakcija između server i klijent aplikacija. **Studio** obezbjeđuje interfejs da se izvršava bilo kao klijent ili server. Ovo se ostvaruje aktiviranjem bilo **DDE Client Runtime** i **DDE Server** u **Runtime Tasks** pod **Status** opcijom **Project**.

Da bi se izvršavao kao Server, jednostavno treba startati DDE ili NetDDE Server task u **Runtime Tasks**. Da bi se izvršavao kao DDE klijent, treba konfigurirati DDE interface radni list u **Communication** tabu.

**Network Dynamic Data Exchange ( NetDDE )** je proširenje DDE koje radi između računara u mreži. Da bi se izvršavao **IWS** kao server za NetDDE konekciju, treba startati aplikaciju **DDE Server**. Da bi se izvršavao IWS kao klijent za NetDDE konekciju, koristiti isti DDE interfejsni radni list sa odgovarajućom konfiguracijom da bi se aktivirala **Studio** aplikacija.

**Opaska:** Kada se izvršava NetDDE , samo WRITE trigeri se prihvataju. Da bi se iščitavali podaci, treba konfigurirati write komandu na serverskom računaru.

### DDE radni list

Desno kliknuti DDE folder da se insertira novi radni list, koji izgleda kao na slijedećoj slici:

|   | Tag Name | Item |  |
|---|----------|------|--|
| 1 |          |      |  |
| 2 |          |      |  |
| 3 |          |      |  |
| 4 |          |      |  |
| 5 |          |      |  |



## Zaglavlje DDE radnog lista

Zaglavlje DDE klijent radnog lista omogućava nam da definišemo tagove koji će startati očitavanje i upisivanje, a također i tagove koji će primiti konekcijske statuse.

**Description field** - opis radnog lista za dokumentovanje

**Application name Field** – Aplikaciono ime DDE servera.

**Topic Field** – Topic u serverskoj aplikaciji

**Connect Field** - Tag koji kontroliše konekciju IWS DDE klijenta i DDE serverske aplikacije. Kada se ovaj tag setuje na 1, on zahtjeva konekciju na server. Ako konekcija nije moguća ili ona se prekine, IWS će setovati ovaj flag na 0 ponovo. Ako je konekcija OK, ova vrijednost će ostati setovana na 1.

**Read Trigger Field** - Tag koji komanduje iščitavanjem tabele. Kada ovaj tag promjeni vrijednost, generira se poling ka DDE serveru. Ova opcija može biti korištena samo sa lokalnim DDE, ne i sa NetDDE serverima.

**Enable Read when Idle Field**- Kada je tag koji je unesen u ovo polje iznad vrijednosti 0, očitavanje uređaja je omogućeno.

**Read Status Field** – Status komande očitavanja

**Write Trigger Field** – Tag koji omogućava pokretanje komande da se generiše na serveru.

**Enable Write on Tag Change Field** – Kada god je vrijednost taga unesenog u ovo polje veća od nule (0), komunikacioni drajver kontinualno provjerava da li postoji promjena u vrijednosti tagova unesenih u radnom listu. Ako se promjena pojavi, promjenjeni tag se upisuje na uređaj zajedno sa njegovom adresom.

**Write Status Field** – Status od komande za upisivanje

## Tijelo DDE radnog lista

Tijelo radnog lista DDE klijenta dozvoljava nam da konfiguriramo da se svaki tag poveže sa svakim ITEM dijelom na DDE serverskoj adresi.

**Tag Name Field** – Tag Studio baze koji treba da se očita ili upiše na DDE serverskoj aplikaciji.

**Item Field** - ITEM dio DDE adrese na serveru. Potrebno je pogledati u softwaresku dokumentaciju servera da bi se dobila informacija o korektnoj sintaksi za APP, TOPIC, i ITEM. Mi možemo konfigurirati Topic i Item polja sa

tagovima na adresi koristeći sintaksu : **text { tag}**. Vrijednost od **{ tag}** se evaluira u string i koristi za adresu. Primjer:

Topic : **topic\_{ tag\_topic\_name}**

Item : **{ tag\_item\_name}** ili **A{ tag\_number}**

### Konfigurisanje NetDDE klijenta za NetDDE Servera

Konfigurisanje **NetDDE** konekcije je kao i konfigurisanje DDE konekcije, izuzev što se razlikuje za Header Application name i topic. Prije nego što startamo da testiramo, moramo verifikovati da DDE server je omogućen na stanici sa kojom želimo da razmjenjujemo podatke.

**Opaska:** Kada se povezujemo sa serverima koji nisu IWS , potrebno je konsultovati serversku dokumentaciju da bi saznali o korektnoj sintaksi za **APP**, **TOPIC** i **ITEM**.

### Zaglavlje DDE radnog lista za NetDDE Server

**Application Name Field:** < ime računara>\NDDE\$,

Gdje < ime računara> mora biti validno ime računara u mreži

**Topic Field :** ako se spajamo sa drugom IWS stanicom, treba koristiti topic ime : UNISOFT\$

### Tijelo DDE radnog lista ka NetDDE Serveru

**Tag Name Field** – IWS lokalno ime taga, koji se linkuje sa nekim udaljenim tagom putem DDE komunikacije

**Item Field** - Ime taga na udaljenom računaru koje je u relaciji sa lokalnim tagom.

## Komunikacije

### PLC Drajveri

Drajver je dio IWS softwarea. Njegova funkcija je da uspostavi komunikaciju između procesnog uređaja i IWS softwarea.

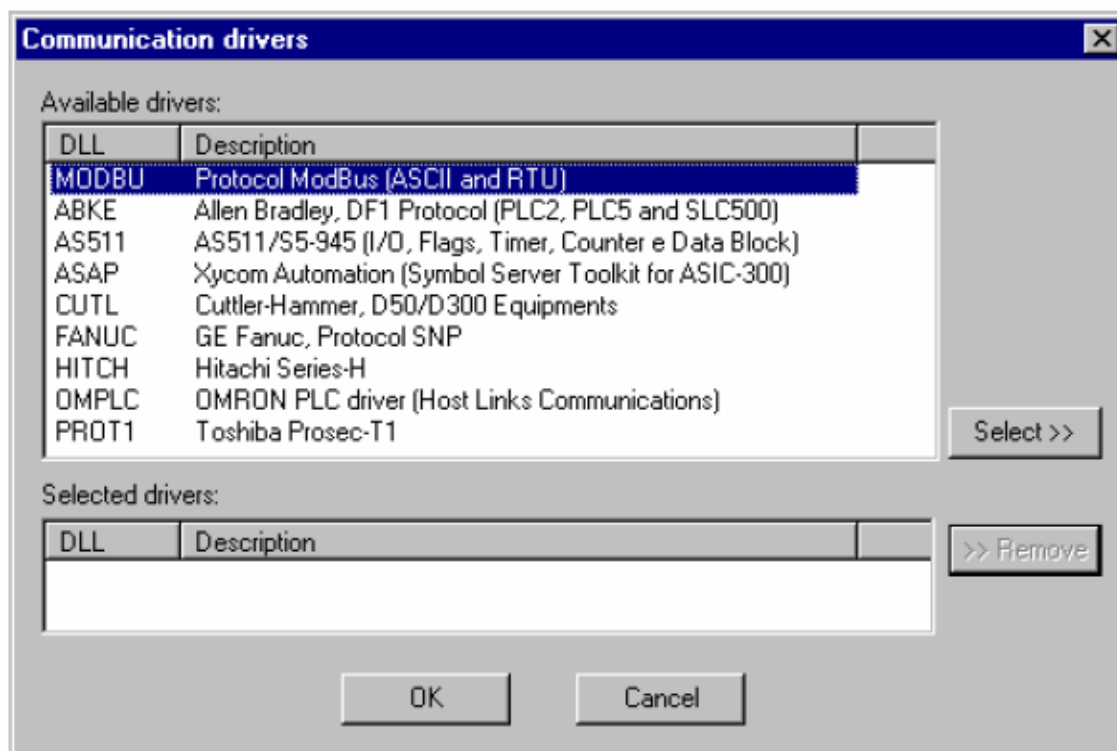
Komunikacija u ovom slučaju znači čitanje vrijednosti iz memorije procesnih uređaja u aplikacione varijable koje se nazivaju "Tagovi", ili upisivanje vrijednosti iz aplikacionih tagova u memoriju procesnih uređaja.

Slijedi primjeri nekih drajvera koji su raspoloživi za WINNT/2K/XP i WinCE Operativne sisteme :

- Allen Bradley - DF1
- Siemens S5 – AS511 PG Port
- Profibus DP Master and Slave
- Allen Bradley – ControlNet Slave
- OMRON – Host Link
- GE FANUC – SNP, 90-30 i 90-70 seriju
- Modbus – Schneider 984 Series
- Profibus DP Master
- Hitachi – H Series
- Toshiba – Prosec T1/T2

### Selekcija drajvera

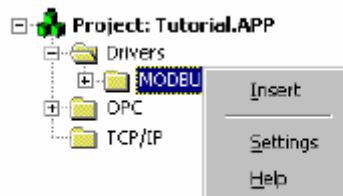
Kada se IWS instalira, sa njim su instalirani i svi drajveri. Da bi se startalo konfigurisanje bilo kojeg drajvera, treba desno kliknuti na folder **Drivers** a onda na **Add/Remove**. U dijalog boksu koji se otvori treba izabrati željeni drajver, ili njegov opis i kliknuti na **Select**. U našem primjeru izabraćemo **ModBus** drajver.



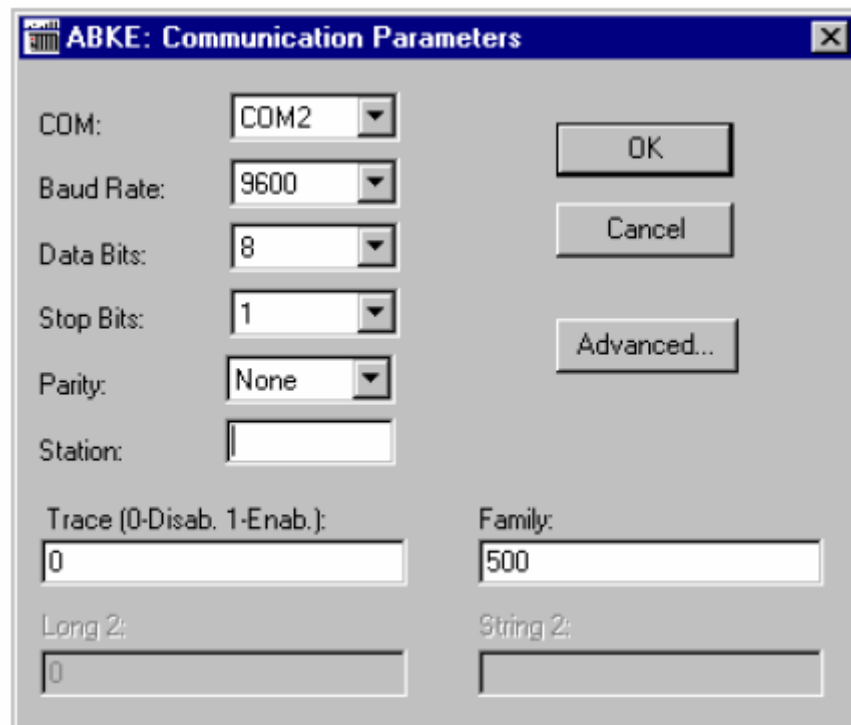
## Konfigurisanje komunikacionih parametara

Ovaj modul konfigurira serijski kanal i/ili specifične parametre drajvera. Veličine koje se unesu ili promjene koje se izvrše, će se preuzeti samo nakon inicijalizacije drajvera. Zbog toga, ako se drajver izvršava a korisnik unese promjene u setinzima, drajver mora biti prvo zatvoren a zatim ponovno otvoren da bi promjene bili uzete u obzir.

Da bi specificirali komunikacione parametre desno kliknimo na MODBU podfolder u Driver folderu, na opciju **Settings**:



## Komunikacioni parametri



| Parameter | Default Value | Valid values                   | Description  |
|-----------|---------------|--------------------------------|--|
| COM       | COM1          | COM1 to COM8                   | Serial port of the PC used to communication with the device, if it is a serial driver. |
| Baud Rate | 19200         | 110 to 57600bps                | Communication data's rate  |
| Data Bits | 8             | 5 to 8                         | Number of Data bits used in the protocol   |
| Stop Bits | 1             | 1 or 2                         | Number of stop bits used in the protocol   |
| Parity    | Odd           | even, odd, none, space or mark | Protocol's Parity  |
| Station   | 0             | 0                              | Number or Name of the Computer, or Unit in a Network, if the protocol needs it.        |

**Opaska:** Uredjaj mora biti konfigurisan sa istim vrijednostima koje su definirane u prozoru **Communication Parameters**.

### Long1, Long2, String1 i String2 polja

Ova polja se konfigurišu sa različitim funkcijama za svaki drajver. Na primjer, za Drajver ModBus, polje **String1** se zove **Protocol**, i ovdje treba unjeti ili **ASCII** ili **RTU**.

Za druge tipove drajvera , ova polja su različita.

Ako korisnik unese nevalidne unose u ova polja, oni će biti prihvaćeni, ali kada pokušamo da zatvorimo ovaj prozor, pojaviće se poruka o grešci i neće dozvoliti zatvaranje prozora.

### Napredna setovanja ( advanced settings )

Klikanjem na taster **Advanced..** u prozorima **Communication parameters** , biće moguće konfigurisati druge parametre serijske komunikacije:

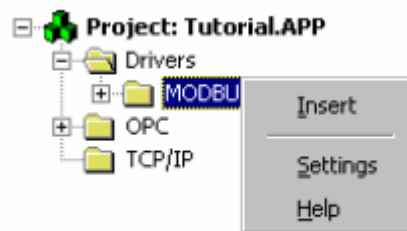
| Parametar             | Default vrijednost | Validne vrijednosti  | Opis  |
|-----------------------|--------------------|----------------------|---|
| Start message (ms)    | 100                | 0 - 10000            | Maksimalno vrijeme da se primi početak odgovora sa uređaja ( time-out vrijeme)  |
| End message (ms)      | 0                  | 0 do 10000           | Maksimalno vrijeme da se primi kraj odgovora sa uređaja od početka odgovora. ( Vrijednost 0 znači da drajver neće provjeravati ovo vrijeme) |
| Interval between char | 500                | 0 do 10000           | Maksimalno vrijeme između karaktera poslanih sa uređaja   |
| Wait CTS (ms)         | 100                | 0 do 10000           | Maksimalno vrijeme da se primi CTS signal poslije setovanja RTS signala.  |
| Control RTS           | No                 | No, yes ili yes+echo | Definiše da li handshake signal RTS mora biti setovan prije komunikacije i da li će imati eho u komunikaciji                                |
| Verify CTS            | No                 | No ili yes           | Definiše da li drajver mora čekati za handshake signal CTS ,  |

|                     |             |                         |  |
|---------------------|-------------|-------------------------|--|
|                     |             |                         | prije slanja poruke  |
| Disable DTR         | Not checked | Not checked ili checked | Ako je chekiran, drajver neće setovati DTR signal prije starta komunikacije.                             |
| Retries             | 0           | 0 do 5                  | Broj ponovnih pokušaja komunikacije za svaki konfigurisani tag u drajverskom listu, u slučaju neuspjeha. |
| Tx Buffers ( bytes) | 512         | 0 do 512                | Maksimalna veličina informacionog bafera koja će se poslati sa drajvera                                  |
| Rx Buffer ( bytes)  | 512         | 0 do 512                | Maksimalna veličina informacionog bafera koja će se primiti sa hosta.                                    |

**Opaska:** Setinzi komunikacionih parametara su isti za sve konfiguracione listove drajvera.

### Dodavanje novog lista drajvera.

Da bi se dodao novi list drajvera , desno kliknuti na folder sa imenom drajvera, i onda kliknuti na **Insert**.



U našem primjeru ćemo dodati novi list drajvera za Modbus drajver ( Modbu). List kao i prethodni se sastoji iz dva djela, zaglavlja ( header) i tijela ( body). U zaglavlju unosimo informacije za komande očitavanja i upisivanja a u tijelu su operatorske adrese:

### Zaglavlje

Description:  
  Increase read priority

Read Trigger:  Enable Read when Idle:  Read Completed:  Read Status:

Write Trigger:  Enable Write on Tag Change:  Write Completed:  Write Status:

Station:  Header:   Min:   
 Max:

|  | Tag Name | Address | Div | Add |
|--|----------|---------|-----|-----|
|  | TAG_0001 | 1       |     |     |

Zaglavlje sadrži potrebne informacije za funkcije drajvera. Kao inicijalna informacija, za svaku oblast sa kojom želimo komunicirati, moramo kreirati novi drajverski konfiguracioni list.

## Description

U ovo polje unosimo podatke o tipovima oblasti, opsegu, i da li iščitavamo ili i upisujemo.

## Increase read priority

Ako je ovo list iščitavanja, i ima ih više koji su sa istim trigerom za očitavanje, i dogodi se trenutak za čitanje, tada list sa ovim povećanim prioritetom će biti prvi kod slijedećeg čitanja kada ih pozove triger za očitavanje.

## Read trigger

Ovo polje može sadržavati tag koji će generisati događaj očitavanja, svaki put kada promjeni svoju vrijednost.

## Enable Read When Idle

Ovo polje može sadržavati tag ili vrijednost koja će omogućiti kontinualno čitanje, kada je vrijednost u tom polju veća od nule.

## Read completed

Ovo polje može sadržavati tag čija će se vrijednost toglovati ( naizmjenično mjenjati ), kada se događaj očitavanja završio.



### Read status

Ovo polje može sadržavati tag čija će se vrijednost popuniti sa cjelobrojnomo vrijednošću, kada god se događaj očitavanja završio. Ako je vrijednost jednaka nuli, događaj je bio u redu . U suprotnom, događaj je završen sa greškom. Poruke grešaka mogu se vidjeti u **Logwin** modulu , ili da se pregledaju u MODBU.MSG fajlu, u IWS **DRV** direktoriju.

### Write Trigger

Ovo polje može sadržavati tag koji će generisati događaj upisivanja cijelog spreadsheetsa, svaki put kada se njegova vrijednost promjeni. Ovdje je vrlo važno pomenuti slijedeće:

Kada koristimo ovu karakteristiku, drajver će upisivati vrijednosti TAGs u memoriju PLC. Ova operacija upisivanja koristi blokove, od prvog operatorskog lista do posljednjeg. Ako postoji operator koji je bio deklarisan u takvom listu, i njegova adresa je između prve i posljednje, on će primiti vrijednost nula ( 0 ).

Zbog toga, korisnik mora biti siguran šta želi da upisuje kada koristi ovaj triger, i provjeri da li postoji neka rupa u spreadsheetu, koja može dovesti do problema u sistemu ili u PLC programu.

### Enable Write on Tag Change

Ovo polje može sadržavati tag koji, kada je njegova vrijednost veća od nule, omogućava upisivanje samo tagova iz spreadsheeta koji su imali promjenu, što je različito nego kod triggera za upisivanje ( write trigger).

### Write Complete

Ovo polje može sadržavati tag čija će se vrijednost toglovati kada je događaj upisivanja završen.

### Write Status:

Ovo polje može sadržavati tag čija će se vrijednost popuniti sa cjelobrojnomo vrijednošću, svaki put kada se događaj očitavanja završio. Ako je ova vrijednost jednaka nuli, događaj je prošao u redu. U suprotnom, događaj se završio sa greškom. Poruke grešaka se mogu vidjeti u **Logwin** modulu, ili iščitane u OMPLC.MSG fajlu, u **DRV** direktoriju IWS-a.

### Station

Ovo polje mora, ako je tako indicirano u help fajlu za drajver, sadržavati ID za CPU, ili broj uredjaja, ili PLC adresu, koji se odnosi na ovaj specifični list. Svaki drajver ima različitu sintaksu za ovo polje.

Naprimjer, GE Fanuc SNP drajver omogućava da se identificira PLC sa ASCII karakterima, dok OMRON-ov Host Link protocol, dozvoljava kao adrese samo numeričke vrijednosti 1 do 31.

U najvećem broju slučajeva, ovo je adresa u mreži PLC uređaja.

Može biti također popunjena sa tagom, oko kojeg su vitičaste ( curly) zagrade / Naprimjer : {tag} ).

### Header field

Ovo je jedno od najvažnijih polja u zaglavlju lista. Svaki drajver ima različiti sintaksu za ovo polje. Obično, ovdje treba unjeti nešto što liči na tip operatora i nakon toga početnu adresu.

Neki od primjera za ovo su:

| Drajver            | Header | Značenje   |
|--------------------|--------|--|
| MODBUS             | 4X:100 | 4X označava da će ovaj list komunicirati sa Holding registrima, od adrese 100 nagore. U slučaju PLC AEG tip 984, to će biti naprimjer od adrese 400100 nagore. |
| OMPLC ( Host link) | IR:0   | IR označava da će list komunicirati sa I/O internim relejima, od adrese 0 na gore.   |
| FANUC ( SNP)       | %M     | %M označava da će ovaj list komunicirati sa %M diskretnim internim operatorom. Nema početne adrese za ovaj drajver.  |
| ABKE (DF1)         | N7:0   | N7 indicira da će list komunicirati sa N7 fajlom, od adrese 0 nagore. U slučaju PLC-5-40 ( Allen Bradley ), od adrese N7:0.                                    |
| AS511 ( Siemens)   | DB5:10 | DB5 označava da ovaj list će komunicirati sa Data blokom broj 5. od riječi broj 10.  |

U našem primjeru u Tutorialu, korist ćemo sintaksu MODBUS-a.

**Sintaksa je : < reference>: < initial address> .**

Naprimjer, ako je header **4X:1**, list će čitati od adrese 4000001 do najvećeg ofseta koji je konfigurisan u koloni adrese.

Korektne reference su:

**0X: status namotaja ( očitavanje digitalnog izlaza)**

**1X: status digitalnog ulaza ( read only)**

**3X: očitavanje ulaznog registra ( read only ) – analognog ulaza**

**4X : holding register ( analogni izlaz )**

ID: ( izvještaj sa slave-a ) – read only.

Gdje u gornjem pregledu je označeno read only , funkcija upisivanja neće raditi ( tj. ne može biti izlaz ).

Ovo polje može također biti popunjeno tagom u vitičastim zagradama ( naprimjer {tag}).

### Min/Max

Ova polja su omogućena samo onda ako je omogućen check boks sa lijeve strane ovih polja. Ako se izabere, tada omogućava opseg vrijednosti koje onda omogućavaju konverziju u inženjerske jedinice.

Ova polja određuju minimalnu i maksimalnu vrijednost opsega. Naprimjer: vrijednosti u memoriji su 0 do 4095 i znače 0% do 100%.

### Tijelo komunikacionog lista

Ovaj list ima četiri kolone: **Tag name, Address, Add i Div.**

#### Tag Name

Ime taga kojeg će koristiti komunikacioni drajver.

#### Address

Adresa da se čita ili upisuje vrijednost taga u uređaj. Kao i kod polja u Headeru, ovo polje je različito za različite drajvere. U slučaju našeg Modbus drajvera, mi ovdje unosimo offset od inicijalne adrese koja je konfigurisana u header polju. Nije dozvoljeno unositi negativnu vrijednost ovog offseta. Vrijednost 0 će prepisati takvu vrijednost ako se pokuša unjeti.

#### Div/Add/Max/Min

| Kolona | Opseg vrijednosti                           | Značenje  |
|--------|---|---|
| Div    | Bilo koja cjelobrojna ili realna vrijednost | Kod read komande: $\text{Tag} = (\text{Host value}) / \text{DIV}$<br>Kod write komande: $\text{Host value} = \text{Tag} * \text{DIV}$                         |
| Add    | Bilo koja cjelobrojna ili realna vrijednost | Kod read komande: $\text{Tag} = (\text{Host value}) + \text{ADD}$<br>Kod write komande: $\text{Host value} = \text{Tag} - \text{DIV}$                         |
| Min    | Bilo koja cjelobrojna ili realna vrijednost | Definira minimalnu vrijednost doznačenu tagu, kada je odgovarajuća vrijednost hosta jednaka vrijednosti definiranoj u polju Min u zaglavlju drajverskog lista |
| Max    | Bilo koja                                   | Definira maksimalnu vrijednost doznačenu  |

|  |                                   |   |
|--|-----------------------------------|---|
|  | cjelobrojna ili realna vrijednost | tagu, kada je odgovarajuća vrijednost hosta jednaka vrijednosti definiranoj u polju Max u zaglavlju drajverskog lista |
|--|-----------------------------------|---|

## Priprema aplikacije za primjer runtime drajvera

### Tagovi zaglavlja

Slijedeći tagovi će se unjeti u polja zaglavlja konfiguracionog lista drajvera. Svi su tipa polja ( array) , i svaki element bit će unesen na svakom listu. Naprimjer, RdTr[1] u polju "Read Trigger" , na listu "ABKE001.DRV", a RdTr[5] na listu "ABKE005.DRV", itd.

| Ime Taga         | Veličina | Tip     | Komentar   |
|------------------|----------|---------|--|
| RdTr             | 0        | Bool    | Bool—ov tag koji će biti u polju "Read Trigger"              |
| RdEn             | 0        | Bool    | Bool—ov tag koji će biti u polju "Enable read when Idle"     |
| RdClp            | 0        | Bool    | Bool—ov tag koji će biti u polju "Read Complete"             |
| RdSt             | 0        | Integer | Integer tag koji će biti u polju "Read Status"               |
| WrTr             | 0        | Bool    | Bool—ov tag koji će biti u polju "Write Trigger"             |
| WrEn             | 0        | Bool    | Bool—ov tag koji će biti u polju "Enable Write when Idle"    |
| WrCpl            | 0        | Bool    | Bool—ov tag koji će biti u polju "Write Complete"            |
| WrSt             | 0        | Integer | Integer tag koji će biti u polju "Write Status"              |
| Station "Header" | 0        | String  | String tag koji će biti u polju "on in the test's beginning" |
| Header "Station" | 0        | String  | String tag koji će biti u polju "on in the test's beginning" |

Komunikacioni tagovi bit će TAG\_DRV , dimenzije 10. Ekran će izgledati kao na slijedećoj slici:

**header = #####** **PLC**

|             |               |                     |
|-------------|---------------|---------------------|
| <b>RDTR</b> | rdtr = #####  | Tag_DRV[1] = #####  |
|             | rden = #####  | Tag_DRV[2] = #####  |
| <b>RDEN</b> | rdst = #####  | Tag_DRV[3] = #####  |
|             |               | Tag_DRV[4] = #####  |
|             |               | Tag_DRV[5] = #####  |
| <b>WRTR</b> | wtrtg = ##### | Tag_DRV[6] = #####  |
|             | wren = #####  | Tag_DRV[7] = #####  |
| <b>WREN</b> | wrst = #####  | Tag_DRV[8] = #####  |
|             |               | Tag_DRV[9] = #####  |
|             |               | Tag_DRV[10] = ##### |

a drajverski list:

Description:  
  Increase read priority

Read Trigger:     Enable Read when Idle:     Read Completed:     Read Status:

Write Trigger:     Enable Write on Tag Change:     Write Completed:     Write Status:

Station:     Header:      Min:     Max:

|    | Tag Name    | Address | Div | Add |
|----|-------------|---------|-----|-----|
| 1  | Tag_DRV[1]  | 1       |     |     |
| 2  | Tag_DRV[2]  | 2       |     |     |
| 3  | Tag_DRV[3]  | 3       |     |     |
| 4  | Tag_DRV[4]  | 4       |     |     |
| 5  | Tag_DRV[5]  | 5       |     |     |
| 6  | Tag_DRV[6]  | 6       |     |     |
| 7  | Tag_DRV[7]  | 7       |     |     |
| 8  | Tag_DRV[8]  | 8       |     |     |
| 9  | Tag_DRV[9]  | 9       |     |     |
| 10 | Tag_DRV[10] | 10      |     |     |
| 11 |             |         |     |     |

## TCP/IP

IWS moduli TCP/IP Klijent-Server omogućavaju da dvije ili više IWS aplikacija održavaju njihove baze podataka sinhronizovanim.

Ovi moduli koriste TCP/IP protokol da ostvare komunikaciju između aplikacija. Prije korištenja IWS TCP/IP Klijent-Server modula, moramo instalirati i konfigurirati TCP/IP protokol na računarima na kojima će se izvršavati ovi moduli.

## Konfiguracija Servera

Na serverskom računaru nije potrebno ništa posebno konfigurirati. Potrebno je samo izvršavati modul TCP/IP Server. Da bi ovo ostvarili, treba u **Project Settings** prozoru setovati da TCP/IP Server se izvršava automatski.

Dok se program izvršava, u sistemskom dijelu police u donjem desnom uglu ekrana ( system tray) pojaviće se mala ikona **TCP/IP Server**.

## Konfiguracija klijenta

Na klijentskom računaru, potrebno je koristiti program **TCP/IP Klijent** konfiguracioni program da se konfigurira **Serverska IP adresa i tagovi koje želimo da dijelimo sa serverom**.

## Kako da provedemo TCP/IP konfiguraciju klijenta

Na IWS-u izabrati COMM tabelu, i izabrati novi TCP radni list, desno kliknuvši na TCP folder.

## Opis polja

**Description:** ovo polje se koristi samo za dokumentovanje.

**Connection status** : Ovo polje treba sadržavati **ime nekog taga**. TCP/IP modul konfigurisanja klijenta će ažurirati ovaj tag u skladu sa stanjem konekcije. Ako je **vrijednost taga 0 ( nula )**, tada **je konekcija OK**. U suprotnom, vratiće se kod greške od strane biblioteke za Windows sockete.

**Server IP address:** ovo polje treba da sadrži IP adresu servera. To može biti i string ( ime noda) , ili možemo koristiti tag uključen u vitičastim zagradama, ili standardna 4 bajtna IP adresa. Naprimjer, ako popunimo polje sa {tag\_name} , TCP/IP klijent modul će pokušati da se spoji na server koji ima ime označeno sa tag\_name.

**Tag name:** ova polja trebaju sadržavati tagove koje želimo da djelimo sa serverom. Ako je tag varijabla polja ili klasa ( ili obadvoje), svaki element i član klase će se dijeliti. Korisnik treba samo da stavi ime taga u ovo polje, bez da specificira indeks polja ili člana klase. Ako to i uradi, TCP/IP klijentski modul će to ignorisati.

**Remote tag:** ova polja trebaju sadržavati ime taga koji će biti linkovan sa tagom specificiranim u polju Tag name. Ovo polje je opciono. Ako ga ostavimo nepopunjenim, isto ime taga će biti korišteno i kod klijenta i servera.

Opaska: Ako djelimo polje, tada tag u serveru treba da ima isti broj elemenata kao i kod klijenta. Ako je tag tipa klase, tada i definicija klase treba biti ista i kod servera i klijenta.

### **Izvršavanje TCP/IP klijent modula.**

Treba izabrati da se ovaj modul izvršava u meniju Project>>Status, Runtime Tasks Table, TCP/IP klijent.

Nakon što se ovaj program starta pojaviće se mala ikona u system trayu.

## **OPC ( OLE za upravljanje procesima)**

### *Uvod*

IWS OPC klijent modul omogućava IWS sistemu da komunicira sa biko kojim uređajem na kojem je implementiran OPC Server. Ovaj modul implementira OPC standard kao što je opisano u dokumentu "OLE for process Control Data Access standard", objavljen od strane OPC Foundation.

### **Pripremanje OPC Server baze podataka**

Prije nego što počnemo koristiti OPC klijent konfigurator, moramo instalirati neki OPC server.

Uzmimo kao primjer da koristimo Software Toolbox OPC Power Server.

Nakon njegovog instaliranja i aktiviranja, treba da konfiguriramo OPC Server bazu podataka.

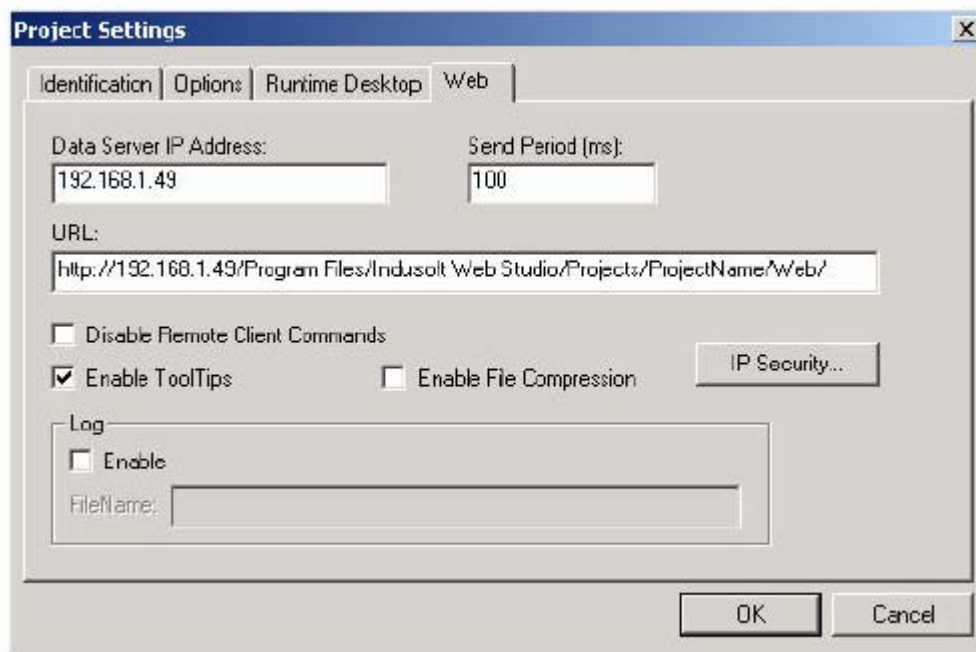
## WEB

IWS dozvoljava da pohranimo ekrane u HTML formatu i izvezemo ih u Internet browserima kao što je naprimjer Internet Explorer ( IE ).

Kada je aplikacija razvijena , treba da setujemo parametre u Web tabu u Program Settings dijalog prozoru i pohranimo svaki ekran kao HTML file ( koristeći opciju **File – Save As HTML** )

**Opaska:** svi tagovi koji se koriste za prikazivanje podataka ili unošenje komandi treba da imaju svoje setinge promjenjene u server , da bi mogli da komuniciraju preko Web-a.

Računar na kojem su pohranjeni HTML fajlovi ( Page Server) mora biti Web Server ( tj. HTTP Server drajver), a računar na kojem se aplikacija izvršava treba imati fiksnu IP adresu. Primjetimo da Page Server i Web Server može, ali ne mora da bude, isti računar. U našem tutorijalu mi ćemo pretpostaviti da su Page i WEB Serveri na istom računaru.



Da bi mogli da gledamo web stranice, moramo prvo konfigurirati Web setinge. Njih ćemo naći u **Project Settings** prozoru pod tabom **Web**. Prvo treba da unesemo **Data Server IP Address** , tj. adresu mašine na kojoj se izvršava aplikacija. Nakon toga treba da unesemo **URL** u slijedećem formatu:

**<http://< IP adresa računara na kojem je WEB server>/< path od servera do direktorija gdje je web stranica instalirana >/>**



Kada se ova dva polja unesu, treba kliknuti na OK taster. Nakon toga otići na Tools Menu i izabrati **Verify Application**.

**Opaska:** Pošto Web pages prikazuju informaciju pomoću **WEB Servera, RunTime System**, i **TCP/IP Server** na mašini na kojoj se izvršava aplikacija moraju se izvršavati da bi se mogle gledati Web stranice na nekom od računara u mreži koristeći IE.

Kada su setinzi korektni, treba startati WEB Server a zatim aplikaciju na računaru na kojem se izvršava.

Sa ovo dvoje koje se izvršava, sada korisnik može da koristi IE da se spoji sa aplikacijom time što će izabrati **URL** kao

**http://< IP adresa računara na kojem je WEB Server/ < path od servera d direktorija Web stranica>/< ime aplikacionog ekrana>.html.**

Pojaviće se prompt za ime korisnika i njegovu lozinku, koje treba unjeti iz liste konfigurisanih korisnika.

Kada se pojavi ekran, korisnik može interaktivirati sa njim i sa aplikacijom u potpunosti kao i sa terminala računara na kojem se izvršava aplikacija.