Importera larm från WinCC till Nimbus

WinCC kan skicka larm till Nimbus på två sätt:

1) Med en textfil som skapas av ett globalt *C-script*
2) Med en separat applikation (*WinCC2Nimbus*) som använder *ODK* och kopplar sig till *WinCC messaging service*

Vilken metod man väljer beror på applikationen. Det enklaste att konfigurera och felsöka är metod 1 via en textfil, men man måste i så fall in på varje larm som ska sändas vidare och kryssa i en ruta, vilket är viktigt att komma ihåg om man lägger till larm i framtiden.

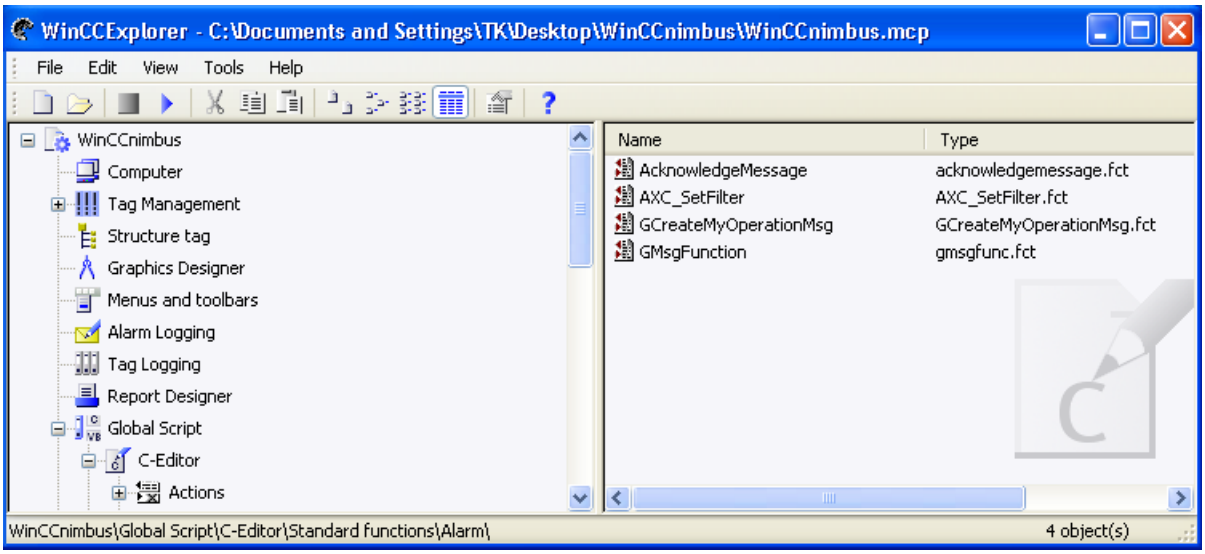
Båda metoderna beskrivs i detta dokument.

Metod 1: Konfigurera WinCC för att exportera en textfil vid larm

Leta reda på filen *C:\Program Files (x86)\Siemens\WinCC\aplib\Alarm\gmsgfunc.fct*. Byt namn på filen till *gmsgfunc.org*. Vilken mapp filen ligger i kan variera beroende på WinCC version och operativsystem.

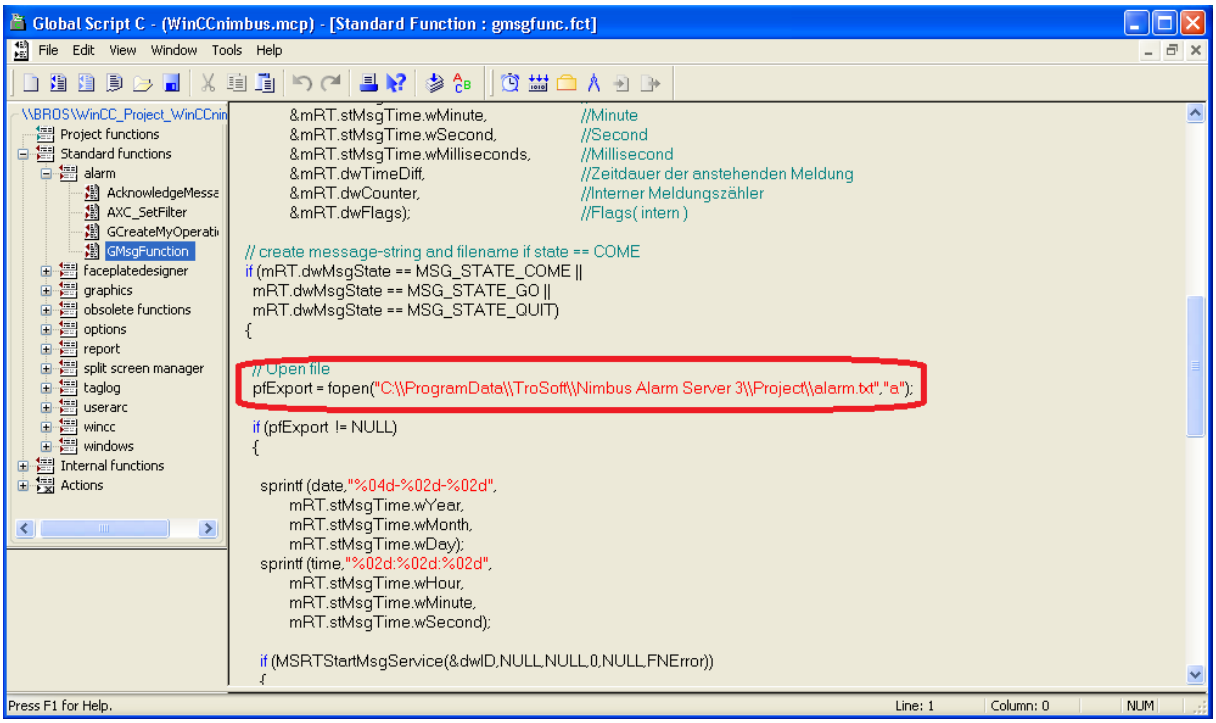
Kopiera istället in filen *gmsgfunc.fct* från levererad media, den finns i mappen *..\Tools and documents\WinCC*. Filen finns också på TroSoft hemsida att ladda hem. Den finns också i .txt-format om man själv vill klippa och klistra in programkoden, eftersom *fct*-filer inte kan öppnas med ex *Notepad*. Scriptet finns också med längre ner i detta dokument.

Starta *WinCC Explorer*.



Gå in på *Global Script -> C-Editor -> Standard functions -> Alarm*

Öppna *GMsgFunction*



Redigera raden med sökvägen om det behövs. Den pekar på Nimbus version 3 projektmapp. Mappen skapas när *Nimbus* installeras. Man kan välja i princip vilken mapp man vill så länge både *Nimbus* och *WinCC* har rätt att skriva och radera filer i mappen.

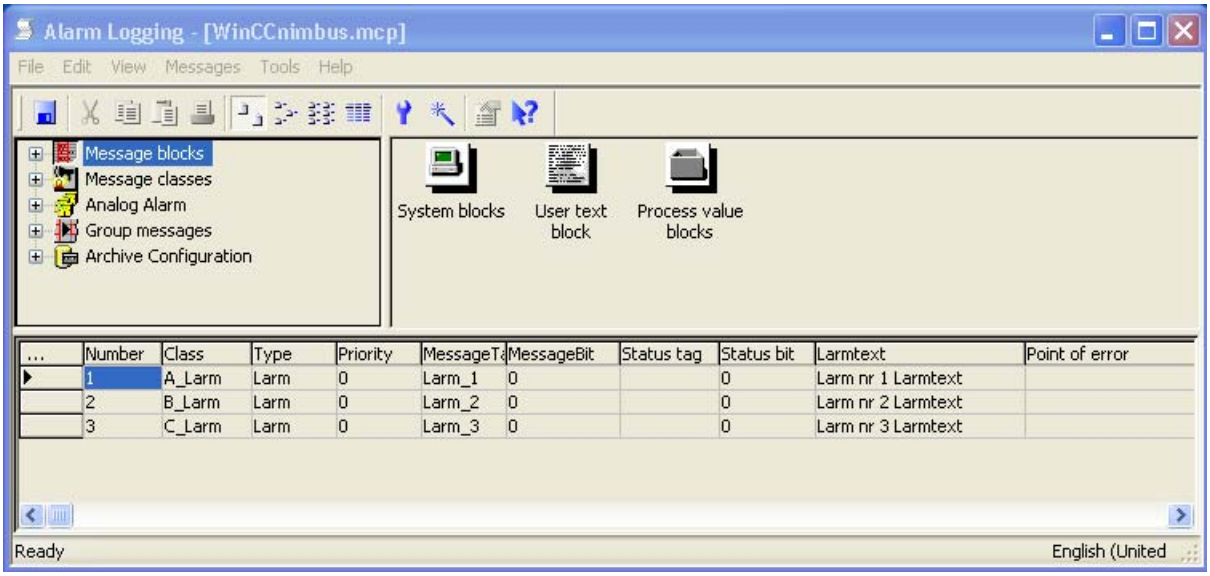
Om orginalet *gmsgfunc.fct* används till något annat i larmhanteringsväg så måste man såklart baka ihop befintliga funktioner med det nya scriptet.

Välj *File -> Close*. Kompilera om ifall *WinCC* föreslår det.

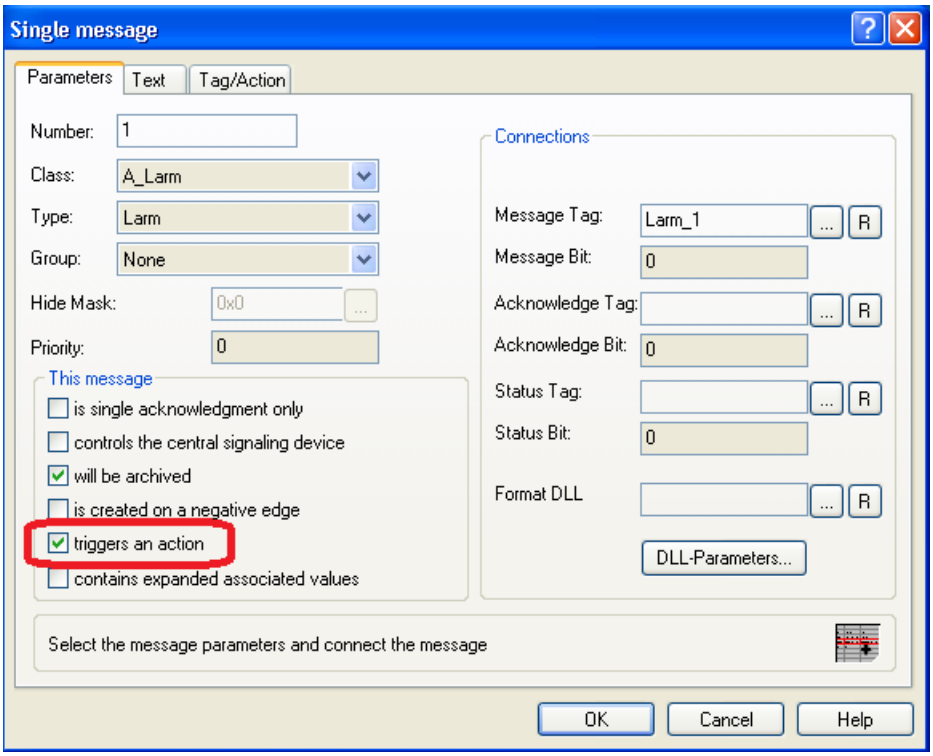
Välj *Tools -> Regenerate Header*.

Stäng *Script editorn*.

Öppna *Alarm Logging*



På alla de larm som ska hanteras och skickas till textfilen och därmed Nimbus måste man in och ändra en parameter. Högerklicka och välj *Properties*.



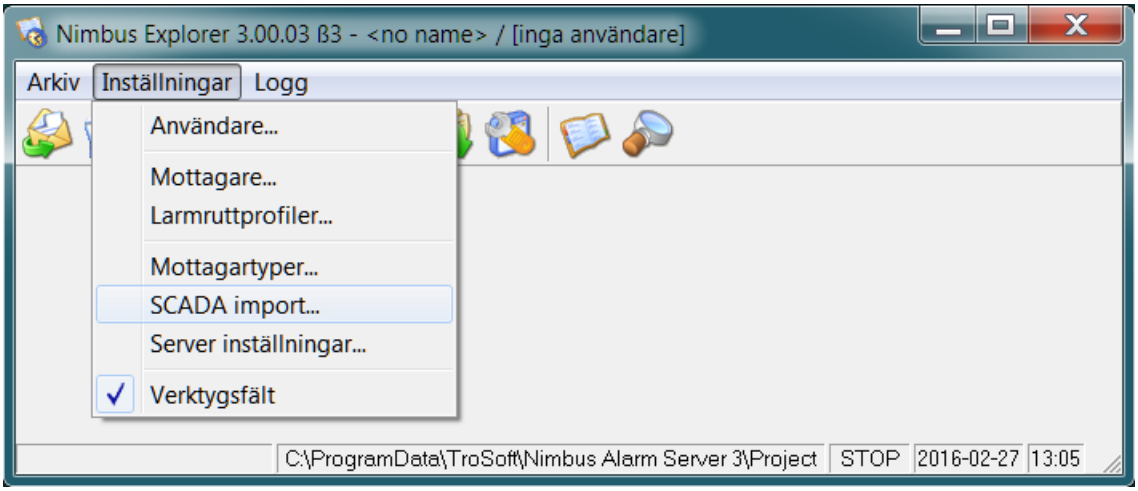
Kryssa i *Triggers an action*. Därmed kommer *WinCC* aktivera scriptet för denna larmpunkt.

Observera att om ni använder PCS7 och CFC så är det där som man ska kryssa i eftersom denna inställning kopieras därifrån över vid kompilering/download.

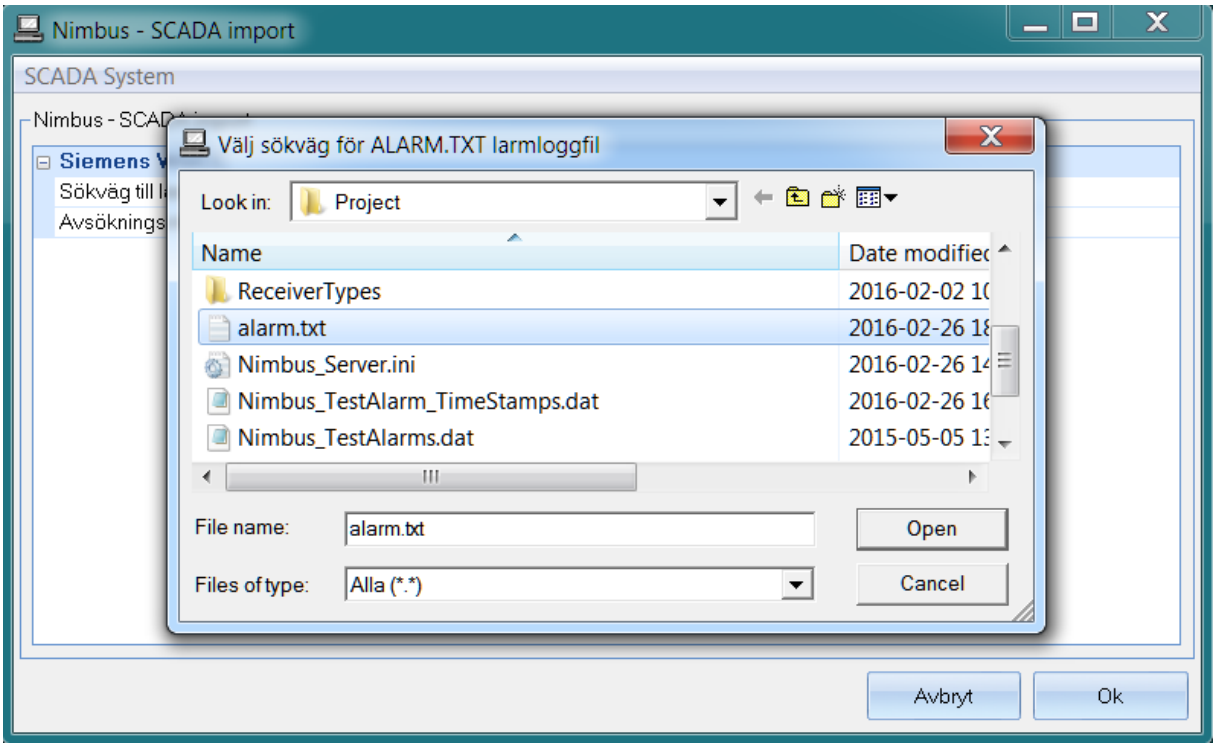
Starta *WinCC runtime* och testa ett larm. Textfilen ska då skapas.

Metod 1: Konfigurera Nimbus för att importera textfilen

Starta *Nimbus Explorer* (högerklicka och välj *Kör som Administratör*) via genvägen. *Nimbus Explorer* ska alltid startas som Administratör.



Välj *Inställningar -> SCADA import.*



Välj *SCADA System -> Lägg till SCADA Import -> Siemens WinCC*

Välj *alarm.txt* i den mapp som angavs i scriptet. Om filen inte finns ännu räcker det med att klicka *Open*.

När du startar *Nimbus Alarm Server* kommer programmet ta bort textfilen eftersom den innehåller gamla larmhändelser. Skapa ett nytt larm, det ska nu komma in i *Nimbus Explorer*.

Metod 1: gmsgfunc.fct

```
//
// Date     / Vers  / Sign / Comment
// ------------------------------------------------------------------------------------
// 16.02.26 / 1.0.0.0 /  TR  / Major changes
// 16.02.27 / 1.0.0.1 /  TR  / Added priority and Class moved to  Area
// 16.03.18 / 1.0.0.2 /  TR  / Added Text block 2
//
//
//
//
//
// ------------------------------------------------------------------------------------

#include "msrtapi.h";

BOOL GMsgFunction( char* pszMsgData)
{

  extern char g_Msg[];

  MSG_RTDATA_STRUCT mRT;
  PCMN_ERROR pError;
  DWORD dwID = 0;
  CMN_ERROR Error;
  MSG_CSDATA_STRUCT MsgData;
  MSG_CLASS_STRUCT MsgClass;
  MSG_TEXT_STRUCT mtsClass;

  MSG_TEXT_STRUCT mtsBlock1;
  MSG_TEXT_STRUCT mtsBlock2;
  MSG_TEXT_STRUCT mtsBlock3;
```

```
char lpszMsgState[256];
char date[20];
char time[20];
char errcode[20] = "\r\n";
long lPriority = 0;

FILE *pfExport;
pError=&Error;

memset (&mRT, 0, sizeof(MSG_RTDATA_STRUCT));
memset (&MsgData, 0, sizeof(MSG_CSDATA_STRUCT));
memset (&MsgClass, 0, sizeof(MSG_CLASS_STRUCT));

if (pszMsgData != NULL )
{
  //Read messagedata
  sscanf( pszMsgData, "%ld,%ld,%04d.%02d.%02d,%02d:%02d:%02d:%03d,%ld, %ld, %ld",
    &mRT.dwMsgNr,                   //Messagenr
    &mRT.dwMsgState,                //Status MSG_STATE_COME, .._GO, .._QUIT, .._QUIT_SYSTEM
    &mRT.stMsgTime.wYear,           //Day
    &mRT.stMsgTime.wMonth,          //Month
    &mRT.stMsgTime.wDay,            //Year
    &mRT.stMsgTime.wHour,           //Hour
    &mRT.stMsgTime.wMinute,         //Minute
    &mRT.stMsgTime.wSecond,         //Second
    &mRT.stMsgTime.wMilliseconds,   //Millisecond
    &mRT.dwTimeDiff,                //Zeitdauer der anstehenden Meldung
    &mRT.dwCounter,                 //Interner Meldungszähler
    &mRT.dwFlags);                  //Flags( intern )

  // create message-string and filename if state == COME
  if (mRT.dwMsgState == MSG_STATE_COME ||
    mRT.dwMsgState == MSG_STATE_GO ||
    mRT.dwMsgState == MSG_STATE_QUIT)
  {

    // Open file, this file must be located somewhere where we have read/write/delete
    // access rights
    pfExport =
      fopen("C:\\ProgramData\\TroSoft\\Nimbus Alarm Server 3\\Project\\alarm.txt","a");

    if (pfExport  != NULL)
    {

      sprintf (date,"%04d-%02d-%02d",
      mRT.stMsgTime.wYear,
      mRT.stMsgTime.wMonth,
      mRT.stMsgTime.wDay);
      sprintf (time,"%02d:%02d:%02d",
      mRT.stMsgTime.wHour,
      mRT.stMsgTime.wMinute,
      mRT.stMsgTime.wSecond);

      if (MSRTStartMsgService(&dwID, NULL, NULL, 0, NULL, pError))
      {

        MSRTGetMsgCSData (mRT.dwMsgNr, &MsgData, pError);
        // These are the Texts as they appear in the Text-tab in alarm logging,
        // block 1 -> dwTextID[0] etc
        MSRTGetMsgText (0, MsgData.dwTextID[0], &mtsBlock1, pError);
        MSRTGetMsgText (0, MsgData.dwTextID[1], &mtsBlock2, pError);
        MSRTGetMsgText (0, MsgData.dwTextID[2], &mtsBlock3, pError);
        MSRTGetClassInfo (mRT.dwMsgNr, &MsgClass, pError);
        MSRTGetMsgText (0,MsgClass.dwName, &mtsClass, pError);
        MSRTGetMsgPriority (MsgData.dwMsgNr, (long*)&lPriority, pError);

        fprintf(pfExport,"%s#", date);
        fprintf(pfExport,"%s#", time);
        fprintf(pfExport,"%s#", mtsBlock1.szText);

        // This text is the only cruical text for Nimbus
        switch (mRT.dwMsgState)
        {
        case MSG_STATE_COME:
          fprintf(pfExport,"%s","Aktivt");
          break;
```

```
          case MSG_STATE_GO:
            fprintf(pfExport,"%s","Avgått");
            break;
          case MSG_STATE_QUIT:
            fprintf(pfExport,"%s","Kvitterat");
            break;
          }
          fprintf(pfExport,"#");

          fprintf(pfExport,"%ld#", lPriority);
          fprintf(pfExport,"%s#", mtsClass.szText);

          // You may add any texts you wish but Nimbus will only import them if you
          // select them in the file found in
          // Nimbus Project folder ..\Project\Import\Import_WinCC.imp
          fprintf(pfExport,"%s#", mtsBlock3.szText);

          fprintf(pfExport,"\n");

          MSRTStopMsgService( dwID, pError);

        } // MSRTStartMsgService

        //Close and save File
        fclose( pfExport );

      } // pfExport  != NULL

    } // mRT.dwMsgState

  } // pszMsgData != NULL

  return TRUE;

}
```

Metod 2: Konfigurera WinCC för att exportera larm via WinCC2Nimbus

Installera applikationen *WinCC2Nimbus* som finns på levererad media, den finns i mappen *..\Tools and documents\WinCC*. Applikationen finns också på TroSoft hemsida att ladda hem.

WinCC2Nimbus kan köras som en vanlig applikation eller som en tjänst. Körs den som vanlig applikation ska den alltid startas som Administratör.

För att lägga till *WinCC2Nimbus* som tjänst så startar man den en gång med startparametern *-i*

Exempelvis:

"C:\Pogram Files (x86)\WinCC2Nimbus\WinCC2Nimbus.exe" -i

Obs! För att installationen ska bli rätt måste man göra detta i ett eleverat kommandofönster (startas som Administratör).

Vill man ta bort applikationen från tjänsterna använder man startparametern *-u*. Se till att applikationen är stoppad först.

När man lagt till applikationen som tjänst måste man starta den med tjänstehanteraren. Applikationen kommer starta automatiskt när datorn startas om. För att säkerställa WinCC hinner starta först, sätt tjänstens *Startup Type* till *Automatic (Delayed start).*

Lämpligt är att köra den som vanlig applikation tills man sett att det kommer in larm. Larmhändelserna visas i fönstret i *WinCC2Nimbus*. Programmet skapar också en LogFiles-mapp i sin installationsmapp där alla händelser loggas. Loggfilerna raderas automatiskt när de blir för gamla, per default lagras de i 90 dagar.

WinCC2Nimbus kan också skapa larmhändelser i Nimbus när det startas och stoppas samt när WinCC startas och stoppas. Inställningarna för detta görs i konfigurationsfilen *WinCC2Nimbus.ini* som ligger i installationsmappen. Varje händelse har en egen sektion, ex för att skapa ett larm när WinCC stoppas, redigera sektionen *[NimbusMessageWhenWinCCIsStopped]*

EventType=1

T0=WinCC
T1=
T2=99
T3=
T4=WinCC2 has stopped

EventType är typen av händelse*, 0 = Inactive (normal), 1 = Active (Larm), 2 = Acknowledge (Kvittens)*

Sätter man *EventType* till 1 enligt ovan så kommer *WinCC2Nimbus* att skapa ett larm och skicka till Nimbus när WinCC stoppas. Alla fält *T0 (Tag)* till *T4 (Beskrivning)* kan fritt anges och därmed kan man skapa profiler som passar just dessa larm eller låta dem passa in i andra profiler man har gjort.

Tips: För att övervaka processer i servern (ex WinCC-services) så kan man använda *Watchdog*-funktionen i Nimbus som finns i *Nimbus Explorer* under *Inställningar -> Serverinställningar -> Watchdog*

WinCC2Nimbus kommer prenumerera på alla larmhändelser i *WinCC* och skicka över dem till *Nimbus Alarm Server*. Det finns två varianter på vilket *WinCC2Nimbus* kan skicka över larm till *Nimbus Alarm Server*:

a) *Genom att lägga larm direkt i Nimbus kommandokö*
b) *Genom en TCP-förbindelse (Generic TCP)*

Variant A är default och kräver inte att man konfigurerar någon *SCADA import* på *Nimbus*, enda förutsättningen är att *Nimbus Alarm Server* är installerat på WinCC-servern.

Ska variant A användas, starta *WinCC* samt *Nimbus Alarm Server* och skapa ett larm. Det ska komma in i både *WinCC2Nimbus* och *Nimbus Explorer*.

Metod 2 - variant B: Konfigurera WinCC2Nimbus för TCP

Variant B är lämplig om man har *Nimbus Alarm Server* installerad på en annan server. För att *WinCC2Nimbus* ska tillåta anslutning via TCP behöver man ändra parametern *ConnectionType* i *WinCC2Nimbus.ini* från 0 till 1.

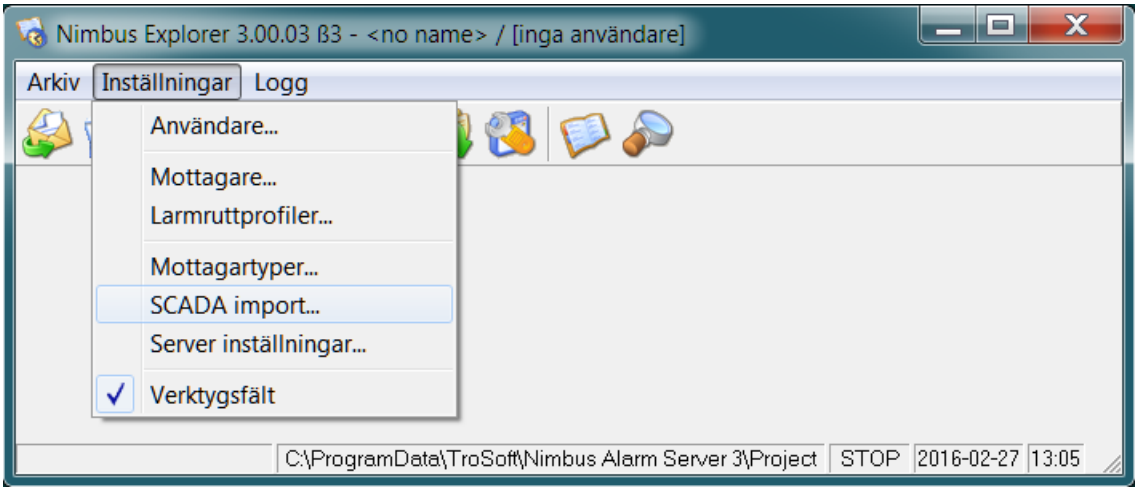
WinCC2Nimbus kommer lyssna som socket server på TCP port *14000* per default. Porten kan ändras med parametern *PortNumber*.

Starta om *WinCC2Nimbus* när något ändras i *WinCC2Nimbus.ini.*

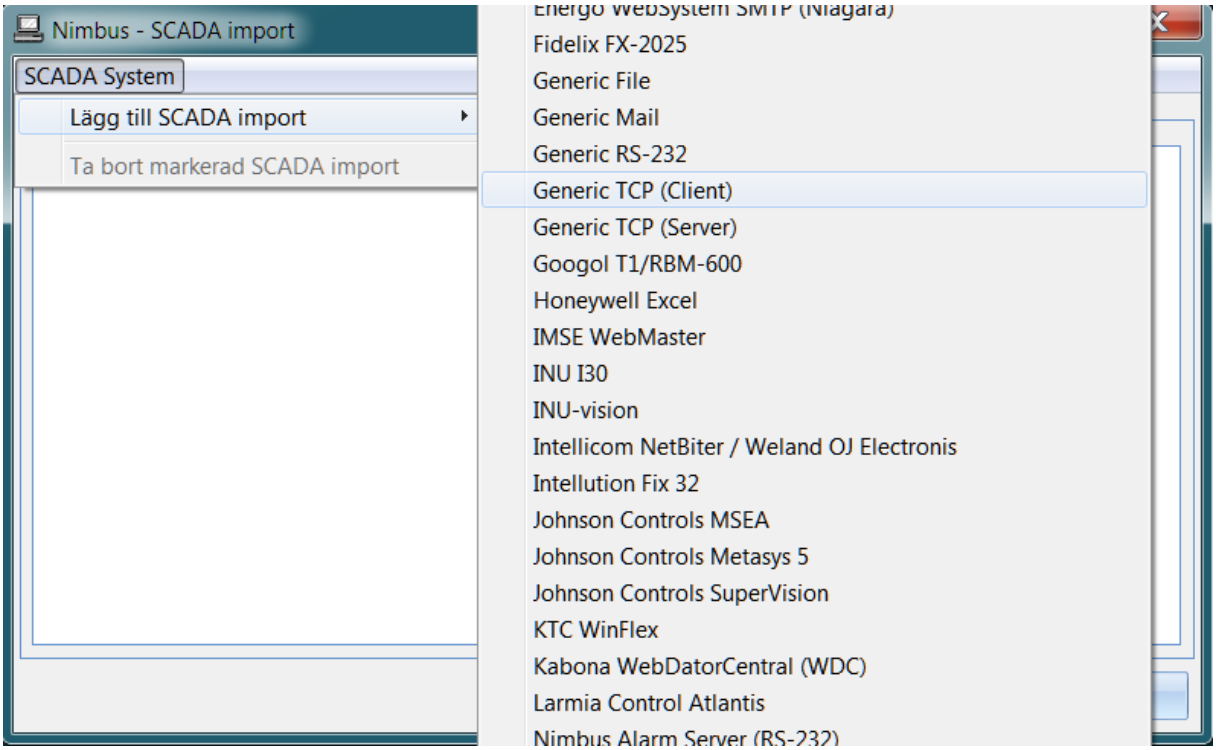
Obs! Se till att brandväggarna i servrarna vid behov har undantag för WinCC2Nimbus (den agerar socket server) och Nimbus Alarm Server (den agerar socket klient) på den port som är vald.

Metod 2 - variant B: Konfigurera Nimbus för att hämta larm från WinCC2Nimbus via TCP

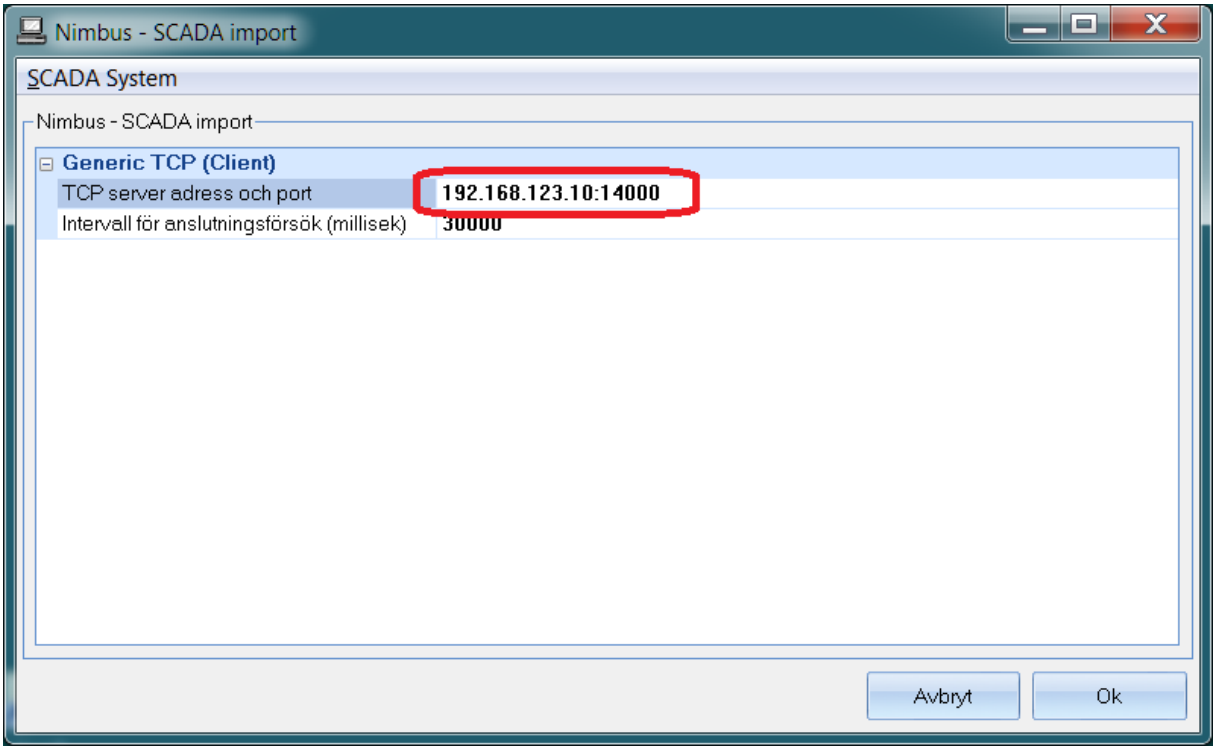
Starta *Nimbus Explorer* (högerklicka och välj *'Kör som Administratör'*) via genvägen på Nimbus servern. *Nimbus Explorer* ska alltid startas som Administratör.



Välj *Inställningar -> SCADA import.*



Välj *SCADA System -> Lägg till SCADA Import -> Generic TCP (Client)*



Ange adress och port till *WinCC*-servern. *Nimbus Alarm Server* är socket klient.

Starta *WinCC* samt *Nimbus Alarm Server* och skapa ett larm. Det ska komma in i både *WinCC2Nimbus* och *Nimbus Explorer*.