

# Import alarms from Siemens TIA Portal (WinCC RT Professional) to Nimbus

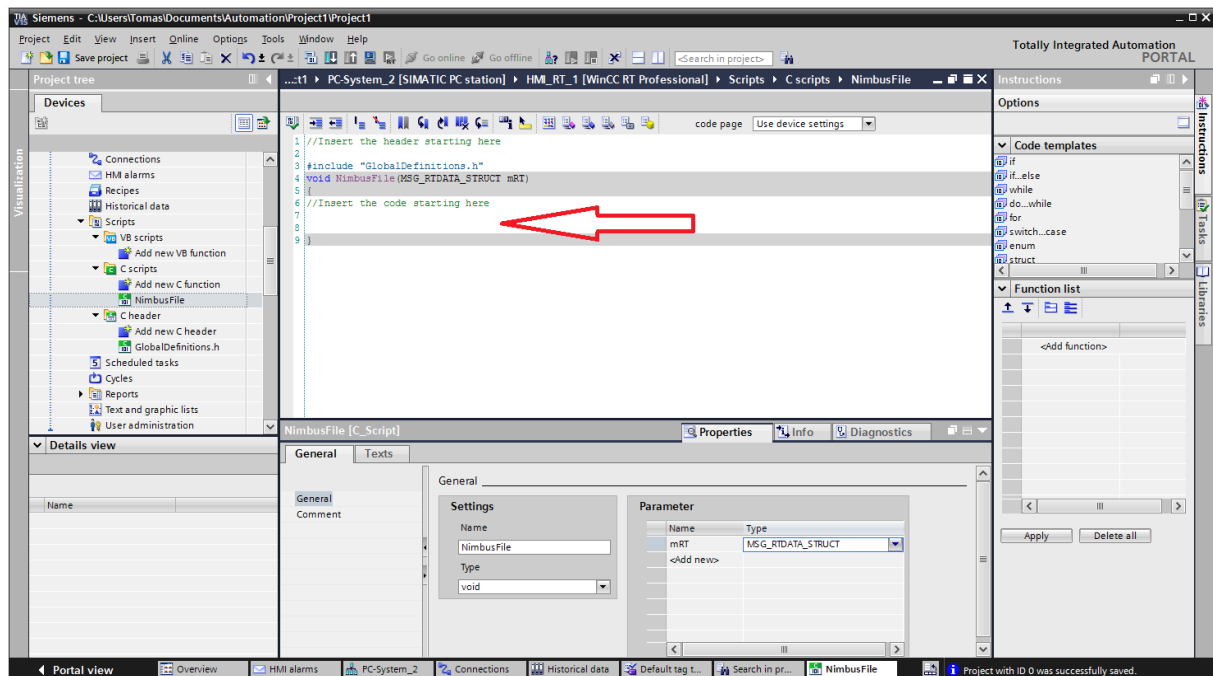
*WinCC RT Professional* can export HMI alarms to Nimbus using a text file created by a *C-script*

## Configure WinCC to create the Nimbus readable alarm event text file

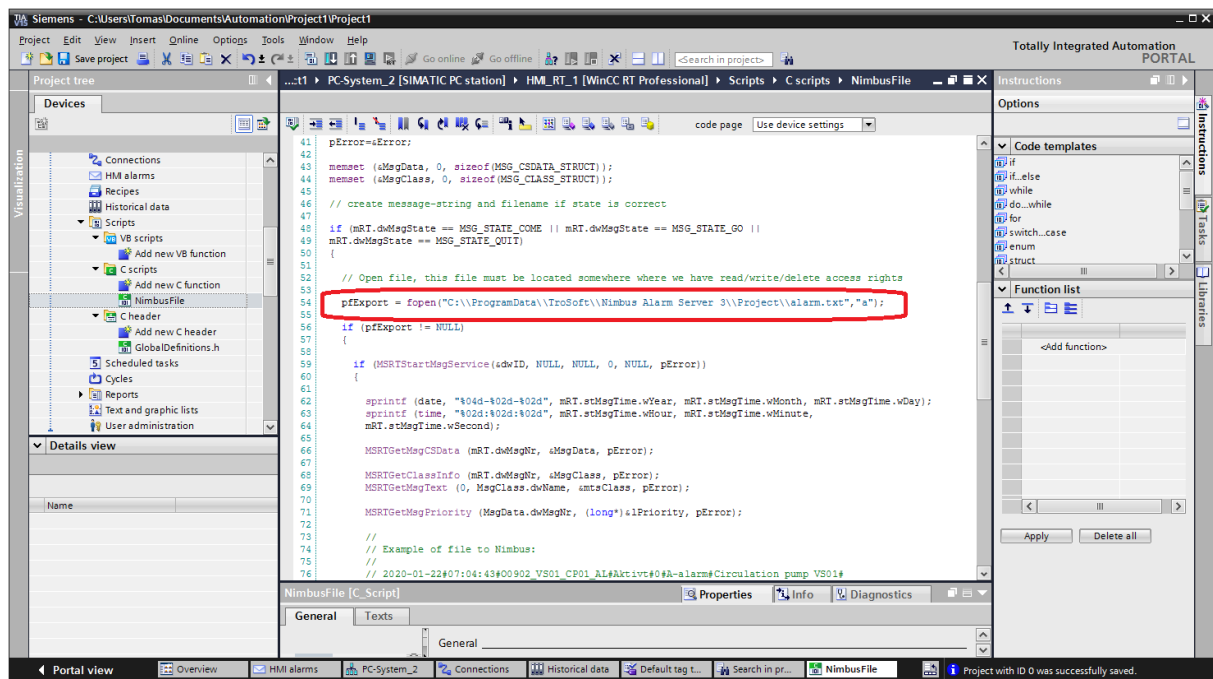
Add a new C-script in the Project view -> *PC-system* -> *HMI\_RT* -> *Scripts* -> *C scripts*. Name the script *NimbusFile*.

Create a new function named *NimbusFile* of type *void*.

Add parameter *mRT* of type *MSG\_RTDATA\_STRUCT*.



Copy the program code from the *NimbusFile* C-script section below and insert it into the void *NimbusFile(MSG\_RTDATA\_STRUCT mRT)* function (red arrow)

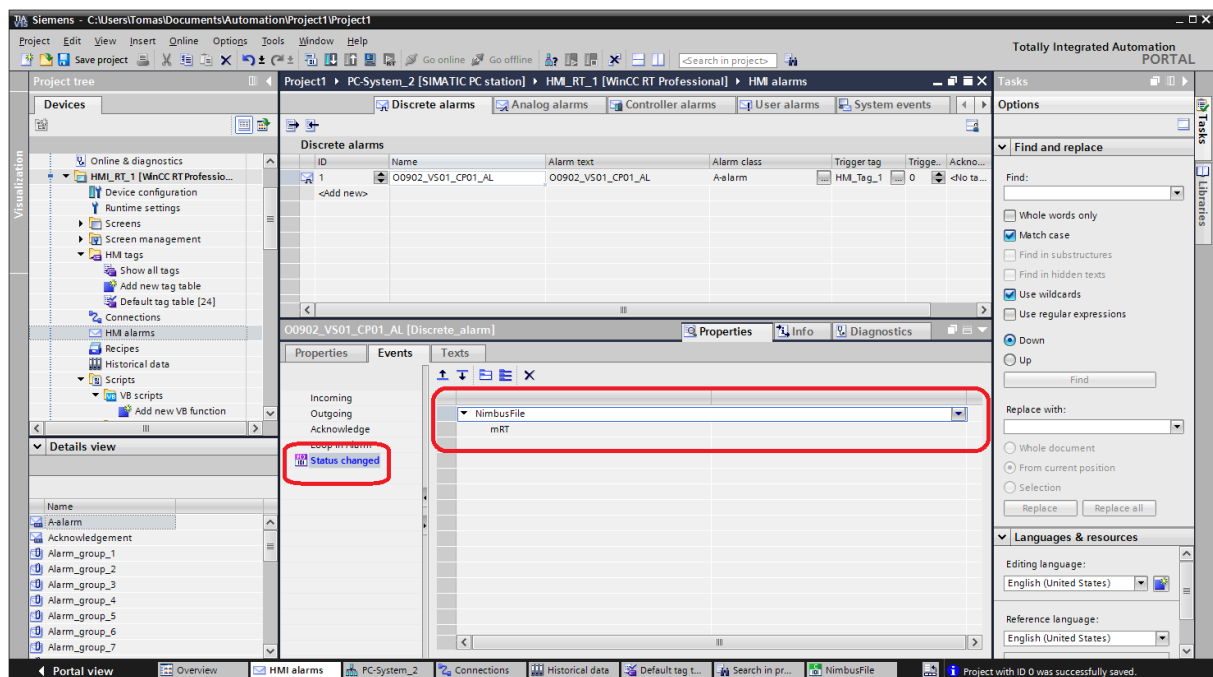


Edit the row with the path and filename if needed. It will by default point to the *Nimbus version 3 project folder*. The folder is created when *Nimbus* is installed. You can choose any folder you like as long as both *Nimbus* and *WinCC* have security settings allowing the applications to create and delete files in that folder.

Use the *Edit -> Check syntax* menu choice to ensure the code is ok and save the project

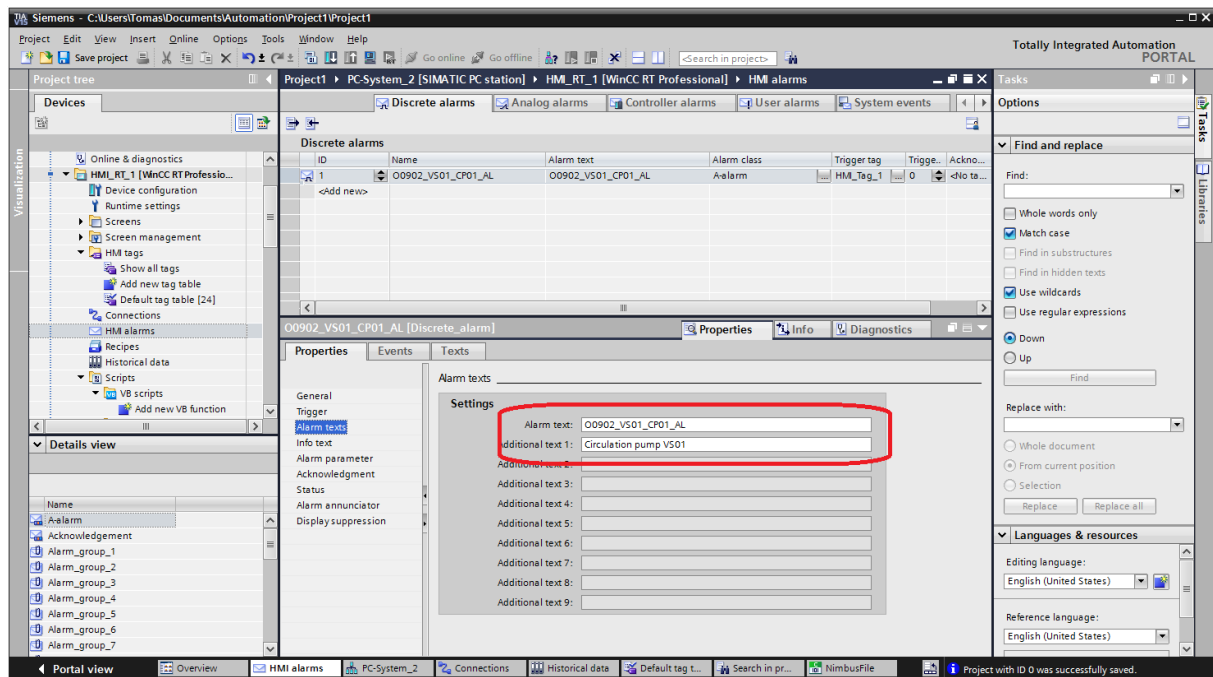
Open *HMI Alarms*.

*Goto Events -> Status changed*. Select the alarms that should send their alarm events to Nimbus and add function *NimbusFile*.



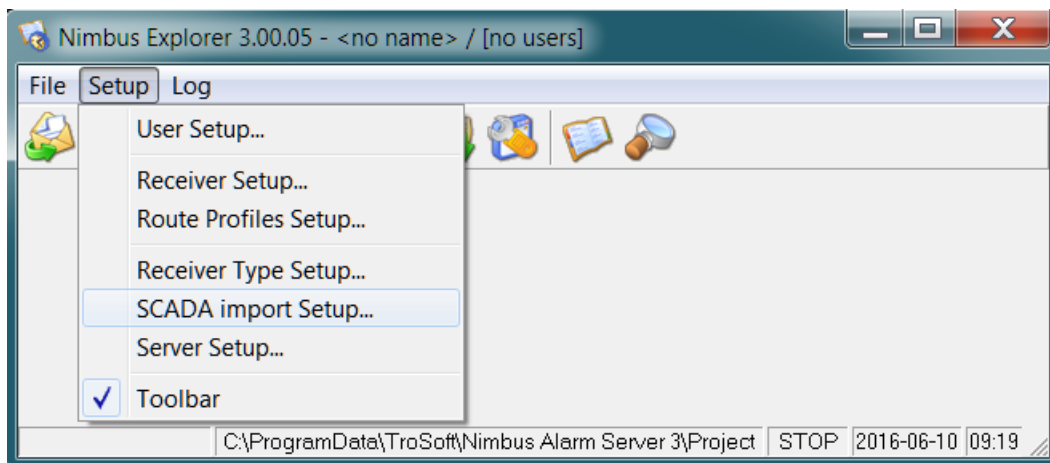
*Obs! Remember to add this function if new HMI alarms are created in the future.*

Nimbus will by default use the first two text fields (inserted into fields *T0 (Tag)* and *T4 (Description)*):

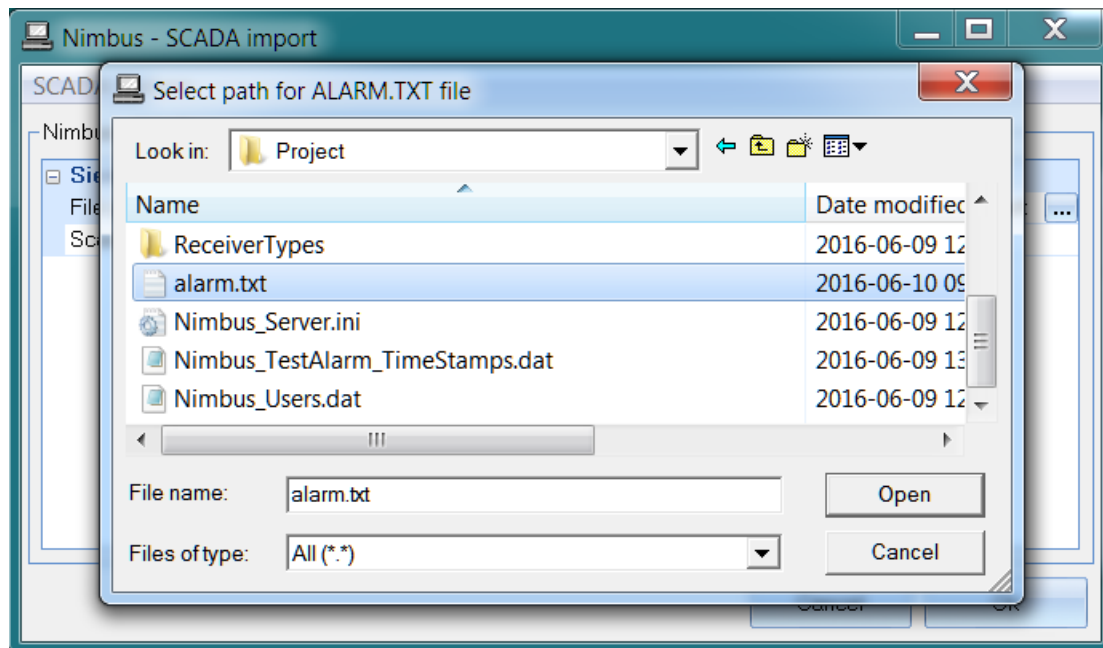


## Configure Nimbus to import the alarm event text file

Run *Nimbus Explorer* (right click and select *Run as Administrator*) using its shortcut. *Nimbus Explorer* shall always be run as *Administrator*.



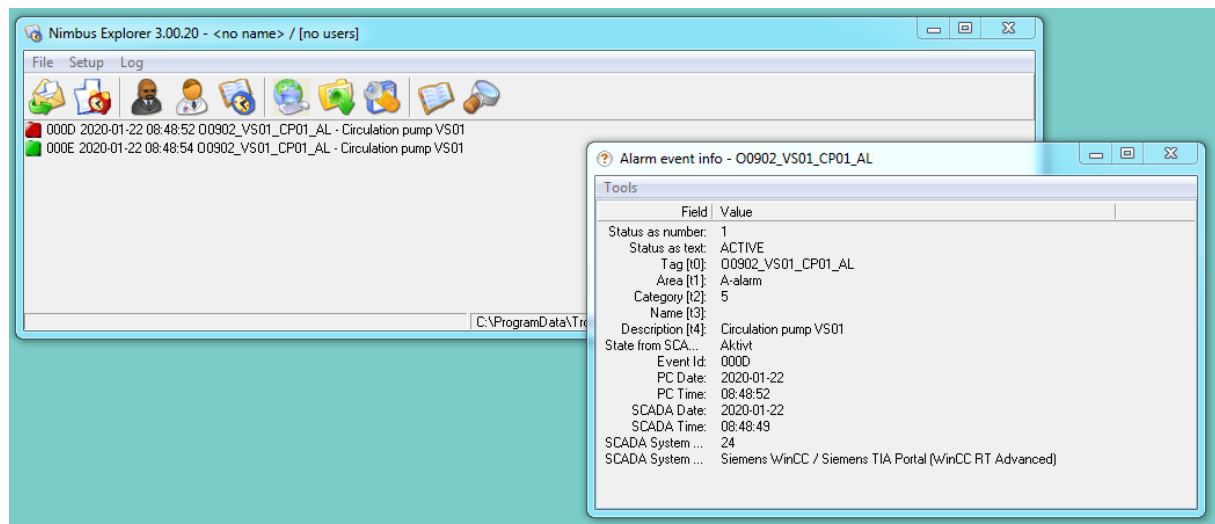
Select *Setup -> SCADA Import Setup*.



Select SCADA System -> Add SCADA System Import -> Siemens WinCC

Select *alarm.txt* found in the same folder that was set in the script. If the file does not exist, just select *Open*. Ensure the *File path to scan* also included the textfile name.

When you run *Nimbus Alarm Server*, the program will remove the textfile because it contains just old alarm events. Create a new alarm in *WinCC*. It should now appear in *Nimbus Explorer*. Below is alarm events from the WinCC example above:



*Nimbus* always remove the file when it has been read

## NimbusFile C-script

```
//
// Date      / Vers  / Sign / Comment
// -----
// 2016.02.26 / 1.0.0.0 / TR  / Major changes
// 2016.02.27 / 1.0.0.1 / TR  / Added priority and Class moved to Area
```

```

// 2016.03.18 / 1.0.0.2 / TR / Added Text block 2
// 2020.01.21 / 1.0.0.3 / TR / Changed WinCC code to TIA WinCC RT Professional
//
//
//
// -----

extern char g_Msg[];

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

MSG_TEXT_STRUCT mtsBlock;

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

FILE *pfExport;
pError=&Error;

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

// create message-string and filename if state is correct
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)
    {

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

            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);

            MSRTGetMsgCSData (mRT.dwMsgNr, &MsgData, pError);

            MSRTGetClassInfo (mRT.dwMsgNr, &MsgClass, pError);
            MSRTGetMsgText (0, MsgClass.dwName, &mtsClass, pError);

            MSRTGetMsgPriority (MsgData.dwMsgNr, (long*)&lPriority, pError);

            //
            // Example of file to Nimbus:
            //
            // 2020-01-22#07:04:43#00902_VS01_CP01_AL#Aktivt#0#A-alarm#Circulation pump VS01#

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

            MSRTGetMsgText (0, MsgData.dwTextID[0], &mtsBlock, pError);
            fprintf(pfExport,"%s#", mtsBlock.szText);

            // This text is the only crucial 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

MSRTGetMsgText (0, MsgData.dwTextID[1], &mtsBlock, pError);
fprintf(pfExport, "%s#", mtsBlock.szText);

fprintf(pfExport, "\n");

MSRTStopMsgService( dwID, pError);

} // MSRTStartMsgService

//Close and save File
fclose(pfExport);

} // pfExport != NULL

} // if (mRT.dwMsgState...
```