CRYSTAL XE
APPLICATION NOTE

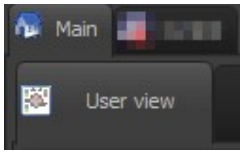
Create a script that runs in background

To create a script that runs in background, you can use a script object on the the main user view.

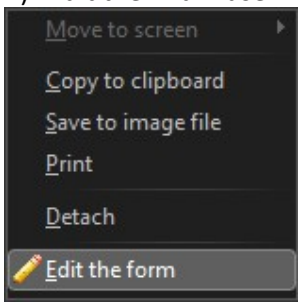
1 Add a script object

Follow these steps:

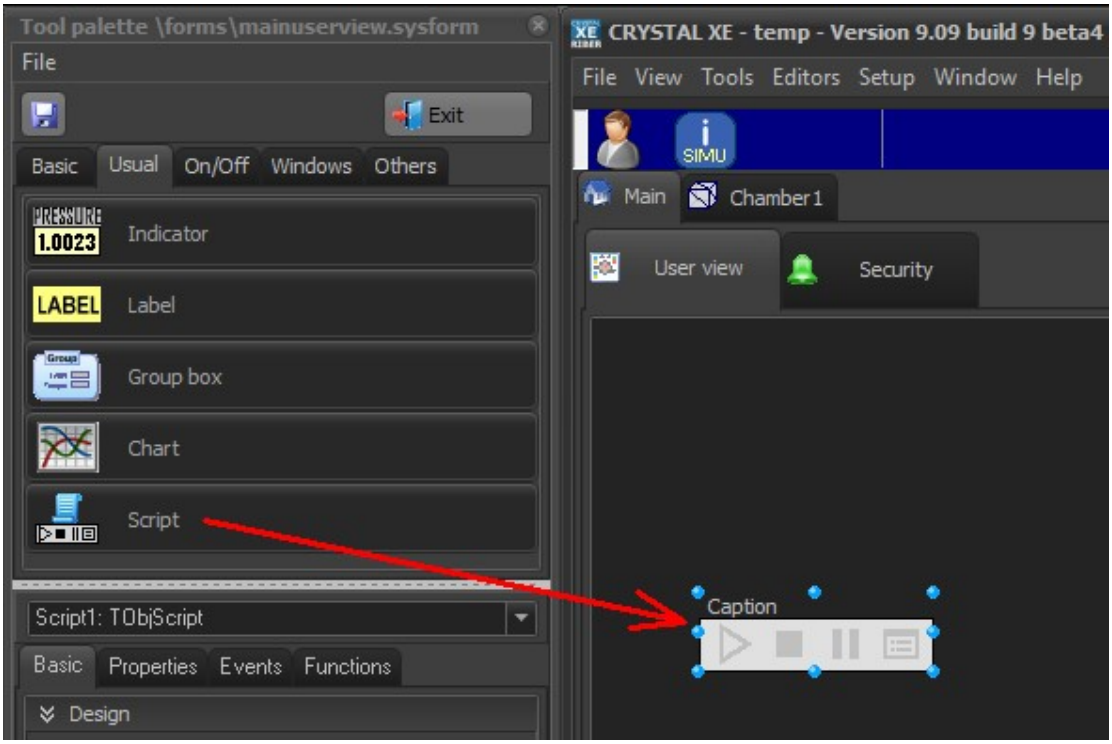
1) Click on the **Main** tab and **User view** tab



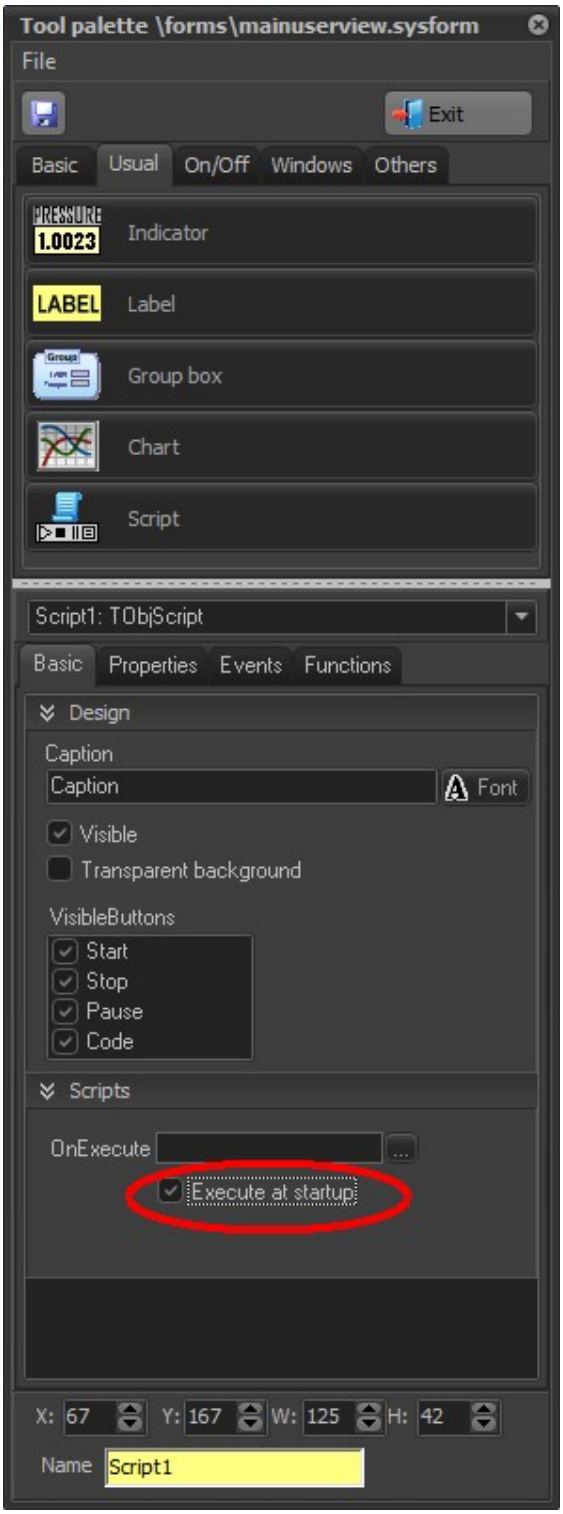
2) Edit the main user view by right clicking on the background and select “Edit the form”



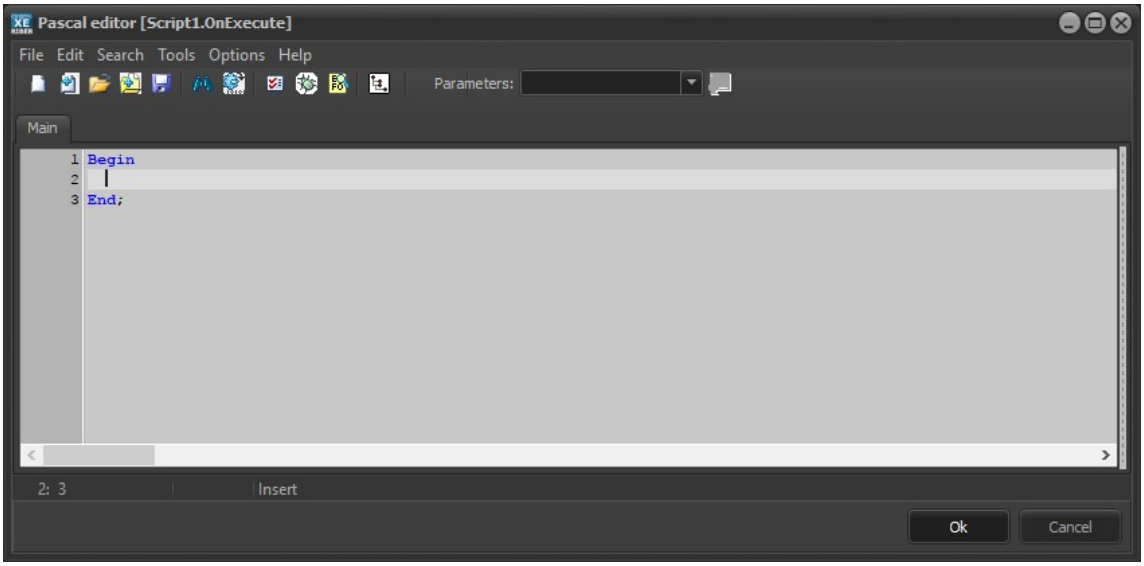
3) In the tool palette, click on the Usual Tab and drag and drop a script object into the main user view.



4) Check the box “Execute at startup”



5) Double click on the script object to open the script editor



Enter the script here and click on the button Ok when done.
Click on the Exit button located in the Tool palette window and answer Yes to keep the changes.

2 Example of scripts

2.1 Send a SMS when the PSSI interlock growth pressure is triggered.

```
Begin
  Repeat
    if Growth.PSSI.pssi.T1_Timer>=Growth.PSSI.pssi.T1_Max then
    Begin
      SendSMS('Growth pressure warn level Alarm');
      Repeat
      until Growth.PSSI.pssi.T1_Timer=0;
    end;
    sleep(500);
  until false;
End;
```

2.2 Create a new recorder file every day

```
Begin
   Repeat
     GROWTH.Recorder1.Start;
     WaitDateTime('00:00');
     GROWTH.Recorder1.Stop;
   Until false;
End;
```

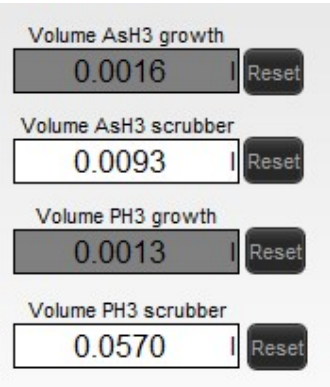
2.3 Log an event when a CVEP valve changes

```
Var
  CryoPump,Trap,TrapTurbo,IonPump,BufferTurbo : boolean;
  //-------------------------------------
```

```
  Function GetStatusStr(b:boolean):String;
  Begin
    if b then Result:='Open'; else Result:='Closed';
  End;
  //-------------------------------------
Begin
  CryoPump    := GROWTH.CVEP_valve.CryoPump.CVEP_Status;
  Trap        := GROWTH.CVEP_valve.Trap.CVEP_Status;
  TrapTurbo   := GROWTH.CVEP_valve.TrapTurbo.CVEP_Status;
  IonPump     := GROWTH.CVEP_valve.IonPump.CVEP_Status;
  BufferTurbo := GROWTH.CVEP_valve.BufferTurbo.CVEP_Status;
  Repeat
    if (CryoPump <>GROWTH.CVEP_valve.CryoPump.CVEP_Status) then
    Begin
      CryoPump    := GROWTH.CVEP_valve.CryoPump.CVEP_Status;
      LogEvent(0, 790, 3, 0,'CVEP_valve.CryoPump '+GetStatusStr(CryoPump));
    End;
    if (Trap <> GROWTH.CVEP_valve.Trap.CVEP_Status) then
    Begin
      Trap    := GROWTH.CVEP_valve.Trap.CVEP_Status;
      LogEvent(0, 790, 3, 0,'CVEP_valve.Trap '+GetStatusStr(Trap));
    End;
    if (TrapTurbo <> GROWTH.CVEP_valve.TrapTurbo.CVEP_Status) then
    Begin
      TrapTurbo := GROWTH.CVEP_valve.TrapTurbo.CVEP_Status;
      LogEvent(0, 790, 3, 0,'CVEP_valve.TrapTurbo '+GetStatusStr(TrapTurbo));
    End;
    if (IonPump <> GROWTH.CVEP_valve.IonPump.CVEP_Status) then
    Begin
      IonPump := GROWTH.CVEP_valve.IonPump.CVEP_Status;
      LogEvent(0, 790, 3, 0,'CVEP_valve.IonPump '+GetStatusStr(IonPump));
    End;
    if (BufferTurbo <> GROWTH.CVEP_valve.BufferTurbo.CVEP_Status) then
    Begin
      BufferTurbo := GROWTH.CVEP_valve.BufferTurbo.CVEP_Status;
      LogEvent(0, 790, 3, 0,'CVEP_valve.BufferTurbo '+GetStatusStr(BufferTurbo));
    End;
    sleep(500);
  until false;
End;
```

2.4 Calculate the volume of the gas

HMI for this script




```
//-------------------------------------------------------------------------------
// This script is always running in background
```

```
// It calculates the volume of the AsH3MFM and PH3MFM
// It uses the following global variables
//  - App.Var.VolAsH3Growth   : When one valve VRun is openned
//  - App.Var.VolAsH3Scrubber : When both valves VRun are closed (Gas cleaning)
//  - App.Var.VolPH3Growth    : When one valve VRun is openned
//  - App.Var.VolPH3Scrubber  : When both valves VRun are closed (Gas cleaning)
//  These global variables are automatically saved when exiting the application
//-------------------------------------------------------------------------------
uses const_colors;
Begin
  ResetTimer(1);
  ResetTimer(2);
  Repeat

    if (GROWTH.AsH3_1.Valves6.VRun or GROWTH.AsH3_2.Valves6.VRun) then
    Begin
      IndicatorAsH3Growth.Color := clWhite;
      IndicatorAsH3Scrubber.Color := clDkGray;
      App.Var.VolAsH3Growth := App.Var.VolAsH3Growth +
(GetTimer(1)/60)*GROWTH.Hydride_MFM.AsH3MFM.MV/1000;
    end else
    Begin
      IndicatorAsH3Growth.Color := clDkGray;
      IndicatorAsH3Scrubber.Color := clWhite;
      App.Var.VolAsH3Scrubber := App.Var.VolAsH3Scrubber +
(GetTimer(1)/60)*GROWTH.Hydride_MFM.AsH3MFM.MV/1000;
    End;
    ResetTimer(1);

    if (GROWTH.PH3_1.Valves6.VRun or GROWTH.PH3_2.Valves6.VRun) then
    Begin
      IndicatorPH3Growth.Color := clWhite;
      IndicatorPH3Scrubber.Color := clDkGray;
      App.Var.VolPH3Growth := App.Var.VolPH3Growth +
(GetTimer(2)/60)*GROWTH.Hydride_MFM.PH3MFM.MV/1000;
    end else
    Begin
      IndicatorPH3Growth.Color := clDkGray;
      IndicatorPH3Scrubber.Color := clWhite;
      App.Var.VolPH3Scrubber := App.Var.VolPH3Scrubber +
(GetTimer(2)/60)*GROWTH.Hydride_MFM.PH3MFM.MV/1000;
    end;
    ResetTimer(2);

    sleep(1000);
  until false;
End;
```

2.5 Start a fail recipe when two conditions are true

```
//---------------------------------------------------------------
Function AlarmCondition:boolean;
Begin
  result := (C21DZ.Growth_Gauge.pressure.MV>1.0E-5) and (C21DZ.Manipulator.FluxMV.MV>1.0E-
6);
end;
//---------------------------------------------------------------
Begin
  Repeat
    if AlarmCondition then
    Begin
      C21DZ.Recipe.LoadFromFile('FailRecipe.rcp',2); //Start the recipe with priority 2
      C21DZ.Recipe.Start(0);
```

```
      LogEvent(0, 790, 0, 3,'Two conditions are true, start the failed recipe');
      Repeat
      until C21DZ.Recipe.Status<>1; // wait for the recipe is not running
      Repeat
      until not(AlarmCondition); // wait for the alarm disappears
    end;
  until false;
End;
```