

SIEMENS

Ingenuity for life

Industry Online Support

Home

如何在 WinCC V7.5 SP1 中 修改报警记录的消息文本

WinCC / V7.5 SP1 / Alarm / ODK

<https://support.industry.siemens.com/cs/ww/en/view/109783954>

Siemens
Industry
Online
Support



This entry is from the Siemens Industry Online Support. The general terms of use (http://www.siemens.com/terms_of_use) apply.

c

Siemens 为其产品及解决方案提供了工业信息安全功能，以支持工厂、系统、机器和网络的安全运行。

为了防止工厂、系统、机器和网络受到网络攻击，需要实施并持续维护先进且全面的工业信息安全保护机制。**Siemens** 的产品和解决方案仅构成此类概念的其中一个要素。

客户负责防止其工厂、系统、机器和网络受到未经授权的访问。只有在必要时并采取适当安全措施（例如，使用防火墙和网络分段）的情况下，才能将系统、机器和组件连接到企业网络或 Internet。

此外，应考虑遵循 **Siemens** 有关相应信息安全措施的指南。更多有关工业信息安全的信息，请访问 <http://www.siemens.com/industrialsecurity>。

Siemens 不断对产品和解决方案进行开发和完善以提高安全性。**Siemens** 强烈建议您及时更新产品并始终使用最新产品版本。如果使用的产品版本不再受支持，或者未能应用最新的更新程序，客户遭受网络攻击的风险会增加。

要及时了解有关产品更新的信息，请订阅 **Siemens** 工业信息安全 RSS 源，网址为 <http://www.siemens.com/industrialsecurity>。

目录

1	<概述>	4
1.1	<解决思路>.....	4
1.2	<MSRTSetComment 函数>.....	4
2	<创建变量>	6
3	<组态报警>	7
4	<组态画面>	8
4.1	<添加画面对象>.....	8
4.2	<添加组合框控件>.....	8
4.3	<设置报警视图属性>.....	9
5	<修改报警注释>	13
6	<运行结果>	14

1 <概述>

<在工业生产中不可避免地会产生很多报警信息，其中有些报警是需要分析原因的，而大部分报警产生的原因是需要事后才能获得，这就需要在报警消息产生一段时间后，把报警产生的原因再写入到报警信息中。例如，图 1 中报警的消息文本“设备 2 故障，原因：电气故障”，这其中“电气故障”信息就是在分析报警产生的原因之后由操作人员选择相应报警并写入原因。

	日期	时间	编号	消息文本
1	2020/11/21	10:41:00.742	2	设备2故障, 原因: 电气故障
2	2020/11/21	10:57:41.358	1	设备1故障, 原因: 机械故障
3	2020/11/21	11:05:20.781	3	设备3故障, 原因: 其它原因
4	2020/11/21	11:05:35.885	1	设备1故障, 原因: 电气故障
5	2020/11/21	11:05:35.885	2	设备2故障, 原因: 机械故障
6	2020/11/21	11:05:35.885	3	设备3故障, 原因: 机械故障
7				
8				

图 1

>

1.1 <解决思路>

<WinCC 报警消息产生后，无法直接修改其消息文本，考虑到消息文本可以通过格式规范来引用消息注释对话框的字段内容，其中，@100%s@ = 计算机名称，@101%s@ = 应用程序名称（最多 32 个字符），@102%s@ = 用户名称（最多 16 个字符），@103%s@ = 消息归档列表中的消息注释（最多 255 个字符）。因此可以通过修改报警注释（Comment）来实现事后写入报警原因的功能。>

1.2 <MSRTSetComment 函数>

<MSRTSetComment 是 WinCC ODK 函数，需要安装 WinCC ODK 库才能使用。安装 ODK 库之后可以查看 ODK 帮助中有关 MSRTSetComment 函数的说明，如图 2 所示。

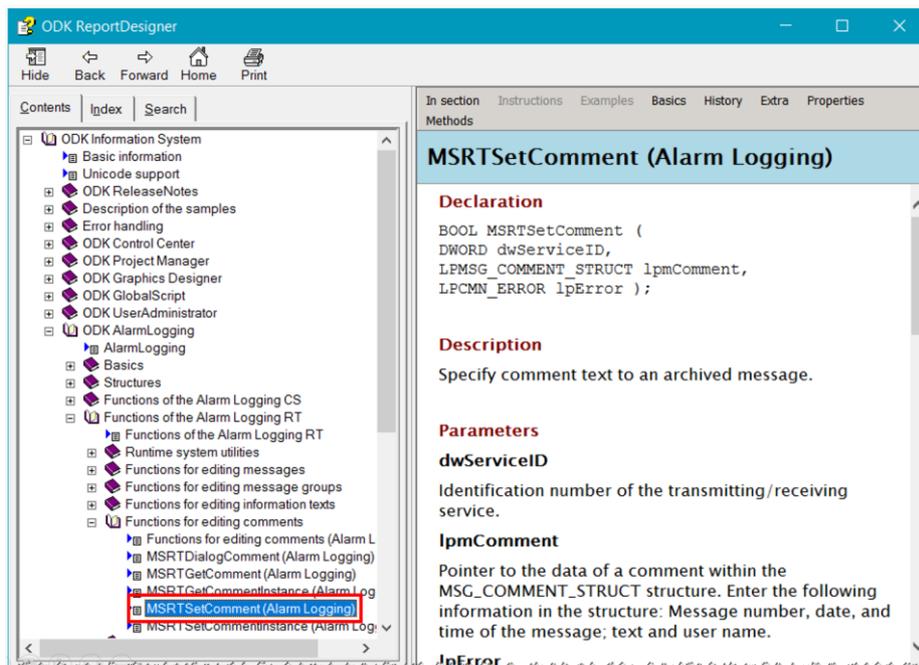


图 2

函数格式：

```
BOOL MSRTSetComment ( DWORD dwServiceID, LPMSG_COMMENT_STRUCT lpmComment, LPCMN_ERROR lpError );
```

各个参数的含义如下：

dwServiceID：连接 ID，可以自定义。

lpmComment：为数据结构变量，类型为 LPMSG_COMMENT_STRUCT，包括以下内容：

- lpmComment.dwMsgNr：报警编号
- lpmComment.stTime：报警时间
- lpmComment.szText：将要写入的注释
- lpmComment.szUser：操作人员

lpError：函数执行返回的错误信息。

MSRTSetComment 函数通过 dwMsgNr 和 stTime 参数来定位报警记录，然后把指定的注释内容 (szText) 和操作人员信息 (szUser) 写入到归档报警中。

下面内容详细介绍组态步骤及细节。

>

2 <创建变量>

<在 WinCC 创建以下内部变量，用于触发报警并保存所选报警的时间信息。>

变量管理		变量 [alarmText]			
!!! 变量管理		名称	注释	数据类型	
内部变量	alarmText	1	alarm1	报警变量	无符号的 16 位值
	NewGroup_1	2	alarmDay	报警时间-日	无符号的 8 位值
	NewGroup_2	3	alarmHour	报警时间-小时	无符号的 8 位值
	Performance	4	alarmMin	报警时间-分钟	无符号的 8 位值
	ProcessHistorian	5	alarmMonth	报警时间-月	无符号的 8 位值
	Script	6	alarmMsec	报警时间-毫秒	无符号的 16 位值
	TagLoggingRt	7	alarmNum	报警编号	无符号的 32 位值
结构变量		8	alarmSec	报警时间-秒	无符号的 8 位值
		9	alarmTim	报警时间	日期/时间
		10	alarmYear	报警时间-年	无符号的 16 位值
		11	✖		

图 3

这其中，变量“alarm1”用于触发 WinCC 报警。变量“alarmNum”和“alarmTim”用于保存所选报警的编号和产生时间。

>

3 <组态报警>

<在 WinCC 中创建如下报警，消息等级选择“系统，无确认”，消息类型选择“过程控制系统”。目的是无需对报警消息进行确认，且此消息类型不带“离开”状态，结果是报警消息只有到达状态。

报警记录		消息 [选择]				
消息	编号	消息变量	消息位	消息等级	消息类型	消息文本
错误	1	alarm1	0	系统，无确认	过程控制系统	设备1故障，原因：@103%s@
系统，需要确认	2	alarm1	1	系统，无确认	过程控制系统	设备2故障，原因：@103%s@
系统，无确认	3	alarm1	2	系统，无确认	过程控制系统	设备3故障，原因：@103%s@
消息块	4	✘	✘			
消息组	5					
系统消息	6					
限值监视	7					

图 4

在每条报警的消息文本中以“@103%s@”格式引用报警注释内容。

>

4 <组态画面>

4.1 <添加画面对象>

<在画面中添加报警视图、输入/输出域、静态文本、组合框以及写入按钮，如图 5 所示。

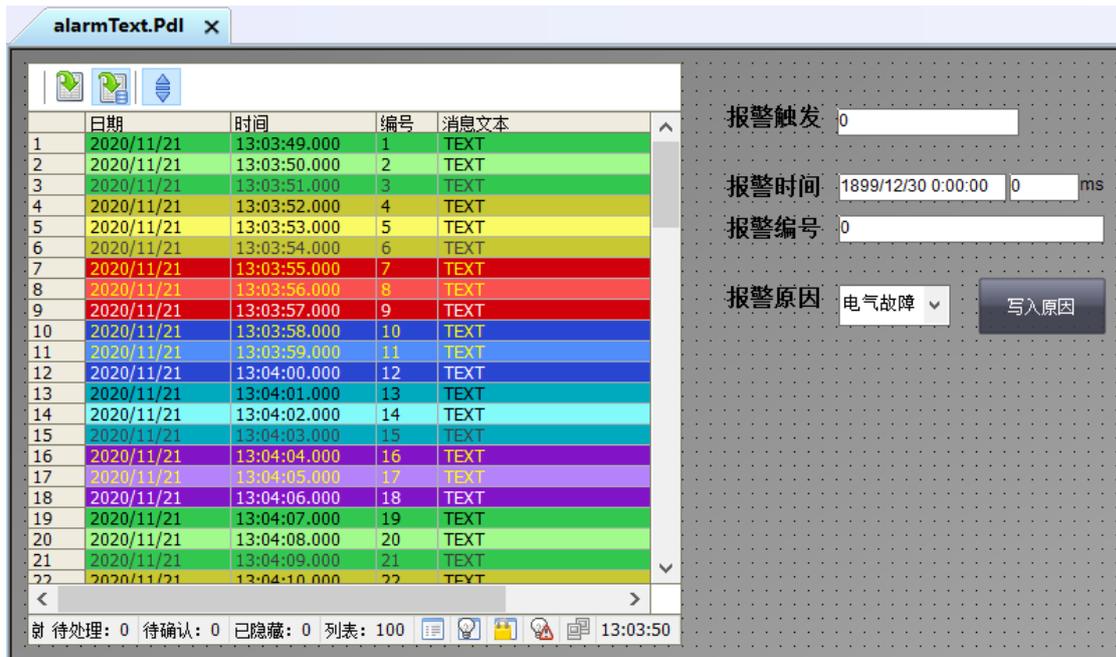


图 5

这其中，报警触发连接变量“alarm1”，报警时间连接变量“alarmTim”和“alarmMsec”，报警编号连接变量“alarmNum”。>

4.2 <添加组合框控件>

<在画面上添加组合框控件，用来选择相应报警产生的原因，此处选择原因将会写入到报警信息中。可以从智能对象下拖拽组合框控件直接添加到画面，并设置其“字体”属性下的“索引”和“文本”项属性。具体的步骤如图 6 所示。

4 <组态画面>



图 6

也可以直接从 excel 中拖拽文本到 WinCC 直接生成组合框。如图 7 所示。

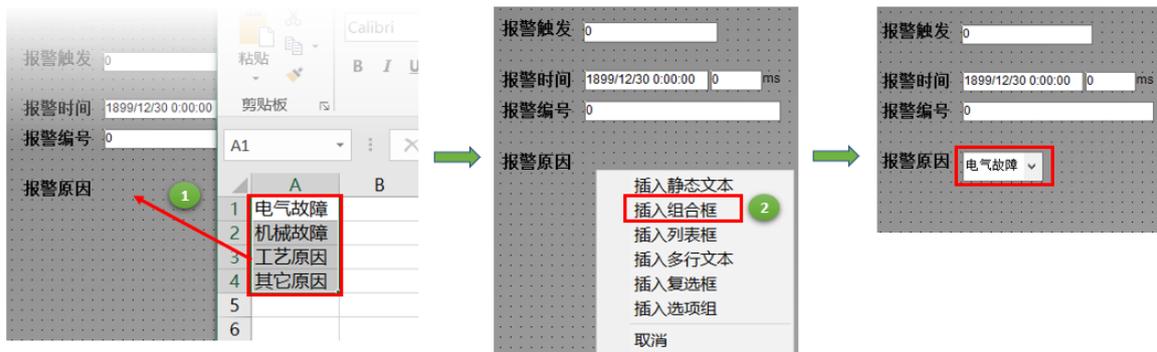


图 7

首先在 excel 输入需要的文本，选择所有文本，移动鼠标到选中区域的右下角，直到鼠标变成十字光标形状。

然后按着鼠标右键拖拽文本到 WinCC 画面，松开鼠标，在弹出对话框中选择“插入组合框”。

最后，具有四项原因文本的组合框被插入到 WinCC 画面。

>

4.3 <设置报警视图属性>

<双击报警视图控件，在“常规”栏下为打开画面是显示的列表选择“1-短期归档列表”，并取消“自动滚动”选项，如图 8 所示。



图 8

切换到“消息列表”栏，根据需要选择要显示的消息块。本例选择如图 9 所示。



图 9

切换到“工具栏”栏，只保留“消息列表”、“短期归档列表”和自动滚动三个工具。如图 10 所示。

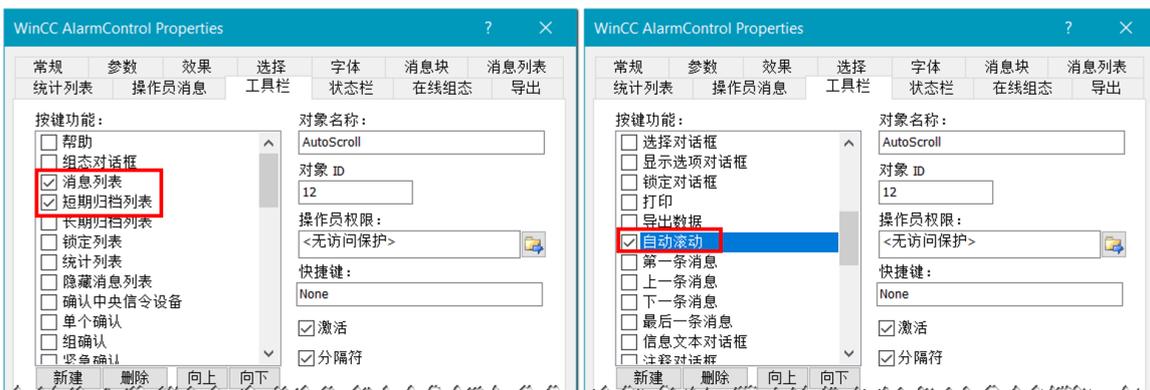


图 10

报警触发之后，需要在消息列表中选择要写入报警原因的报警，并且要获取所选报警的编号及时间信息。在报警视图控件中选择不同的报警，会触发控件的“OnSelectedMsgChanged”事件，在此事件脚本中可以直接读取所选报警的报警编号，然后通过读取报警视图中单元格的内容来获取报警时间。如图 11 所示。

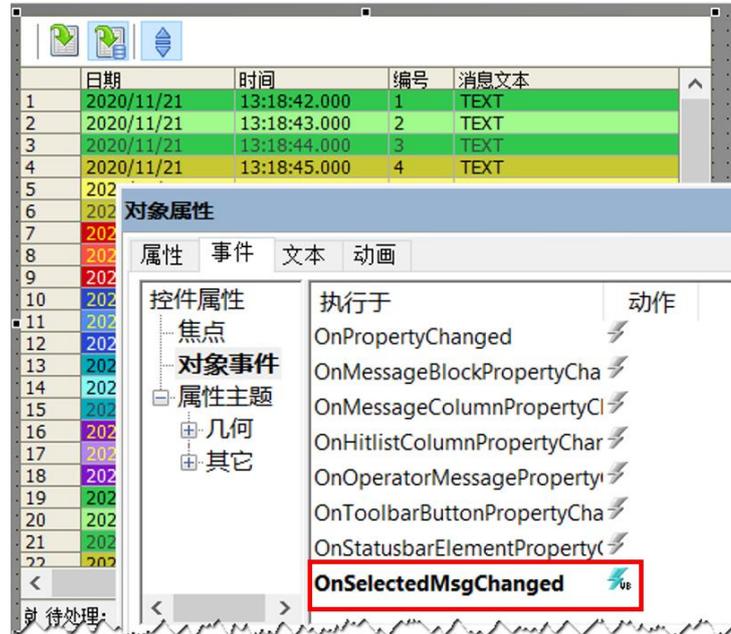


图 11

VBS 脚本如下：

```
Sub OnSelectedMsgChanged(Byval Item, Byval msgNr)
    Dim alarmNum
    Set alarmNum = HMIRuntime.Tags("alarmNum")
    alarmNum.Write msgNr '写入报警编号
    Dim alarmTim
    Set alarmTim = HMIRuntime.Tags("alarmTim")
    Dim alarmYear
    Set alarmYear = HMIRuntime.Tags("alarmYear")
    Dim alarmMonth
    Set alarmMonth = HMIRuntime.Tags("alarmMonth")
    Dim alarmDay
    Set alarmDay = HMIRuntime.Tags("alarmDay")
    Dim alarmHour
    Set alarmHour = HMIRuntime.Tags("alarmHour")
    Dim alarmHour0
    Set alarmHour0 = HMIRuntime.Tags("alarmHour")
    Dim alarmMin
    Set alarmMin = HMIRuntime.Tags("alarmMin")
    Dim alarmSec
```

```
Set alarmSec = HMIRuntime.Tags("alarmSec")
```

```
Dim alarmMsec
```

```
Set alarmMsec = HMIRuntime.Tags("alarmMsec")
```

```
Dim almTimt
```

```
'读取选中的报警消息的日期和时间
```

```
almTimt=FormatDateTime(ITEM.GETROW(ITEM.GetSelectedRow).CELLTEXT  
(1)&" "&
```

```
Left(ITEM.GETROW(ITEM.GetSelectedRow).CELLTEXT(2),Len(ITEM.GETRO  
W(ITEM.GetSelectedRow).CELLTEXT(2))-4))
```

```
alarmTim.Write almTimt
```

```
alarmYear.Write Year(almTimt)
```

```
alarmMonth.Write Month(almTimt)
```

```
alarmDay.Write Day(almTimt)
```

```
alarmHour.Write Hour(almTimt)
```

```
alarmMin.Write Minute(almTimt)
```

```
alarmSec.Write Second(almTimt)
```

```
'读取选中的报警消息的毫秒值
```

```
alarmMsec.Write
```

```
Right(ITEM.GETROW(ITEM.GetSelectedRow).CELLTEXT(2),3)
```

```
End Sub
```

```
>
```

5 <修改报警注释>

<在“写入原因”按钮中使用 MSRTSetComment 函数来修改报警注释。>



图 12

C 脚本如下：

```
#include "msrtapi.h"
#include "apdefap.h"

static DWORD I_svID = 0; //Connection ID to Alarmlogging
CMN_ERROR g_sErr;
LPCMN_ERROR pError = &g_sErr;
MSG_COMMENT_STRUCT MsgComment;
SYSTEMTIME almTime =
{GetTagWord("alarmYear"),GetTagByte("alarmMonth"),GetTagWord("alarmDay"),
GetTagWord("alarmDay"),GetTagByte("alarmHour"),GetTagByte("alarmMin"),GetT
agByte("alarmSec"),GetTagWord("alarmMsec")}; //获取报警时间
MsgComment.dwMsgNr =GetTagDWord("alarmNum"); //报警编号
MsgComment.stTime = almTime; //报警时间
strcpy( MsgComment.szText,GetPropChar(lpszPictureName,"R1","SelText")); //将
要写入的注释
strcpy( MsgComment.szUser, GetTagChar("@CurrentUser")); //操作人员
MSRTSetComment( I_svID, &MsgComment, pError ); //写入注释
```

脚本中要引用“msrtapi.h”和“apdefap.h”头文件，这需要安装 ODK 之后才支持。另外，脚本 GetPropChar(lpszPictureName,"R1","SelText")是获取组合框“R1”所选择的文本。

>

6 <运行结果>

<在 WinCC 启动列表中选择“报警记录运行系统”和“图形运行系统”，并激活 WinCC 运行系统。如图 13 所示。



图 13

项目运行后的初始状态如图 14 所示。



图 14

通过设置报警变量“alarm1”来触发报警，如图 15 所示。可以看到此时的报警消息文字中并不包含报警原因。



图 15

6 <运行结果>

接下来在报警视图中选择一条报警，可以看到所选报警的时间（包括毫秒）和编号信息已经被读出。如图 16 所示。接下来，选择报警原因，如图中②所示，最后点击“写入原因”按钮。



图 16

可以看到报警原因“电气故障”已经写入到所选报警的消息文本中，如图 17 所示。注意，需要先切换到“消息列表”视图，再切换到“短期归档列表”视图才能刷新出已经写入的报警原因。

日期	时间	编号	消息文本
2020/11/21	13:49:36.179	1	设备1故障，原因:
2020/11/21	13:49:36.179	2	设备2故障，原因: 电气故障
2020/11/21	13:49:36.179	3	设备3故障，原因:

图 17

同样，可以为其它的报警写入报警原因，如图 18 所示。

日期	时间	编号	消息文本
2020/11/21	13:49:36.179	1	设备1故障，原因: 机械故障
2020/11/21	13:49:36.179	2	设备2故障，原因: 电气故障
2020/11/21	13:49:36.179	3	设备3故障，原因: 工艺原因

图 18

>