

**SIEMENS**

常问问题 • 02/2015

# SIMOTION 编译器如何设置

SIMOTION, Compiler

<https://support.industry.siemens.com/cs/cn/zh/view/109477021>

---

# 目录

<b>1</b>	<b>概述.....</b>	<b>3</b>
1.1	简介.....	3
1.2	编译器的设定位置.....	3
<b>2</b>	<b>设置说明.....</b>	<b>5</b>
2.1	报警显示设置.....	5
2.2	选择连接和预处理器的设置.....	6
2.3	其他全局设置参数.....	7
2.4	本地配置.....	9

# 1 概述

## 1.1 简介

在 SIMOTION 的编程软件 SCOUT 中，可以设置程序编译器的功能和选项。这些选项对于程序的正确编译以及程序的执行有很重要的意义。对于编译器的设置可以使用全局设置或者本地设置。全局设置是针对此项目中所有编程语言的代码进行设置控制，而本地设置则为针对独立的程序单元进行编译器的设置。

## 1.2 编译器的设定位置

SIMOTION 的编译器全局设定位于软件的菜单中，即选择 SCOUT 软件菜单的 Options > Settings，选择 Compiler 标签对编译器选项进行设置。

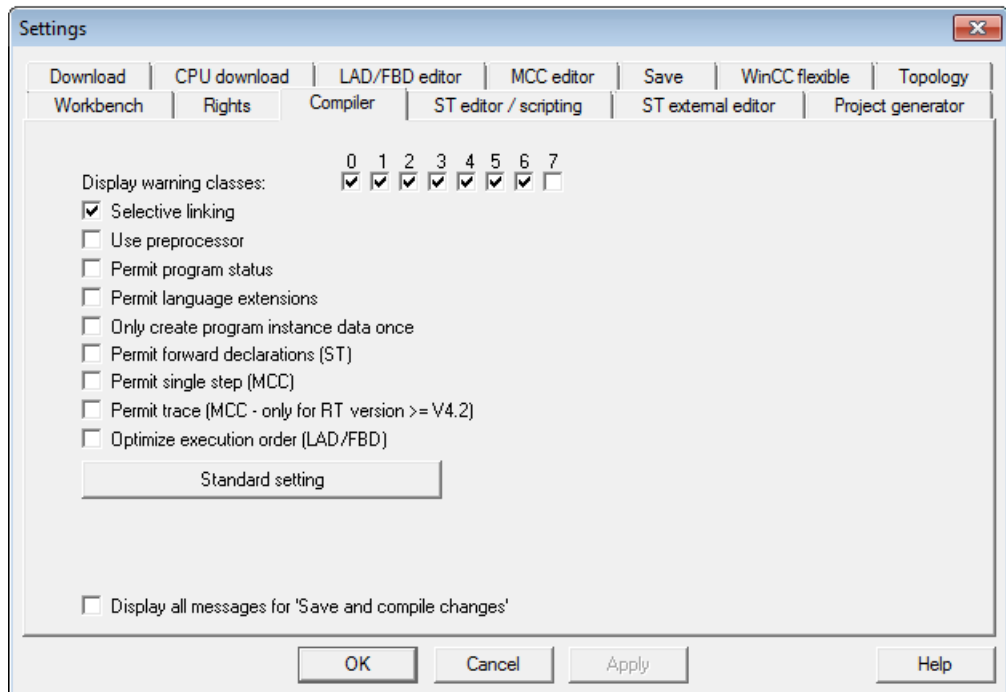


图 1-1 SCOUT 软件的编译全局设置

在单独的每一个程序单元创建时或者在程序单元上点击右键选择“Properties...”均可以进行编译器的本地设置，如图 1-2 所示：

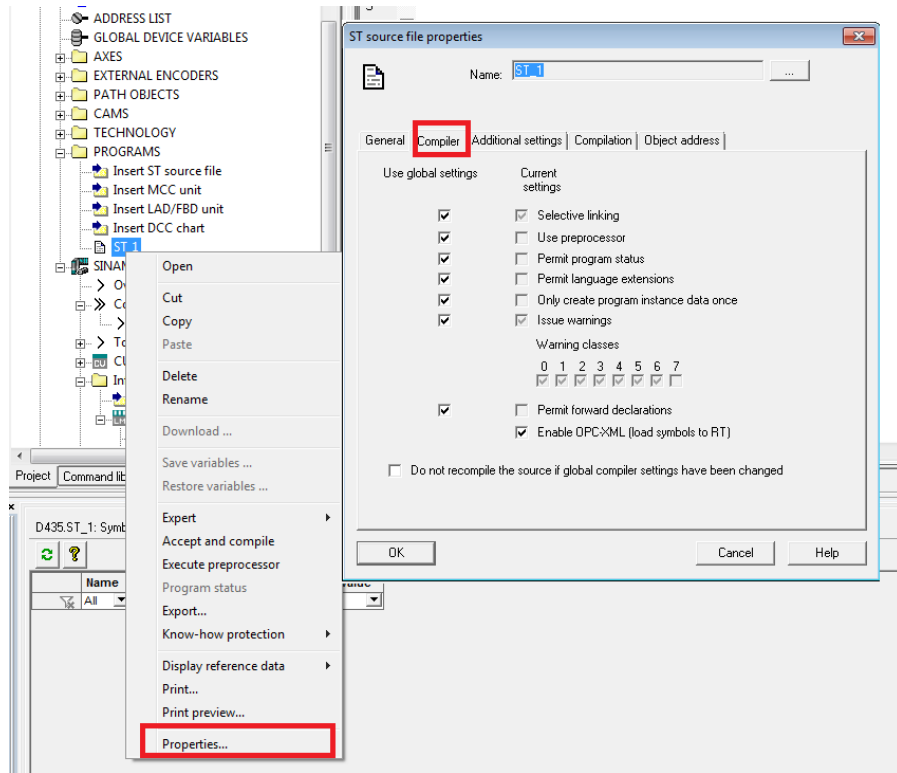


图 1-2 SCOUT 程序单元的编译本地设置

## 2 设置说明

### 2.1 报警显示设置

全局编译器的报警显示设置的选项说明如下：

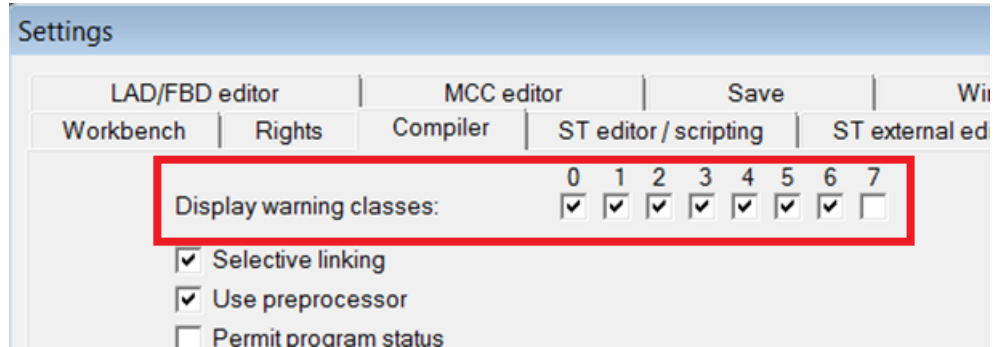


图 2-1 报警等级设定

报警等级 **Warning classes**，当把对应的报警等级进行勾选时，则编译器输出此等级的报警信息。例如显示或者不显示当前代码存在没有使用的变量。

等级 0 为程序单元中存在没有使用的变量或者代码提示。

等级 1 为隐藏的识别符提示。比如 `case` 指令中使用多个相同的识别符，仅有第一个有效，其他的识别符无效。

等级 2 为类型转换提示。

等级 3 为编译器选项设置提示。比如当前语法不支持，需要进行编译器的设置。

等级 4 为信号灯（潜在的故障功能）提示，比如报警 16200，提示需要全局变量用于信号灯功能。

等级 5 为 `alarm` 功能提示。比如调用了 `alarm s` 功能，其消息的配置无法进行检测。

等级 6 为使用库中的结构变量提示。比如库中定义了全局变量，此库在反复调用时会有变量冲突。

等级 7 为使用了预处理器的提示。关于预处理器在下文中有介绍。

## 2.2 选择连接和预处理器的设置

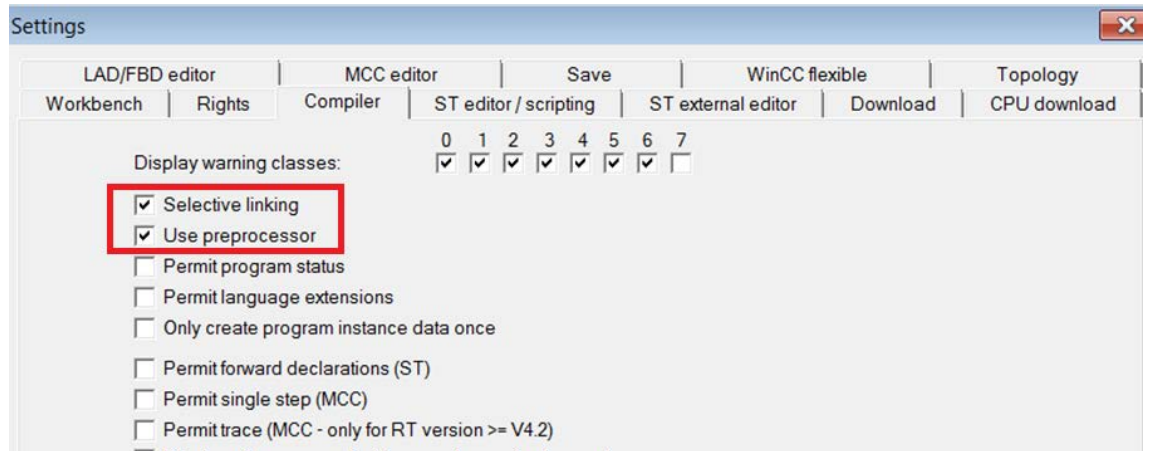


图 2-2 选择性链接和预处理器

**选择性链接 Selective linking:** 如果选择此选项，则生成可执行程序代码时删除未使用的代码。未选中: 生成可执行程序代码时保留未使用的代码。

**使用预处理器 Use preprocessor:** 在 ST 代码中可以在属性中的“Additional settings”中添加特殊的识别符。例如: `My_define=g_var` 意味着 `My_define` 和 `g_var` 同含义:

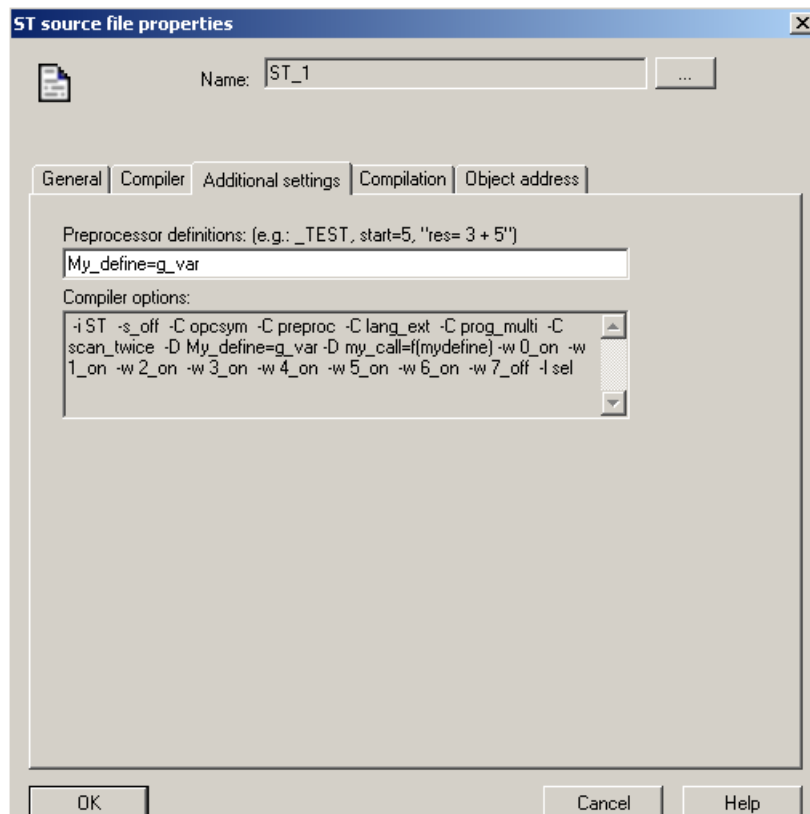


图 2-3 预编译设置

则可以编写如下代码:

```
INTERFACE  
  
PROGRAM A;  
VAR_GLOBAL  
  
    G_VAR :INT;  
  
    END_VAR  
END_INTERFACE  
  
IMPLEMENTATION  
  
PROGRAM A  
  
    My_define:=4;  
  
    END_PROGRAM  
END_IMPLEMENTATION
```

图 2-4 预编译变量的设置

## 2.3 其他全局设置参数

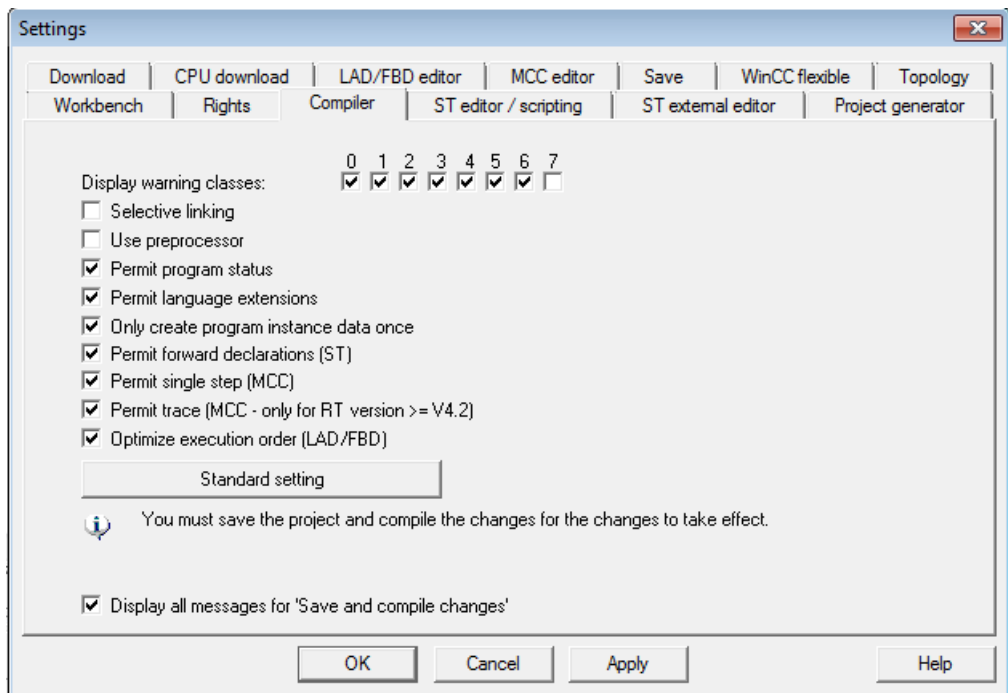


图 2-5 其他全局设置参数

- 使能程序状态监控 **Permit program status**: 如果选择则生成附加的监控程序代码，从而允许在线监控程序的状态和变量数值，否则无法监控程序状态和数值，此选项对于调试非常重要。
- 允许语言扩展 **Permit language extensions**: 选中: 允许与 IEC 61131-3 不一致的语言。未选中: 只运行符合 IEC 61131-3 的编程语言。

如果允许扩展，则可以使用如下的三种功能：

- 1.使用位方式访问变量，如

---

```
// Only with compiler option "Permit language extensions"
```

```
FUNCTION f : VOID  
  VAR CONSTANT  
    BIT_7 : INT := 7;  
  END_VAR  
  VAR  
    dw : DWORD;  
    b: BOOL;  
  END_VAR  
  b := dw.BIT_7; // Access to bit 7  
  b := dw.3;    // Access to bit 3  
END_FUNCTION
```

2.在程序块外部访问程序块的输入变量。

3.使用一个程序调用另外一个程序，例如：

```
INTERFACE  
  PROGRAM A;  
END_INTERFACE
```

```
IMPLEMENTATION  
  PROGRAM A  
  B(); //B IS OTHER PROGRAM  
  END_PROGRAM  
END_IMPLEMENTATION
```

● 只生成程序实例一次 **Only create program instance data once**: 选中: 程序的本地变量只在程序的用户存储区中存储一次。未选中: 程序的本地变量存储在相应的任务的用户存储器中。注意: 在使用了允许语言扩展 **Permit language extensions** 的第 3 种功能 “使用一个程序调用另外一个程序” 时, 必须勾选此选项 “Only create program instance data once”。

● 允许 MCC 的单步调试 **Permit single step(MCC)**: 选中时产生额外的代码从而可以使能程序的单步监控。

● 允许 ST 语言的向前声明 **Permit forward declarations (ST)**: 允许 ST 语言编程时的调用和声明的顺序颠倒。

● 允许 MCC 的跟踪功能 **Permit Trace (MCC)**: 允许 MCC 语言编译时生成附加代码用于程序分支的执行监控。

● 显示 “保存并编译修改” 的所有消息 **Display all messages with “save and compile changes”**: 控制软件在使用 “保存并编译修改” 功能时输出的错误记录的范围, 如果选择此功能, 则显示编译的详细信息, 例如每个 ST 语言的编译信息被显示出来。

● 优化执行次序 **Optimize execution order (LAD/FBD)**: 对于梯形图和功能块编程的语言执行进行优化, 此设置对于多个并行分支的执行的先后顺序有影响。



---

## 2.4 本地配置

本地配置的功能和全局配置一致，但是增加了选项“Use global settings”（图 1-2）。

**Use global settings:** 不选中: 本地设置起作用，全局设置被忽略。选中: 全局设置起作用，本地设置的复选框用灰色标记。