



Products: R&S CRTU-G

# Migration to Visual Studio .NET 2005 of proprietary Test Cases on the GSM Protocol Tester R&S<sup>®</sup> CRTU-G

## Application Note

Due to the fact that Microsoft does not maintain the Visual Studio .NET 2003 anymore Rohde & Schwarz has to use the successor compiler product Visual Studio .NET 2005 for building the test case packages.

All CRTU-G customers having proprietary test case packages, developed based on Visual Studio .NET 2003, can migrate them to Visual Studio .NET 2005.

This is required when the proprietary test case packages are going to be enhanced on a platform with installed Visual Studio .NET 2005.



## **Contents**

1	Overview .....	3
2	Software Requirements.....	3
3	Procedure of Migration.....	3
	Convert Solution and Project Files.....	3
	Rename Solution Files .....	4
	Rename Project Files.....	5
	Edit * _VS05.SLN Files .....	5
	Avoid 'C4966 Warnings' .....	6
	Rework the Code .....	6
	Check if your modifications are backward compatible.....	6
	Solutions for fixing frequent Compiler Errors .....	7
4	Additional Information .....	10
5	Ordering Information .....	10

## 1 Overview

Due to the fact that Microsoft does not maintain the Visual Studio .NET 2003 anymore Rohde & Schwarz has to use the successor compiler product Visual Studio .NET 2005 for building the test case packages.

All CRTU-G customers having proprietary test case packages, developed based on Visual Studio .NET 2003, can migrate them to Visual Studio .NET 2005. This is required when the proprietary test case packages are going to be enhanced on a platform with installed Visual Studio .NET 2005.

The GSM Protocol Tester R&S® CRTU-G is abbreviated as CRTU-G for the remainder of this Application Note.

## 2 Software Requirements

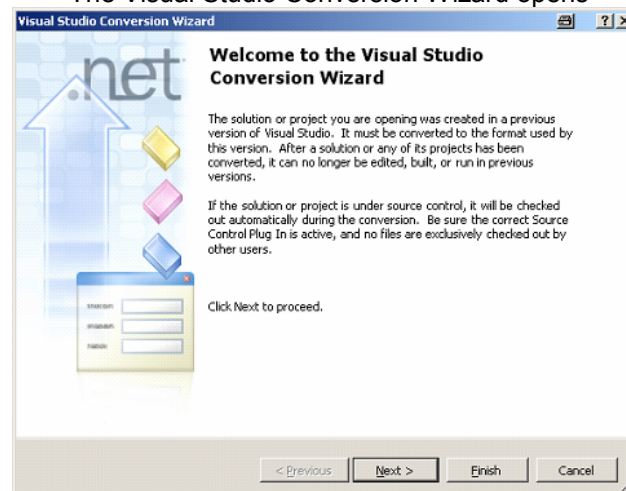
<b>Compiler</b>	Microsoft Visual Studio .NET 2005
<b>TC Package</b>	Proprietary GSM Test Case Package based on .NET 2003

## 3 Procedure of Migration

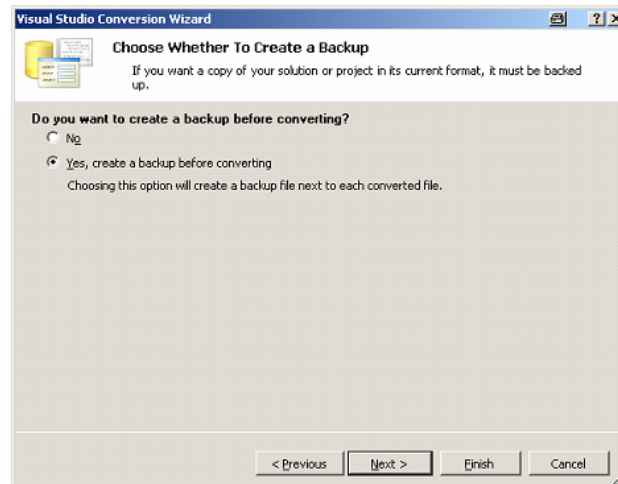
This procedure describes how to create a second set of solution and project files, which can be built with Visual Studio .NET 2005.

### Convert Solution and Project Files

- Open the existing solution file (\*.SLN) using Visual Studio .NET 2005. example: open CRTUGC31.sln
- The Visual Studio Conversion Wizard opens



- Press the Next button
- Select 'Yes, create a backup before converting' and press the Next button



## Rename Solution Files

**Note:** This procedure is only required when it is intended still using the old \*.SLN and \*.VCPROJ files (.NET 2003) in parallel to the .NET 2005 files.

- Add to the solution (\*.SLN) file name the suffix '\_VS05'  
example: CRTUGC31.sln → CRTUGC31\_VS05.sln

02g8502p.dat	1 KB	DAT File	14.05.2003 09:40	A
02gsm2p.dat	1 KB	DAT File	14.05.2003 09:41	A
02pcn2p.dat	1 KB	DAT File	14.05.2003 09:41	A
02pcs2p.dat	1 KB	DAT File	14.05.2003 09:41	A
CRTUGC31.ccscc	1 KB	CCSCC File	23.11.2005 15:04	A
CRTUGC31.sln	9 KB	Microsoft Visual Stu...	13.09.2006 16:46	A
initenv.bat	1 KB	MS-DOS Batch File	12.02.2003 13:55	A
MakeAll.bat	1 KB	MS-DOS Batch File	25.11.2003 16:38	A
CRTUGC31.sln.old	9 KB	OLD File	26.11.2004 12:27	A
UpgradeLog.XML	40 KB	XML Document	13.09.2006 16:46	A
_UpgradeReport_Files		File Folder	13.09.2006 16:46	
CRTUGC31.ncb	11 KB	VC++ Intellisense D...	13.09.2006 16:46	A

- Remove the extension 'old' from the file \*.SLN.OLD  
example: CRTUGC31.sln.old → CRTUGC31.sln

02g8502p.dat	1 KB	DAT File	14.05.2003 09:40	A
02gsm2p.dat	1 KB	DAT File	14.05.2003 09:41	A
02pcn2p.dat	1 KB	DAT File	14.05.2003 09:41	A
02pcs2p.dat	1 KB	DAT File	14.05.2003 09:41	A
CRTUGC31.ccscc	1 KB	CCSCC File	23.11.2005 15:04	A
CRTUGC31_vs05.sln	9 KB	Microsoft Visual Stu...	13.09.2006 16:46	A
initenv.bat	1 KB	MS-DOS Batch File	12.02.2003 13:55	A
MakeAll.bat	1 KB	MS-DOS Batch File	25.11.2003 16:38	A
CRTUGC31.sln	9 KB	Microsoft Visual Stu...	26.11.2004 12:27	A
UpgradeLog.XML	40 KB	XML Document	13.09.2006 16:46	A
_UpgradeReport_Files		File Folder	13.09.2006 16:46	
CRTUGC31.ncb	11 KB	VC++ Intellisense D...	13.09.2006 16:46	A

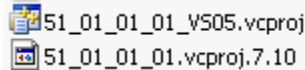
### Rename Project Files

**Note:** This procedure is only required when it is intended still using the old \*.SLN and \*.VCPROJ files (.NET 2003) in parallel to the .NET 2005 files.

- Add to each project (\*.VCPROJ) file name the suffix '\_VS05'  
example: 51\_01\_01\_01.vcproj → 51\_01\_01\_01\_VS05.vcproj



- Remove the extension 'old' from all files \*.VCPROJ.7.10.OLD  
example: 51\_01\_01\_01.vcproj.7.10.old → 51\_01\_01\_01.vcproj.7.10



**Note:** The project files of the used ApplComm integrated in this solution also have to be modified in the same way.

### Edit \*\_VS05.SLN Files

**Note:** This procedure is only required when it is intended still using the old \*.SLN and \*.VCPROJ files (.NET 2003) in parallel to the .NET 2005 files.

- Open the solution (\*.SLN) file with the suffix '\_VS05' with an editor (e.g. Notepad)
- Replace the old project references '\*.VCPROJ' by '\*\_VS05.VCPROJ'

example: '51\_01\_01\_01.vcproj' → '51\_01\_01\_01\_VS05.vcproj'

```
# Visual Studio 2005

Project("{8BC9CEB8-8B4A-11D0-8D11-00A0C91BC942}") = "51_01_01_01",
"..\\51\\01\\51_01_01_01.vcproj", "{5420739A-EE2C-421A-9977-8E92A1D0F90C}"

...

# Visual Studio 2005

Project("{8BC9CEB8-8B4A-11D0-8D11-00A0C91BC942}") = "51_01_01_01",
"..\\51\\01\\51_01_01_01_VS05.vcproj", "{5420739A-EE2C-421A-9977-8E92A1D0F90C}"

...
```

### Avoid 'C4966 Warnings'

When compiling with Visual Studio .NET 2005 a lot of the following warnings appear:

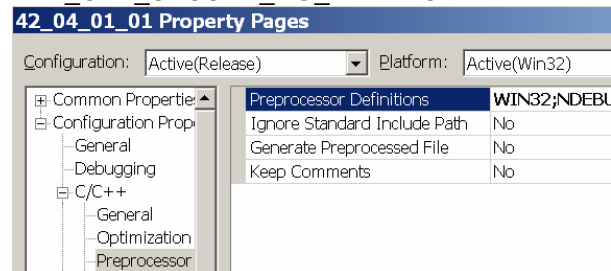
#### C4996: 'function': was declared deprecated

The code 'C4996' means, that the compiler encountered a function marked with deprecated. A deprecated function marked may no longer be supported in a future release. Visual Studio .NET 2005 offers a pre-compiler setting to disable this warning. Therefore you have to add the following entry to your preprocessor definitions.

`_CRT_SECURE_NO_DEPRECATED`

example:

- Open Visual Studio .NET 2005
- Open the solution file of CRTUGC74 (CRTUGC74\_VS05.sln)
- Select the project 42\_04\_01\_01 in the Solution Explorer
- Right mouse-click and select "Properties"
- Select 'Configuration Properties → C/C++ → Preprocessor → Preprocessor Definitions' and enter '`_CRT_SECURE_NO_DEPRECATED`'



### Rework the Code

- Compile the test case package with the Visual Studio .NET 2005 solution file

example: CRTUGC31\_VS05.sln

- Correct the errors

**Note:** *The chapter "Solutions for fixing frequent Compiler Errors" proposes solutions for frequent compiler errors.*

### Check if your modifications are backward compatible

**Note:** *This procedure is only required when it is intended still using the old \*.SLN and \*.VCPROJ files (.NET 2003) in parallel to the .NET 2005 files.*

- Compile the test case package using the old solution file (Visual Studio .NET 2003 should open)

**Note:** *Every time you use another compiler, you have to rebuild the whole solution, because otherwise linking of the libraries will not work.*

### **Solutions for fixing frequent Compiler Errors**

#### **C2668: 'function' : ambiguous call to overloaded function**

The specified overloaded function call could not be resolved. You may want to explicitly cast one or more of the actual parameters.

You can also get this error through template use. If, in the same class, you have a regular member function and a templated member function with the same signature, the templated one must come first. This is a limitation of the current implementation of Visual C++.

Example: CRTUGC31 function pow()used:

```
int main()
{
...
NonDrxTimerValue = (tLongword)pow(2,
AppFnCtl.PagingInfo.DRXParameter.NonDrxTimer - 1)// C2668
...
NonDrxTimerValue = (tLongword)pow((long double)2,
(long double)AppFnCtl.PagingInfo.DRXParameter.NonDrxTimer -
1)// OK
...
}
```

Note 1: tLongword = unsigned long → (long double) for best precision

Note 2: You have to cast both parameters to be able to compile the source with Visual Studio .NET 2003 and Visual Studio .NET 2005.

#### **C4430: missing type specifier - int assumed. Note: C++ does not support default-int**

This error can be generated as a result of compiler conformance work that was done for Visual C++ 2005: all declarations must now explicitly specify the type; int is no longer assumed.

example: CRTKLU1 (SimAuto.c)

The line

```
static SimuUsed = LEGACY_MODE; // Legacy SIM
```

changed to

```
static int SimuUsed = LEGACY_MODE; // Legacy SIM
```

Note: The value is an enum so it should be better to select the enum type for this variable. But it is an unnamed enum so it is changed to int.

### **C2065: 'identifier' : undeclared identifier**

A variable's type must be specified in a declaration before it can be used. The parameters that a function uses must be specified in a declaration, or prototype, before the function can be used. The most common reason for this error is:

Declaring an iterator variable in a **for** loop, and then trying to use that iterator variable outside the scope of the **for** loop.

example: CRTUGC08, Project 26\_6\_3\_7, File 637.c

```
...
for(int i=0; i<8; i++)
{
...
}
...
for(int i=0; i<8; i++)
{
...
}
...
for(i=0; i<8; i++) <- error C2065
```

### **Solution:**

```
...
int i;
for(i=0; i<8; i++)
{
...
}
...
for(i=0; i<8; i++) <- removed the "int" because of optimization
{
...
}
...
for(i=0; i<8; i++) <- no error now
```



### **C2050: switch expression not integral**

The **switch** expression evaluates to a non integer value. To resolve the error, use only integral values in switch statements.

This is a resulting error from C2065 (undeclared identifier). Because of the use of a variable that is outside its scope the switch can't evaluate the value of the variable.

example: CRTUGC08, Project 26\_6\_3\_7, File 637.c

```
...
for(int i=0; i<8; i++)
{
...
}
...
switch(i)    <- error C2050
```

Solution: Refer to the description for error C2065. If you remove the reason for C2065 then C2050 should also disappear.

### **C2228: left of '.identifier' must have class/struct/union**

The operand to the left of the period (.) is not a class, structure, or union.

This is a resulting error from C2065 (undeclared identifier). Because of the use of a variable that is outside its scope the same variable used as an index value can't be evaluated.

example: CRTUGC08, Project 26\_6\_3\_7, File 637.c

```
...
for(int i=0; i<8; i++)
{
...
}
...
BcchCarrier[i].SysInfoType6    <- error C2228
```

Solution: Refer to the description for error C2065. If you remove the reason for C2065 then C2050 should also disappear.

## **4 Additional Information**

Please send any comments or suggestions about this application note to [TM-Applications@rsd.rohde-schwarz.com](mailto:TM-Applications@rsd.rohde-schwarz.com).

## **5 Ordering Information**

**GSM Protocol Tester**  
CRTU-G

1140.0009.02

For additional information about GSM protocol testing, see the Protocol Testing Web area on the Rohde & Schwarz GLORIS website <https://gloris.rohde-schwarz.com>.



ROHDE & SCHWARZ GmbH & Co. KG · Mühlendorfstraße 15 · D-81671 München · Postfach 80 14 69 · D-81614 München ·  
Tel (089) 4129 -0 · Fax (089) 4129 - 13777 · Internet: <http://www.rohde-schwarz.com>

*This application note and the supplied programs may only be used subject to the conditions of use set forth in the download area of the Rohde & Schwarz website.*