



SIEMENS

Ingenuity for life



Industry Online Support

Home



## 如何在 C# 应用程序中读取 WinCC 变量归档数据

WinCC / V7.5 / C# tag logging

<https://support.industry.siemens.com/cs/ww/en/view/109766732>

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](http://www.siemens.com/terms_of_use)) apply.

## 安全性信息

**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>。

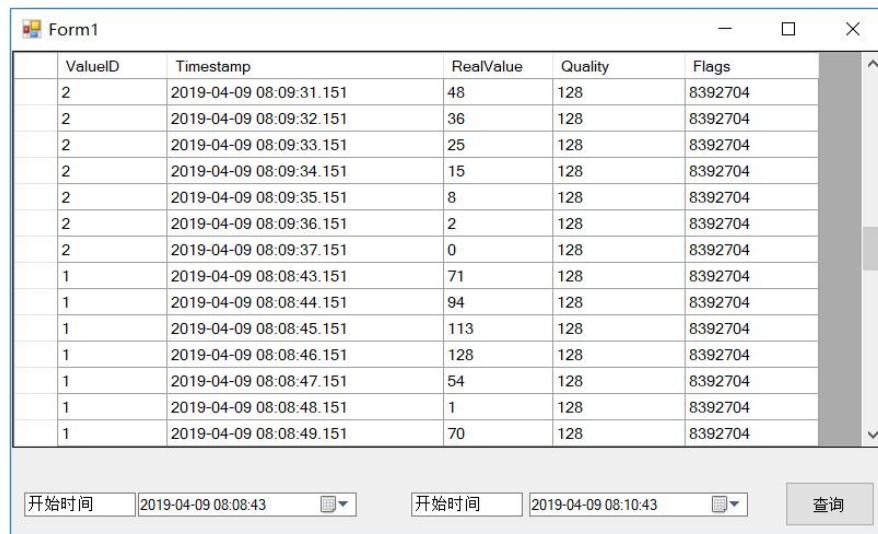
# 目录

1	<概述> .....	4
1.1	<功能说明> .....	4
1.2	<Connectivity Pack 介绍> .....	4
1.3	<软件版本> .....	5
2	<访问本地计算机的 WinCC 归档数据> .....	6
2.1	<WinCC 组态> .....	6
2.2	<C#应用程序编写> .....	8
3	<访问远程计算机的 WinCC 归档数据> .....	15

# 1 <概述>

## 1.1 <功能说明>

<编写一个 C# 应用程序，通过 WinCC 的 OLE DB 接口读取设定时间范围内的 WinCC 变量归档数据，效果如图 1 所示：>



The screenshot shows a Windows application window titled "Form1". Inside, there is a data grid table with columns: ValueID, Timestamp, RealValue, Quality, and Flags. The data grid contains 15 rows of data. Below the data grid are two date/time input fields: "开始时间" (Start Time) with the value "2019-04-09 08:08:43" and "结束时间" (End Time) with the value "2019-04-09 08:10:43". To the right of these fields is a "查询" (Query) button. The application has a standard Windows-style title bar and a scroll bar on the right side of the data grid.

ValueID	Timestamp	RealValue	Quality	Flags
2	2019-04-09 08:09:31.151	48	128	8392704
2	2019-04-09 08:09:32.151	36	128	8392704
2	2019-04-09 08:09:33.151	25	128	8392704
2	2019-04-09 08:09:34.151	15	128	8392704
2	2019-04-09 08:09:35.151	8	128	8392704
2	2019-04-09 08:09:36.151	2	128	8392704
2	2019-04-09 08:09:37.151	0	128	8392704
1	2019-04-09 08:08:43.151	71	128	8392704
1	2019-04-09 08:08:44.151	94	128	8392704
1	2019-04-09 08:08:45.151	113	128	8392704
1	2019-04-09 08:08:46.151	128	128	8392704
1	2019-04-09 08:08:47.151	54	128	8392704
1	2019-04-09 08:08:48.151	1	128	8392704
1	2019-04-09 08:08:49.151	70	128	8392704

图 1

## 1.2 <Connectivity Pack 介绍>

<WinCC 变量归档数据是以压缩的形式存储在数据库中，第三方开发的应用程序需要通过 WinCC/Connectivity Pack 提供的 OLE-DB 接口才能够解压并读取这些数据。

### 1.2.1 连接字符串

"Provider=WinCCOLEDBProvider.1; Catalog= \*\*\*; Data Source= \*\*\*;"

其中“Catalog”是 WinCC 运行数据库的名称，它是由项目名称以及创建日期组成的。并且当修改项目名称或在其它计算机上打开此项目时，这个数据库名称会发生变化。WinCC 提供系统变量“@DatasourceNameRT”保存当前的 WinCC 运行数据库名称。因此建议在脚本中使用这个变量来获得当前项目的 Catalog

“Data Source”是服务器名称。

当应用程序和 WinCC 在同一台计算机时，Data Source 写法为“.\WinCC”或者“<WinCC 计算机名称>\WinCC”。

当应用程序和 WinCC 不在同一台计算机时，Data Source 写法只能为“<WinCC 计算机名称>\WinCC”。

### 1.2.2 查询语句

## 1 <概述>

"TAG:R,<ValueID or  
ValueName>,<TimeBegin>,<TimeEnd>[,<SQL\_clause>][,<TimeStep>]"

其中：

ValueName：格式为“ArchiveName\Value\_Name”，可以使用多个名称。

TimeBegin,TimeEnd：时间范围，格式“yyy-MM-dd HH:mm:ss.msc”。

SQL\_Clause：SQL 语法中的过滤标准。

### 1.2.3 查询结果

不同版本 WinCC 返回的数据记录集结果可能会有所不同。表 1 是 WinCC V7.5  
OLEDB 返回的记录集。

表 1 WinCC V7.5 返回的记录集

域名称	类型	注释
ValueID	整型 (4 字节) 或 整型 (8 字节)	值的唯一标识。 长度取决于查询的类型。
TimeStamp	日期时间	时间戳
RealValue	实型 (8 字节)	变量值
Quality	整型 (4 字节)	值的质量代码 (例如“好”或“劣”)
Flags	整型 (4 字节)	内部控制参数

注意，WinCC 的归档数据是使用 UTC ( 协调世界时 ) 时间保存的。>

## 1.3 <软件版本>

< WinCC 版本: WinCC V7.5 Update2

操作系统: Windows 10

Visual Studio: Visual Studio 2019>

## 2 <访问本地计算机的 WinCC 归档数据>

### 2.1 <WinCC 组态>

<在 WinCC 中组态归档变量，这些变量的归档数据将显示在 C# 应用程序中。具体步骤如下：

第一步：新建项目并创建变量

根据实际情况创建变量。本例中，创建两个内部变量“tag1”和“tag2”。

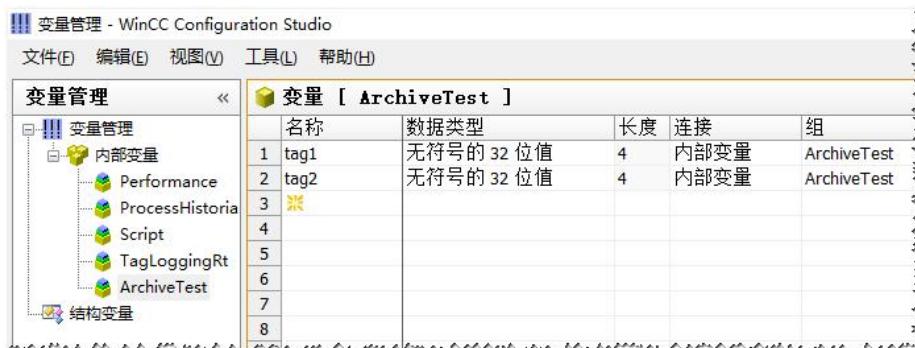


图 2

第二步：组态变量归档

打开 WinCC 变量记录编辑器，创建过程值归档，并添加需要归档的变量。

本例中，创建过程值归档“Archive”，并在“Archive”下添加变量“tag1”和“tag2”，归档周期都设置为 1 秒。如图 3 所示：



图 3

第三步：组态画面

在 WinCC 中创建画面并显示归档变量的数据。

第四步：激活 WinCC 项目并启动变量模拟

在启动列表中选择“变量记录运行系统”，并激活 WinCC 项目。

## 2 <访问本地计算机的 WinCC 归档数据>

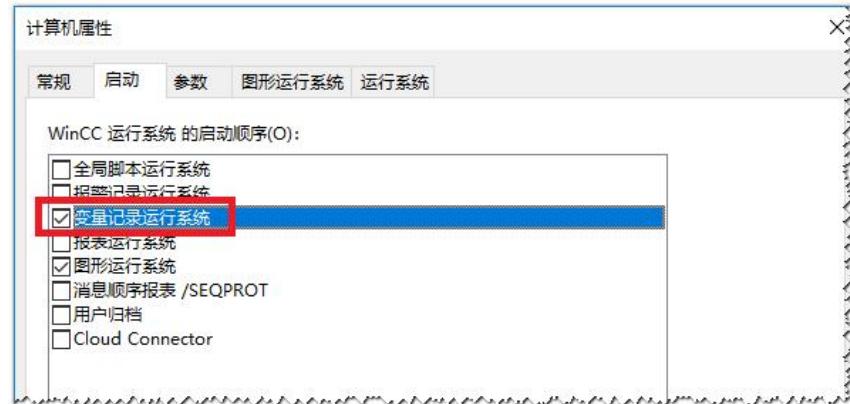


图 4

启动变量模拟，并按图 5 所示配置模拟属性并启动变量模拟。

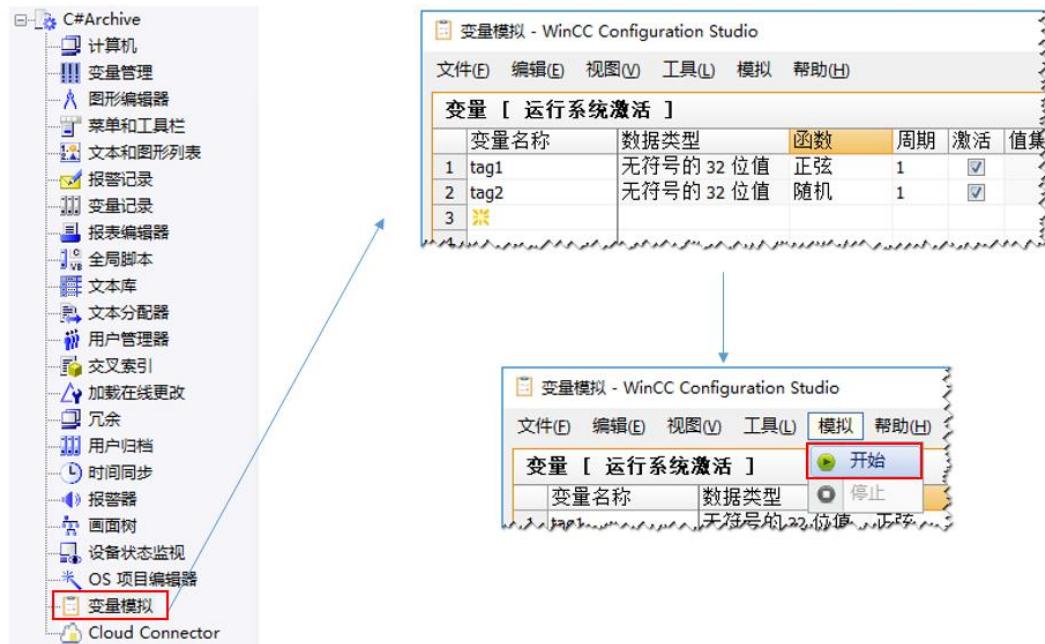


图 5

最终的 WinCC 的运行效果如图 6 所示：

## 2 <访问本地计算机的 WinCC 归档数据>

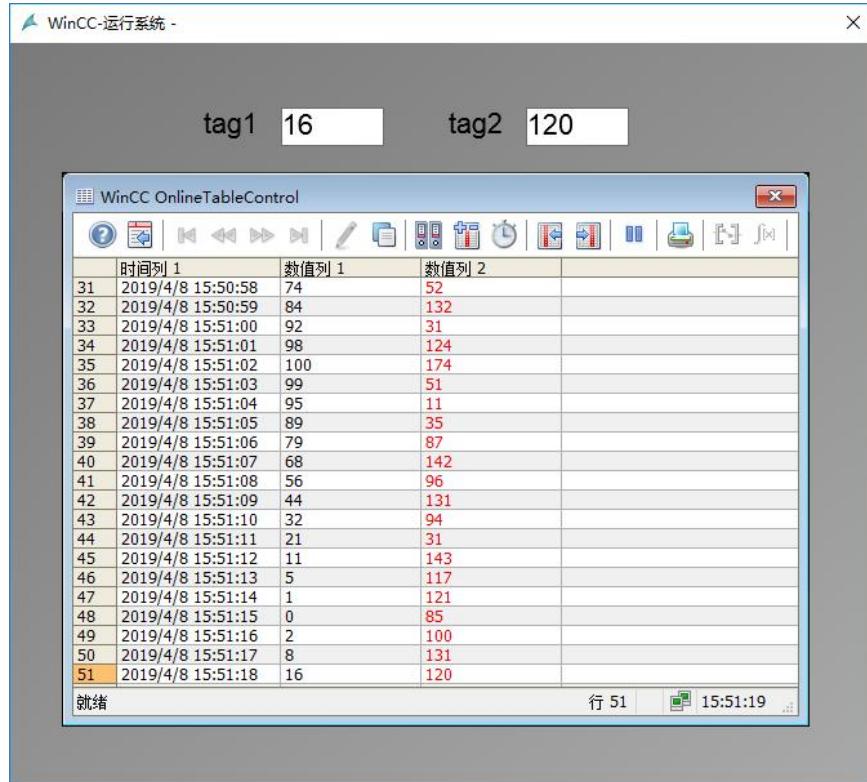


图 6

>

## 2.2 <C#应用程序编写>

<接下来使用 Visual Studio 2019 编写应用程序，读取 WinCC 的归档数据。

第一步：创建项目

在 Visual Studio 2019 中新建项目，项目模板选择 C# 的“Windows 窗体应用”。如图 7 所示：

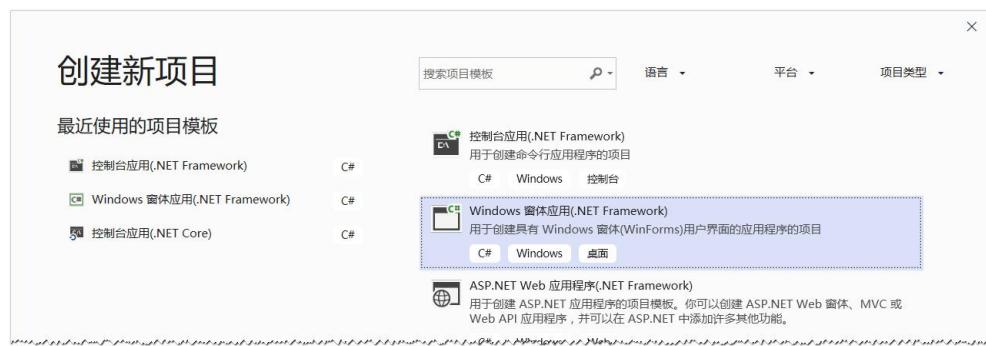


图 7

第二步：界面组态

在窗体中添加一个 DataGridView 控件用来显示查询到的数据，添加两个 DateTimePicker 控件用来设置查询开始和结束时间，以及一个查询按钮。

## 2 <访问本地计算机的 WinCC 归档数据>

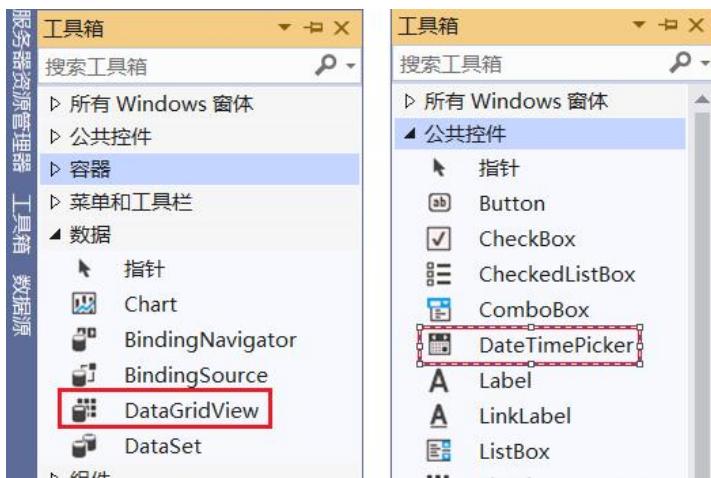


图 8

最终界面的结构如图 9 所示：

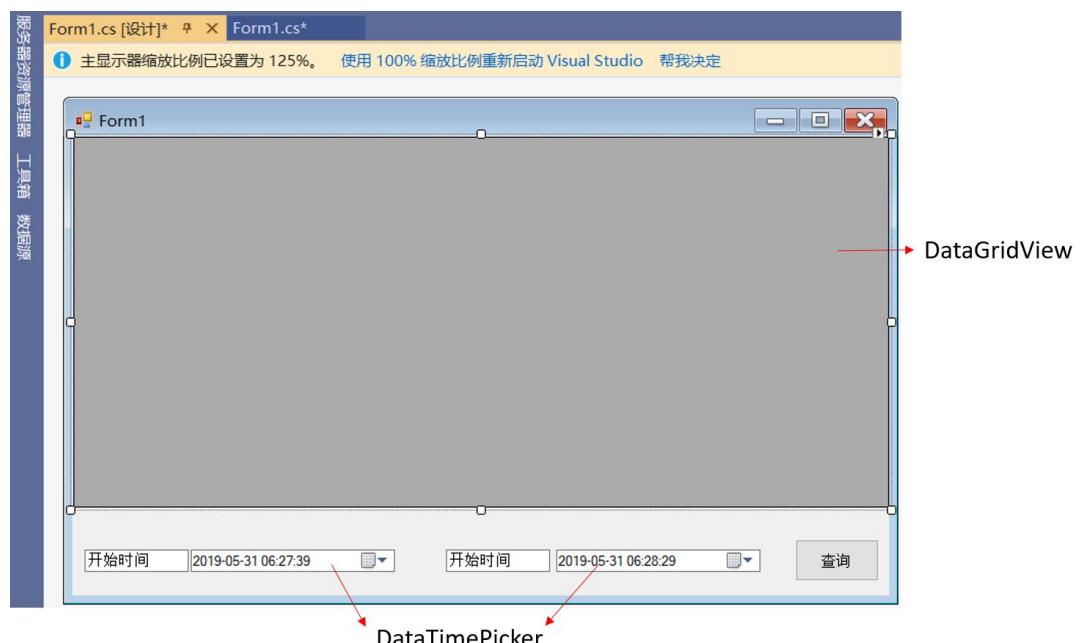


图 9

设置两个 DateTimePicker 控件的属性。

( 1 ) 开始时间 DateTimePicker 的属性：

对象名称：“DT\_begain”。

Format 属性：“Custom”。

CustomFormat 属性：“yyyy-MM-dd HH:mm:ss”。

( 2 ) 结束时间 DateTimePicker 的属性：

对象名称：“DT\_end”。

“Format”属性：“Custom”。

## 2 <访问本地计算机的 WinCC 归档数据>

“CustomFormat”属性：“yyyy-MM-dd HH:mm:ss”。

### 第三步：编写查询代码

#### (1) 连接字符串

查询脚本中需要用到 OLEDB 的连接字符串和查询语句。由于连接字符串中的 Catalog (WinCC 的运行数据) 名称当修改项目名称或在其它计算机上打开原项目时, Catalog 会发生变化。因此建议使用 WinCC 系统变量 “@DatasourceNameRT” 获得当前的 Catalog。

如果要在 C# 中读取 WinCC 变量的值, 需要引用“WinCC HMI Runtime 1.0 Type Library”接口。首先在“引用”上右键选择“添加引用”, 如图 10 所示:

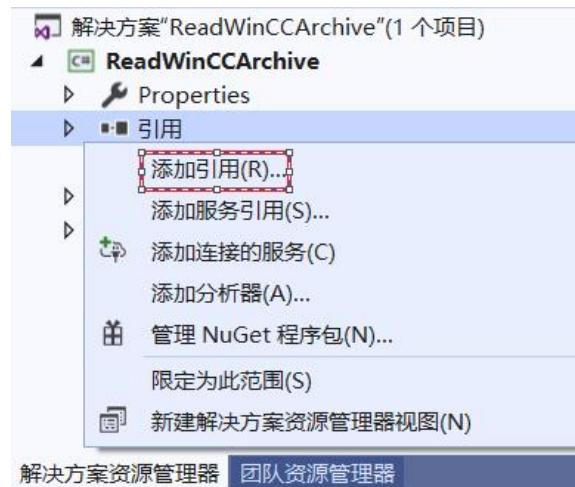


图 10

然后选择“COM”, 在列表中选择“WinCC HMI Runtime 1.0 Type Library”。如图 11 所示:

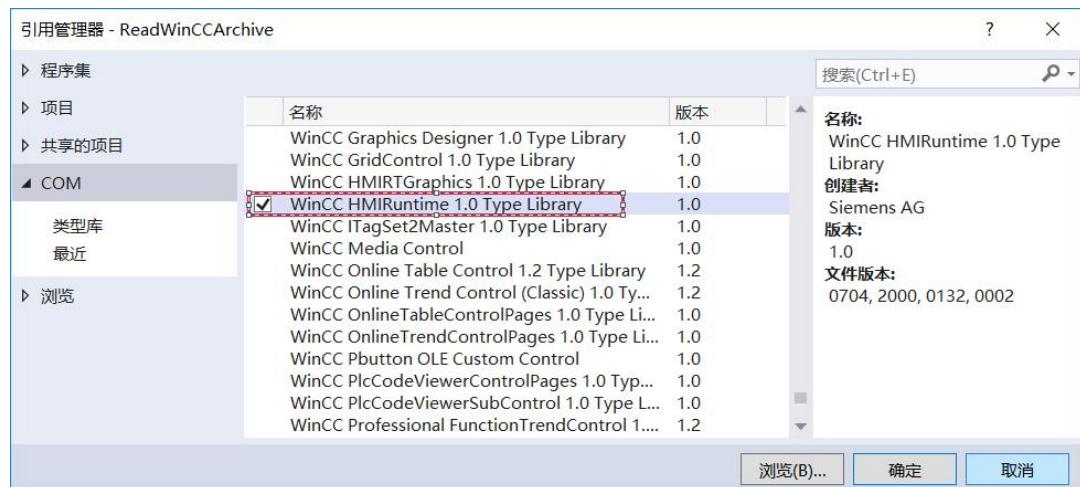


图 11

命名空间中使用“using CCHMIRUNTIME”引用 CCHMIRUNTIME。如图 12 所示:

## 2 <访问本地计算机的 WinCC 归档数据>

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.OleDb; //OLEDB
using CCHMIRUNTIME; //WinCC CCHMIRUNTIME
```

图 12

然后就可以使用 WinCC 系统变量 “@DatasourceNameRT” 和  
“@LocalMachineName” 来组成 WinCC OLEDB 的连接字符串。如图 13 所示：

```
//读取WinCC系统变量“@DatasourceNameRT”和“@LocalMachineName”
HMIRuntime HMIRT = new HMIRuntime();
IHMITag Catlog = HMIRT.Tags["@DatasourceNameRT"];
IHMITag PCname = HMIRT.Tags["@LocalMachineName"];
//连接字符串
string ConnectionStr = @"Provider=WinCCOLEDBProvider.1;Catalog=" + Catlog.Read() + ";Data Source=" + PCname.Read() + "\\\WINCC";
```

图 13

### ( 2 ) 查询语句

需要按照设定的时间范围查询 WinCC 的归档数据，因此 WinCC OLEDB 的查询语句如图 14 所示：

```
string DT_Begain = DT_begain.Value.ToString();
string DT_End = DT_end.Value.ToString();
//查询语句
comm.CommandText = @"TAG:R,('Archive\tag1','Archive\tag2')," + DT_Begain + "," + DT_End + """;
```

图 14

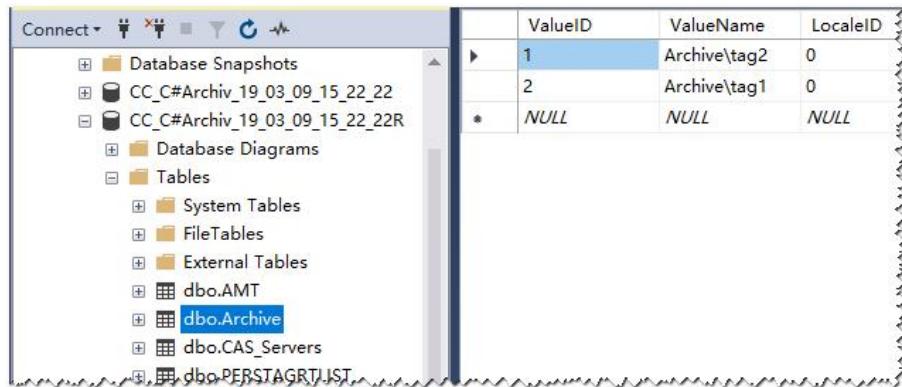
查询语句中可以使用 ValueName 也可以使用 Value ID，本例中使用的 ValueName。  
如果在本例中使用 ValueID，则查询语句如下：

"TAG:R,(1;2)," + DT\_Begain + "," + DT\_End + """;

其中归档变量'Archive\tag1'对应的 ValueID 为 1，归档变量'Archive\tag2'对应的 ValueID 为 2。

可以从 WinCC 运行数据库下的“Archive”数据表查询 ValueName 和 Value ID 的对应关系，如图 15 所示：

## 2 <访问本地计算机的 WinCC 归档数据>



	ValueID	ValueName	LocaleID
*	1	Archive\tag2	0
	2	Archive\tag1	0
	NULL	NULL	NULL

图 15

注意：

默认情况下，WinCC V7.5 没有安装 SQL Server Management Studio。

WinCC V7.5 有三张安装盘：WinCC V7.5 安装盘、SIMATIC NET V15 安装盘以及“Additional Content”安装盘，其中在“Additional Content”安装盘中就有 SQL Server Management Studio 的安装程序。

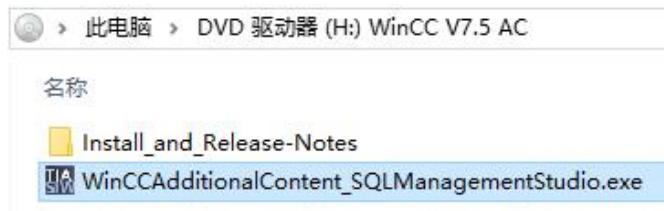


图 16

安装完成后，SQL Server Management Studio 会出现在 Microsoft SQL Server Tools 下。如图 17 所示：



图 17

### ( 3 ) 完整代码

命名空间部分，除了引用“CCHMIRUNTIME”外还要引用“System.Data.OleDb”。如图 18 所示：

## 2 <访问本地计算机的 WinCC 归档数据>

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.OleDb; //OLEDB
using CCHMIRUNTIME; //WinCC CCHMIRUNTIME
```

图 18

查询代码如图 19 所示：

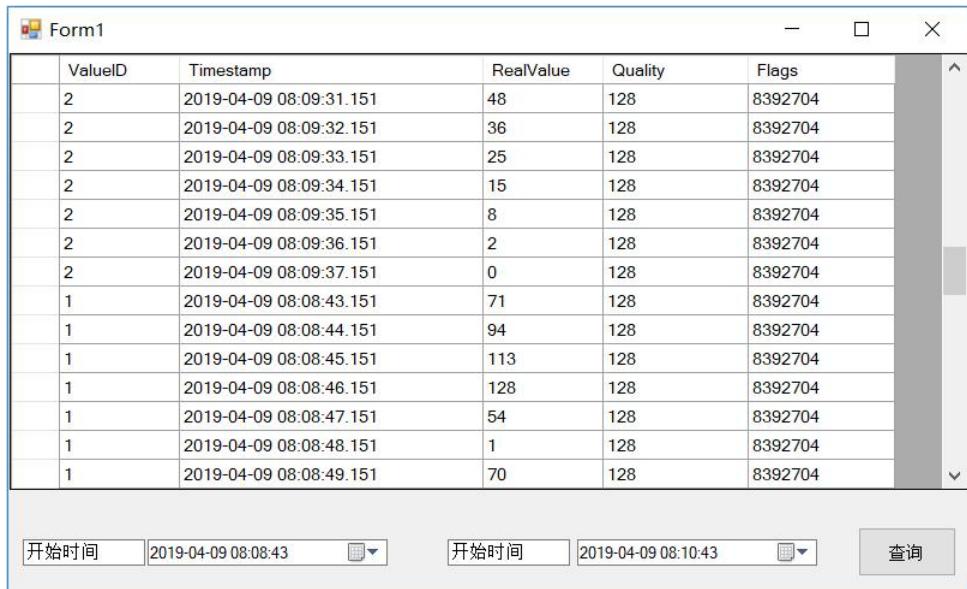
```
private void Button4_Click(object sender, EventArgs e)
{
    //读取WinCC系统变量"@DatasourceNameRT"和"@LocalMachineName"
    HMIRuntime HMIRT = new HMIRuntime();
    IHMITag Catalog = HMIRT.Tags["@DatasourceNameRT"];
    IHMITag PCname = HMIRT.Tags["@LocalMachineName"];
    //声明OLEDB对象
    DataSet ds = new DataSet();
    OleDbCommand comm = new OleDbCommand();
    OleDbDataAdapter da = new OleDbDataAdapter();
    //连接字符串
    string ConnectionStr = @"Provider=WinCCOLEDBProvider.1;Catalog=" + Catalog.Read() + ";Data Source=" + PCname.Read() + "\WINCC";
    OleDbConnection conn = new OleDbConnection(ConnectionStr);
    conn.Open();
    comm.Connection = conn;
    comm.CommandType = CommandType.Text;
    //查询语句
    string DT_Begin = DT_begin.Value.ToString();
    string DT_End = DT_end.Value.ToString();
    comm.CommandText = @"TAG:R,('arcv\tag1','arcv\tag2'),'" + DT_Begin + "','" + DT_End + "'";

    da.SelectCommand = comm;
    da.Fill(ds);
    this.dataGridView1.DataSource = ds.Tables[0].DefaultView; //表从起始行显示在dataGridView里
    this.dataGridView1.Columns["Timestamp"].DefaultCellStyle.Format = "yyyy-MM-dd HH:mm:ss.fff"; //设置时间格式
    conn.Close();
}
```

图 19

选择查询时间范围后，查询效果如图 20 所示：

## 2 <访问本地计算机的 WinCC 归档数据>



The screenshot shows a Windows application window titled 'Form1'. The main content is a data grid with the following columns: ValueID, Timestamp, RealValue, Quality, and Flags. The data grid contains 18 rows of data. Below the grid are two date/time input fields and a 'Query' button.

ValueID	Timestamp	RealValue	Quality	Flags
2	2019-04-09 08:09:31.151	48	128	8392704
2	2019-04-09 08:09:32.151	36	128	8392704
2	2019-04-09 08:09:33.151	25	128	8392704
2	2019-04-09 08:09:34.151	15	128	8392704
2	2019-04-09 08:09:35.151	8	128	8392704
2	2019-04-09 08:09:36.151	2	128	8392704
2	2019-04-09 08:09:37.151	0	128	8392704
1	2019-04-09 08:08:43.151	71	128	8392704
1	2019-04-09 08:08:44.151	94	128	8392704
1	2019-04-09 08:08:45.151	113	128	8392704
1	2019-04-09 08:08:46.151	128	128	8392704
1	2019-04-09 08:08:47.151	54	128	8392704
1	2019-04-09 08:08:48.151	1	128	8392704
1	2019-04-09 08:08:49.151	70	128	8392704

开始时间   开始时间

图 20

说明：查询结果中的时间是 UTC 时间，因此时间范围需要按照 UTC 时间来选择。  
如果需要使用本地时间，那就需要把查询时间范围转成 UTC 时间，并把查询结果中的时间从 UTC 时间转成本地时间显示。>

### 3 <访问远程计算机的 WinCC 归档数据的注意事项>

<如果 C#应用程序和 WinCC 项目不在同一台计算机，那么需要满足以下条件：

- 1、如果 C#应用程序所在计算机没有安装 WinCC 或者 DataMonitor 软件，则需要安装 Connectivity Pack 软件来提供 OLE-DB 接口。
- 2、C#应用程序所在计算机当前登陆的用户需要在 WinCC 项目所在计算机的用户管理中注册（用户名和密码都要相同），并且此用户需要具有 SQL Server 的管理权限。另外“WinCC HMIRuntime 1.0 Type Library”接口不支持读取远程 WinCC 的变量。这种情况下，连接字符串中的计算机名直接写 WinCC 项目所在计算机名即可。对于 WinCC 项目的 Catalog，可以通过以下两种方法来获取：

- ❖ 在 C#应用程序中编写 OPC 客户端脚本读取 WinCC 变量“@DatasourceNameRT”的值。
- ❖ 在 C#应用程序中遍历 SQL Server 中所有的数据库名称，从中选择以字母“CC”开头并以字母“R”结尾的数据库名称。

>