

CXH高山青竹的博客

<http://blog.sina.com.cn/cxh1988186> [订阅] [手机订阅][首页](#) [博文目录](#) [图片](#) [关于我](#)

个人资料

正文

字体大小: 大 中 小



CXH高山青竹

微博

加好友

发纸条

写留言

加关注



博客等级: **14**
博客积分: **636**
博客访问: **41,061**
关注人气: **9**
获赠金笔: **5**
赠出金笔: **1**
荣誉徽章:

JD.COM 京东

¥ 39.00

1/8

相关博文

他在卫生间强行进入我的身体，一
用户378897396

NBA地震！豪华四巨头兄弟篮球，詹
用户371812474

"台湾迎来最黑暗一天"：岛内爆
用户372546788

佩莱格里尼一语点中为何恒大的霸
用户379092213

VC ADO操作数据库（转） (2015-02-27 11:01:25)

转载 ▼

标签: ado 数据库 分类: MFC

来源: http://blog.163.com/cgq_i/blog/static/7287792320083300241390/

一、ADO简介

ADO(ActiveX Data Object)是Microsoft数据库应用程序开发的新接口，是建立在OLE DB之上的高层数据库访问技术，请不必为此担心，即使你对OLE DB，COM不了解也能轻松对付ADO，因为它非常简单易用，甚至比以往所接触的ODBC API、DAO、RDO都要容易使用，并不失灵活性。本文将详细地介绍在VC下如何使用ADO来进行数据库应用程序开发，并给出示例代码。

二、基本流程

- (1)引入ADO库定义文件、初始化COM库
- (2)用Connection对象连接数据库
- (3)利用建立好的连接，通过Connection对象的Execute方法、Command对象执行SQL命令，或利用Recordset对象取得结果记录集进行查询、处理。
- (4)使用完毕后关闭连接释放对象。

准备工作:

为了大家都能测试本文提供的例子，我们采用Access数据库，您也可以直接在我们提供的示例代码中找到这个test.mdb。

下面我们将详细介绍上述步骤并给出相关代码。

【1】用#import指令引入ADO类型库

我们在stdafx.h中加入如下语句：(stdafx.h这个文件哪里可以找到?你可以在FileView中的Header Files里找到) #import "c:\program files\common files\system\ado\msado15.dll" no_namespace rename ("EOF","adoEOF")

这一语句有何作用呢？其最终作用同我们熟悉的#include类似，编译的时候系统会为我们生成msado15.tlh,ado15.tli两个C++头文件来定义ADO库。

几点说明:

- (1) 您的环境中msado15.dll不一定在这个目录下，请按实际情况修改
- (2) 在编译的时候可能会出现如下警告，对此微软在MSDN中作了说明，并建议我们不要理会这个警告。msado15.tlh(405): warning C4146: unary minus operator applied to unsigned type, result still unsigned

【2】COM库的初始化

在MFC中可以用AfxOleInit()来初始化COM库，这项工作通常在CWinApp::InitInstance()的重载函数中完成，请看如下代码:

```
BOOL CADOTest1App::InitInstance()
{
    AfxOleInit();
    .....
```

在非MFC环境中用:

```
CoInitialize(NULL);
CoUnInitialize();
```

妻子晚上总是要强行与我同房，查
用户379020986

阿联帮手！紫色西装给杜峰再次带
用户371812474

轰炸空中突发大火两枚400万吨堡
用户372766156

男友不顾同屋姐妹与我发生关系
用户379098152

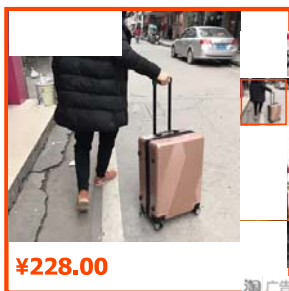
小姨长大后实在太漂亮，那晚上挑
用户378897396

贼心不死！菲南海再挑事，中国不
用户372348096

更多>>



便携式打印机



¥228.00

推荐博文

女房东：上戏学生租房拍摄，结果

中国孩子一代不如一代？这群“小

北京检方对雷洋案5涉案警务人员

周小平 |

宗庆后乘高铁坐二等座被拍到，员

曹德旺没跑路，真问题要改革

国外网友交换礼物结果碰上了盖茨

雾霾天不戴口罩的人是怎么想的？

见义勇为者何以非“重大过失”而

小新眼中的2016



印度地铁绝对让你刮目相看



2016年我去过的秘境



赤峰小雪驱散了雾霾



高棉的微笑是苦涩的

【3】创建Connection对象并连接数据库

首先我们需要添加一个指向Connection对象的指针:

```
_ConnectionPtr m_pConnection;
```

下面的代码演示了如何创建Connection对象实例及如何连接数据库并进行异常捕捉。

```
BOOL CADOTest1Dlg::OnInitDialog()  
{  
    CDialog::OnInitDialog();  
    HRESULT hr;  
    try  
    {  
        hr = m_pConnection.CreateInstance("ADODB.Connection");//创建Connection对象  
        //hr = m_pConnection.CreateInstance(_uuidof(Connection));  
        if(SUCCEEDED(hr))  
        {  
            m_pConnection->ConnectionTimeout = 0;  
            hr = m_pConnection->Open("Provider=Microsoft.Jet.OLEDB.4.0;Data  
Source=test.mdb","","",adModeUnknown);//连接数据库  
            ///上面一句中连接字符串中的Provider是针对ACCESS2000环境的，对于ACCESS97,需要改  
为:Provider=Microsoft.Jet.OLEDB.3.51;  
        }  
    }  
    catch(_com_error e)///捕捉异常  
    {  
        CString errormessage;  
        errormessage.Format("连接数据库失败!\r\n错误信息:%s",e.ErrorMessage());  
        AfxMessageBox(errormessage);///显示错误信息  
    }  
}
```

在这段代码中我们是通过**Connection对象的Open方法**来进行连接数据库的，下面是该方法的原型
HRESULT Connection15::Open (_bstr_t ConnectionString, _bstr_t UserID, _bstr_t Password,
long Options)

ConnectionString为连接字符串,UserID是用户名, Password是登陆密码,Options是连接选项,用于指定
Connection对象对数据的更新许可权,

Options可以是如下几个常量:

- adModeUnknown:缺省。当前的许可权未设置
- adModeRead:只读
- adModeWrite:只写
- adModeReadWrite:可以读写
- adModeShareDenyRead:阻止其它Connection对象以读权限打开连接
- adModeShareDenyWrite:阻止其它Connection对象以写权限打开连接
- adModeShareExclusive:阻止其它Connection对象打开连接
- adModeShareDenyNone:允许其它程序或对象以任何权限建立连接

我们给出一些常用的连接方式供大家参考:

(1)通过JET数据库引擎对ACCESS2000数据库的连接

```
m_pConnection->Open("Provider=Microsoft.Jet.OLEDB.4.0;Data  
Source=C:\\test.mdb","","",adModeUnknown);
```

(2)通过DSN数据源对任何支持ODBC的数据库进行连接:

```
m_pConnection->Open("Data Source=adotest;UID=sa;PWD=;","","",adModeUnknown);
```

(3)不通过DSN对SQL SERVER数据库进行连接: m_pConnection->Open("driver={SQL
Server};Server=127.0.0.1;DATABASE=vckbase;UID=sa;PWD=139","","",adModeUnknown);
其中Server是SQL服务器的名称，DATABASE是库的名称

(4)通过DSN对ORACLE数据库进行连接

```
m_pConnection->Open("Provider=OraOLEDB.Oracle;Data Source=orcl;User  
Id=scott;Password=m123;");
```

Connection对象除Open方法外还有许多方法，我们先介绍Connection对象中两个有用的属性
ConnectionTimeOut与State

ConnectionTimeOut用来设置连接的超时时间，需要在Open之前调用，例如: m_pConnection-
>ConnectionTimeout = 5;///设置超时时间为5秒

```
m_pConnection->Open("Data Source=adotest","","",adModeUnknown);
```

State属性指明当前Connection对象的状态，0表示关闭，1表示已经打开，我们可以通过读取这个属性来作相应的处理，例如:

```
if(m_pConnection->State)
```

```
    m_pConnection->Close(); ///如果已经打开了连接则关闭它
```



凿个洞洞就可以
捕大鱼



混血萝莉神似昆
凌

[查看更多>>](#)

谁看过这篇博文

暂无访客

【4】执行SQL命令并取得结果记录集

为了取得结果记录集，我们定义一个指向Recordset对象的指针:_RecordsetPtr m_pRecordset; 并为其创建Recordset对象的实例: m_pRecordset.CreateInstance("ADODB.Recordset"); 或者 m_pRecordset.CreateInstance(__uuidof(Recordset));

SQL命令的执行可以采用多种形式，下面我们一进行阐述。

(1)利用Connection对象的Execute方法执行SQL命令

Execute方法的原型如下所示:

_RecordsetPtr Connection15::Execute (_bstr_t CommandText, VARIANT * RecordsAffected, long Options) 其中CommandText是命令字符串，通常是SQL命令。参数RecordsAffected是操作完成后所影响的 行数, 参数Options表示CommandText中内容的类型，Options可以取如下值之一：

adCmdText:表明CommandText是文本命令
adCmdTable:表明CommandText是一个表名
adCmdProc:表明CommandText是一个存储过程
adCmdUnknown:未知

Execute执行完后返回一个指向记录集的指针，下面我们给出具体代码并作说明。

示例

```
///执行SQL命令：CREATE TABLE创建表格users,users包含四个字段:整形ID,字符串username,整形old, 日期型
birthday
m_pConnection->Execute("CREATE TABLE users(ID INTEGER,username TEXT,old INTEGER,birthday
DATETIME)",&RecordsAffected,adCmdText);
///往表格里面添加记录
m_pConnection->Execute("INSERT INTO users(ID,username,old,birthday) values
(1,Washington,25,1970/1/1)",&RecordsAffected,adCmdText);
///将所有记录old字段的值加一
m_pConnection->Execute("UPDATE users SET old = old+1",&RecordsAffected,adCmdText);
///执行SQL统计命令得到包含记录条数的记录集
m_pRecordset = m_pConnection->Execute("SELECT COUNT(*) FROM
users",&RecordsAffected,adCmdText);
_variant_t vIndex = (long)0;
_variant_t vCount = m_pRecordset->GetCollect(vIndex);///取得第一个字段的值放入vCount变量
m_pRecordset->Close();///关闭记录集
CString message;
message.Format("共有%d条记录",vCount.IVal);
AfxMessageBox(message);///显示当前记录条数
```

(2)利用Command对象来执行SQL命令

```
_CommandPtr m_pCommand;
m_pCommand.CreateInstance("ADODB.Command");
或者m_pCommand.CreateInstance(__uuidof(Command));
_variant_t vNULL;
vNULL.vt = VT_ERROR;
vNULL.scode = DISP_E_PARAMNOTFOUND;///定义为无参数
m_pCommand->ActiveConnection = m_pConnection;///非常关键的一句，将建立的连接赋值给它
m_pCommand->CommandText = "SELECT * FROM users";///命令字符串
m_pRecordset = m_pCommand->Execute(&vNULL,&vNULL,adCmdText);///执行命令，取得记录集
```

在这段代码中我们只是用Command对象来执行了SELECT查询语句，Command对象在进行存储过程的调用中能真正体现它的作用。下次我们将详细介绍。

(3)直接用Recordset对象进行查询取得记录集

例如

```
m_pRecordset->Open("SELECT * FROM users",_variant_t((IDispatch *))
m_pConnection,true),adOpenStatic,adLockOptimistic,adCmdText);
```

Open方法的原型是这样的:

HRESULT Recordset15::Open (const _variant_t & Source, const _variant_t & ActiveConnection, enum CursorTypeEnum CursorType, enum LockTypeEnum LockType, long Options)

其中：

- ①Source是数据查询字符串
- ②ActiveConnection是已经建立好的连接（我们需要用Connection对象指针来构造一个_variant_t对象）
- ③CursorType光标类型，它可以是以下值之一,请看这个枚举结构:

enum CursorTypeEnum

{

adOpenUnspecified = -1,///不作特别指定

adOpenForwardOnly = 0,///前滚静态光标。这种光标只能向前浏览记录集，比如用MoveNext向前滚动,这种方式可以提高浏览速度。但诸如Bookmark,RecordCount,AbsolutePosition,AbsolutePage都不能使用

adOpenKeyset = 1,///采用这种光标的记录集看不到其它用户的新增、删除操作，但对于更新原有记录的 操作对你是可见的。
adOpenDynamic = 2,///动态光标。所有数据库的操作都会立即在各用户记录集上反应出来。
adOpenStatic = 3,///静态光标。它为你的记录集产生一个静态备份，但其它用户的新增、删除、更新操作对你的记录集来说是不可见的。
};
④LockType锁定类型，它可以是以下值之一，请看如下枚举结构：
enum LockTypeEnum
{
adLockUnspecified = -1,///未指定
adLockReadOnly = 1,///只读记录集
adLockPessimistic = 2,悲观锁定方式。数据在更新时锁定其它所有动作，这是最安全的锁定机制
adLockOptimistic = 3,乐观锁定方式。只有在你调用Update方法时才锁定记录。在此之前仍然可以做数据的更新、插入、删除等动作
adLockBatchOptimistic = 4，乐观分批更新。编辑时记录不会锁定，更改、插入及删除是在批处理模式下完成。
};
⑤Options请参考本文中对Connection对象的Execute方法的介绍

【5】记录集的遍历、更新

根据我们刚才通过执行SQL命令建立好的users表，它包含四个字段:ID,username,old,birthday
以下的代码实现：打开记录集，遍历所有记录，删除第一条记录，添加三条记录，移动光标到第二条记录，更改其年龄，保存到数据库。

```
_variant_t vUsername,vBirthday,vID,vOld;  
_RecordsetPtr m_pRecordset;  
m_pRecordset.CreateInstance("ADODB.Recordset");  
m_pRecordset->Open("SELECT * FROM users",_variant_t((IDispatch*)  
m_pConnection,true),adOpenStatic,adLockOptimistic,adCmdText);  
while(!m_pRecordset->adoEOF)///这里为什么是adoEOF而不是EOF呢?还记得rename("EOF","adoEOF")这一句吗?  
{  
vID = m_pRecordset->GetCollect(_variant_t((long)0));///取得第1列的值,从0开始计数，你也可以直接给出列的名称，如下一行  
vUsername = m_pRecordset->GetCollect("username");///取得username字段的值  
vOld = m_pRecordset->GetCollect("old");  
vBirthday = m_pRecordset->GetCollect("birthday");  
///在DEBUG方式下的OUTPUT窗口输出记录集中的记录  
if(vID.vt != VT_NULL && vUsername.vt != VT_NULL && vOld.vt != VT_NULL && vBirthday.vt != VT_NULL)  
TRACE("id:%d,姓名:%s,年龄:%d,生日:%s\r\n",vID.IVal,(LPCTSTR)(_bstr_t)vUsername,vOld.IVal,(LPCTSTR)(_bstr_t)vBirthday);  
m_pRecordset->MoveNext();///移到下一条记录  
}  
m_pRecordset->MoveFirst();///移到首条记录  
m_pRecordset->Delete(adAffectCurrent);///删除当前记录  
///添加三条新记录并赋值  
for(int i=0;i< 3;i++)  
{  
m_pRecordset->AddNew();///添加新记录  
m_pRecordset->PutCollect("ID",_variant_t((long)(i+10)));  
m_pRecordset->PutCollect("username",_variant_t("叶利钦"));  
m_pRecordset->PutCollect("old",_variant_t((long)71));  
m_pRecordset->PutCollect("birthday",_variant_t("1930-3-15"));  
}  
m_pRecordset->Move(1,_variant_t((long)adBookmarkFirst));///从第一条记录往下移动一条记录,即移动到第二条记录处  
m_pRecordset->PutCollect(_variant_t("old"),_variant_t((long)45));///修改其年龄  
m_pRecordset->Update();///保存到库中  
【6】关闭记录集与连接  
记录集或连接都可以用Close方法来关闭  
m_pRecordset->Close();///关闭记录集  
m_pConnection->Close();///关闭连接
```

三、ADO 数据类型转换表

	ADO			
--	-----	--	--	--

ADO DataType Enum	DataType Enum Value	.NET Framework	SQL Server!	Size	Access文1
adBigInt	20	Int64 SqlDbType.BigInt ¹⁰ OleDbType.BigInt ¹¹	bigint ⁹	8	
adBinary	128	Byte[] SqlDbType.VarBinary ¹⁰ OleDbType.Binary ¹¹	binary timestamp	50 8	
adBoolean	11	Boolean SqlDbType.Bit ¹⁰ OleDbType.Boolean ¹¹	bit	1 2	YesNo
adBSTR	8	String OleDbType.BSTR ¹¹			
adChapter	136	(DataReader)			
adChar	129	String SqlDbType.Char ¹⁰ OleDbType.Char ¹¹	char	X	
adCurrency	6	Decimal SqlDbType.Money ¹⁰ OleDbType.Currency ¹¹	money smallmoney	8 4 0	Currency
adDate	7	DateTime OleDbType.DBDate ¹¹		0	DateTime
adDBDate	133	DateTime OleDbType.DBDate ¹¹			
adDBFileTime	137	DBFileTime ¹¹			
adDBTime	134	DateTime OleDbType.DBTime ¹¹			
adDBTimeStamp	135	DateTime SqlDbType.DateTime ¹⁰ OleDbType.DBTimeStamp ¹¹	datetime smalldatetime	8 4 0	DateTime
adDecimal	14	Decimal OleDbType.Decimal ¹¹			
adDouble	5	Double SqlDbType.Float ¹⁰ OleDbType.Double ¹¹	float	8 0	Double
adEmpty	0	Empty ¹¹			
adError	10	External-Exception OleDbType.Error ¹¹			
adFileTime	64	DateTime OleDbType.Filetime ¹¹			
adGUID	72	Guid SqlDbType.UniqueIdentifier ¹⁰ OleDbType.Guid ¹¹	uniqueidentifier ⁵	16	ReplicationID
adIDispatch	9	Object OleDbType.IDispatch ¹¹			
adInteger	3	Int32 SqlDbType.Int ¹⁰ OleDbType.Integer ¹¹	identity ⁴ int	4 4	AutoNumb Long Integ
adIUnknown	13	Object OleDbType.IUnknown ¹¹			
adLongVarBinary	205	Byte[] SqlDbType.VarBinary ¹⁰ OleDbType.LongVarBinary ¹¹	image	2147483647	OLEObject
adLongVarChar	201	String SqlDbType.VarChar ¹⁰ OleDbType.LongVarChar ¹¹	text	2147483647	Memo ¹ , Hyperlink ¹
adLongVarWChar	203	String SqlDbType.NText ¹⁰ OleDbType.VarWChar ¹¹	ntext ⁵	1073741823	Memo ³ Hyperlink

adNumeric	131	Decimal SqlDbType.Decimal ¹⁰ OleDbType.Decimal ¹¹	decimal numeric	9	Decimal ³
adPropVariant	138	Object OleDbType.PropVariant ¹¹			
adSingle	4	Single SqlDbType.Real ¹⁰ OleDbType.Single ¹¹	real	4	Single
adSmallInt	2	Int16, SqlDbType.SmallInt ¹⁰ OleDbType.SmallInt ¹¹	smallInt	2	Integer
adTinyInt	16	Byte OleDbType.TinyInt ¹¹			
adUnsignedBigInt	21	UInt64 OleDbType.UnsignedBigInt ¹¹			
adUnsignedInt	19	UInt32 OleDbType.UnsignedInt ¹¹			
adUnsignedSmallInt	18	UInt16 OleDbType.UnsignedSmallInt ¹¹			
adUnsignedTinyInt	17	Byte SqlDbType.TinyInt ¹⁰ OleDbType.UnsignedTinyInt ¹¹	tinyInt	1	Byte
adUserDefined	132				
adVarBinary	204	Byte[] SqlDbType.VarBinary ¹⁰ OleDbType.VarBinary ¹¹	varbinary	50	ReplicationI
adVarChar	200	String SqlDbType.VarChar ¹⁰ OleDbType.VarChar ¹¹	varchar	X	Text ^{1, 2}
adVariant	12	Object SqlDbType.Variant ¹⁰ OleDbType.Variant ¹¹	sql_variant ⁹	8016	
adVarNumeric	139	OleDbType.VarNumeric ¹¹			
adVarWChar	202	String SqlDbType.NVarChar ¹⁰ OleDbType.VarWChar ¹¹	nvarchar ⁵	X	Text ³
adWChar	130	String SqlDbType.NChar ¹⁰ OleDbType.WChar ¹¹	nchar ⁵	X	

PS:

- 1 - ODBC Driver (3.51.171300): Microsoft Access Driver (*.mdb), Access 97 (3.5 format)
- 2 - OLE DB Provider: Microsoft.Jet.OLEDB.3.51, Access 97 (3.5 format)
- 3 - OLE DB Provider: Microsoft.Jet.OLEDB.4.0 , Access 2000 (4.0 format)
- 4 - OLE DB Provider: SQLOLEDB, SQL Server 6.5
- 5 - OLE DB Provider: SQLOLEDB, SQL Server 7.0 +
- 6 - The VB Decimal data type can only be used within a Variant, that is, you cannot declare a VB variable to be of type Decimal.
- 7 - Oracle 8.0.x Note: DECIMAL and INT are synonyms for NUMBER and NUMBER(10) respectively.
- 8 - Oracle 8.1.x
- 9 - OLE DB Provider: SQLOLEDB, SQL Server 2000 +
- 10 - SQL Server .NET Data Provider (via System.Data.SqlTypes)
- 11 - OLE DB .NET Data Provider (via System.Data.OleDb)

四、其他

1、_variant_t

- (1)、一般传给这3个指针的值都不是MFC直接支持的数据类型,而要用_variant_t转换一下
_variant_t(XX)可以把大多数类型的变量转换成适合的类型传入：
- (2)、_variant_t var;
_variant_t -> long: (long)var;
_variant_t -> CString: CString strValue = (LPCSTR)_bstr_t(var);

CString -> _variant_t: _variant_t(strSql);

2、BSTR宽字符串与CString相互转换

BSTR bstr;

CString strSql;

CString -> BSTR: bstr = strSql.AllocSysString();

BSTR -> CString: strSql = (LPCSTR)bstr;

3、_bstr_t与CString相互转换

_bstr_t bstr;

CString strSql;

CString -> _bstr_t: bstr = (_bstr_t)strSql;

_bstr_t -> CString: strSql = (LPCSTR)bstr;

4、关于时间

Access:表示时间的字符串#2004-4-5#

Sql:表示时间的字符串" 2004-4-5"

DateField(时间字段) select * from my_table where DateField > #2004-4-10#

```
try {  
    m_pCommand->CommandText = "INSERT INTO tTest(age) VALUES('23f2') ";  
    m_pRecordset = m_pCommand->Execute(NULL,NULL, adCmdText);  
} catch(_com_error e)///捕捉异常  
{ CString errormessage; errormessage.Format("连接数据库失败!\r\n错误信息:%s",e.ErrorMessage());  
    AfxMessageBox(errormessage);///显示错误信息 }
```

0

喜欢

0

赠金笔

分享:

阅读 (24) | 评论 (0) | 收藏 (0) | 转载 (0) | 喜欢 ▼ | 打印 | 举报

已投稿到: 排行榜

推荐: 世上健在的最长寿的人 多谈几次恋爱就不会要死要活 ×

新浪首页 登录 注册

前一篇: 动态链接库 (转)

后一篇: VC ADO操作存储过程 (转)

评论

重要提示: 警惕虚假中奖信息

[发评论]

评论加载中,请稍候...

发评论



哈尔滨自由



恭喜你：获得一次免费诊股机会

输入股票代码即可 3秒预测明日涨跌



投资有风险

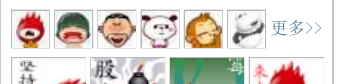
登录名: 密码: 找回密码 注册 ☒ 记住登录状态

☒ 分享到微博 ☐ 评论并转载此博文

按住左边滑块，拖动完成上方拼图

发评论

以上网友发言只代表其个人观点，不代表新浪网的观点或立场。



郭德纲不尊重
为何日本人骂
他割喉陌生女
上戏学生租房

- 年入30万算有钱人吗 • 36岁蔡依林
- 韩国圣诞竟这么美 • 女人失恋了
- 张艺谋终生没摆脱那电影 •
- 老公对前任备注老婆 • 手把手教

[< 前一篇](#)
[动态链接库（转）](#)

[后一篇 >](#)
[VC ADO操作存储过程（转）](#)

新浪BLOG意见反馈留言板 不良信息反馈 电话：4006900000 提示音后按1键（按当地市话标准计费） 欢迎批评指正
[新浪简介](#) | [About Sina](#) | [广告服务](#) | [联系我们](#) | [招聘信息](#) | [网站律师](#) | [SINA English](#) | [会员注册](#) | [产品答疑](#)

Copyright © 1996 - 2016 SINA Corporation, All Rights Reserved
新浪公司 版权所有