开发技术整合应用丛书

Visual C++ .NET 网络数据库整合应用

精锐创作组　　编著

人民邮电出版社

需要全本请访问 www.ertongbook.com



第5章 DO技术在V 中的应用



5.1 CDaoWorkspac 类

5.1.1 CDaoWorkspac 类概述

CDaoWorkspac 类是对 DAO 工作区对象的封装，工作区主要是一个事务管理器，管理在同一个事务区内的一系列打开的数据库。它在 MFC DAO 体系结构中处于最顶层，负责管理从登录到离开期间的数据库会话全过程。

一般来说，在应用程序中只需管理一个数据库会话，因此在大多数情况下不需要使用多个工作区，也不需创建明确的工作区对象。例如，在打开数据库和记录集对象时，所使用的是 DAO 默认工作区。但是，如果需要，可以通过创建额外工作区对象同时运行多个会话。每个工作区包含其 Databas 收集中的多个打开的数据库对象。

5.1.2 CDaoWorkspac 类的构造函数

CDaoWorkspac 类的构造函数为 CDaoWorkspac ，调用它将构造一个 CDaoWorkspac 对象。

它的原型如下所示：

```
CDaoWorkspac() ;
```

构造了 CDaoWorkspac 对象后，一般有下面两种初始化方式：

- 调用对象的 Open 成员函数打开默认工作区对象，或者打开 Workspac 收集中已有对象。
- 调用对象的 Create 成员函数创建一个新 DAO 工作区对象。

初始化完毕后，将启动一个新工作区会话。

5.1.3 工作区属性函数

CDaoWorkspac 类的工作区属性函数包括 GetIsolateODBCTrans、GetNam 、GetUsr - Nam 、IsOpen 和 SetIsolateODBCTrans，它们可以完成查询工作区名、设置 ODBC 事务等操作。

GetIsolateODBCTran 函数

调用该函数以获得工作区的 ODBC 事务隔离特性（即 DAO IsolateODBCTrans 属性值），它的原型如下所示：

```
BOOL GetIsateODBCTran  ();
```

如果 ODBC 事务被隔离，则返回非 0 值，否则返回值为 0。

示例程序代码如下：

```
BOOL flag;
car * tepStr;
CDaoWorkspac_MyWorkSpac;
tepStr=nar;
m_MyWorkSpacOpe(NULL);
flag=m_MyWorkSpacGetIsateODBCTran();
MeageBox(itoa(flag,tepStr,10)," 信息",MB_OK);
```

GetNam 函数

调用该函数可以得到工作区对象的用户定义名，它的原型如下所示：

```
CStrig GetNam ( );
```

如果函数调用成功，则返回包含用户定义的 DAO 工作区对象名。

示例程序代码如下：

```
CStrig flag;
CDaoWorkspac_MyWorkSpac;
m_MyWorkSpacOpe(NULL);
flag=m_MyWorkSpacGetNam();
MeageBox(flag," 信息",MB_OK);
```

GetUserNam 函数

调用该函数可以得到工作区对象的所有者名，它的原型如下所示：

```
CStrig GetUsrNam ( );
```

如果函数调用成功，则返回工作区对象所有者名。

示例程序代码如下：

```
CStrig flag;
car * tepStr;
CDaoWorkspac_MyWorkSpac;
tepStr=nar;
m_MyWorkSpacOpe(NULL);
flag=m_MyWorkSpacGetUsrNam();
MeageBox(flag," 信息",MB_OK);
```



IsOpe 函数

调用该函数以确定工作区对象是否被打开，它的原型如下所示：

```
BOOL IsOpe ( ) ct;
```

如果工作区对象打开，则返回非 0 值，否则返回值为 0。

示例程序代码如下：

```
BOOL flag;
CDaoWorkspac_MyWorkSpac;
m_MyWorkSpacOpe(NULL);
flag=m_MyWorkSpacIsOpe();
if (flag)
{
    MeageBox("  打开","信息",MB_OK);
}
e
{
    MeageBox("  没有打开","信息",MB_OK);
}
```

SetIsolateODBCTran 函数

调用该函数以指定在工作区中是否允许事务隔离，它的原型如下所示：

```
vod SetIsateODBCTran( BOOL    bIsateODBCTran    );
```

如果希望隔离 ODBC 事务，则设置 bIsolateODBCTrans 参数为 true，否则设置该参数为 fal 。

5.1.4 工作区操作函数

CDaoWorkspac 类中的工作区操作函数包括Append、BeginTrans、Cose 、CtTrans 、CpactDatabas 、Create 、GetDatabasount 、GetDatabasInfo 、GetWorkspacount 、GetWorkspacInfo 、Open、RepairDatabas 、Rollback和Idle，它们可以完成打开工作区、创建工作区、开始和提交事务等操作。

Apped 函数

调用该函数可以将新创建的工作区对象添加到数据库引擎的 Workspac 收集，它的原型如下所示：

```
vod Apped( );
```


需要全本请访问 www.ertongbook.com

工作区存在于内存而不是在磁盘中。即使不将工作区添加到 Workspac 收集中，也可以对其进行使用。如果工作区已经被添加到 Workspac 收集中，则在调用其 Cose 成员函数之前，它以活动打开状态保持在 Workspac 收集中。

BegiTran 函数

调用该函数以开始事务操作，它的原型如下：

```
vod BegiTran( );
```

由于工作区被定义为单个事务空间，因此事务将应用于工作区中所有打开的数据库。

完成事务有下面两种途径：

- 调用 CtTrans 成员函数提交事务，把变化保存到数据源。
- 调用 Rollback 函数取消事务。

在事务未完成期间，关闭工作区对象或数据库对象，所有未完成事务将回滚。

C 函数

调用该函数以关闭工作区对象，它的原型如下：

```
virtual vod C( );
```

关闭打开的工作区对象将释放基础 DAO 对象，如果工作区是 Workspac 收集的一个成员，则将被从收集中移去。关闭工作区对象将关闭工作区中任何打开的数据库，这导致同时关闭数据库中打开的任何记录集，并且回滚没完成的编辑或更新。

当数据库引擎会话结束时，工作区及其 Databas 收集也将被关闭。进行下一个会话时，必须再次打开工作区和相应的数据库。

CtTran 函数

调用该函数以提交事务——保存对工作区中一个或多个数据库的一组编辑或更新操作，它的原型如下所示：

```
vod CtTran( );
```

事务包含从调用 BeginTrans 开始的对数据库数据或它的结构所做的一系列改动。完成事务时可以提交它或者用 Rollback 回滚（取消改动）。

在默认情况下（不使用事务时），将立即提交对记录的更新，而调用 BeginTrans 导致更新延迟到调用 CtTrans 时提交。

在一个工作区内，事务总是全局的，而不仅仅局限在一个数据库或记录集中。如果在工作区事务内执行一个以上数据库或记录集操作，CtTrans 提交所有未完成更新，Rollback 恢复这些数据库和记录集上的所有操作。

在关闭有未完成事务的数据库或工作区时，事务全部回滚。



对其中一个数据库的更新提交失败并不影响对其他数据库的更新提交。

CpactDatabas　　函数

调用该函数以压缩 Mirosoft Jet 数据库，它的原型如下：

```
stati vod PASCL CpactDatabas
(
    LPCTSTR lpszSrcNam ,
    LPCTSTR lpszDetNam  ,
    LPCTSTR lpszLoal  = dbLangGeral,
    it  nOpti  = 0
);
stati vod PASCL CpactDatabas
(
    LPCTSTR lpszSrcNam ,
    LPCTSTR lpszDetNam  ,
    LPCTSTR lpszLoal  ,
    it  nOpti  ,
    LPCTSTR lpszPasrd
);
```

参数的说明如下：

- lpszSrcNam ： 现有关闭的工作区名字。
- lpszDetNam ： 创建的压缩数据库完整路径，可以像 lpszSrcNam 那样指定网络路径，但是不能使用 lpszDetNam 参数指定与 lpszSrcNam 相同的数据库文件。
- lpszPasord ： 数据库的密码，在使用支持密码保护的 CpactDatabas 版本时，必须提供所有参数，其格式如下：

  ;PWD=lpszPasrd

  例如：

  ;PWD="Happy"

- lpszLocal ： 创建压缩数据库时的对照顺序，如果省略该参数，则将使用默认值 dbLangGeral 。该参数的取值如下所示：

  dbLangGeral ： Engl 、German、Fre 、Portugue 、Italan 和 Modern Span ；

  dbLangArabi ： Arabi ；

  dbLangCyri ： Rusan ；

dbLangCze ：Cze ；

dbLangDutc ：Dutc ；

dbLangGrek ：Grek ；

dbLangHebre ：Hebre ；

dbLangHungarian：Hungarian；

dbLangIcandi ：Icandi ；

dbLangNordi ：Nordianguage （只用于 Miroft Jet 数据库引擎 1.0 版）；

dbLangNorwdan：Norwgian 和 Dan ；

dbLangPo ：Po ；

dbLangSpan ：Traditial Span ；

dbLangSwdfi ：Swdi 和 Fi ；

dbLangTurki ：Turki 。

- nOptions：指定了创建压缩数据库时所使用的选项。如果省略该参数，则所创建的压缩数据库与源数据库具有相同的加密性与版本号。该参数的取值如下所示（可以使用位操作符"OR"组合其中的一些值）：

  dbEnrypt ：压缩时加密数据库；

  dbDerypt ：压缩时解密数据库；

  dhVers10 ：压缩时使用 Miroft Jet 数据库引擎 1.0 版创建数据库；

  dbVers11 ：压缩时使用 Miroft Jet 数据库引擎 1.1 版创建数据库；

  dbVers20 ：压缩时使用 Miroft Jet 数据库引擎 2.0 版创建数据库；

  dbVers30 ：压缩时使用 Miroft Jet 数据库引擎 3.0 版创建数据库。

Create 函数

调用该函数以创建一个新 DAO 工作区对象，它的原型如下：

```
virtual vod Create
(
    LPCTSTR lpszNam ,
    LPCTSTR lpszUsrNam ,
    LPCTSTR lpszPasrd
);
```

参数的说明如下：

- lpszNam ：指定了新工作区对象的名称，最多可包含 14 个字符。
- lpszUsrNam ：指定了工作区所有者名称。
- lpszPasord ：指定了新工作区对象的密码，最多可包含 14 个字符，密码中可以使用除 ASCII0（空）以外的任何字符，并且它对大小写敏感。



创建工作区对象的整个创建过程是：

- 构造一个 CDaoWorkspac 对象。
- 调用对象的 Create 成员函数，创建基础 DAO 工作区。
- 调用 Append 函数将工作区添加到数据库引擎的 Workspac 收集。

调用 Create 函数之后工作区对象处于打开状态，因此不需要再调用 Ope 函数。如果工作区已经在 Workspac 收集中存在，则不能为其调用 Create 函数。

GetDatabaseunt 函数

调用该函数以获取工作区的 Databas 收集中 DAO 数据库对象数量—— 工作区中打开的数据库数量，它的原型如下：

```
srt GetDatabasunt( )      ;
```

如果函数调用成功，则返回工作区中打开的数据库数目。

在需要分别得到工作区 Databas 收集中的所有数据库的信息时，可以首先调用 GetDatabasount 函数以得到打开的数据库数目，然后以该数目作为循环索引，重复调用 GetDatabasInfo 函数即可通过遍历获得工作区中所有打开的数据库信息。

示例程序代码如下：

```
it flag;
car *tepStr;
CDaoWorkspac_MyWorkSpac;
tepStr=nar;
m_MyWorkSpacOpe(NULL);
flag=m_MyWorkSpacGetDatabasunt();
MeageBox(itoa(flag,tepStr,10),NULL,MB_OK);
```

GetDatabasefo 函数

调用该函数以获得工作区中指定数据库的信息，它的原型如下所示：

```
vod GetDatabasInfo
(
    it  nIndex,
    CDaoDatabasInfo&  dbifo ,
    DWORD dwInfoOpti     = AFX_DAO_PRIMARY_INFO
);
vod GetDatabasInfo
(
    LPCTSTR lpszNam ,
```

```
    CDaoDatabasInfo&  dbifo ,
    DWORD dwInfoOpti      = AFX_DAO_PRIMARY_INFO
);
```

参数的说明如下：

- nIndex：工作区的 Databas 收集中数据库对象的索引，以 0 为基值。
- dbinfo：返回指定的 CDaoDatabas 对象信息，该参数为 CDaoDatabasnfo 结构，其定义如下：

```
struct CDaoDatabasInfo
{
    CStrig m_strNam;
    //数据库名称
    BOOL m_bUpdatabl;
    //是否可以更新
    BOOL m_bTranacti;
    //是否可以执行事物
    CStrig m_strVers;
    //版本
    lg m_latigOrder;
    //排序规则
    srt m_nQueryTiut;
    //登录超时
    CStrig m_strCt;
    //连接字符串
};
```

- dwInfoOptions：指定调用函数将返回的信息，参数的取值和函数的返回值如下：

AFX_DAO_PRIMARY_INFO：指定该默认选项，则函数将返回包括数据库名、可更新性和是否可执行事务在内的主要信息。

AFX_DAO_SECONDARY_INFO：指定该选项，则函数将返回包括主要信息、版本号、排序规则以及登录超时在内的辅助信息。

AFX_DAO_ALL_INFO：指定该选项，则函数将返回包括辅助信息以及连接字符串在内的所有信息。

- lpszNam ：数据库对象的名字，最多可包含 14 个字符。

GetWorkspacunt 函数

调用该函数以获得数据库引擎的 Workspac 收集中 DAO 工作区对象数量，它的原型如下所示：



srt GetWorkspacunt( );

如果函数调用成功，则返回 Workspac 收集中的打开工作区数目。

为获取收集中给定工作区的有关信息，首先调用 GetworksPacount 函数得到打开工作区数目，然后以该数目作为循环索引，重复调用 GetWorkspac Info 函数即可通过遍历获得收集中的所有工作区的信息。

示例程序代码如下：

```
it flag;
car *tepStr;
CDaoWorkspac_MyWorkSpac;
tepStr=nar;
flag=m_MyWorkSpacGetWorkspacunt();
MeageBox(itoa(flag,tepStr,10),NULL,MB_OK);
```

GetWorkspacInfo 函数

调用该函数以获得数据库引擎收集会话中指定工作区的信息，它的原型如下：

```
vod GetWorkspacInfo
(
    it   nIndex,
    CDaoWorkspacInfo&   wkspcfo   ,
    DWORD dwInfoOpti      = AFX_DAO_PRIMARY_INFO
);
vod GetWorkspacInfo
(
    LPCTSTR lpszNam  ,
    CDaoWorkspacInfo&   wkspcfo   ,
    DWORD dwInfoOpti      = AFX_DAO_PRIMARY_INFO
);
```

它的参数说明如下：

- nIndexWorkspac ：收集中工作区对象的索引，以 0 为基值。
- wkspcinfo：返回指定的工作区对象信息的 CDaoWorkspacInfo 结构，该结构的定义如下：

```
struct CDaoWorkspacInfo
{
   CStrig m_strNam;
   //工作区名称
```


需要全本请访问 www.ertongbook.com

```
    CStrig m_strUsrNam;
    // 工作区所有者名称
    BOOL m_bIsateODBCTran;
    // 工作区事务是否被隔离
};
```

- dwInfoOption：指定函数调用将返回的信息，该参数的取值和函数返回值如下：

AFX_DAO_PRIMARY_INFO：指定该默认选项，则函数将返回包括工作区名在内的主要信息；

AFX_DAO_SECONDARY_INFO：指定该选项，则函数将返回包括主要信息以及工作区所有者名在内的辅助信息；

AFX_DAO_ALL_INFO：指定该选项，则函数将返回包括辅助信息以及工作区中的事务是否被隔离在内的所有信息。

- lpszNam ：指定了工作区对象的名字，最多可包含 14 个字符。

示例程序代码如下：

```
it flag;
car *tepStr;
CDaoWorkspac_MyWorkSpac;
it idex;
CDaoWorkspacInforkspacfo;
DWORD InfoOpti;

tepStr=nar;
idex=0;
InfoOpti=AFX_DAO_ALL_INFO;
m_MyWorkSpacOpe(NULL);
flag=m_MyWorkSpacGetWorkspacunt();
for(it i=0;i<flag;i++)
{
    m_MyWorkSpacGetWorkspacInfo(idex,wrkspacfo,InfoOpti);
    MeageBox(wrkspacfo_strNam,NULL          ,MB_OK);
}
```

Idl 函数

调用该函数以允许数据库引擎执行后台任务，它的原型如下所示：

```
stati vod PASCL Idl( it      nti      = dbFreLoks );
```

参数 ntion 中存储的是空闲处理期间引擎应执行的操作，当前该参数的唯一取值为 dbFreLocks 。



Ope 函数

调用该函数以显式打开指定的工作区对象，它的原型如下所示：

```
virtual vod Ope( LPCTSTR  lpszNam  = NULL );
```

参数 lpszNam 指定了要打开的 DAO 工作区对象名称，最多可包含 14 个字符。如果使用默认值 NULL，则可以显式打开默认工作区。

该函数将工作区对象置为打开状态，同时如果尚未对应用初始化，则初始化数据库引擎。

RepairDatabase 函数

调用该函数以修复一个受损的数据库，它的原型如下：

```
stati vod PASCL RepairDatabas( LPCTSTR     lpszNam  );
```

参数 lpszNam 指定了要修复的 Mirosoft Jet 引擎数据库文件的路径和文件名，如果省略路径则只查找当前目录。

在进行修复之前，必须先关闭由 lpszNam 指定的数据库。在多用户环境中，其他用户不能在修复期间打开该数据库。如果指定数据库未关闭或者被独占使用，则调用 Repair-Databas 函数将产生错误。

该函数试图通过不完整的写操作，修复被作了“可能损坏”标记的数据库。由于电源断电或计算机硬件问题，使 Mirosoft Jet 数据库应用程序意外关闭，就可能对数据库造成损坏。如果是以正常途径退出应用程序，则不对数据库作“可能损坏”记号。

Roback

调用该函数以回滚事务，它的原型如下：

```
vod Roback( );
```

在一个工作区对象内，事务对工作区总是全局的，而不仅仅局限于一个数据库或记录集，因此调用 Rollback 函数将回滚所有数据库和记录集上的操作。如果关闭工作区对象时未提交或回滚事务，则事务自动回滚。如果调用 CtTrans 或 Rollback 函数之前未调用 BeginTrans 函数，则会产生错误。

5.1.5 数据库引擎属性函数

CDaoWorkspac 类中数据库引擎属性函数包括 GetVerson 、GetIniPath、GetLogin-Tiout 、SetDefaultPasord 、SetDefaultUsr 、SetIniPath 和 SetLoginTiout ，它们能够完成查询 ODBC 数据库登录超时、设置数据库引擎密码等操作。

GetInPath 函数

调用该函数以得到 Mirosoft Jet 数据库引擎的初始化设置在 Widow 注册表中的位置，它的原型如下所示：

```
statiStrig PASCL GetInPath( );
```

如果函数调用成功，将返回包含注册位置的 CString。

```
it flag;
CStrig tepStr;
CDaoWorkspac_MyWorkSpac;
m_MyWorkSpacOpe(NULL);
flag=m_MyWorkSpacGetWorkspacunt();
for(it i=0;i<flag;i++)
{
    tepStr=m_MyWorkSpacGetInPath();
    MeageBox(tepStr,NULL,MB_OK);
}
```

GetLogiTiut 函数

调用该函数以得到工作区中登录 ODBC 数据库的超时设置（即 DAO LoginTmout 属性值），它的原型如下所示：

```
statirt PASCL GetLogiTiut( );
```

如果函数调用成功，则返回 ODBC 数据库登录超时值，以秒为单位。

示例程序代码如下：

```
it flag;
it tep;
car *tepStr;
CDaoWorkspac_MyWorkSpac;
tepStr=nar;
m_MyWorkSpacOpe(NULL);
flag=m_MyWorkSpacGetWorkspacunt();
for(it i=0;i<flag;i++)
{
    tep=m_MyWorkSpacGetLogiTiut();
    MeageBox(itoa(tep,tepStr,10),NULL,MB_OK);
}
```



GetVersio 函数

调用该函数以获得与工作区相联系的 Mirosoft Jet 数据库引擎的版本，它的原型如下所示：

statiStrig PASCL GetVers( );

如果函数调用成功，则返回相应的数据库引擎版本。

示例程序代码如下：

```
it flag;
CStrig tepStr;
CDaoWorkspac_MyWorkSpac;
m_MyWorkSpacOpe(NULL);
flag=m_MyWorkSpacGetWorkspacunt();
for(it i=0;i<flag;i++)
{
    tepStr=m_MyWorkSpacGetVers();
    MeageBox(tepStr,NULL,MB_OK);
}
```

SetDetaultPasswod

调用该函数以设置数据库引擎使用的默认密码，它的原型如下所示：

stati vod PASCL SetDefaultPasrd(LPCTSTR *lpszPasrd*);

参数 lpszPasord 指定了默认密码，密码最多可包含 14 个字符，其中除了 ASCII 0（空）以外的任何字符，并且密码对大小写敏感。

使用该成员函数应该遵循以下步骤：

- 构造 CDaoWorkspac 对象。
- 调用 SetDefaultPasord 函数。
- 不指定密码，对该工作区对象调用 Create 函数。

默认情况下，DefaultPasord 属性值被设置为空字符串（""）。

示例程序代码如下：

```
it flag;
car *tepStr1;
car *tepStr2;
CDaoWorkspac_MyWorkSpac;
tepStr1=nar;
tepStr2=nar;
```

```
tepStr1="Togzh";
tepStr2="";
m_MyWorkSpacOpe(NULL);
flag=m_MyWorkSpacGetWorkspacunt();
for(it i=0;i<flag;i++)
{
    m_MyWorkSpacSetDefaultPasrd(tepStr1);
    m_MyWorkSpacSetDefaultPasrd(tepStr2);
}
```

SetDefaultUser 函数

调用该函数以设置数据库引擎使用的默认名，它的原型如下：

stati vod PASCL SetDefaultUsr(LPCTSTR *lpszDefaultUsr*);

参数 lpszDefaultUs 为默认用户名，可包含为 1~20 个字符。

使用该成员函数应该遵循以下步骤：

- 构造 CDaoWorkspac 对象组但不调用 Create 函数。
- 调用 SetDefaultUsr 函数。
- 不指定用户名，对该工作区对象调用 Create 函数。

默认情况下，DefautUsr 属性值被设置为 admi 。

示例程序代码如下：

```
it flag;
car *tepStr1;
car *tepStr2;
CDaoWorkspac_MyWorkSpac;
tepStr1=nar;
tepStr2=nar;
tepStr1="Togzh";
tepStr2="adm";
m_MyWorkSpacOpe(NULL);
flag=m_MyWorkSpacGetWorkspacunt();
for(it i=0;i<flag;i++)
{
    m_MyWorkSpacSetDefaultUsr(tepStr1);
    m_MyWorkSpacSetDefaultUsr(tepStr2);
}
```



SetInPath

调用该函数以指定 Mirosoft jet 数据库引擎的 Windows 造成设置的位置，它的原型如下所示：

stati vod PASCL SetInPath(LPCTSTR *lpszRegitrySubkey*);

参数 lpszRegistrySubkey 指定了用于可安装 ISAM 数据库所需 Mirosoft Jet 数据库引擎或参数位置的字符串，包含 Widow 登记子码名字。

调用该成员函数，可以指定为只在需要指定特殊设置时调用 SetIniPath 函数。必须在程序安装期间而不是运行期间调用 SetIniPath 函数；必须在打开任何工作区、数据库或记录集前调用 SetIniPath 函数，否则 MFC 将产生异常。

5.1.6 CDaorkSpac 类的数据成员

CDaoWorkspac 类中带有数据成员 m_pDAOWorkspac ，该数据成员为指向基础 DAO 工作区对象的指针。

5.2 CDaoDatabase 类

5.2.1 CDaoDatabas 类

CDaoDatabas 类是对 DAO 数据库对象的封装，它代表了一个数据库连接，使用它可以操纵数据库中的数据。在一个由 CDaoworspac 对象指定的工作区中，可以同时拥有多个活动的 CDaoDatabas 对象。每个 CDaoDatabas 对象都拥有自己的表定义、查询定义、记录集和关系对象的收集，同时 CDaoDatabas 类也提供了操纵这些对象的成员函数。不过上述对象都存储在 DAO 中，而不是存储在 MFC 数据库对象中。

与 CDaoWorkspac 工作区对象相似，可以隐式地建立数据库对象，也可以显式地创建数据库对象。要显式地通过 CDaoDatabas 利用现有的数据库，只需构造一个 CDaoDatabas 对象，并将已打开的 CDaoworkspac 对象指针作为参数传递给它，如果为 CDaoDatabas 对象的构造函数指定工作区，则 MFC 会自动为数据库对象建立一个临时的默认工作区。调用 Open 函数可以打开一个现存的数据库，而调用 Create 函数则可以创建一个新的 Mirosoft Jet 数据库（.MDB）。

所有的这些方法都将所创建的 CDaoDatabas 对象添加到工作区的 Databas 收集中，并且建立了与数据库的连接。此后，可以通过该 CDaoDatabas 对象指针，来构造 CDaoReordset 、CDaoTableDef 或 CDaoQueryDef 对象，并通过它们对所连接的数据库进行操作。数据库对象使用完毕后，调用其成员函数 C 关闭连接，然后删除 CDaoDatabas

需要全本请访问 www.ertongbook.com