

25. 数据库操作

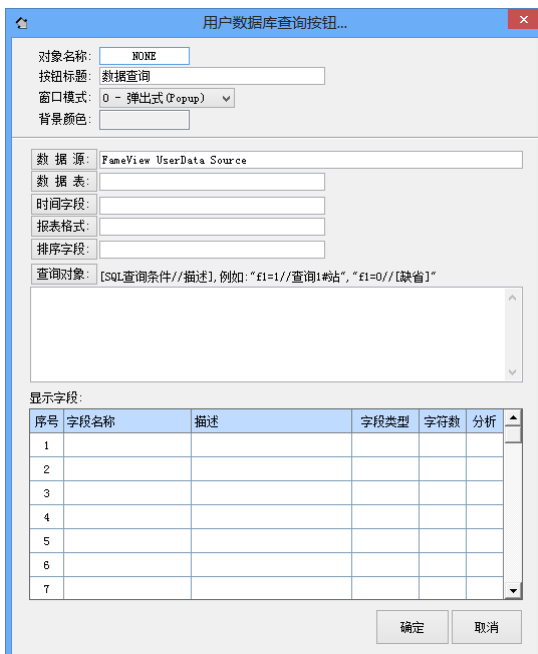
序号	内容	页码
25.1	数据查询	25-2
25.2	曲线查询	25-6
25.3	表格显示	25-10
25.4	字段列表	25-14
25.5	删除数据表记录	25-17
25.6	数据库脚本编程	25-18
25.7	存储过程脚本编程	25-19
25.8	数据表平均统计	25-20
25.9	数据表导出 Excel	25-21
25.10	ADO 访问数据库	25-22
25.11	查询数据表最近时间段记录	25-23
25.12	获取数据表最近时间段统计值	25-23
25.13	获取数据表某时间段字段总和	25-24
25.14	分钟数据表生成小时和日数据表	25-25
25.15	数据库动态增加数据表	25-26
25.16	异步批量删除数据库记录	25-26
25.17	异步批量修正数据库记录	25-27
25.18	判断数据库文件尺寸, 异步批量删除数据库记录	25-27
25.19	查询数据表某月日平均值并通过 Excel 柱状图显示	25-28
25.20	以可变时间间隔查询显示数据表记录	25-30
25.21	设置数据查询组件查询时间间隔	25-32
25.22	数据库增补记录	25-33
25.23	通过触发器计算增加值	25-34
25.24	脚本使用 ADO 访问数据库判断是否成功	25-36
25.25	执行脚本备份 SQL 数据库	25-37
25.26	脚本使用 ADO 执行存储过程	25-37
25.27	得到数据表时间字段最早及最新日期	25-38
25.28	查询月数据表(每月对应 1 个数据表)	25-39
25.29	数据库转发	25-41
25.30	使用脚本管理数据库	25-46

25.1 数据查询

- 画面提供“数据查询”组件：



- 组态对话框：



- 执行<数据源>按钮, 选择或定义数据源：



- [1]. “控制面板->管理工具”中定义ODBC数据源；
- [2]. 远程数据源须输入用户名称、口令才允许访问；

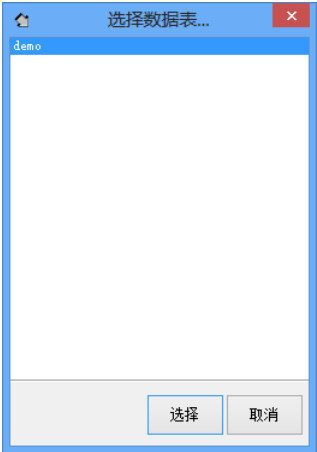
[3]. 执行<自定义数据源>按钮, 定义DSN描述型数据源:



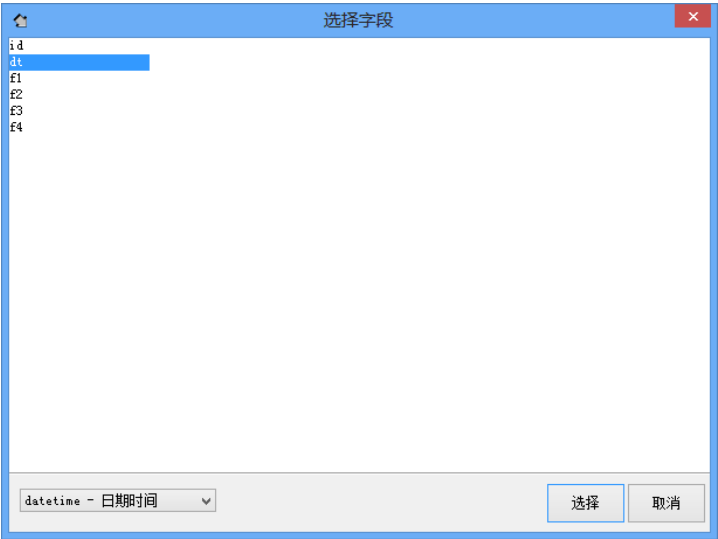
[4]. 或直接输入DSN描述型数据源:

数据库	数据源描述
SQL Server	Driver={SQL Server};Server=(local);Database=UserDatabase;Uid=;Pwd=;
Access	Driver={Microsoft Access Driver (*.mdb)};Dbq=c:\mydb.mdb;Uid=;Pwd=;
Oracle	Driver={Microsoft ODBC for Oracle};Server=192.168.1.100;Uid=;Pwd=;

□ 数据表, 必选项, 选择存储查询记录的数据表或视图:



□ 时间字段, 必选项, 查询数据表所使用的时间字段:



- 报表格式, 可选项, 预先制作并选择Crystal报表格式, 将查询结果打印或浏览;
- 排序字段, 可选项, 将查询结果进行特殊排序;

例1. 根据f3字段升序排列:

排序字段: f3

例2. 根据f3字段降序排列:

排序字段: f3 DESC

- 查询对象, 可选项, 预设过滤条件, 实现分组查询;

查询对象: [SQL查询条件//描述], 例如: "f1=1//查询1#站", "f1=0//[缺省]"

f1=1//1#站
f1=2//2#站
f1=3//3#站
f1=?//某站
f4= ABC //文本查询

- 鼠标双击<显示字段>表格, 通过字段列表选择查询显示字段, 最多支持500个字段:

显示字段:

序号	字段名称	描述	字段类型	字符数	分析
1	f1	f1	real	10	×
2	f2	f2	real	10	×
3	f3	f3	real	10	×
4	f4	f4	string	10	×
5					
6					
7					

- [1]. 无需再选择时间字段, 缺省显示在查询结果的第二列;
- [2]. 选择字段自动识别字段类型, 需设定字段描述和其内容的字符长度;
- [3]. 按键删除列表字段;

- 查询窗口模式: 弹出式、嵌入式

数据查询

起始时间: 2013-10-29 10:00:00 查询 < 上页 选择 下页 >

查询长度: 1 小时

序号	时间	f1	f2	f3	f4
1	2013-10-29 10:00:00	1	7240.21143	5488	(null)
2	2013-10-29 10:00:00	2	7240.21143	4513	(null)
3	2013-10-29 10:00:00	3	7240.21143	0.95209	(null)
4	2013-10-29 10:00:10	1	1114.23083	5617	(null)
5	2013-10-29 10:00:10	2	1114.23083	4384	(null)
6	2013-10-29 10:00:10	3	1114.23083	-0.36712	(null)
7	2013-10-29 10:00:20	1	9110.99609	5744	(null)
8	2013-10-29 10:00:20	2	9110.99609	4257	(null)
9	2013-10-29 10:00:20	3	9110.99609	-0.51687	(null)
10	2013-10-29 10:00:30	1	4054.38403	5872	(null)
11	2013-10-29 10:00:30	2	4054.38403	4129	(null)
12	2013-10-29 10:00:30	3	4054.38403	0.9958	(null)
13	2013-10-29 10:00:40	1	2711.2644	6000	(null)
14	2013-10-29 10:00:40	2	2711.2644	4001	(null)
15	2013-10-29 10:00:40	3	2711.2644	-0.66958	(null)
16	2013-10-29 10:00:50	1	863.36865	6128	(null)
17	2013-10-29 10:00:50	2	863.36865	3873	(null)
18	2013-10-29 10:00:50	3	863.36865	-0.14022	(null)
19	2013-10-29 10:01:00	1	8192.08398	6255	(null)
20	2013-10-29 10:01:00	2	8192.08398	3746	(null)
21	2013-10-29 10:01:00	3	8192.08398	0.88851	(null)
22	2013-10-29 10:01:10	1	8777.42871	6384	(null)
23	2013-10-29 10:01:10	2	8777.42871	3617	(null)
24	2013-10-29 10:01:10	3	8777.42871	-0.91816	(null)
25	2013-10-29 10:01:20	1	5845.51514	6511	(null)
26	2013-10-29 10:01:20	2	5845.51514	3490	(null)
27	2013-10-29 10:01:20	3	5845.51514	0.20846	(null)

查询对象
1#站
2#站
3#站
某站
文本查询
(全部)

起始时间: 2013-10-29 10:00:00		查询		< 上页		选择	下页 >	
查询长度: 1 小时								
序号	时间	f1	f2	f3	f4			
1	2013-10-29 10:00:00	1	7240.21143	5488	(null)			
2	2013-10-29 10:00:00	2	7240.21143	4513	(null)			
3	2013-10-29 10:00:00	3	7240.21143	0.95209	(null)			
4	2013-10-29 10:00:10	1	1114.23083	5617	(null)			
5	2013-10-29 10:00:10	2	1114.23083	4384	(null)			
6	2013-10-29 10:00:10	3	1114.23083	-0.36712	(null)			
7	2013-10-29 10:00:20	1	9110.99609	5744	(null)			
8	2013-10-29 10:00:20	2	9110.99609	4257	(null)			
9	2013-10-29 10:00:20	3	9110.99609	-0.51687	(null)			
10	2013-10-29 10:00:30	1	4054.38403	5872	(null)			
11	2013-10-29 10:00:30	2	4054.38403	4129	(null)			
12	2013-10-29 10:00:30	3	4054.38403	0.9958	(null)			
13	2013-10-29 10:00:40	1	2711.2644	6000	(null)			
14	2013-10-29 10:00:40	2	2711.2644	4001	(null)			
15	2013-10-29 10:00:40	3	2711.2644	-0.68958	(null)			
16	2013-10-29 10:00:50	1	863.36865	6128	(null)			
17	2013-10-29 10:00:50	2	863.36865	3873	(null)			
18	2013-10-29 10:00:50	3	863.36865	-0.14022	(null)			
19	2013-10-29 10:01:00	1	8192.08398	6255	(null)			
20	2013-10-29 10:01:00	2	8192.08398	3746	(null)			
21	2013-10-29 10:01:00	3	8192.08398	0.88851	(null)			
22	2013-10-29 10:01:10	1	8777.42871	6384	(null)			
23	2013-10-29 10:01:10	2	8777.42871	3617	(null)			

□ 数据查询组件脚本函数:

1	设置弹出式查询窗口显示位置
	UserDbSQLObj.SetWindowPosition objName, sx, sy
2	设置弹出式查询窗口显示尺寸
	UserDbSQLObj.SetWindowSize objName, cx, cy
3	显示弹出式查询窗口
	UserDbSQLObj.ShowWindow objName
4	设置查询缺省时间长度
	UserDbSQLObj.SetTimeLen objName, len, unit
5	设置或追加查询内容, 或设置附加过滤条件
	UserDbSQLObj.SetQueryObject objName, desc
6	设置查询窗口的选项
	UserDbSQLObj.SetWindowOptions objName, options

25.2 曲线查询

- 画面提供“曲线查询”组件：



- 组态对话框：

对象名称: NONE

按钮标题: 曲线查询

窗口模式: 0 - 弹出式 (Popup)

背景颜色:

数据源: FameView UserData Source

数据表:

时间字段:

查询对象: [SQL查询条件//描述], 例如: "f1=1//查询1#站", "f1=0//[缺省]"

曲线字段: [字段//最小值, 最大值//描述], 例如: "f1//0, 100//温度", "f1//0, VA. VA1//温度"

确认 取消

- 执行<数据源>按钮, 选择或定义数据源:

选择数据源...

dBASE Files
Excel Files
FameView DataBase Source
FameView DocData Source
FameView ReportData Source
FameView UserData Source
MS Access Database
Visual FoxPro Database
Visual FoxPro Tables

自定义数据源...

用户名称:

口令:

选择 取消

- [1]. “控制面板->管理工具”中定义ODBC数据源;
- [2]. 远程数据源须输入用户名称、口令才允许访问;

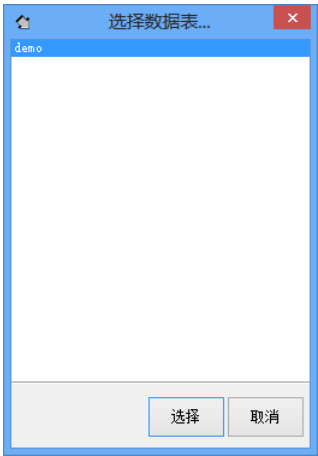
[3]. 执行<自定义数据源>按钮, 定义DSN描述型数据源:



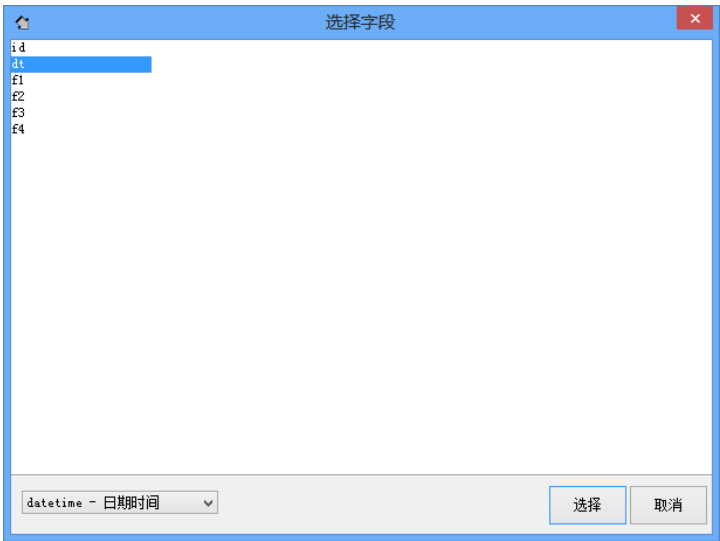
[4]. 或直接输入DSN描述型数据源:

数据库	数据源描述
SQL Server	Driver={SQL Server};Server=(local);Database=UserDatabase;Uid=;Pwd=;
Access	Driver={Microsoft Access Driver (*.mdb)};Dbq=c:\mydb.mdb;Uid=;Pwd=;
Oracle	Driver={Microsoft ODBC for Oracle};Server=192.168.1.100;Uid=;Pwd=;

□ 数据表, 必选项, 选择存储曲线记录的数据表或视图:



□ 时间字段, 必选项, 查询数据表曲线记录所使用的时间字段:



- 查询对象, 可选项, 预设过滤条件, 实现分组查询;

查询对象: [SQL查询条件//描述], 例如: "f1=1//查询1#站", "f1=0//[缺省]"

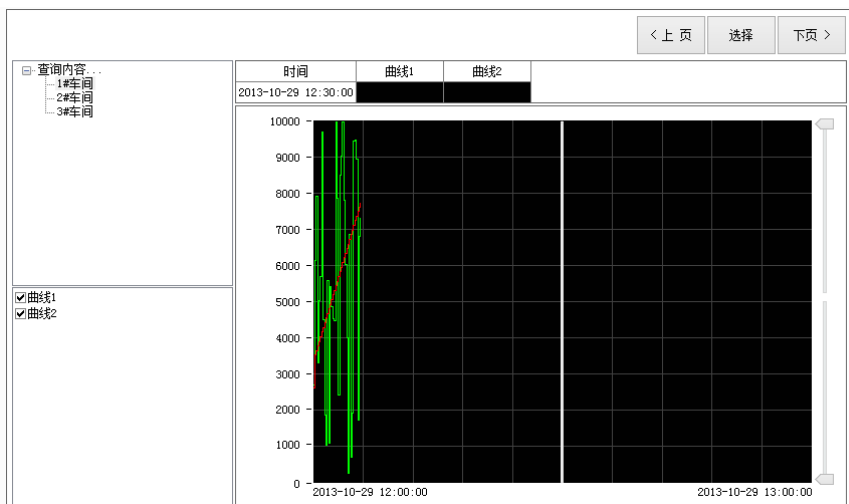
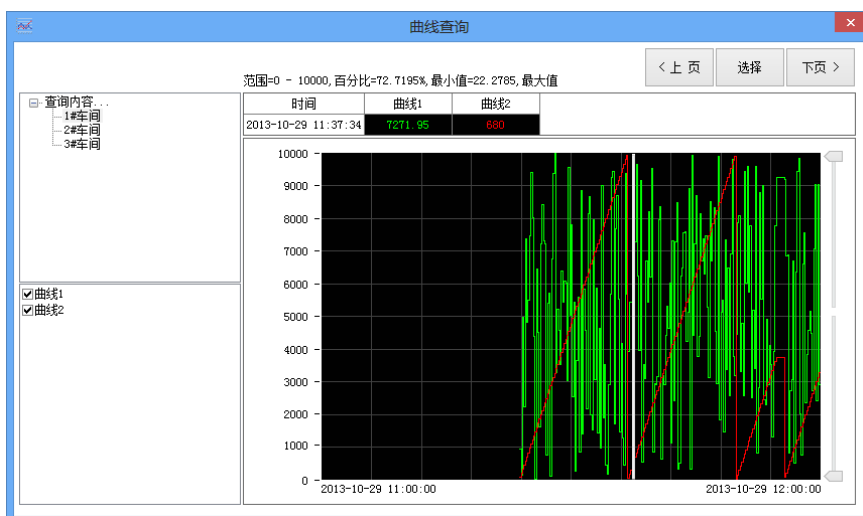
f1=1//1#车间
f1=2//2#车间
f1=3//3#车间

- 曲线字段, 必选项, 最多设置 10 条曲线字段:

曲线字段: [字段//最小值, 最大值//描述], 例如: "f1//0, 100//温度", "f1//0, VA, VA1//温度"

f2//0, 10000//曲线1
f3//0, 10000//曲线2

- 查询窗口模式: 弹出式、嵌入式



□ 曲线查询组件脚本函数：

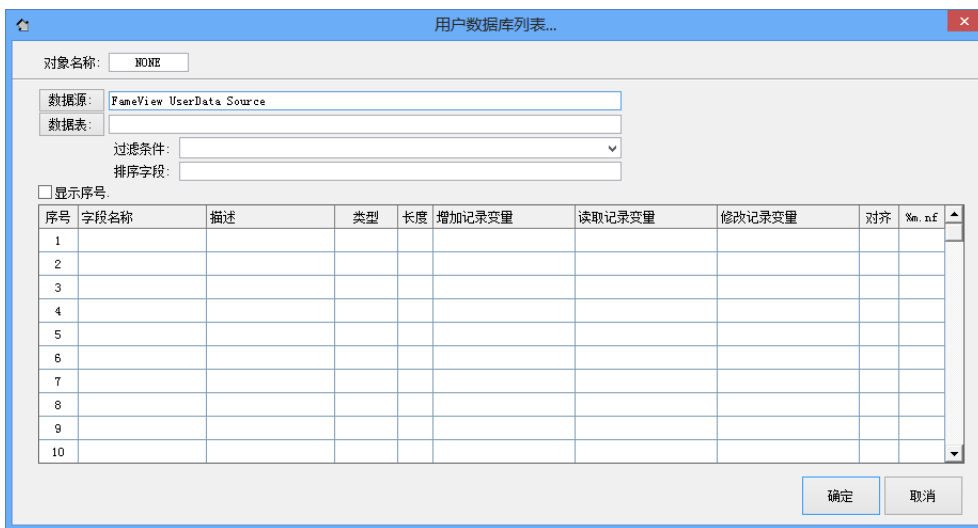
1	设置弹出式查询窗口显示位置
	SQLCurveObj.SetWindowPosition objName, sx, sy
2	设置缺省查询时间长度和分辨率
	SQLCurveObj.SetTimeLen objName, len, unit
3	添加附加参考曲线
	SQLCurveObj.ShowWindow objName, bValue, len, unit
4	设置某曲线颜色
	SQLCurveObj.SetCurveColor objName, n, r, g, b
5	设置弹出式查询窗口参数
	SQLCurveObj.SetCurveParam objName, Title, db, Table, TimeField, EndTime, CurveMode, SqlDesc, CurveDesc
6	设置控件是否显示树形选择条
	SQLCurveObj.ShowTreeBar objName, n
7	以独占方式显示弹出式曲线查询窗口
	SQLCurveObj.ShowWindow objName
8	以共享方式显示弹出式曲线查询窗口
	SQLCurveObj.ShowWindowEx objName
9	设置弹出式查询窗口尺寸
	SQLCurveObj.SetWindowSize objName, cx, cy
10	设置曲线缺省为自动刷新
	SQLCurveObj.SetAutoUpdate objName
11	设置或追加查询对象内容
	SQLCurveObj.SetQueryObject objName, desc

25.3 表格显示

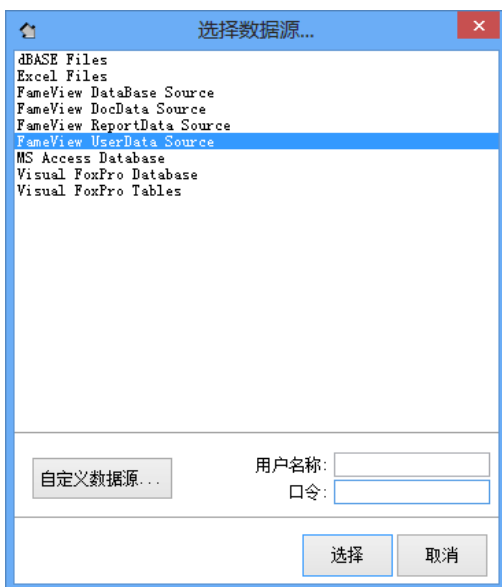
- 画面提供“表格显示”组件：



- 组态对话框：



- 执行<数据源>按钮, 选择或定义数据源：



- [1]. “控制面板->管理工具”中定义ODBC数据源；
[2]. 远程数据源须输入用户名称、口令才允许访问；

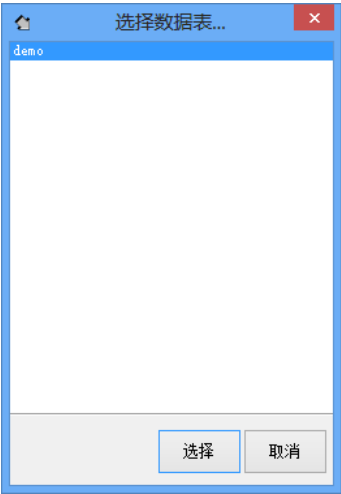
[3]. 执行<自定义数据源>按钮, 定义DSN描述型数据源:



[4]. 或直接输入DSN描述型数据源:

数据库	数据源描述
SQL	Driver={SQL Server};Server=(local);Database=UserDatabase;Uid=;Pwd=;
Access	Driver={Microsoft Access Driver (*.mdb)};Dbq=c:\mydb.mdb;Uid=;Pwd=;
Oracle	Driver={Microsoft ODBC for Oracle};Server=192.168.1.100;Uid=;Pwd=;

□ 数据表, 必选项, 选择数据表或视图:



□ 过滤条件, 可选项, 过滤显示数据表部分内容;

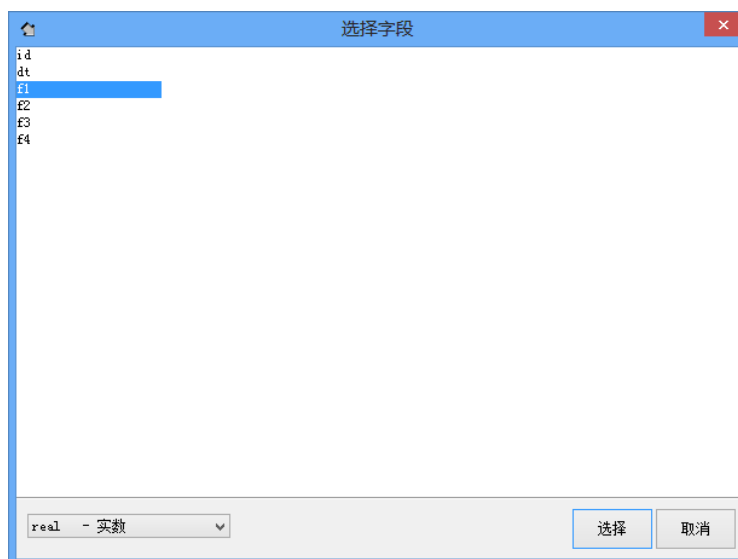
例如:		[dt时间字段]
最近2小时内记录	dt>DateAdd(HOUR, -2, GetDate())	
最近1天内记录	dt>DateAdd(DAY, -1, GetDate())	
最近30分钟内记录	dt>DateAdd(MINUTE, -30, GetDate())	
间隔1分钟记录	DATEDIFF(SECOND, '2013-1-1', dt)%60=0	
间隔10分钟记录	DATEDIFF(SECOND, '2013-1-1', dt)%600=0	
间隔1小时整点记录	DATEDIFF(SECOND, '2013-1-1', dt)%3600=0	
符合某条件记录内容	f1=1 and f2>100	
	f1=1 and dt>DateAdd(HOUR, -2, GetDate())	
	f1=1 and DATEDIFF(SECOND, '2013-1-1', dt)%3600=0	

□ 排序字段, 可选项, 根据某字段使记录进行排序;
排序字段名段名称后加DESC, 进行降序排序, 否则进行升序排序;

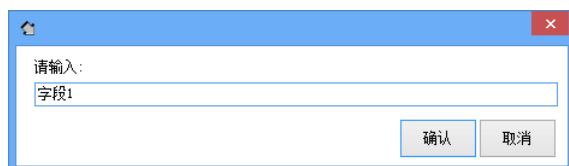
- 设置字段表格, 显示数据表内容;

序号	字段名称	描述	类型	长度	增加记录变量	读取记录变量	修改记录变量	对齐
1	dt	时间	datetime	26				自动
2	f1	字段1	real. 2	15				左
3	f2	字段2	real. 3	15				中
4	f3	字段3	real	15				右
5								

- [1]. 鼠标双击表格第2列, 从字段列表中选择字段:



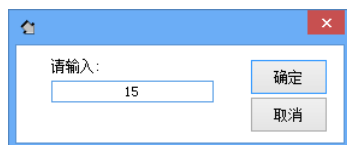
- [2]. 鼠标双击表格第3列, 设置字段描述或表格标题:



- [3]. 表格第4列为字段类型, 选择字段名称时自动匹配字段类型;

字段类型为real或double浮点数时, 双击鼠标切换设置数值小数位数(0-5);

- [4]. 鼠标双击表格第5列, 设置字段内容最大字符长度:



- [5]. 表格第9列为字段内容水平对齐方式: 自动、左、中、右, 双击鼠标切换对齐方式;

- [6]. 运行状态时表格显示:

时间	字段1	字段2	字段3
2013-10-29 09:53:00	1.00	8216.804	113
2013-10-29 09:53:00	2.00	8216.804	9988
2013-10-29 09:53:00	3.00	8216.804	0.92521
2013-10-29 09:53:10	1.00	1842.708	241
2013-10-29 09:53:10	2.00	1842.708	9760

[7]. 执行脚本函数, 更新表格内容:

```
UserDbListObj.Update "objName", ""
```

```
UserDbListObj.Update "objName", "dt>=DateAdd(Hour, -1, GetDate())"
```

[8]. 执行脚本函数, 删除但前选择 (n=0) 或指定 (n>0) 记录:

```
UserDbListObj.Delete "objName", n
```

□ 设置字段表格, 追加数据表内容:

序号	字段名称	描述	类型	长度	追加记录变量	读取记录变量	修改记录变量	对齐
1	dt	时间	datetime	26	VT. %DATETIME			自动
2	f1	字段1	real. 2	15	VA. %VA1			左
3	f2	字段2	real. 3	15	VA. %VA2			中
4	f3	字段3	real	15	VA. %VA3			右

[1]. 鼠标双击表格第6列, 设置各字段对应变数;

[2]. 调用脚本函数, 把变量值作为记录内容, 追加到数据表中:

```
UserDbListObj.AddNewFromVarValue "objName"
```

□ 设置字段表格, 读取数据表内容:

序号	字段名称	描述	类型	长度	追加记录变量	读取记录变量	修改记录变量	对齐
1	dt	时间	datetime	26		VT. %VT1		自动
2	f1	字段1	real. 2	15		VA. %VA1		左
3	f2	字段2	real. 3	15		VA. %VA2		中
4	f3	字段3	real	15		VA. %VA3		右

[1]. 鼠标双击表格第7列, 设置各字段对应变数;

[2]. 调用脚本函数, 把当前选择记录 (n=0) 或指定记录 (n>0) 内容, 赋值到变量:

```
UserDbListObj.SetVarValueFromField "objName", n
```

□ 设置字段表格, 修改数据表内容:

序号	字段名称	描述	类型	长度	追加记录变量	读取记录变量	修改记录变量	对齐
1	dt	时间	datetime	26			VT. %DATETIME	自动
2	f1	字段1	real. 2	15			VA. %VA1	左
3	f2	字段2	real. 3	15			VA. %VA2	中
4	f3	字段3	real	15			VA. %VA3	右

[1]. 鼠标双击表格第8列, 设置各字段对应变数;

[2]. 调用脚本函数, 把变量值作为记录内容, 修改当前选择记录 (n=0) 或指定记录 (n>0) 内容:

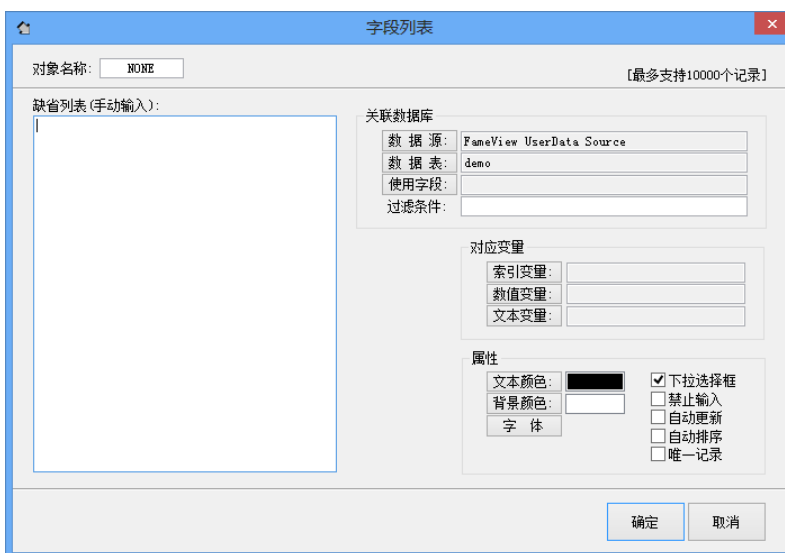
```
UserDbListObj.EditFromVarValue "objName", n
```

25.4 字段列表

- 画面提供“字段列表”组件：



- 组态对话框：



- 缺省列表, 手动输入多行文本(回车换行), 作为缺省列表内容, 可不关联数据库;
- 执行<数据源>按钮, 选择或定义数据源:



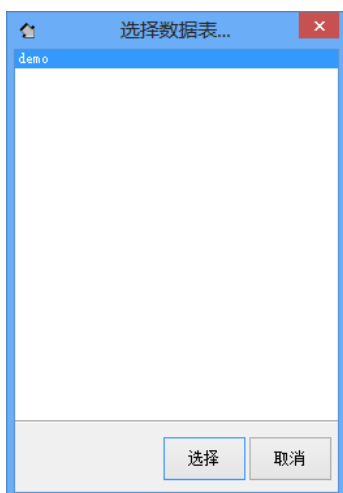
- [1]. “控制面板->管理工具”中定义ODBC数据源;
- [2]. 远程数据源须输入用户名称、口令才允许访问;
- [3]. 执行<自定义数据源>按钮, 定义DSN描述型数据源:



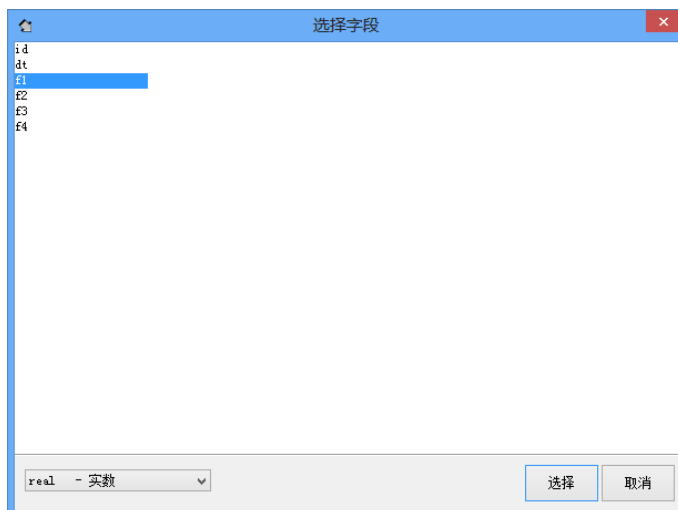
[4]. 或直接输入DSN描述型数据源:

数据库	数据源描述
SQL	Driver={SQL Server};Server=(local);Database=UserDatabase;Uid=;Pwd=;
Access	Driver={Microsoft Access Driver (*.mdb)};Dbq=c:\mydb.mdb;Uid=;Pwd=;
Oracle	Driver={Microsoft ODBC for Oracle};Server=192.168.1.100;Uid=;Pwd=;

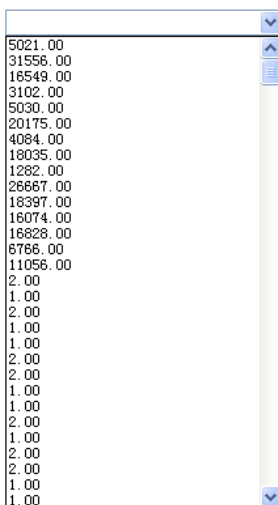
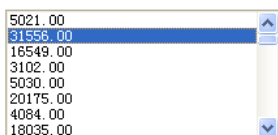
□ 数据表, 必选项, 选择数据表或视图:



□ 使用字段, 选择字段列表对应的字段名称:



- 过滤条件, 可选项, 过滤显示数据表部分内容;
- 对应变量的, 获取字段列表被选择项的内容;
 - [1]. 索引变量, 须是AO/AR/VA变量, 对应字段列表的索引号 (0-10000);
 - [2]. 数值变量, 须是AO/AR/VA变量, 对应字段列表中被选择项的数值内容;
 - [3]. 文本变量, 须是VT变量, 对应字段列表中被选择项的文本内容;
- 显示样式: 列表框、下拉选择框



- [1]. 自动排序, 列表内容进行排序处理;
- [2]. 自动更新, 显示样式为下拉选择框时, 当手动下拉列表框时, 自动更新列表内容;
- [3]. 唯一记录, 过滤列表中重复内容;
- [4]. 调用脚本更新列表内容:

```
DataboxObj.Update objName,odbcName,tableName,fieldName,fieldType,filter
```


25.5 删除数据表记录

□ 实例;

2002-05-05	22:07:51	删除数据...
------------	----------	---------

- [1]. 现有数据库, ODBC 数据源 db1, 包含数据表 T1, 时间字段 dt, 根据时间字段删除记录;
- [2]. 运行数据库添加两个文本变量(VT): VT1、VT2;
- [3]. 画面中增加日期拾取器和时间拾取器, 并把拾取内容保存到 VT1 和 VT2;
- [4]. 画面中增加按钮, 执行脚本删除某时间以前的记录;

□ 数据库类型为 Access 时:

' 得到删除日期、时间

s1=RunSys.GetVTtext("VT1",-1)

s2=RunSys.GetVTtext("VT2",-1)

' 组成格式文本: "Delete * From T1 Where dt<= #2002-1-1 0:0:0#"

s="Delete * From T1 Where dt<= #" : s=s+s1 : s=s+" " : s=s+s2 : s=s+"#"

' 打开数据库DB1

SQLObj.OpenSQLConnect "DB1", "", ""

' 执行SQL语句, 删除数据

SQLObj.SQLExecute s

' 关闭DB1

SQLObj.CloseSQLConnect

□ 数据库类型为SQL Server时:

' 得到删除日期、时间

s1=RunSys.GetVTtext("VT1",-1)

s2=RunSys.GetVTtext("VT2",-1)

' 组成格式文本: "Delete * From T1 Where dt<= '2002-1-1 0:0:0'"

s="Delete * From T1 Where dt<= " ' : s=s+s1 : s=s+" " : s=s+s2 : s=s+"'"

' 打开数据库DB1

SQLObj.OpenSQLConnect "DB1", "", ""

' 执行SQL语句, 删除数据

SQLObj.SQLExecute s

' 关闭DB1

SQLObj.CloseSQLConnect

25.6 数据库脚本编程

- 往记录集中加入记录;

```
UserDB.OpenConnect "My Database","",""
UserDB.OpenRecordset "t1",2,2,2
UserDB.AddNew
UserDB.SetFieldValue "f1",7,"AAA"
UserDB.SetFieldValue "f2",1,1000
UserDB.Update
UserDB.CloseRecordset
UserDB.CloseConnect
```

- 得到数据库中字段的值;

```
UserDB.OpenConnect "My Database","",""
UserDB.OpenRecordset "t1",2,2,2
UserDB.MoveFirst
UserDB.Move(10)
Value=UserDB.GetFieldValue "f2",1,1000
UserDB.CloseRecordset
UserDB.CloseConnect
```

- 执行 SQL 语句:

```
SQLObj.OpenSQLConnect "TestDB","",""
SQLObj.SQLExecute "DELETE FROM demo"
SQLObj.CloseSQLConnect
```

- 数据查询

```
UserDB.OpenConnect "FameView UserData Source","",""
UserDB.SetFilter "f1>=1000"
UserDB.OpenRecordset "demo",2,2,2
If UserDB.IsEmpty()=0 Then
    retValue=UserDB.GetFieldValue("f2",4)
End If
UserDB.CloseRecordset
UserDB.CloseConnect
```

25.7 存储过程脚本编程

- 假设存储过程内容如下：

```
CREATE PROCEDURE [test]
    (@f1 [real], @f2 [real], @f3 [datetime], @f4 [real] output)
AS INSERT INTO [UserDatabase].[dbo].[demo] ( [f1],[f2],[f3])
    VALUES ( @f1,@f2,@f3)
if @@error=0
    set @f4=12.11
else
    set @f4=0.12
GO
```

- 过程调用如下：

```
n=StoredProc.OpenConnect("FameView UserData Source","","")
If n=1 Then
    StoredProc.SetCommandText "test"

    StoredProc.SetInputParamCount 3
    StoredProc.SetInputParam 1,"f1",4,1.23
    StoredProc.SetInputParam 2,"f2",4,2.34
    StoredProc.SetInputParam 3,"f3",7,"2005-12-11 05:00:00"

    StoredProc.SetOutputParamCount 1
    StoredProc.SetOutputParam 1,"f4",4
    StoredProc.Execute

    m=StoredProc.GetOutputValue(1)
    RunSys.SetVarValue VA,"%VA1",-1,m

    StoredProc.CloseConnect
End If
```

25.8 数据表平均统计

- 数据表某天的记录内容进行平均统计, 结果存入其他数据表
- 假设两个数据表 t1 和 t2, 有相同结构:

字段名称	类型
Dt	datetime
f1	real
f2	real
d3	real

- 针对 t1 中近 30 天内的记录, 将每天的数据进行平均统计, 结果存入 t2, 并删除 t1 记录;

[1]. SQL Server 编写存储过程:

```
CREATE PROCEDURE [test]
as
Declare @n int
Declare @ct datetime
set @n=-30
while @n<0
begin
    set @ct=DATEADD(DAY,@n, GETDATE())
    INSERT INTO t2
    SELECT AVG(f1) AS f1, AVG(f2) AS f2, AVG(f3) AS f3, @ct AS dt
    FROM t1
    WHERE (YEAR(dt) = YEAR(@ct) AND (Month(dt) = MONTH(@ct) AND (Day(dt) = DAY(@ct)
    set @n=@n+1
end
delete from t1
GO
```

[2]. 组态软件中调用脚本执行存储过程:

```
n=StoredProc.OpenConnect("FameView UserData Source","","")
If n=1 Then
    StoredProc.SetCommandText "test"
    StoredProc.Execute
    StoredProc.CloseConnect
End If
```

25.9 数据库导出Excel

' 允许输入查询的年/月

MyYear = InputBox("请输入查询年份","")

MyMonth = InputBox("请输入查询月份","")

MyDay = 1

' 检查时间是否合法

If IsDate(MyMonth & "/" & MyDay & "/" & MyYear) = True Then

 NOW_DATE = MyYear + "年" + MyMonth + "月"

 ' 得到开始时间

 StartTime = MyMonth & "/" & MyDay & "/" & MyYear & " 00:00:00"

 ' 得到结束时间

 If MyMonth < 12 Then

 MyMonth = MyMonth + 1

 EndTime = MyMonth & "/" & MyDay & "/" & MyYear & " 00:00:00"

 Else

 MyYear = MyYear + 1

 MyMonth = 1

 EndTime = MyMonth & "/" & MyDay & "/" & MyYear & " 00:00:00"

 End If

' 打开数据库及数据表demo, dt为时间字段

UserDB.OpenConnect "FameView UserData Source","",""

UserDB.SetFilter "dt>=" + StartTime + " And dt<" + EndTime + ""

UserDB.OpenRecordset "demo", 2, 2, 2 ' 打开记录集;

' 检查查询内容是否为空

If UserDB.IsEmpty()=0 Then

 ' 打开Excel文件

 Set ExcelObj = CreateObject("Excel.Application")

 ExcelObj.Visible = True

 ExcelFile = "C:\组态系统\ExcelFile\test.xls"

 ExcelObj.Workbooks.Open ExcelFile

 ExcelObj.Sheets("Sheet1").Select

 ExcelObj.Worksheets(1).Select

 ExcelObj.Cells(1, 3) = NOW_DATE

 ' 导出记录到Excel中

 row=2

 While UserDB.IsEOF()=0

 ExcelObj.Cells(row, 1).Value=UserDB.GetFieldValue("dt", 8)

```

        ExcelObj.Cells(row, 2).Value=UserDB.GetFieldValue("f1", 4)
        ExcelObj.Cells(row, 3).Value=UserDB.GetFieldValue("f2", 4)
        row=row+1
        UserDB.MoveNext
    Wend

    ExcelObj.ActiveWorkbook.SaveAs "C:\组态系统\ExcelFile\"&NOW_DATE&".xls"
    ExcelObj.Quit
    Set ExcelObj=Nothing
End If
' 关闭数据库
UserDB.CloseRecordset
UserDB.CloseConnect
End If

```

25.10 使用ADO访问数据库

' 与数据库(tempdb)建立连接

```

Set conn = CreateObject("ADODB.Connection")
strConn="Driver={SQL Server};Server=(local);Database=tempdb;Uid=sa;Pwd;"
conn.Open strConn

```

' 打开记录集(t1)

```

Set rs= CreateObject("ADODB.Recordset")
strSQL="SELECT * FROM t1"
rs.Open strSQL, conn, 2, 2
' 添加记录
rs.AddNew
rs("f1")="222"
rs("f2")=222
rs.Update

```

' 得到记录

```

f1=rs("f1")
f2=rs("f2")

```

' 关闭连接与记录集

```

rs.Close
conn.Close
Set rs=Nothing
Set conn=Nothing

```

25.11 查询数据表最近时间段记录

例 1(查询最近 3 分钟数据):

```
Select * from demo where dt>DateAdd(MINUTE,-3,GetDate())
```

例 2(查询最近 3 小时数据):

```
Select * from demo where dt>DateAdd(HOUR,-3,GetDate())
```

例 3(查询最近 3 天数据):

```
Select * from demo where dt>DateAdd(DAY,-3,GetDate())
```

25.12 获取数据表最近时间段统计值

'[1]得到分钟时间段

```
x=RunSys.GetVarValue(VA,"%VA1",-1)
```

'[2]打开数据库

```
Set conn = CreateObject("ADODB.Connection")
```

```
strConn="Driver={SQL Server};Server=(local);Database=UserDatabase;Uid=;Pwd=;"
```

```
conn.Open strConn
```

```
Set rs= CreateObject("ADODB.Recordset")
```

'[3]进行统计查询

```
strSQL="SELECT Avg(f1) as f11 FROM demo where dt>=DateAdd(Minute,-"&CStr(x)&",GetDate())"
```

```
rs.Open strSQL, conn, 2, 2
```

```
If rs.EOF=False Or rs.BOF=False Then
```

```
    x=rs("f11")
```

```
else
```

```
    x=0
```

```
End If
```

'[4]关闭数据库

```
rs.Close
```

```
conn.Close
```

```
Set rs=Nothing
```

```
Set conn=Nothing
```

'[5]获取统计值

```
bValue=RunSys.OpenDatabase(3000)
```

```
If bValue=1 Then
```

```
    RunSys.SetVarValue VA,"%VA2",-1,x
```

```
    RunSys.CloseDatabase
```

```
End If
```

25.13 获取数据表某时间段字段总和

' [1]初始化变量

```
strStartDate="" : strStartTime="" : strEndDate="" : strEndTime="" : strPt="" : ntype=0
```

' [2]得到时间及选择

```
bValue=RunSys.OpenDatabase(3000)
```

```
If bValue=1 Then
```

```
    strStartDate=RunSys.GetVTtext("开始日期",-1)
```

```
    strStartTime=RunSys.GetVTtext("开始时间",-1)
```

```
    strEndDate=RunSys.GetVTtext("结束日期",-1)
```

```
    strEndTime=RunSys.GetVTtext("结束时间",-1)
```

```
    strPt=RunSys.GetVTtext("%VT1",-1)
```

```
    ntype=RunSys.GetVarValue(VA,"%VA1",-1)
```

```
    ntype=ntype+1
```

```
    RunSys.CloseDatabase
```

```
End If
```

' [3]打开数据库

```
Set conn = CreateObject("ADODB.Connection")
```

```
strConn="Driver={SQL Server};Server=(local);Database=UserDatabase;Uid=;Pwd=;"
```

```
conn.Open strConn
```

```
Set rs= CreateObject("ADODB.Recordset")
```

' [4]执行查询

```
strSQL="SELECT SUM(f2) as f22 FROM demo "
```

```
strSQL=strSQL & "Where dt>=" & strStartDate & " " & strStartTime & " AND dt<=" & strEndDate & " " & strEndTime & " AND "
```

```
strSQL=strSQL & "pt=" & strPt & " AND f1=" & CStr(ntype)
```

```
' ret=MsgBox(strSQL,vbYesNo+vbInformation+vbSystemModal,"标题...")
```

```
rs.Open strSQL, conn, 2, 2
```

' [5]获得结果

```
If rs.EOF=False And rs.BOF=False Then
```

```
    RA1=rs("f22")
```

```
End If
```

' [6]结束

```
rs.Close
```

```
conn.Close
```

```
Set rs=Nothing
```

```
Set conn=Nothing
```

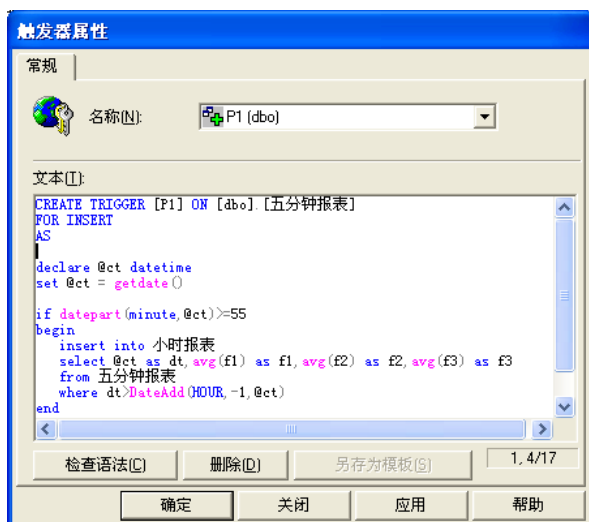

25.14 分钟数据表生成小时和日数据表

□ 分钟、小时、日数据表格式:

	列名	数据类型	长度	允许空
▶	id	bigint	8	
	dt	datetime	8	✓
	f1	real	4	✓
	f2	real	4	✓
	f3	real	4	✓

□ 分钟表建立触发器 P1:

- [1]. 某小时最后 1 次追加记录时, 将最近 1 小时内的统计结果插入小时数据表;
- [2]. 选择分钟表, 执行右键菜单“所有任务->管理触发器”, 编写触发器 P1:



□ 小时表建立触发器 P2:

- [1]. 某天最后 1 次追加记录时, 将最近 1 天内统计结果插入日数据表;
- [2]. 选择小时表, 执行右键菜单“所有任务->管理触发器”, 编写触发器 P2:



25.15 数据库动态增加数据表

' 打开数据库

```
Set conn = CreateObject("ADODB.Connection")
strConn="Driver={SQL Server};Server=(local);Database=UserDatabase;Uid=;Pwd=;"
conn.Open strConn
```

' 增加表: TableName (ID, f1, f2, f3, f4)

```
sqlDesc="CREATE TABLE TableName "& _
    "("& _
    "ID Int IDENTITY (1,1), "& _
    "f1 nVarChar(50) default '默认值' null, "& _
    "f2 real default 0, "& _
    "f3 datetime, "& _
    "f4 Int default 0"& _
    ")"
conn.Execute sqlDesc
```

' 建立时间索引: Index1 (f3)

```
sqlDesc="CREATE CLUSTERED INDEX index1 On Tablename(f3)"
conn.Execute sqlDesc
```

' 关闭数据库

```
conn.Close
Set conn=Nothing
```

25.16 异步批量删除数据库记录

后台异步删除某数据库中某表的批量记录:

```
UserDB.DeleteBatchRecord dbName, tableName, filter, orderby, maxRows
```

dbName, 数据库名称;
 tableName, 数据表名称;
 filter, 过滤条件, 可选项;
 orderby, 排序字段, 可选项;
 maxRows, 允许删除最大行数, 取值范围1-100000;

例1: UserDB.DeleteBatchRecord "FameView UserData Source", "demo", "", "", 1000

例2: UserDB.DeleteBatchRecord "FameView UserData Source", "demo", "dt<'2009-1-1'", "", 1000

例3: dbName="Driver={SQL Server};Server=(local);Database=UserDatabase;Uid=;Pwd=;"

```
UserDB.DeleteBatchRecord dbName, "demo", "dt<'2009-1-1' AND f1=1", "", 1000
```

25.17 异步批量修正数据库记录

在后台修正某数据库中某表的批量记录(每次可最多修改5个数值字段):

```
UserDB.ModifyBatchRecord dbName, tableName, filter, orderby, maxRows,
    f1, f1MinValue, f1MaxValue, f2, f2MinValue, f2MaxValue, f3, f3MinValue, f3MaxValue,
    f4, f4MinValue, f4MaxValue, f5, f5MinValue, f5MaxValue
```

dbName, 数据库名称;

tableName, 数据表名称;

filter, 过滤条件, 可选项;

orderby, 排序字段, 可选项;

maxRows, 允许查询最大行数, 取值范围1-1000000;

f1, 数值字段名称, 只对填写名称的字段进行修改;

f1MinValue和f1MaxValue, 最小值和最大值, 限定字段范围,

如果字段值不在此范围内, 则用此范围内的某随机数进行修正替代;

如果希望用某固定值替代, 使最小值和最大值相等, 并为某固定值即可;

最小值和最大值格式, 决定数值小数位数(0-6):

如最小值为"200", 则自适应小数位;

如最小值为"200.", 则无小数位;

如最小值为"200.0", 则保留1个小数位;

如最小值为"200.00", 则保留2个小数位;

27.18 判断数据库文件尺寸, 异步批量删除数据库记录

根据某数据库文件是否大于某限制尺寸, 后台异步删除数据库某表批量记录:

```
UserDB.DeleteBatchRecord dbFilename, maxSize, dbName, tableName, filter, orderby, maxRows
```

dbFileName, 数据库对应的文件路径名称, 如果不填写根目录则认为在组态系统目录下;

maxSize, 数据库限制尺寸, 单位为MB;

dbName, 数据库名称;

tableName, 数据表名称;

filter, 过滤条件, 可选项;

orderby, 排序字段, 可选项;

maxRows, 允许删除的最大行数, 取值范围1-100000;

例1: dbFile="c:\temp\db1.mdf"

dbName="FameView UserData Source"

```
UserDB.DeleteBatchRecordOnDBSize dbFile, 1500, dbName, "demo", "", "", 1000
```

例2: dbFile="MyFile\userdatabase.mdf"

dbName="FameView UserData Source"

```
UserDB.DeleteBatchRecordOnDBSize dbFile, 1500, dbName, "demo", "dt<' 2009-1-1' ", "", 1000
```

例3: dbFile="MyFile\userdatabase.mdf"

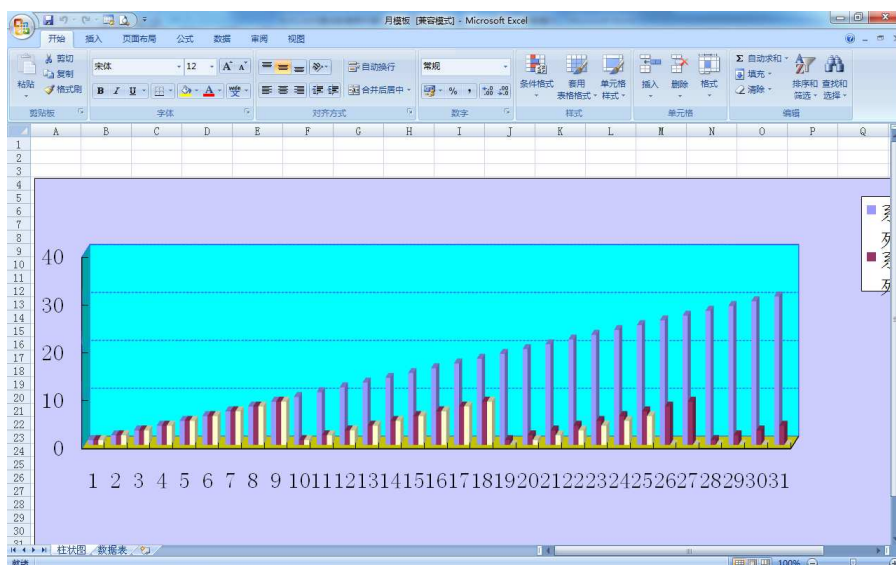
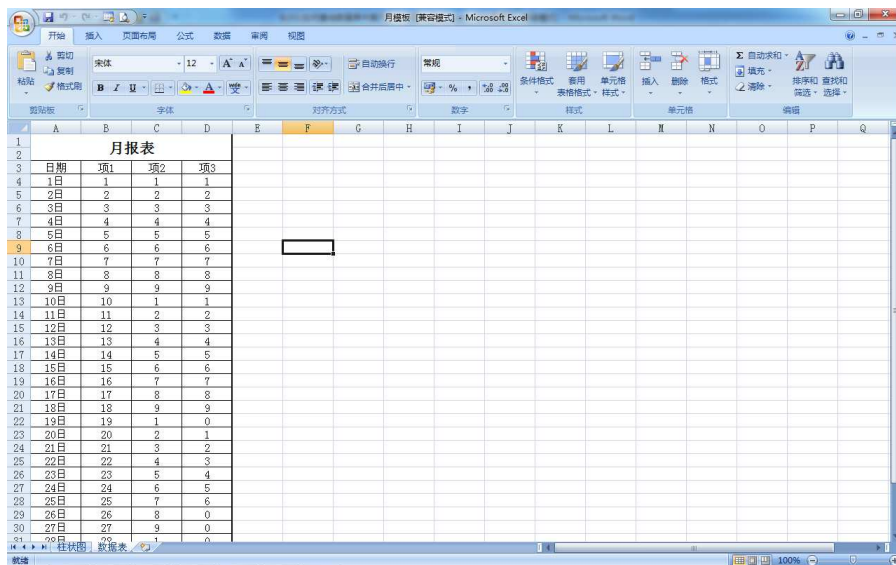
dbName=" Driver={SQL Server};Server=(local);Database=UserDatabase;Uid=;Pwd=;"

```
UserDB.DeleteBatchRecord dbFile, 1500, dbName, "demo", "dt<' 2009-1-1' AND f1=1", "", 1000
```

27.19 查询数据表某月日平均值并通过Excel柱状图显示

□ 制作Excel模板文件(如月模板.xls):

包含两个表, 1个表名称为数据表暂放数据, 另1个表名称为柱状图显示柱状图:



□ 画面中通过脚本实现数据库查询和Excel数据显示:

On Error Resume Next

' [1]得到年份和月份

YearNo=2010

MonthNo=7

' [2]打开数据库

Set conn = CreateObject("ADODB.Connection")

strConn="Driver={SQL

```
Server};Server=JKGONG_FAMEVIEW\SQLEXPRESS;Database=UserDatabase;Uid=;Pwd=;"
conn.Open strConn
```

' [3]打开Excel模板文件

```
Set excelObj = CreateObject("Excel.Application")
excelObj.Visible = False
excelObj.Application.DisplayAlerts=False
excelObj.Workbooks.Open "C:\FameView\MyFile\月模板.xls"
excelObj.Sheets("数据表").Select
```

' [4]执行查询,得到每天的平均值(v1,v2,v3),并写入Excel文件的数据表

```
Set rs= CreateObject("ADODB.Recordset")
For n=1 To 31
    s1=""&CStr(YearNo)&"-"&CStr(MonthNo)&"-"&CStr(n)&" 00:00:00" ' 开始时间
    s2=""&CStr(YearNo)&"-"&CStr(MonthNo)&"-"&CStr(n)&" 23:59:59" ' 结束时间
    strSQL="SELECT Avg(f1) as f11, Avg(f2) as f21, Avg(f3) as f31 FROM demo where dt>="&s1&"
    and dt<="&s2
    rs.Open strSQL, conn, 2, 2
    v1=0:v2=0:v3=0
    If rs.BOF=False And rs.EOF=False Then
        If IsNull(rs("f11"))=False Then v1=rs("f11")
        If IsNull(rs("f21"))=False Then v2=rs("f21")
        If IsNull(rs("f31"))=False Then v3=rs("f31")
    End If
    rs.Close
    excelObj.Cells(n, 1) = v1
    excelObj.Cells(n, 2) = v2
    excelObj.Cells(n, 3) = v3
Next
' [5]关闭数据库
conn.Close
Set rs=Nothing
Set conn=Nothing
' [6]显示Excel窗口
excelObj.Sheets("柱状图").Select
excelObj.Visible = True
excelObj.Application.WindowState=-4137
Set excelObj=Nothing
```

25.20 以可变时间间隔查询显示数据表记录

□ 制作画面：

查询开始时间: 2010-08-05 08:28:29

查询时间长度: 1 天

查询时间间隔: 6 秒

开始查询

序号	dt	f1	f2	f3
----	----	----	----	----

□ 查询按钮调用脚本:

’得到输入的参数

```
m=RunSys.GetVarValue(VA,"%VA1",-1) ' 查询长度数值
```

`x=RunSys.GetVarValue(VA, "%VA3", -1)` ' 查询长度单位:0=小时, 1=天, 2=周, 3=月

’得到以小时为基本单位的查询长度

Select Case x

Case 0 $m=m*1$

Case 1 $m=m*24$

Case 2 $m=m*24*7$

Case 3 $m=m*24*30$

End Select

```
n=RunSys.GetVarValue(VA,"%VA2",-1) ' 查询间隔数值
```

x=RunSys.GetVarValue(VA,"%VA4",-1) ' 查询间隔单位:0=秒,1=分钟,2=小时,3=天

Select Case **x**

Case 0 $n=n*1$

Case 1 $n=n*60$

Case 2 $n=n*3600$

Case 3 $n=n*3600*24$

End Select

' 得到开始日期和时间

```
s=RunSys.GetVTtext("%VT1",-1)
```

```
t1=s+" "
```

```
s=RunSys.GetVTtext("%VT2",-1)
```

```
t1=t1+s
```

```
If m<=0 Then m=1
```

```
If n<=0 Then n=1
```

' 得到开始时间

```
DatetimeObj.SetInitTimeEx t1
```

```
t1=DatetimeObj.GetFormatTime("' %Y-%m-%d %H:%M:%S' ")
```

' 得到结束时间

```
DatetimeObj.MoveBack 0,m,0,0
```

```
t2=DatetimeObj.GetFormatTime("' %Y-%m-%d %H:%M:%S' ")
```

' 进行查询,并依据间隔挑选

```
s="dt>="&t1&" AND dt<="&t2&" AND DATEDIFF(SECOND,"&t1&",dt)%"&CStr(n)&"=0"
```

```
UserDbListObj.Update "AAA",s
```

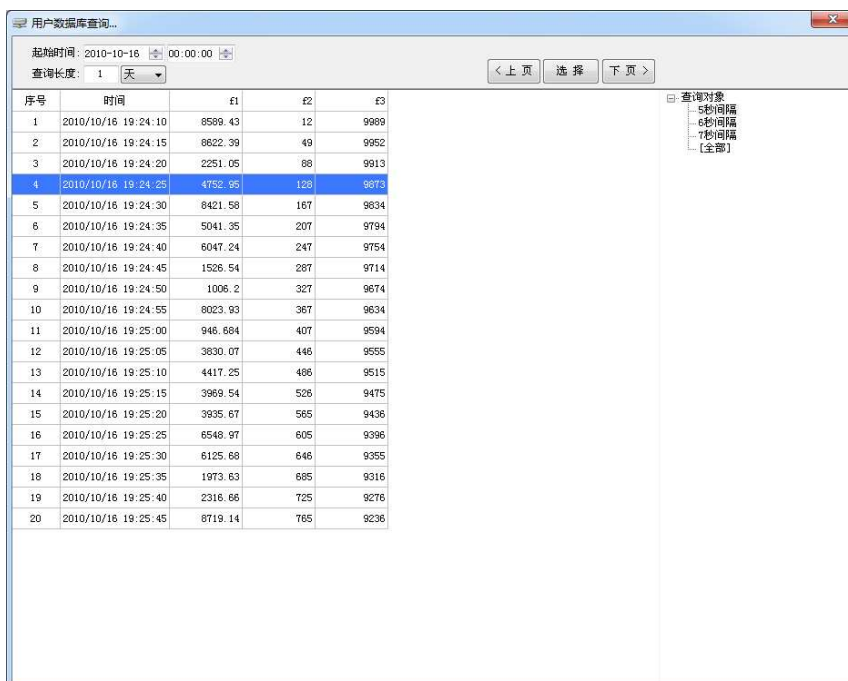
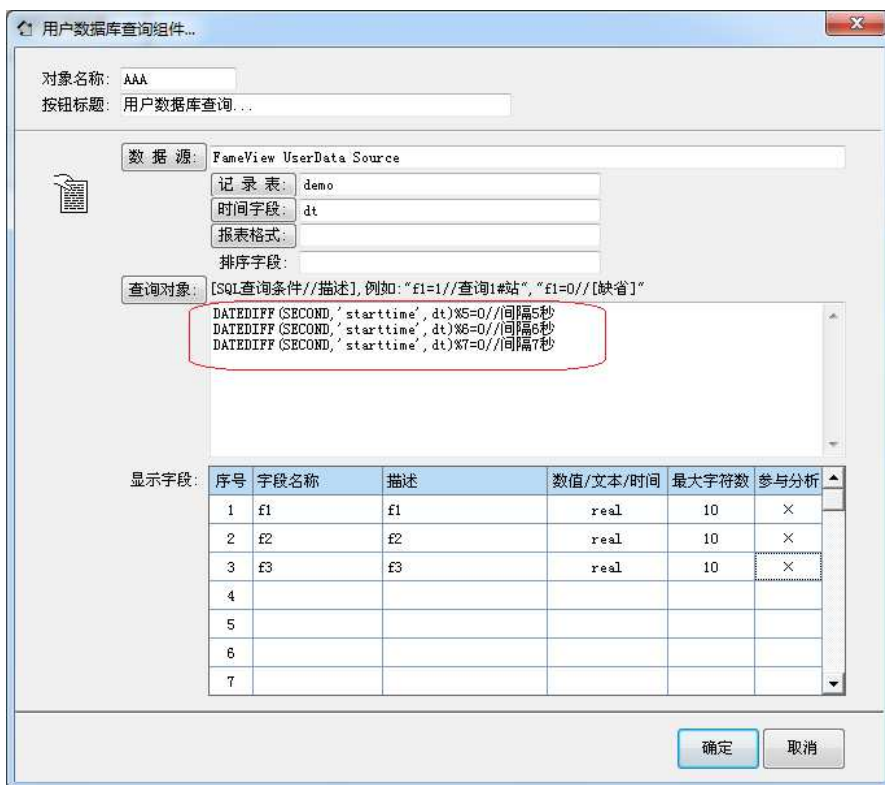
□ 执行查询结果:

查询开始时间:	2010-08-05	08:00:00	
查询时间长度:	1	天	
查询时间间隔:	6	秒	
			开始查询

序号	dt	f1	f2	f3
9	2010/8/5 20:25:38	7796.56	510	9491
10	2010/8/5 20:25:44	9038.67	567	9434
11	2010/8/5 20:25:50	3105.56	623	9378
12	2010/8/5 20:25:56	5355.69	681	9320
13	2010/8/5 20:26:02	4165.17	738	9263
14	2010/8/5 20:26:08	2425.92	795	9206
15	2010/8/5 20:26:14	2030.09	852	9149
16	2010/8/5 20:26:20	2112.8	908	9093
17	2010/8/5 20:26:26	7041.54	965	9036
18	2010/8/5 20:26:32	6230.35	1022	8979
19	2010/8/5 20:26:38	3978.7	1079	8922
20	2010/8/5 20:26:44	3836.18	1136	8865
21	2010/8/5 20:26:50	4948.88	1193	8808
22	2010/8/5 20:26:56	1347.39	1250	8751
23	2010/8/5 20:27:02	3481.55	1307	8694
24	2010/8/5 20:27:08	2627.64	1364	8637
25	2010/8/5 20:27:14	6718.65	1421	8580
26	2010/8/5 20:27:20	8084.05	1478	8523
27	2010/8/5 20:27:26	3105.56	1534	8467

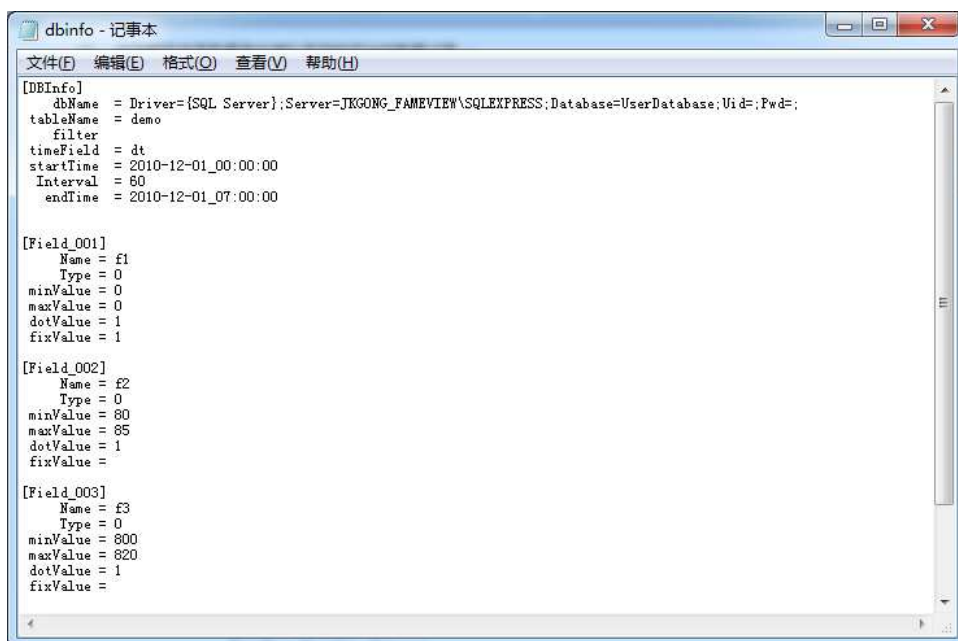
25.21 设置数据查询组件查询时间间隔

- ☐ 数据表存储以1秒为间隔的数据, 需要查询间隔5秒或6秒的记录内容;
- ☐ 数据查询控件定义查询条件:



25.22 数据库增补记录

- ☐ 向某数据表中增补某段时间内的数据记录;
- ☐ 此功能需要具有数据增补授权的加密狗支持, 否则运行速度较慢, 间隔10秒;
- ☐ 制作文本文件, 如dbInfo.txt, 内容格式必须为:



[DBInfo]下定义数据库基本信息:

dbName, 数据库连接串, 支持SQL, Oracle, Access数据库;

可为DSN串或ODBC数据源, 如:

Driver={SQL Server};Server=(local);Database=UserDatabase;Uid=;Pwd=;

tableName, 数据表名称;

filter, 过滤条件, 如f1=1;

timeField, 时间字段名称;

startTime, 开始时间, 格式必须为YYYY-mm-dd_HH:MM:SS

Interval, 追加记录的时间间隔, 以秒为单位;

如果在间隔时间内存在记录则不增补记录;

endTime, 结束时间, 格式必须为YYYY-mm-dd_HH:MM:SS;

开始与结束时间不建议大于1年;

[Field_xxx]下可定义多个字段信息, xxx的取值范围为1-512, 即最多支持512个字段:

Name, 字段名称;

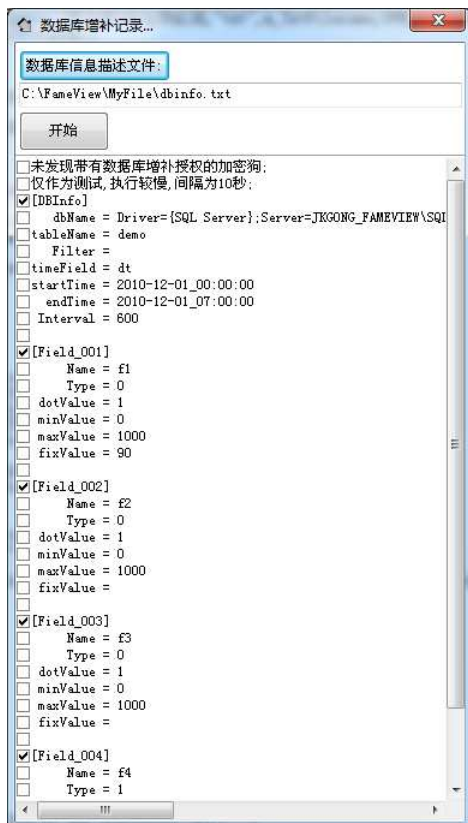
Type, 字段类型, 0=数值, 100=文本;

minValue和maxValue, 最小值和最大值, 取之间的随机数作为存储值;

dotValue, 存储值的小数点精度, 取值为0-6, 如果不填则尽量多的小数位;

fixValue, 固定值, 此值不为空时作为存储值;

- 通过实用工具“[24]. 增补数据库数据”, 并调用上面文本文件进行增补操作:



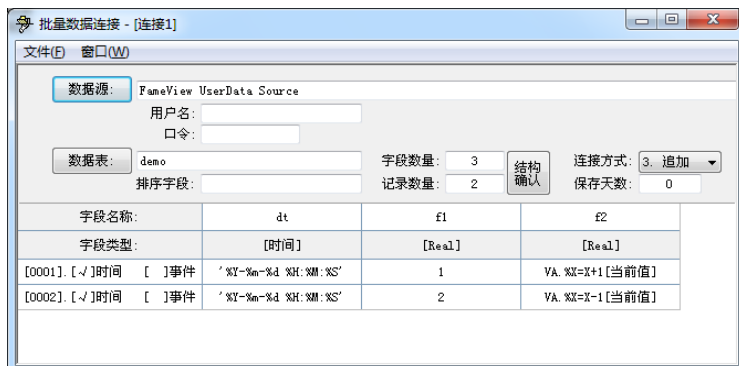
- 或画面或全局脚本中执行脚本: UserDB.AddBatchRecord "dbInfo.txt"

25.23 通过触发器计算增加值

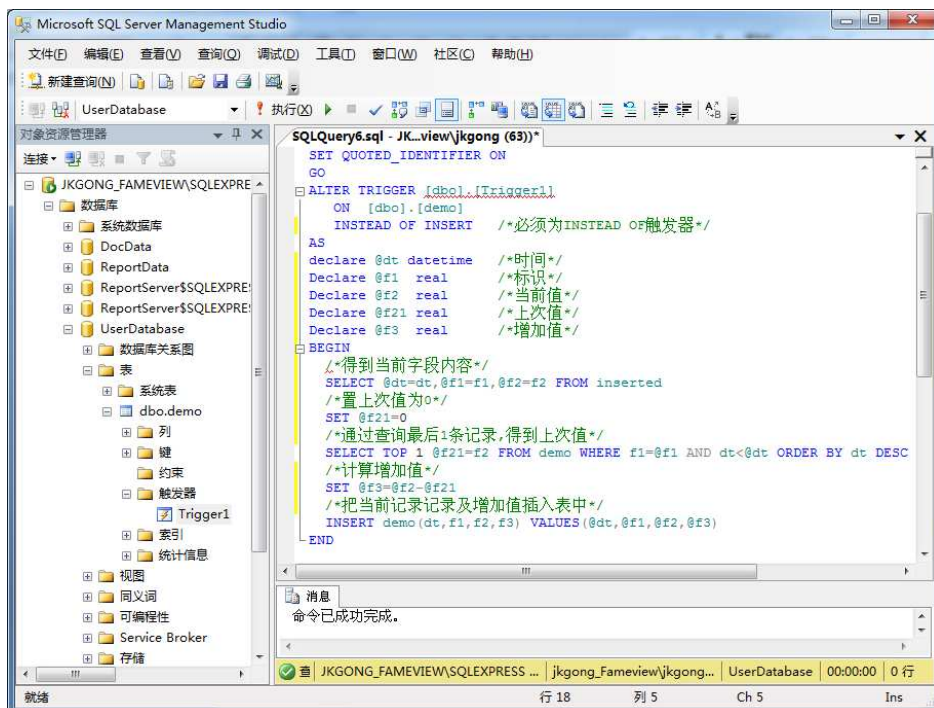
- 假设有 demo 表, 字段 dt (时间), f1 (标识), f2 (当前值), f3 (增加值);

列名	数据类型	允许 Null 值
id	bigint	<input type="checkbox"/>
dt	datetime	<input type="checkbox"/>
f1	real	<input checked="" type="checkbox"/>
f2	real	<input checked="" type="checkbox"/>
f3	real	<input checked="" type="checkbox"/>

- 建立批量数据库连接, 以 60 秒为间隔定期把两批数据当前值追加到 demo 表:



- Demo 数据表建立触发器, 存储前执行此触发器, 计算得到增加值, 存入数据表:



触发器详细内容:

```
CREATE TRIGGER [dbo].[Trigger1] ON [dbo].[demo]
INSTEAD OF INSERT      /*必须为INSTEAD OF触发器*/
AS
Declare @dt datetime    /*时间*/
Declare @f1 real        /*标识*/
Declare @f2 real        /*当前值*/
Declare @f21 real       /*上次值*/
Declare @f3 real        /*增加值*/
BEGIN
    /*得到当前字段内容*/
    SELECT @dt=dt,@f1=f1,@f2=f2 FROM inserted
    /*置上次值为0*/
    SET @f21=0
    /*通过查询最后条记录,得到上次值*/
    SELECT TOP 1 @f21=f2 FROM demo WHERE f1=@f1 AND dt<@dt ORDER BY dt DESC
    /*计算增加值*/
    SET @f3=@f2-@f21
    /*把当前记录记录及增加值插入表中*/
    INSERT demo(dt, f1, f2, f3) VALUES (@dt, @f1, @f2, @f3)
END
```

25.24 脚本使用ADO访问数据库判断是否成功

- 微软提供的 ADO 访问数据库时, 不能返回访问结果, 如果访问失败则直接导致代码异常;
- 如果在脚本前增加下面语句, 可以使脚本出现异常时仍能够继续运行:

```
On Error Resume Next
```

- 继续运行的脚本通过判断 Err.Number 错误对象值是否为零, 可得知是否出现了错误;

- 举例如下:

```
On Error Resume Next
```

```
Err.Clear
```

```
ErrString=""
```

```
' 打开数据库连接
```

```
Set conn = CreateObject("ADODB.Connection")
```

```
strConn="Driver={SQL
```

```
Server};Server=(local)\SQLEXPRESS;Database=UserDatabase;Uid=;Pwd=;"
```

```
conn.Open strConn
```

```
If Err.Number=0 Then
```

```
' 打开数据表
```

```
Set rs= CreateObject("ADODB.Recordset")
```

```
strSQL="SELECT * FROM demo"
```

```
rs.Open strSQL, conn, 2, 2
```

```
If Err.Number=0 Then
```

```
' 添加新记录
```

```
rs.AddNew:rs("f1")=111:rs("f2")=222:rs("dt")=Now():rs.Update
```

```
If Err.Number=0 Then
```

```
ErrString="存储数据成功!"
```

```
Else
```

```
ErrString="存储数据失败"
```

```
End If
```

```
rs.Close
```

```
Else
```

```
ErrString="打开数据表失败!"
```

```
End If
```

```
Set rs=Nothing
```

```
conn.Close
```

```
Else
```

```
ErrString="打开数据库失败!"
```

```
End If
```

```
Set conn=Nothing
```

```
MsgBox ErrString
```

25.25 执行脚本备份SQL数据库

完全备份:

```
FileObj.CreateFolder("d:\datatemp")
SQLObj.OpenSQLConnect "FameView UserData Source", "", ""
SQLObj.SQLExecute "BACKUP DATABASE UserDatabase TO disk='d:\datatemp\UserDatabase.bak'"
SQLObj.CloseSQLConnect
```

差异化备份:

```
bValue=FileObj.CreateFolder("d:\datatemp")
SQLObj.OpenSQLConnect "FameView UserData Source", "", ""
SQLObj.SQLExecute "BACKUP DATABASE UserDatabase TO disk='d:\datatemp\UserDatabase.bak' WITH
DIFFERENTIAL"
SQLObj.CloseSQLConnect
```

25.26 脚本使用ADO执行存储过程

' 命令类型

```
adCmdStoredProc =4
```

' 参数类型

```
adParamInput = 1
```

```
adParamOutput = 2
```

```
adParamInputOutput = 3
```

```
adParamReturnValue = 4
```

' 参数数值类型

```
adTinyInt = 16
```

```
adSmallInt = 2
```

```
adInteger = 3
```

```
adSingle = 4
```

```
adDouble = 5
```

```
adBoolean = 11
```

```
adBigInt = 20
```

```
adChar = 129
```

```
adVarChar = 200
```

```
On Error Resume Next
```

' 打开数据库连接

```
Set adoConn = CreateObject("ADODB.Connection")
```

```
strConn="Driver={SQL Server};Server=(local);Database=UserDatabase;Uid=;Pwd=;"
```

```
adoConn.Open strConn
```

```
If Err.Number=0 Then
```

```
Set adoComm = CreateObject("ADODB.Command")
```

```
Set adoComm.ActiveConnection = adoConn
```

```

adoComm.CommandText = "Proc1" ' 存储过程名称
adoComm.CommandType = adCmdStoredProc
' 输入参数
Set prm = adoComm.CreateParameter("parameter1", adTinyInt, adParamInput, , "1")
adoComm.Parameters.Append prm
' 输出参数
Set prm = adoComm.CreateParameter("parameter2", adInteger, adParamOutput)
adoComm.Parameters.Append prm
' 执行存储过程
Set mRst = adoComm.Execute
If Err.Number=0 Then
    ReturnValue = adocomm.Parameters(0)
End If
adoComm=Nothing
adoConn.Close
End If
Set adoConn=Nothing

```

25.27 得到数据表时间字段最早及最新日期

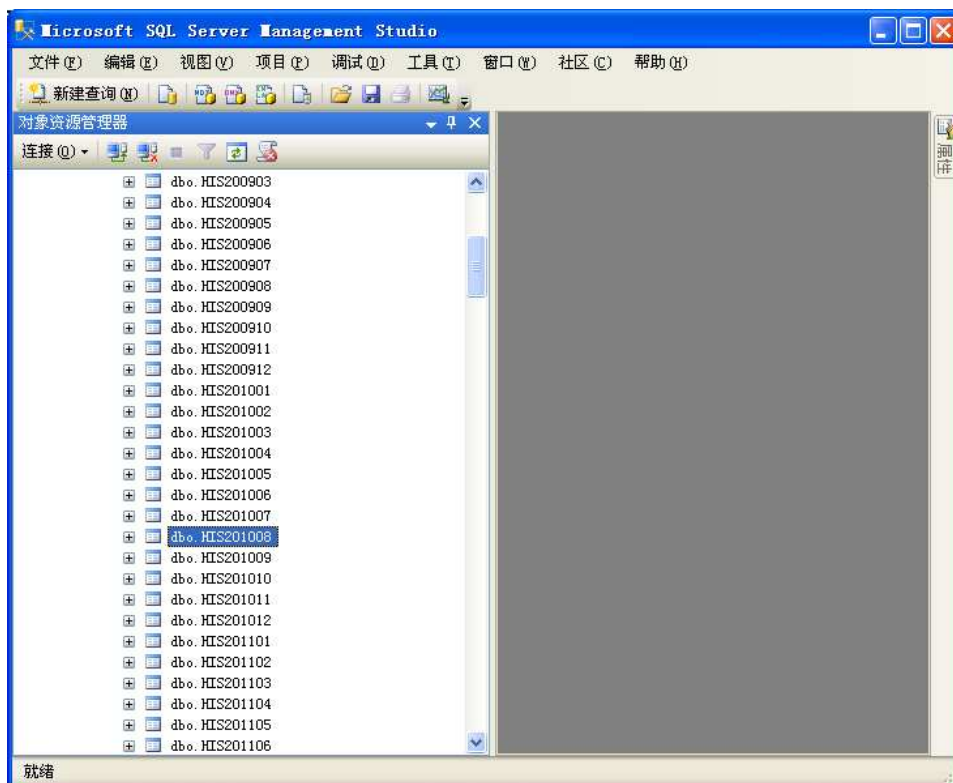
```

' 打开数据库连接
Set conn = CreateObject("ADODB.Connection")
strConn="Driver={SQL
Server};Server=(local)\SQLEXPRESS;Database=UserDatabase;Uid=;Pwd=;"
conn.Open strConn
' 打开数据表
Set rs= CreateObject("ADODB.Recordset")
strSQL="SELECT MIN(dt) as oldTime, MAX(dt) as newTime FROM demo"
rs.Open strSQL, conn, 2, 2
' 判断打开的数据表是否为空
If rs.EOF=False And rs.BOF=False Then
    s1=rs("oldTime")
    s2=rs("newTime")
    MsgBox s1&"", "&s2
End If
' 关闭数据表及连接
rs.Close
conn.Close
Set rs=Nothing
Set conn=Nothing

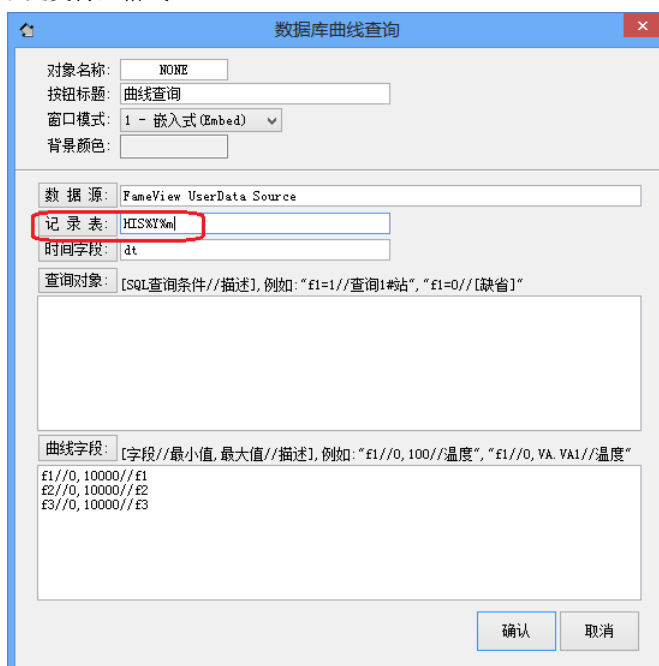
```

25.28 查询月数据表(每月对应1个数据表)

- 数据库结构: 每月生成一个表, 数据表名称格式为HISYYYYmm, YYYY=年, mm=月;



- 组态中对应数据库查询组件的数据表需要设置为HIS%Y%m即可;
- 共4处支持此格式:



数据库曲线查询

对象名称: NONE

按钮标题: 曲线查询

窗口模式: 0 - 弹出式 (Popup) ▼

背景颜色:

数据源: FameView UserData Source

记录表: HISXY%m

时间字段:

查询对象: [SQL查询条件//描述], 例如: "f1=1//查询1#站", "f1=0//[缺省]"

曲线字段: [字段//最小值, 最大值//描述], 例如: "f1//0, 100//温度", "f1//0, VA, VA1//温度"

确认 取消

数据库表格查询...

项目名称:

数据源: FameView UserData Source

数据表: HISXY%m

时间字段:

缺省查询时间长度: 1 小时 ▼

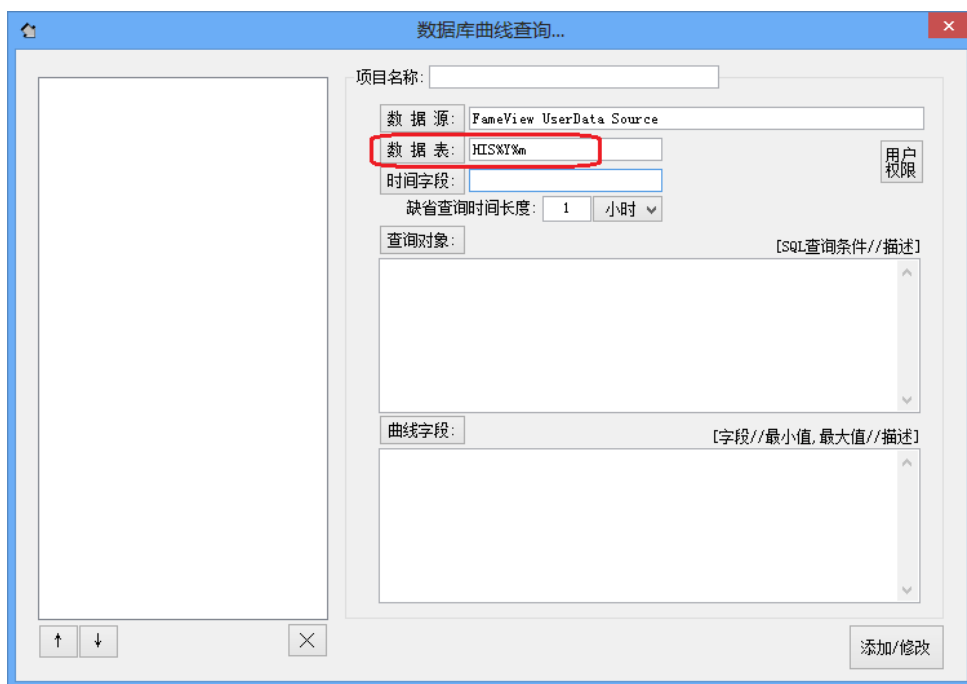
查询内容: [SQL查询条件//描述]

显示字段:

序号	字段名称	描述	字段类型	字符数
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

↑ ↓ ×

添加/修改

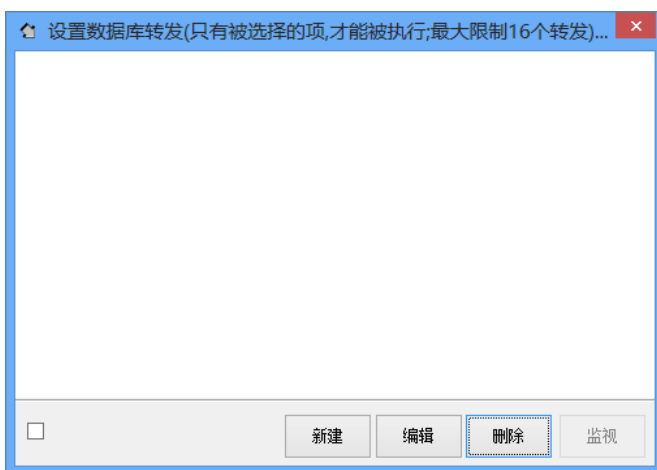


25.29 数据库转发

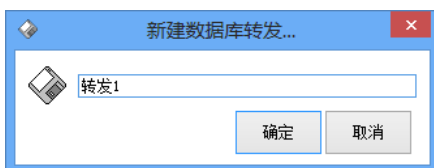
□ 定制数据库转发:



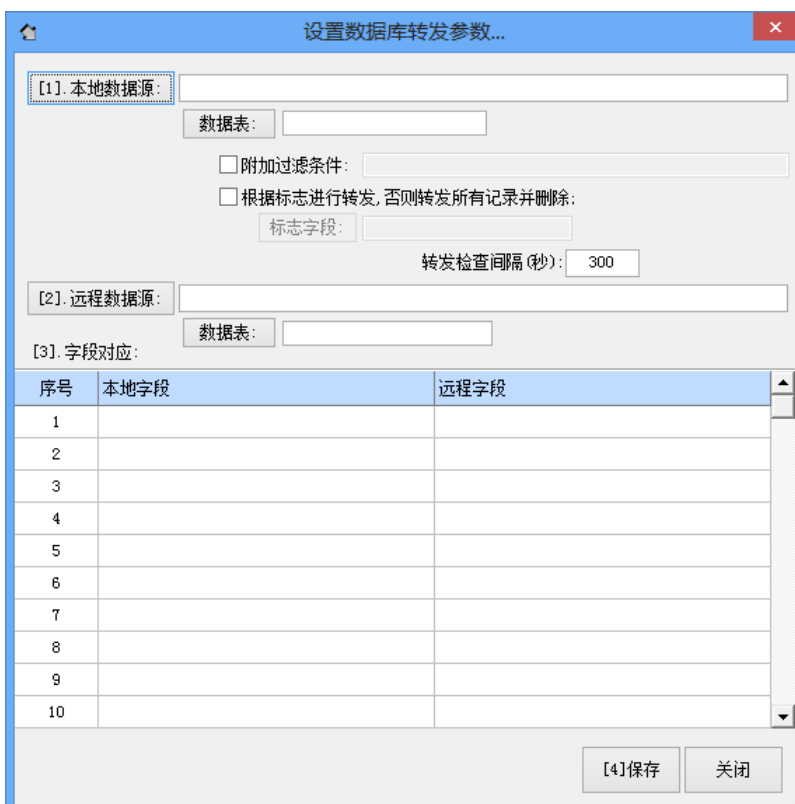
- 选择<数据库转发>功能, 执行<设置数据库转发>:



- 执行<新建>:



- 输入转发名称, 并进行编辑:



[1]. 选择本地数据源:

The dialog box '选择数据源...' (Select Data Source) has the following fields:

- 数据库类型 (Database Type): [1]. SQL Server
- 服务器 (Server): (local)\SQLEXPRESS
- 登录ID (Login ID):
- 口令 (Password):
- 数据库名称 (Database Name): UserDatabase

Buttons: 确定 (OK), 取消 (Cancel)

[2]. 选择远程数据源:

The dialog box '选择数据源...' (Select Data Source) has the following fields:

- 数据库类型 (Database Type): [1]. SQL Server
- 服务器 (Server): 192.168.1.100
- 登录ID (Login ID): sa
- 口令 (Password): *****
- 数据库名称 (Database Name): UserDatabase

Buttons: 确定 (OK), 取消 (Cancel)

[3]. 先后选择远程数据表、本地数据表:

The dialog box '设置数据库转发参数...' (Set Database Transfer Parameters) contains the following sections:

- [1]. 本地数据源 (Local Data Source):**
 - Driver={SQL Server};Server=(local)\SQLEXPRESS;Database=UserDatabase;Uid=
 - 数据表 (Data Table): localTable
 - ☐ 附加过滤条件 (Additional filter conditions):
 - ☐ 根据标志进行转发, 否则转发所有记录并删除 (Transfer according to the flag, otherwise transfer all records and delete):
 - 标志字段 (Flag field):
 - 转发检查间隔 (秒) (Transfer check interval (seconds)): 300
- [2]. 远程数据源 (Remote Data Source):**
 - Driver={SQL Server};Server=(local)\SQLEXPRESS;Database=UserDatabase;Uid=
 - 数据表 (Data Table): remoteTable
- [3]. 字段对应 (Field Correspondence):**

序号	本地字段	远程字段
1	dt	dt
2	f1	f1
3	f2	f2
4	f3	f3
5	flag	flag
6		
7		
8		
9		
10		

Buttons: [4] 保存 (Save), 关闭 (Close)

自动对应远程字段与本地字段相同, 如果远程字段与本地不同, 须相应修改;

[4]. 附加过滤条件, 可选项, 过滤转发本地数据表的部分记录内容;

☒ 附加过滤条件:

[5]. 转发标志字段

[a]. 没有使用转发标志字段, 转发本地数据表全部内容; 转发成功的本地记录, 被自动删除;

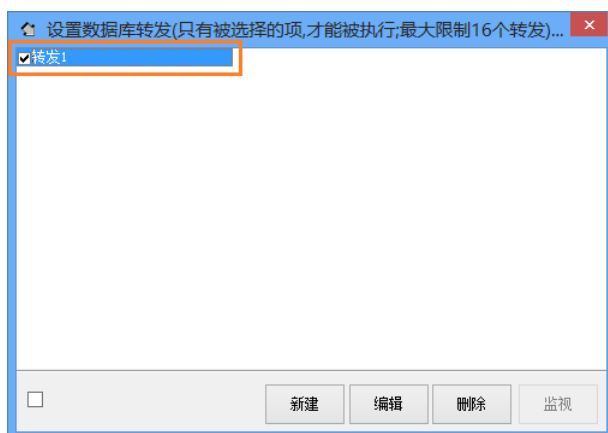
☐ 根据标志进行转发, 否则转发所有记录并删除:
标志字段:

[b]. 使用转发标志字段, 转发本地数据表中标志字段不为零的记录内容;
转发成功的本地记录, 其中标志字段值被修改为零;

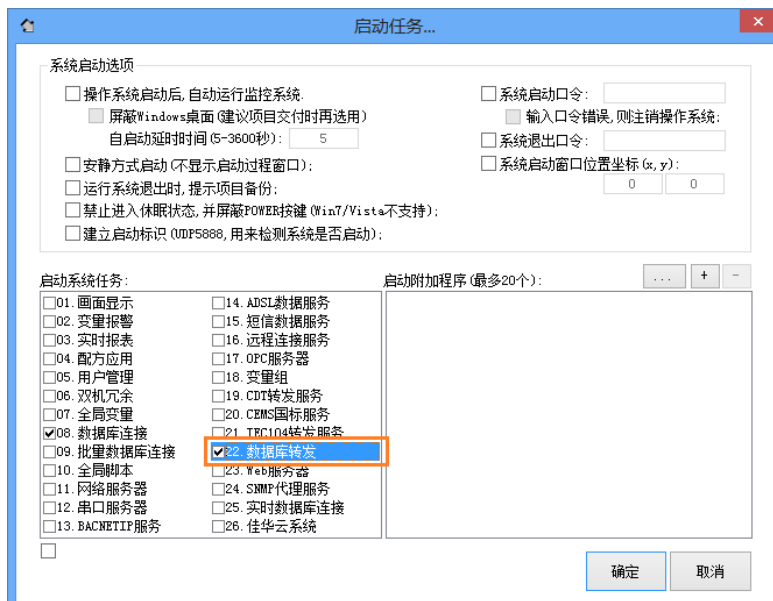
☒ 根据标志进行转发, 否则转发所有记录并删除:
标志字段:

[6]. 转发检查间隔 (N), 某次转发结束, 等待 N 秒后, 启动下次转发任务;

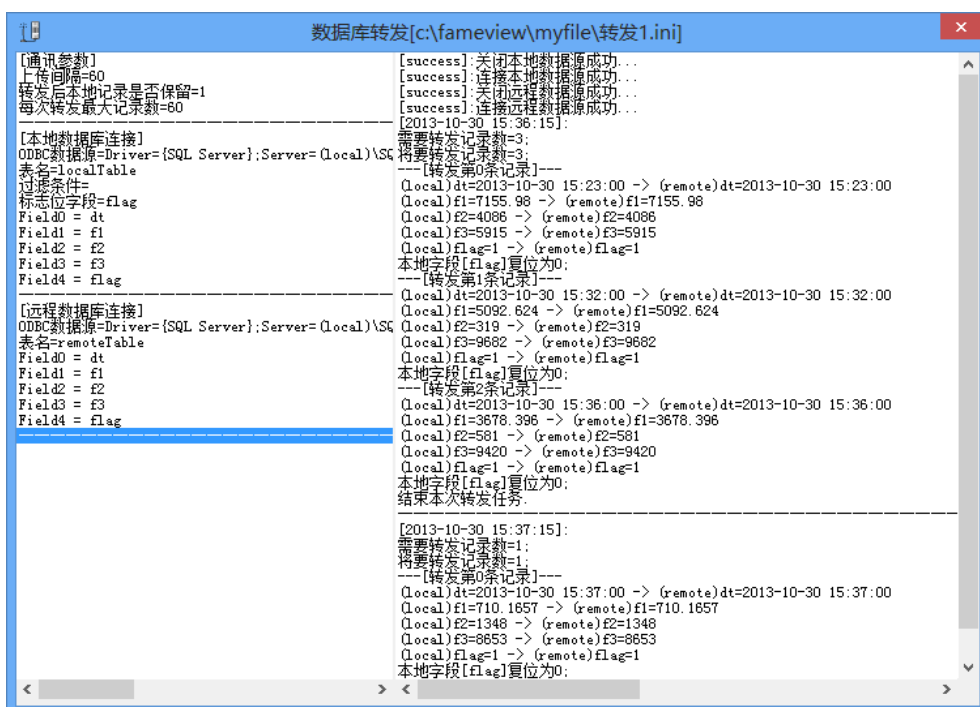
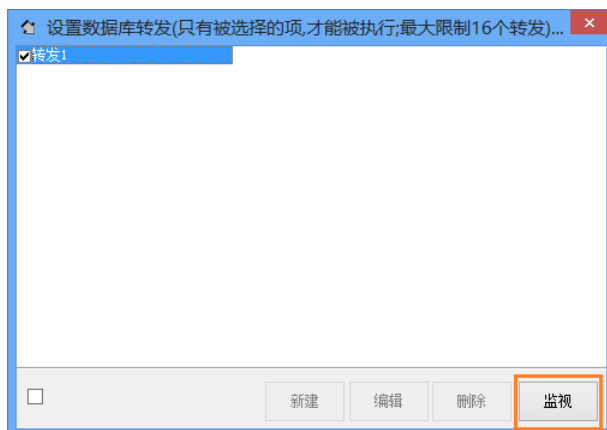
☐ 应用选择数据库转发, 最多支持 16 个数据库转发文件:



☐ 启动数据库转发:



□ 监视数据库转发：



25.30 使用脚本管理数据库

□ 新建数据表

```
' 打开数据库连接
Set conn = CreateObject("ADODB.Connection")
strConn="Driver={SQL
Server};Server=(local)\SQLEXPRESS;Database=UserDatabase;Uid=;Pwd=;"
conn.Open strConn
' 执行SQL命令
s = "CREATE TABLE t1" _
    &"(" _
    &"id bigint IDENTITY (1,1), " _
    &"dt datetime, " _
    &"f1 real, " _
    &"f2 float, " _
    &"f3 nVarChar(50) " _
    &")"
conn.Execute s
' 关闭数据表及连接
conn.Close
Set conn=Nothing
```

□ 删除数据表

```
' 打开数据库连接
Set conn = CreateObject("ADODB.Connection")
strConn="Driver={SQL
Server};Server=(local)\SQLEXPRESS;Database=UserDatabase;Uid=;Pwd=;"
conn.Open strConn
' 执行SQL命令
s = "DROP TABLE t1"
conn.Execute s
' 关闭数据表及连接
conn.Close
Set conn=Nothing
```

□ 增加字段

```
' 打开数据库连接
Set conn = CreateObject("ADODB.Connection")
strConn="Driver={SQL
Server};Server=(local)\SQLEXPRESS;Database=UserDatabase;Uid=;Pwd=;"
conn.Open strConn
```

' 执行SQL命令

```
s = "ALTER TABLE T1 ADD f4 int"
```

```
conn.Execute s
```

' 关闭数据表及连接

```
conn.Close
```

```
Set conn=Nothing
```

□ 修改字段

' 打开数据库连接

```
Set conn = CreateObject("ADODB.Connection")
```

```
strConn="Driver={SQL
```

```
Server};Server=(local)\SQLEXPRESS;Database=UserDatabase;Uid=;Pwd=;"
```

```
conn.Open strConn
```

' 执行SQL命令

```
s = "ALTER TABLE T1 ALTER COLUMN f4 real"
```

```
conn.Execute s
```

' 关闭数据表及连接

```
conn.Close
```

```
Set conn=Nothing
```

□ 删除字段

' 打开数据库连接

```
Set conn = CreateObject("ADODB.Connection")
```

```
strConn="Driver={SQL
```

```
Server};Server=(local)\SQLEXPRESS;Database=UserDatabase;Uid=;Pwd=;"
```

```
conn.Open strConn
```

' 执行SQL命令

```
s = "ALTER TABLE T1 DROP COLUMN f4"
```

```
conn.Execute s
```

' 关闭数据表及连接

```
conn.Close
```

```
Set conn=Nothing
```

□ 增加聚集索引

' 打开数据库连接

```
Set conn = CreateObject("ADODB.Connection")
```

```
strConn="Driver={SQL
```

```
Server};Server=(local)\SQLEXPRESS;Database=UserDatabase;Uid=;Pwd=;"
```

```
conn.Open strConn
```

' 执行SQL命令

```
s = "CREATE CLUSTERED INDEX Index1 ON T1(dt)"
```

```
conn.Execute s
```

' 关闭数据表及连接

conn.Close

Set conn=Nothing

□ 增加普通索引

' 打开数据库连接

Set conn = CreateObject("ADODB.Connection")

strConn="Driver={SQL

Server};Server=(local)\SQLEXPRESS;Database=UserDatabase;Uid=;Pwd=;"

conn.Open strConn

' 执行SQL命令

s = "CREATE INDEX Index2 ON T1(f1)"

conn.Execute s

' 关闭数据表及连接

conn.Close

Set conn=Nothing

□ 增加唯一索引

' 打开数据库连接

Set conn = CreateObject("ADODB.Connection")

strConn="Driver={SQL

Server};Server=(local)\SQLEXPRESS;Database=UserDatabase;Uid=;Pwd=;"

conn.Open strConn

' 执行SQL命令

s = "CREATE UNIQUE INDEX Index3 ON T1(id)"

conn.Execute s

' 关闭数据表及连接

conn.Close

Set conn=Nothing

□ 删除索引

' 打开数据库连接

Set conn = CreateObject("ADODB.Connection")

strConn="Driver={SQL

Server};Server=(local)\SQLEXPRESS;Database=UserDatabase;Uid=;Pwd=;"

conn.Open strConn

' 执行SQL命令

s = "DROP INDEX T1.Index1"

conn.Execute s

' 关闭数据表及连接

conn.Close

Set conn=Nothing