

学校的理想装备

电子图书·学校专集

校园网上的最佳资源

数据库开发

SQL Server7.0 和 Power Bulider7.0

(四)





[返回总目录](#)

第四章

SQL Server 数据操作

本章导读

数据库的任务是管理和存储数据，同时要支持用户对数据的操作。包括查询、添加、删除、更新数据等操作。

SQL（结构化查询语言）一直是国际公认的数据库操作语言。Microsoft SQL Server 在标准 SQL 语言的基础上进行了扩充，建立了 SQL Server 使用的 T-SQL 语言。事实上，SQL Server 在处理任何用户对数据库的操作都是通过执行 T-SQL 语句实现的。

本章将要介绍的是四个最基本的数据操作语句：使用 Insert 语句添加数据；使用 UPDATE 语句更新数据；使用 DELETE 语句删除数据；使用 SELECT 语句查询数据。

添加数据 Insert 语句添加数据

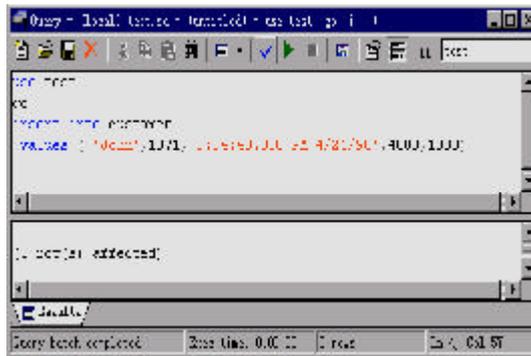
1

```
INSERT INTO table_or_view (column_list)
VALUES (value_list)
```

注：
table_or_view 为添加记录的表格或者视图的名称
column_list 为添加记录的各个字段的名称
value_list 为添加记录的各个字段的值

2

下图所示为在 Query Analyzer 中往表格中添加数据记录的实例，其中执行语句：`INSERT INTO customer VALUES ('John',1071,'2:24:48.000 PM 4/21/98',4000,1000)`。



3

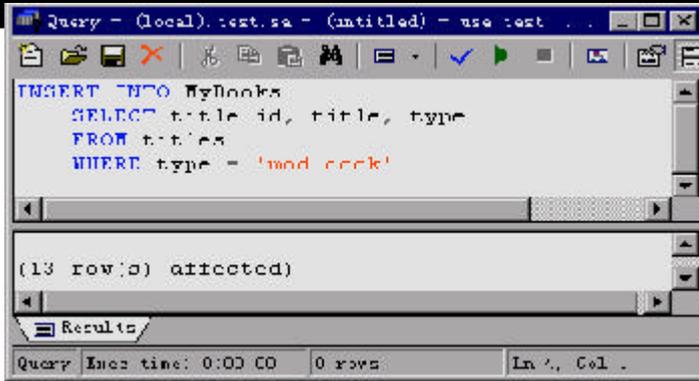
使用 `INSERT..SELECT` 语句可以将 `SELECT` 查询的结果添加到表格中去。其最简单的句法如下图所示：

```
INSERT INTO table_or_view1
SELECT column_list
FROM table_or_view2
```

注：
table_or_view1 为添加记录的目标表格或者视图的名称
table_or_view2 为添加记录的源表格或者视图的名称
column_list 为源表格中字段名称的列表

另外，在 `SELECT` 子句中还可以加上 `WHERE`，`ORDER BY` 等限制，用于控制添加的记录范围或者顺序等

4



5

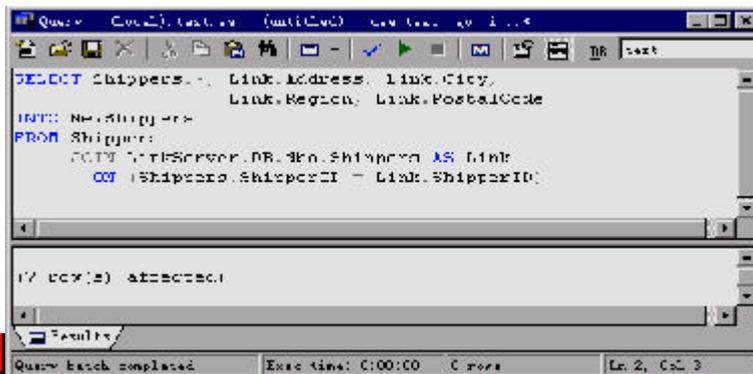
使用 SELECT INTO 语句可以将查询结果添加到新创建的表格中，SELECT INTO 的句法如下图所示：

```
SELECT select_list
[INTO new_table_]
FROM table_source
[WHERE search_condition]
[GROUP BY group_by_expression]
[HAVING search_condition]
[ORDER BY order_expression [ASC | DESC] ]
```

注： select_list 为查询的字段名称列表
 new_table_ 为新建表格的名称
 table_source 为查询表格的名称
 WHERE、GROUP BY、HAVING等限制与SELECT语句中的限制意义相同

6

在 Query Analyzer 中执行 SELECT..INTO 语句往新建表格中添加记录数据的结果如下图所示，执行的实例为：



7

例

更新数据 使用 UPDATE 语句更新数据

解

1

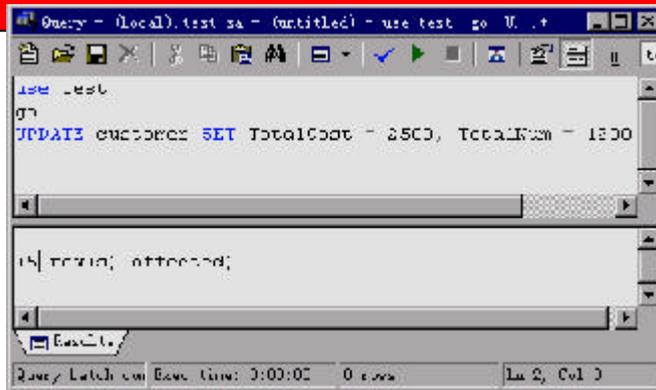
```
UPDATE table_or_view SET column_1 = exp1[,...]
    [where condition]
```

注: table_or_view 带更新的表格或者视图的名称
column_1 为更新的字段名称
exp1 为赋值给字段的表达式

另外, 经常将UPDATE语句与where条件子句连用, 可以限制更新数据的范围

2

在 Query Analyzer 中执行带有 SET 子句的 UPDATE 语句结果如下图所示
执行的实例为: UPDATE customer SET TotalCost = 2500, TotalNum = 1200



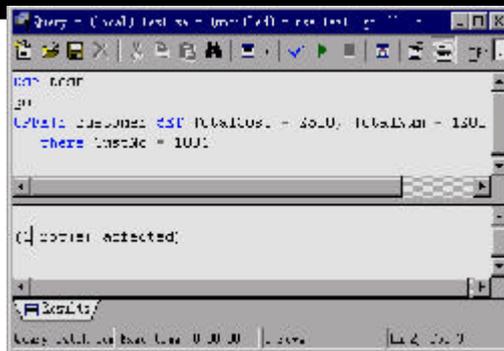
3

带有 Where 子句 UPDATE 的更新语句, 将满足 Where 条件限制的记录修改。其 T-SQL 句法如下图所示:

```
UPDATE table_or_view
SET column = exp [,...]
WHERE conditions
```

注：
table_or_view 要更新数据的表格或者视图的名称
column 为更新的字段名称
exp 为赋值的表达式
conditions 为限制更新范围的条件

4



5

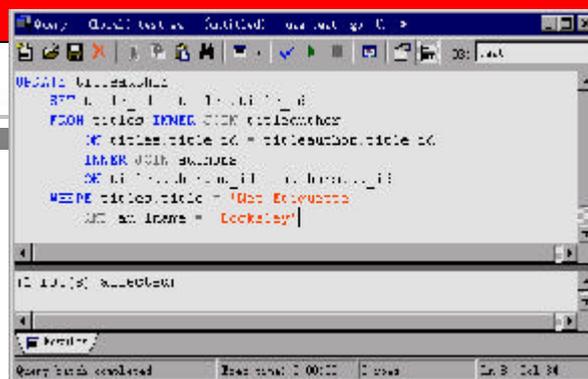
带有 FROM 子句的 UPDATE 语句，可以用源表格中的数据更新目标表格的数据，其句法如下图所示：

```
UPDATE table_or_view
SET column = exp [,...]
FROM source_table
WHERE conditions
```

注：
table_or_view 要更新数据的表格或者视图的名称
column 为更新的字段名称
exp 为赋值的表达式
source table 为源表格名称
conditions 为限制更新范围的条件

6

在 Query Analyzer 中执行带有 FROM 子句的 UPDATE 语句结果如下图所示。

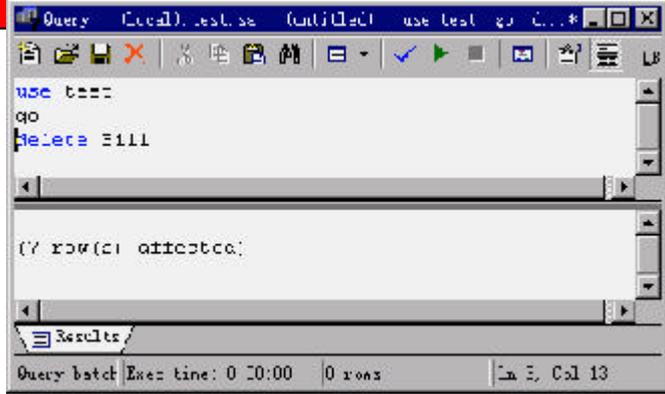


例 删除数据 使用 DELETE、TRUNCATE 解

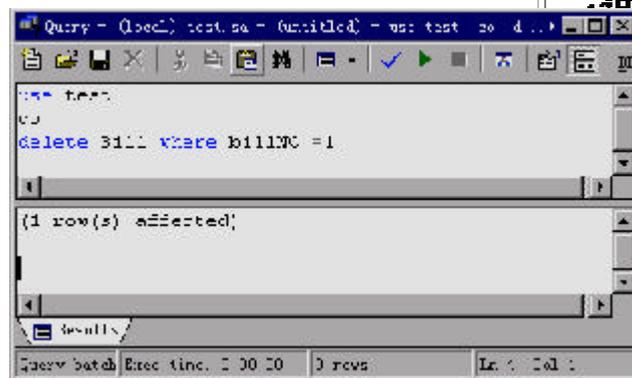
1

```
DELETE
[FROM ]
<
  table_name WITH { <table_hint_limited> [...n]}
  | view name
  | roudset_function_limited
}
[ FROM {<table_source> } [...n] ]
[WHERE
  { <search_condition>
  | < { CURRENT OF
      {
        { [ GLOBAL ] cursor name }
        | cursor_variable_name
      }
    }
  }
]
[OPTION {<query hint> } [...n]]
```

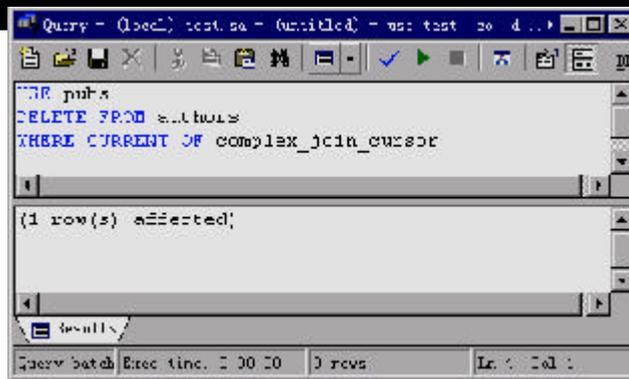
2 在 Query Analyzer 中执行以下实例。例一：“DELETE authors”用于删除表格中的所有数据，其结果如下图所示：



3 实例二：执行 DELETE Bill WHERE BillNo =1 ，用于删除表格中编号为 1 的记录，结果如下图所示：



4



5

使用 TRUNCATE TABLE 语句可以删除数据库表格中的所有记录 其 T-SQL 语句的句法如下图所示：

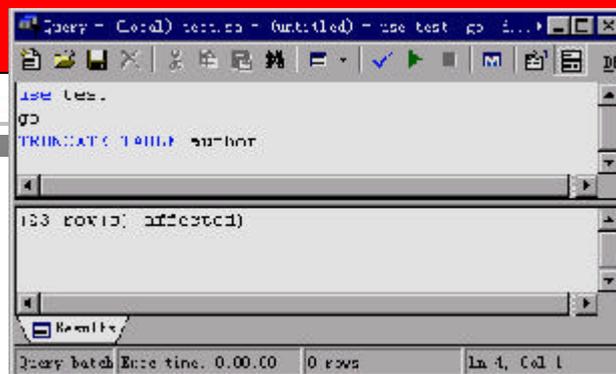
TRUNCATE TABLE name

注：name 为要删除所有数据的表格名称

如要在数据库中删除表格的定义，使用 DROP TABLE 语句。

6

在 Query Analyzer 中执行 TRUNCATE TABLE 的结果如下图所示：

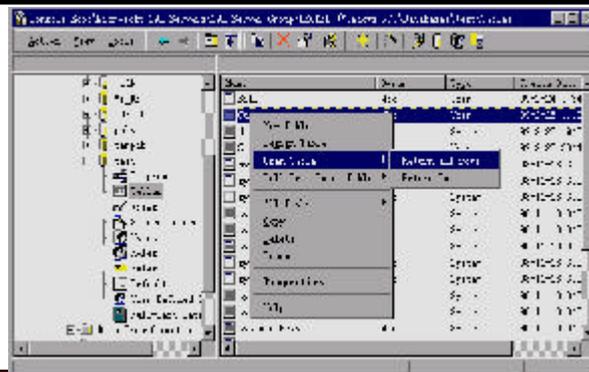


例

查询操作 执行 T-SQL 的环境

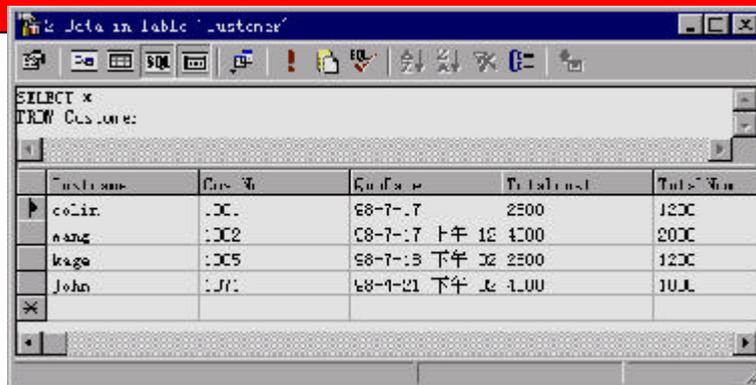
解

1



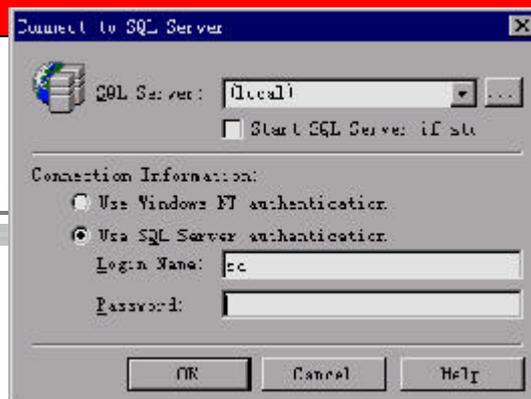
2

在弹出的 Query Designer 中单击 **SQL** 按钮，就可以看见 SQL 输入编辑窗口，输入完 SQL 命令后，单击 **!** 按钮可以执行该 SQL 语句。



3

也可以直接启动 Query Analyzer，这时您需要在 Connect 对话框中，登录用户帐号和口令，并且享有相应的权限。



查询操作 执行 SELECT 语句

1

```
SELECT select_list
[INTO new_table_]
FROM table_source
[WHERE search_condition]
[GROUP BY group_by_expression]
[HAVING search_condition]
[ORDER BY order_expression [ASC | DESC] ]
```

2

FROM 子句用于指明查询表格的名称。例一：使用 FROM 子句指明表格名称。执行语句为 SELECT * FROM Customer,Bill。在 Query Analyzer 中执行的结果如下图所示：

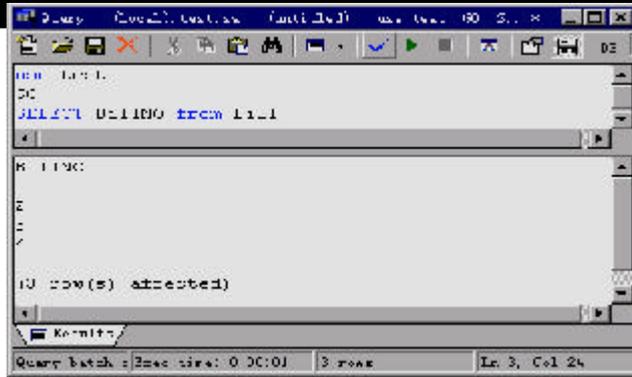
CustomerID	Company	Date Recd	Total Cost	Profit
1001E	1001	1998-07-17 00:00:00.000	2500	1200
1001E	1001	1998-07-17 00:00:00.000	1000	200
1001E	1001	1998-07-17 14:00:00.000	2500	1200
1001E	1001	1998-04-31 14:00:00.000	6000	1000

3

例一在 osql 中执行的结果如下图所示：

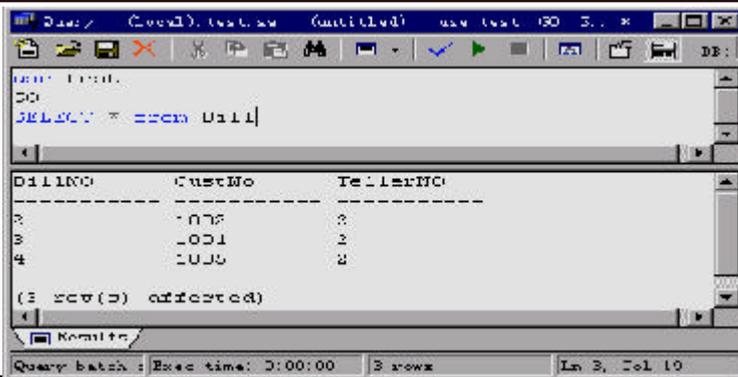
```
> SELECT * FROM CUSTOMER, BILL
CustomerID      Company      Date Recd      Total Cost      Profit
-----
1001E           1001         1998-07-17 00:00:00.000  2500            1200
1001E           1001         1998-07-17 00:00:00.000  1000             200
1001E           1001         1998-07-17 14:00:00.000  2500            1200
1001E           1001         1998-04-31 14:00:00.000  6000            1000
(12 rows affected)
>
```

4



5

SELECT 后的 select_list 用于指明查询返回的字段名称。例二：指明查询字段。执行语句为 SELECT BillNO from Bill。在 osql 中执行的结果如下图所示：



6

在 select_list 中使用 “*” 符号，可以表示返回表格中的所有字段，在 Query Analyzer 中执行 SELECT * FROM Bill 语句结果如下图所示：



7

通过本例，了解了 SELECT 语句的句法，以及如何指定查询的表格名称和字段名称。

查询操作 执行 SELECT 语句 (续)

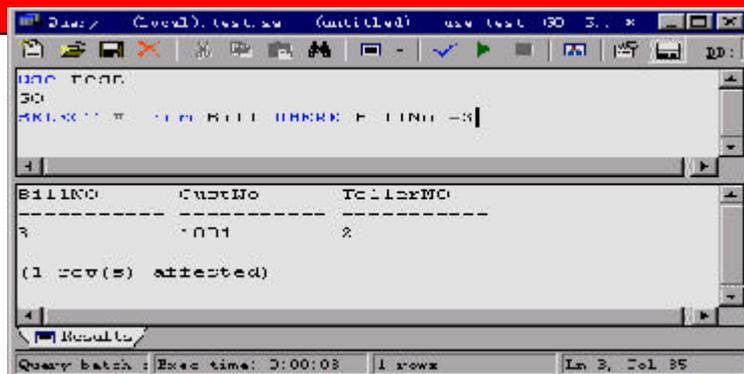
1

```
SELECT column_name_1,..column_name_n
FROM table_name_1,..table_name_n
WHERE column_name comparison_operator value
```

注: column_name_n 为查询的字段名称
table_name_n 为查询的表格或者视图名称
comparison_operator 为比较运算符
WHERE子句的比较运算返回一个布尔型的值

2

例 1 执行带有比较运算的 WHERE 子句的查询结果如下图所示,执行的语句为 SELECT * FROM Bill WHERE BillNo = 2



3

例二: 在 osql 中执行带有比较运算的 WHERE 子句的查询结果如下图所示,执行的语句为 SELECT * FROM bill WHERE BillNo >1



4

Customer	CustNo	GoodName	TotalCost	TotalPay
cake	1001	18-6-27 下午 12	1000	8000
Cary	1002	98-7-23 下午 12	2400	1200
cava	1003	90 4 10 上午 0:	2000	3000
calin	1004	18-1-11	1500	1500

5

例四：在 Query Analyzer 中执行带有 LIKE “_” 比较运算的 WHERE 子句的查询语句结果如下图所示，执行语句的实例为：SELECT * from Customer WHERE CustName LIKE 'c_’。

Customer	CustNo	GoodName	TotalCost	TotalPay
cake	1001	18-6-27 下午 12:00:00	2000	3000
cava	1003	1998-04-10 上午 0:00:00	1500	1500

6

例五：在 osql 命令中执行带有 LIKE “[]” 比较运算的 WHERE 子句的查询语句结果如下图所示，执行语句的实例为：SELECT * FROM Customer WHERE Custname like ‘[a-cA-C]’。

```

(1) you entered:
1) select * from bill where BillNo < 1000
2) go

Customer      CustNo      GoodDate      TotalCost      TotalPay
-----
1001          1001        18-6-27 下午 12:00:00      2000          3000
1002          1002        98-7-23 下午 12:00:00      2400          1200
1003          1003        90 4 10 上午 0:00:00      2000          3000
  
```

7

通过本例，学会了如何在 SELECT 语句的 WHERE 子句中使用比较运算限制查询范围。



查询操作 执行 SELECT 语句 (续)

1

```
SELECT column_name_1,..column_name_n
FROM table_name_1,..table_name_n
WHERE column_name comparison_operator value
Boolean_operator column_name comparison operator
```

注: column_name_n 为查询的字段名称
table_name_n 为查询的表格或者视图名称
comparison_operator 为比较运算符
Boolean_operator 为布尔运算符

2

例一，在 Query Analyzer 中执行带有由 OR 运算符连接的 WHERE 子句的 SELECT 语句，结果如下图。执行的实例为：SELECT * FROM Customer WHERE Totalcost >3000 OR TotalNum >2500。

Custname	CustNo	CustDate	TotalCost
wang	1002	1999-07-17 00:22:00,DCD	3000
zhuo	1004	1998-01-24 12:24:00,DCD	5000
wang	1006	1998-01-17 11:27:00,DCD	5000
wang	1007	1998-01-24 14:04:48,DCD	4000

3

例二，在 Enterprise Manager 中执行带有 AND 运算符连接的 WHERE 子句的 SELECT 语句，结果如下图。执行的实例为：SELECT *FROM Customer WHERE Custname LIKE 'c%' AND TotalNum > 2500。

Custname	CustNo	CustDate	TotalCost	TotalNum
zhuo	1004	98-4-21 下午 > 3000	5000	5000
wang	1006	98-4-17 上午 > 3000	5000	5000

查询操作 执行 SELECT 语句 (续)

7

```
SELECT column_name_1,..column_name_n
FROM table_name_1,...table_name_n
ORDER BY col_1,...col_n [DESC|ASC]
```

注: column_name_n 为查询的字段名称
 table_name_n 为查询的表格或者视图名称
 col_n 为排序的字段名称
 DESC 查询结果按降序排列
 ASC 查询结果按升序排列

2

例一：在 Enterprise Manager 中执行带有 ORDER BY 子句的 SELECT 语句结果如下图所示。执行实例为 SELECT * FROM Customer ORDER BY CustNO。

CustName	CustNo	OrderDate	TotalCost	TotalNum
zhang	1001	98-7-17	2000	1000
wang	1002	98-7-17 上午 10	1000	2000
zhang	1003	98-7-25 上午 10	2400	1200
wang	1004	98-7-21 下午 10	1000	2000
zhang	1005	98-7-18 下午 02	2500	1200
wang	1006	98-7-17 上午 01	2000	2000
zhang	1007	98-7-21 下午 02	5000	1000

3

例二：在 Query Analyzer 中执行带有多个字段排序的 ORDER BY 子句的 SELECT 语句。执行实例为：SELECT * FROM Customer ORDER BY TotalCost, TotalNum。先对 TotalCost 排序，如值相同，再 TotalNum 排序。

CustName	CustNo	OrderDate	TotalCost	TotalNum
zhang	1005	98-7-18 下午 02:00:00.C00	2500	1200
zhang	1007	98-7-21 下午 02:00:00.C00	5000	1000
wang	1004	98-7-21 下午 10:00:00.C00	1000	2000
zhang	1003	98-7-25 上午 10:00:00.C00	2400	1200
wang	1002	98-7-17 上午 10:00:00.C00	1000	2000
zhang	1001	98-7-17	2000	1000
wang	1006	98-7-17 上午 01:00:00.C00	2000	2000

4

```

命令提示符 - sql - 81Legend - 8x --
Z> go
CustName      CustNo      OrderDate      TotalCost      TotalNum
wang          1002 1998 七月 17 12:21AM      4000
kong         1005 1998 七月 18 2:22PM      2500
John         1071 1998 七月 21 2:24PM      1000
edlin        1004 1998 七月 17 12:00AM      2500
cous         1006 1998 七月 13 1:21PM      3000
Casey        1003 1998 七月 23 12:21PM      2400
vake         1004 1998 七月 21 12:24PM      3000

(CY rows affected)
1)
    
```

5

例四：在 Enterprise Manager 中执行带有 ORDER BY 子句的 SELECT 语句，其中排序字段用整数表示。执行实例为：SELECT Custname, CustNO, TotalNum FROM Customer ORDER BY 3, 2。

Custname	CustNO	TotalNum
John	1071	1000
edlin	1004	2500
Casey	1003	2400
wang	1002	4000
vake	1004	3000
cous	1006	3000
kong	1005	2500

6

在 Query Analyzer 中执行带有 DISTINCT 关键字的 SELECT 语句，结果如下图。执行的实例为：

```

Query - (local) - sql - 81Legend - SELECT DISTINCT
SELECT DISTINCT TotalNum FROM Customer
-----
1000
2500
2400
3000

(4 row(s) affected)
    
```

7

通过本例，学会了如何将 SELECT 语句查询的结果排序显示出来，以及如何通过关键字 DISTINCT 设置值相同的多个纪录唯一返回。

查询操作 执行 SELECT 语句 (续)

7

```
SELECT select_list
FROM table_source
GROUP BY [ALL] group_by_expression [...n]
[ WITH { CUBE | ROLLUP } ]
```

注：
 select_list 为字段列表
 table_source 为表格或者视图名称
 ALL 指明显示所有的分组纪录
 group_by_expression 为分组表达式
 CUBE|ROLLUP 显示概括信息

2

例一，在 Enterprise Manager 中执行带有 GROUP BY 子句的 SELECT 语句结果如下图所示，执行的实例为：
 SELECT Custname, [Average cost] = AVG(Totalcost) FROM Customer GROUP BY CustName.

The screenshot shows a window titled "Date on Table 'Customer'". The SQL query is:


```
SELECT Custname, [Average cost] = AVG(Totalcost)
FROM Customer
GROUP BY CustName
```

 The results are displayed in a table with two columns: "Custname" and "Average cost".

Custname	Average cost
lake	3000
larry	2400
rene	3000
solon	1800
julia	3000
rene	2400

3

例二，在 Query Analyzer 中执行带有 GROUP BY 子句的 SELECT 语句结果如下图所示。执行的实例为：
 SELECT Custname, [Average cost] = AVG(Totalcost) FROM Customer GROUP BY CustName HAVING count(*)=1.

The screenshot shows a window titled "Query - (local) AdventureWorks - (untitled) - SELECT Custname...". The SQL query is:


```
SELECT Custname, [Average cost] = AVG(Totalcost)
FROM Customer
GROUP BY CustName
HAVING count(*)=1
```

 The results are displayed in a table with two columns: "Custname" and "Average cost".

Custname	Average cost
lake	3000
larry	3000
rene	3000
solon	1800
julia	3000
rene	2400

4

```

C:\Program Files\Microsoft SQL Server\90\Tools\Binn>sqlcmd -i script.sql -u sa -P 123456
Microsoft SQL Server 9.0.4035.1 Copyright (c) Microsoft Corp.
>
> SELECT Custname, AVG(TotalCost)
> FROM customer
> GROUP BY Custname, TotalCost
>
Custname      Average Cost
-----
colin          17944
john          2118
john          2314
john          2498
john          2488
john          2613
john          2888
john          4698
john          4888
john          4998
john          5000

```

5 例四，执行有关关键字 ALL 的 GROUP BY 子句的 SELECT 语句结果如下图执行的实例为：SELECT Custname, [Average Cost] AVG(TotalCost) FROM customer WHERE CUSTNAME <> 'colin' GROUP BY ALL Custname。

```

SELECT Custname, [Average Cost] = AVG(TotalCost)
FROM customer
WHERE CUSTNAME <> 'colin'
GROUP BY ALL Custname

```

Custname	Average Cost
john	3770
john	3470
john	3770
colin	38000
john	3770
john	3470
john	3770

6 执行有关关键字 ROLLUP 的 GROUP BY 子句的 SELECT 语句，结果如下图所示。执行的实例为：select Custname, "Average cost" = AVG(Totalcost) from customer group by Custname , Totalcost WITH ROLLUP;

```

SELECT Custname, "Average cost" = AVG(Totalcost)
FROM customer
GROUP BY Custname , Totalcost WITH ROLLUP

```

Custname	Average cost
john	3470
john	3000
john	3000
john	3470

7 通过本例，学会了如何使用 GROUP BY 子句来对查询结果进行分组显示和处理。



查询操作 执行 SELECT 语句 (续)

1

Query window showing the following SQL statement:

```
SELECT Custname, Totalcost
FROM Customer
WHERE Totalcost <
  (SELECT AVG(Cost*cost)
   FROM Customer
   WHERE Custname = 'kage')
```

Results table:

Custname	Totalcost
Carry	2000
culin	1000
kage	2000
john	2000

2

例二, 在 Query Analyzer 中执行比较运算 SELECT 子查询语句结果如下图。
语句为: `SELECT Custname, Totalcost FROM Customer WHERE Totalcost < (SELECT AVG(Totalcost) FROM Customer WHERE Custname = 'kage')。`

Query window showing the following SQL statement:

```
SELECT Custname, Totalcost, TotalNum
FROM Customer
WHERE Totalcost <
  (SELECT AVG(Totalcost)
   FROM Customer
   WHERE Custname = 'kage')
```

Results table:

Custname	Custno	TotalCost	TotalNum
culin	00	8000	1000
kage	1000	8000	1000
kage	2000	2000	2000

3

例三, 在 osql 中执行带 "=" 的 SELECT 子查询语句结果如下图。语句为
`SELECT Custname,Custno,totalcost,totalNum FROM Customer WHERE Custno = (SELECT Custno FROM BILL WHERE billno = 3)。`

Command prompt window showing the following SQL statement:

```
2) FROM Customer
3) where custno =
4) (select Custno from bill
5) where billno =3)
6) go
```

Results table:

Custname	CustNo	TotalCost	TotalNum
culin	1000	2500	1200

Output: (1 row affected)

4

```
SELECT Custname, Totalcost
FROM Customer
WHERE EXISTS
  (SELECT *
   FROM bill
   WHERE Bill.Custno = Customer.Custno)
```

Custname	Totalcost
wang	3000
wang	4000
kege	2500

5 例五，执行用子查询替代表达式。语句为 `SELECT Custname, [Average Cost] =(SELECT AVG(Totalcost) FROM Customer), Totalcost - (SELECT AVG(totalcost) FROM customer) AS Difference FROM Customer.`

```
SELECT Custname, [Average Cost] =
  (SELECT AVG(Totalcost)
   FROM Customer), Totalcost -
  (SELECT AVG(totalcost)
   FROM customer) AS Difference
FROM Customer
```

Custname	Average Cost	Difference
wang	2667	333
wang	2667	1333
kege	2667	-967
wang	2667	193

6 在 osql 中执行带有 NOT EXISTS 的子查询，语句为：`Select Custname, Custno, Totalcost, Totalnum from Customer where not exists (Select * from bill where bill.Custno = Customer.Custno)`

```
C1 rows affected
1> select Custname, Custno, Totalcost, Totalnum
2> from Customer
3> where not exists
4> (select * from bill
5> where bill.Custno = Customer.Custno)
6> go
```

Custname	Custno	Totalcost	Totalnum
kege	1001	2400	1200
wang	1004	3000	3000
wang	1006	3000	3000
wang	1007	1200	2400
kege	1008	2310	3400
john	1009	2130	3000
john	1071	1000	1000

7 通过本例，学会了如何在查询语句中使用子查询语句完成更为复杂的操作。





[返回总目录](#)

第五章

数据复制

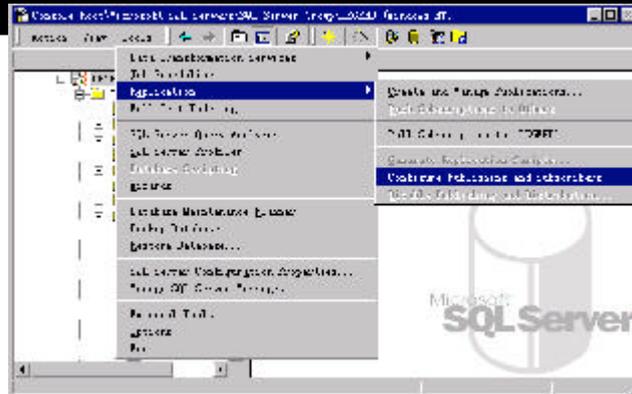
本章导读

数据库复制（分布存储）是企业内分布式数据库用到的重要的和强大的技术。通过它可以在企业内多台服务器上存储数据、执行存储过程。

SQL Server 中的复制(Replication)技术使你能够将数据复制存储在多台服务器上，控制异地数据的同步以保证分布数据的一致性。复制也能实现同一台服务器上多个数据库的数据同步，或者实现在局域网、广域网甚至因特网上多台服务器的数据一致。

本章我们将要介绍复制技术的一些基本概念和使用方法。本章包括的内容有：设置数据复制；修改数据复制设置；删除发布者；出版数据和存储过程；出版物的文章操作；订阅出版物；监视复制；复制代理；监视复制。

1



2

弹出对话框显示设置出版者和订阅者属性的向导程序可以帮助我们完成哪些功能，阅读后，单击下一步继续。

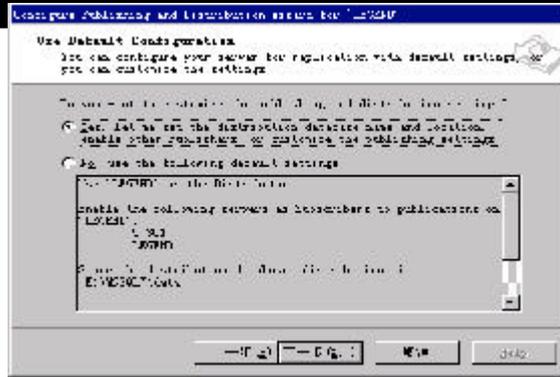


3

在接下来的对话框中，可以设定选择哪个 SQL Server 服务器作为分布者 (Distributor)。可以选择本地机，或者选择在 Enterprise Manager 中注册了其他服务器。在这里，我们选择使用本地机作为分布者。

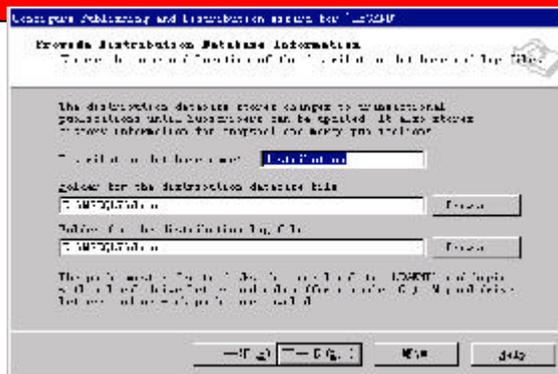


4



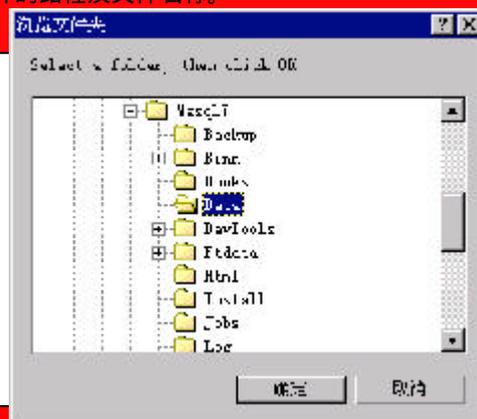
5

下面的对话框，可以选定在服务器中存放分布信息的数据库名称，以及该数据库的数据文件和日志文件的存放路径和名称。



6

在上图中单击 **Browse** 按钮，可以弹出下图所示的对话框，用以设定存放的数据库文件的路径及文件名称。

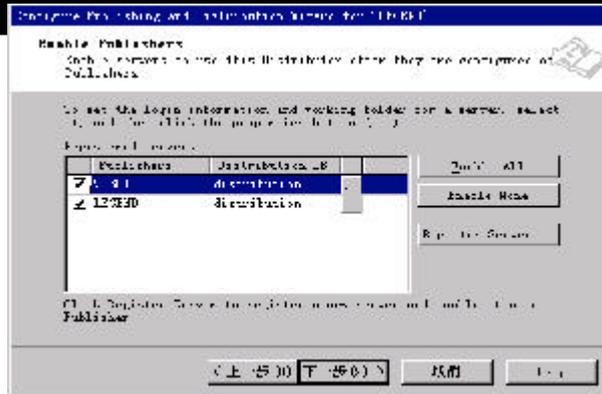


7

本节未完，下节续。

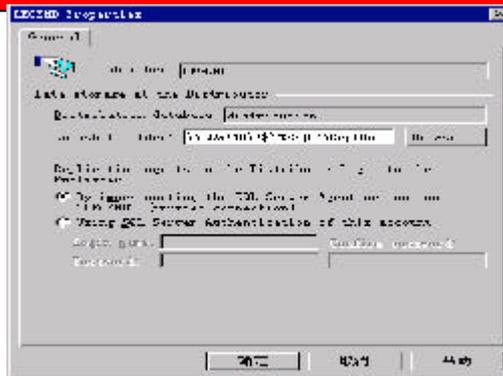
设置数据复制 使用设置复制向导 (续)

7



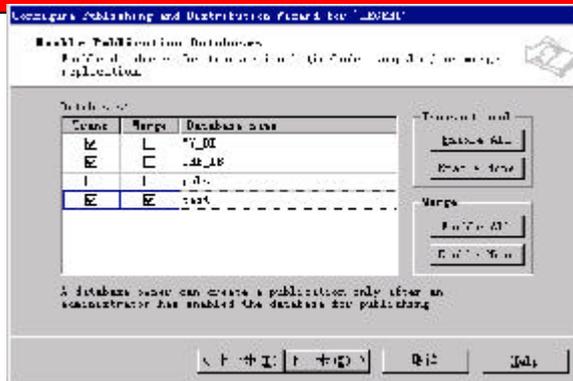
2

单击上图中 Distribution DB 栏右边的  可以设定该出版者的属性，对话框如下图所示。在其中可以设定 Snapshot 的目录，以及分布者登录该服务器的帐号和口令设置。

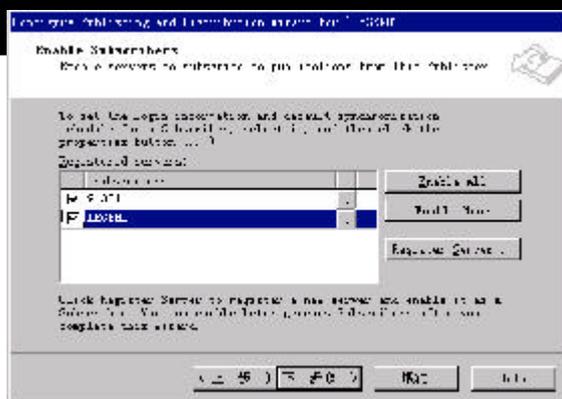


3

在下图对话框中可以选定有哪些用户数据库可以使用数据复制。单击 Trans. Merge 栏下方的复选框分别可以选中是否允许相应的事务型复制或数据库合并操作。



4



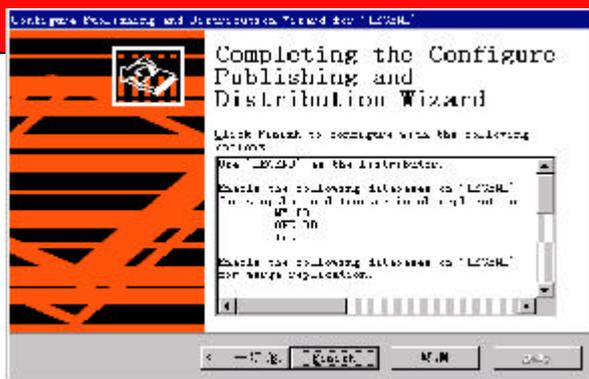
5

在上图 Subscribers 栏右方单击  按钮，可以设置订阅者的属性，弹出对话框如下图所示。在 General 和 Scheduling 页面中分别可以设定一般信息和调度信息。



6

最后，在向导结束时，会弹出如下对话框，显示在向导中设置的概述。检查该信息，确定无误后单击 Finish 结束。如果需要修改可以单击上一步修改。



7

通过本例，学会了如何使用 Configure Publishing and Distribution Wizard 设置数据复制中的出版者和订阅者属性。

4



5

下面的对话框用于设定是否允许立即更新选项。即是否允许订阅者对复制数据进行修改。选定后单击下一步继续。



6

下面的对话框，需要设定订阅者服务器是否都运行在 SQL Server 上。如果有某些订阅者运行其他的数据库引擎，如 ORACLE，SQL Server 会在发布之前，将数据转换成文本格式再发布。



7

本节未完，下节续。



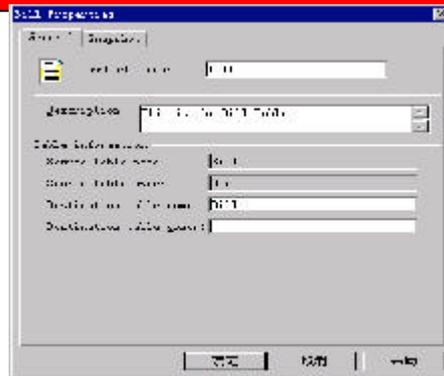
设置数据复制 使用发布向导 (续)

7



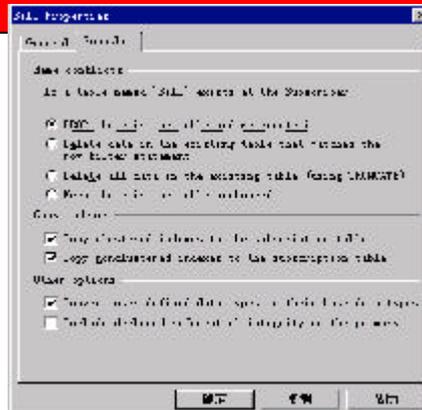
2

在上图中单击 **...** 按钮，会弹出下图对话框，在 General 页面中，输入诸如注释、描述等信息。

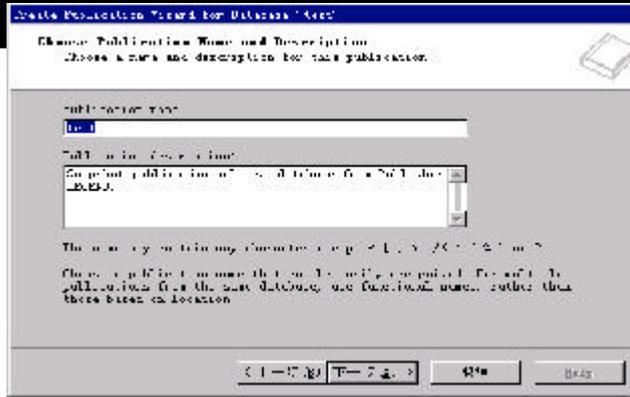


3

在 Snapshot 中，设置对选定的表格复制是采取的操作。包括如果目标服务器上已存在同名表格采取何种处理，以及对表格内的索引如何操作等。

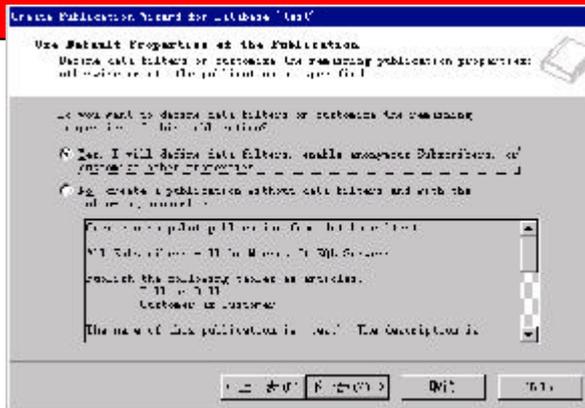


4



5

接下来，该对话框询问是否对过滤数据等使用缺省属性。通常可以使用缺省属性，在这里，为了介绍如何设置。我们选择自己设定。



6

下图对话框，询问是否需要使用数据过滤功能。该功能允许将表格中特定的行或者字段复制到目标服务器，而不需要全部复制。



7

本节未完，下节续。

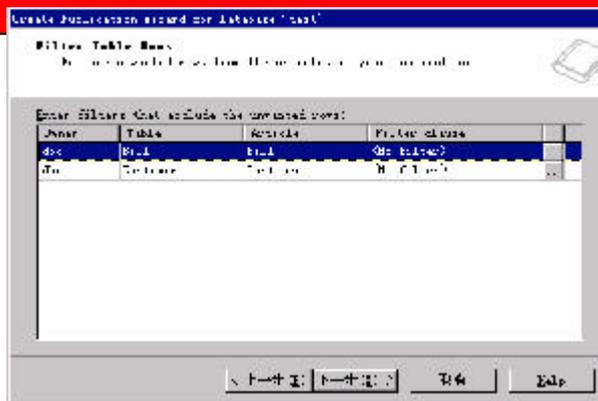
设置数据复制 使用发布向导 (续)

7



2

在下图中，如要对表格中某些特定的纪录进行复制，可以单击图中表格名称右边的按钮。

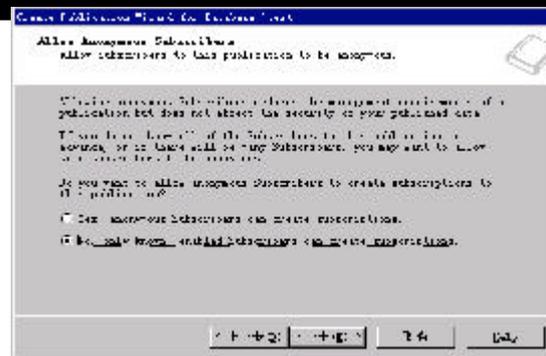


3

在上图中单击，在弹出的对话框中，输入限制特定纪录的 T-SQL 语句如下图所示。

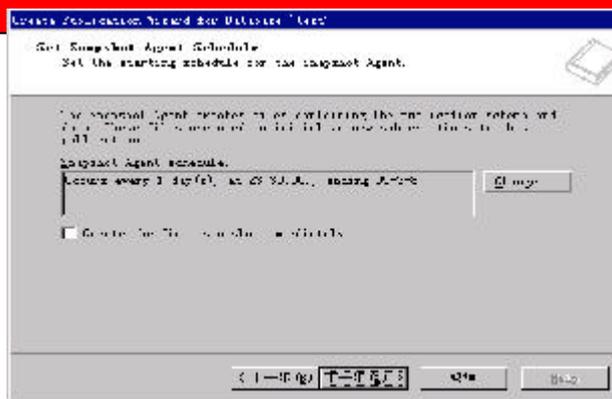


4



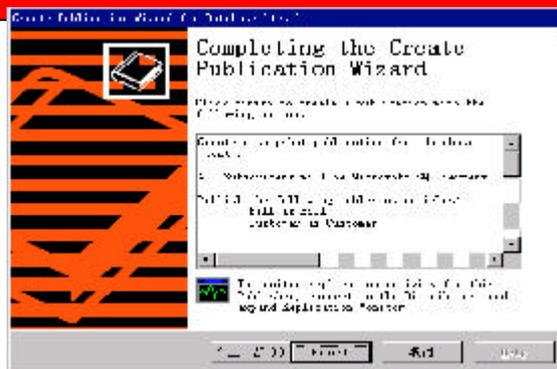
5

在下图中，可以设定 Snapshot Agent 在何时将出版物拷贝到发布者上的 Snapshot 文件夹中。单击 Change 按钮，可以修改调度时间。



6

最后，该向导会显示在向导过程中的所有出版物设定，确定无误后，单击 Finish 键即可。否则，单击上一步返回后修改设置。

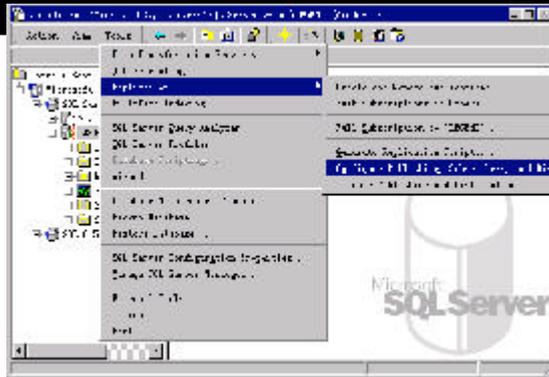


7

通过本例，学会了如何通过使用 Creating Publication Wizard 向导创建出版物，并且设置出版物的属性。

修改数据复制设置 使用向导修改复制设置

7



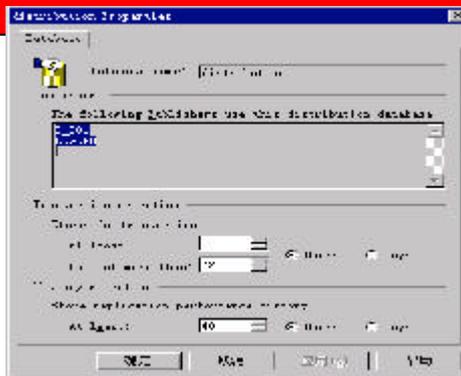
2

在弹出对话框的 Distributor 页面中，可以查看和设置分布者的属性。单击 New 可以创建新的分布数据库，单击 Agent Profiles 可以查看日志纪录，在最下方可以设置连接出版者的口令和帐号设置。

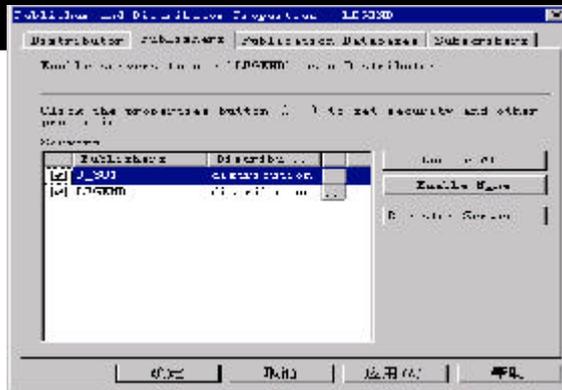


3

在上图中单击 Properties 按钮，可以查看和设定分布者的属性。如设置日志的纪录间隔和日志历史的保留期限。

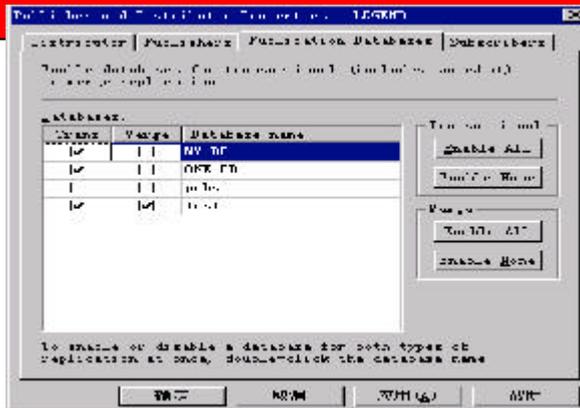


4



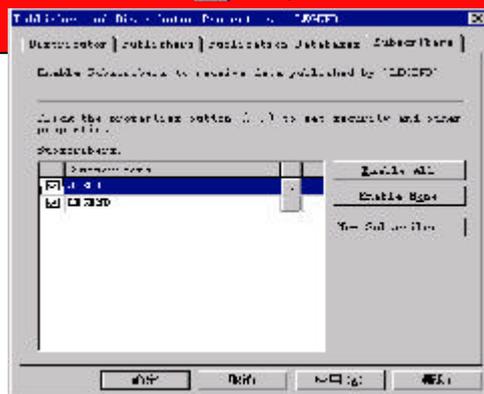
5

在 Publication Databases 页面中，可以查看和设定服务器上有哪些数据库可以被复制。



6

在 Subscriber 页面中，可以设定有哪些服务器可以作为订阅者连接该服务器。单击服务器名称右边的按钮，可以查看和设定该服务器的详细属性。

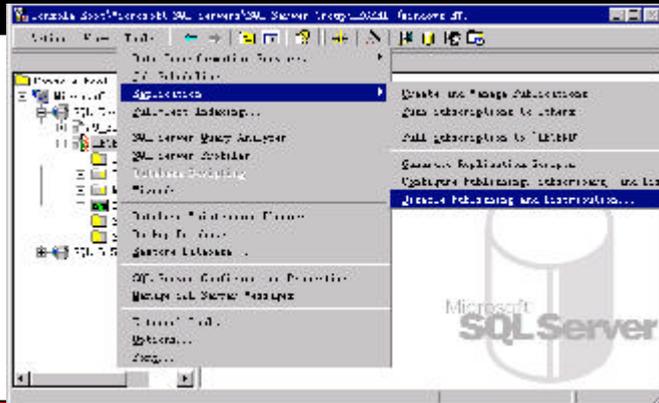


7

通过本例，学会了如何使用 Configure Publishing „Subscribers and Distributions.对话框来查看和设定复制系统中的各个服务器。

删除分布者 使用删除向导删除

1



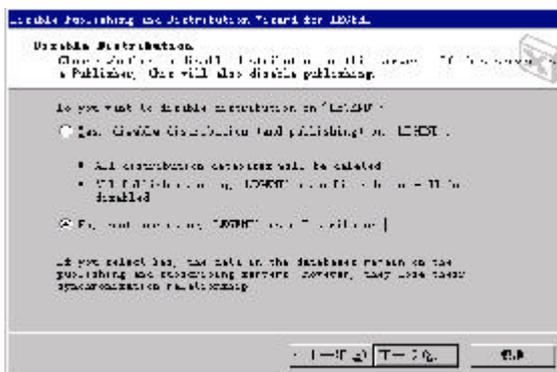
2

接着会弹出向导的起始对话框，该对话框会告诉您，通过该向导您可以完成哪些操作，该向导又会自动地完成哪些操作。按下一步继续。

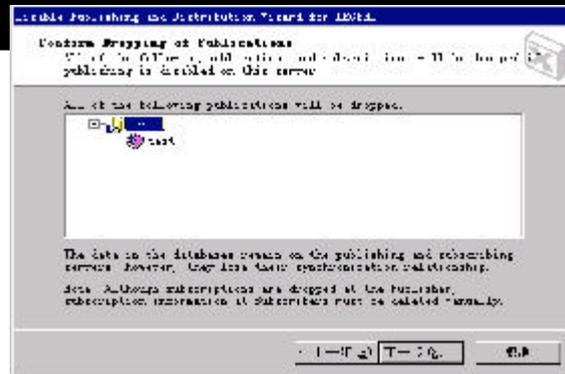


3

下面的对话框让你确认是否要禁止该服务器继续作为分布者，选择 No 后该向导将不会完成任何操作，选择 Yes 后，单击下一步。

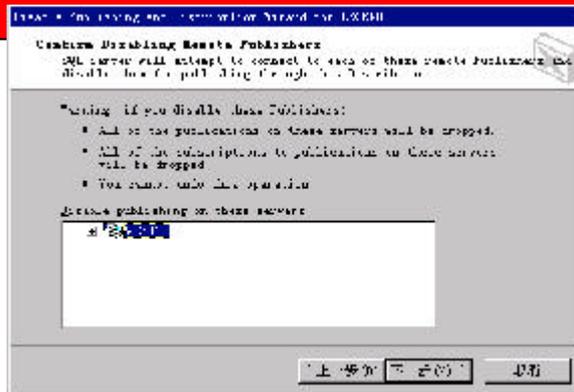


4



5

在下面的对话框中，向向导会显示该向导会继续禁止哪些出版者服务器，确认后单击下一步继续。



6

在下图所示的对话框中，会显示通过设置向导将会执行哪些操作，确认无误后，单击 Finish 完成即可。

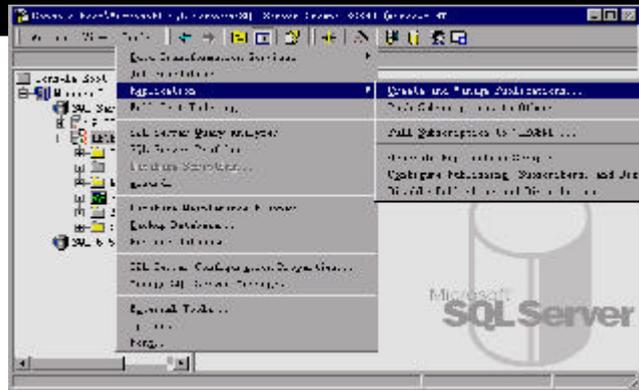


7

通过本例，学会了如何使用 Disable Publishing and Distribution 禁止发布者，删除出版物等。

出版数据和存储过程 使用出版向导

1



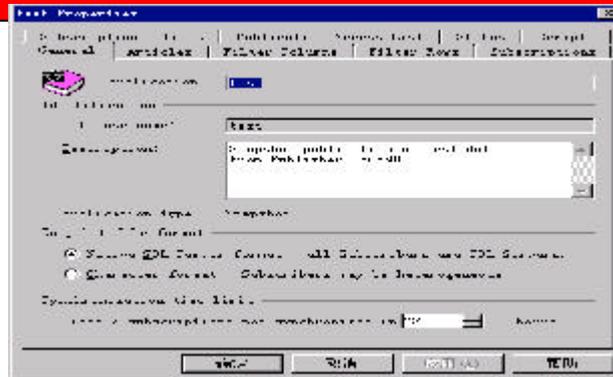
2

在弹出对话框中，展开选定的数据库，在该数据库中选择要查看或者设定的出版物名称，单击 Properties and Subscriptions 按钮。

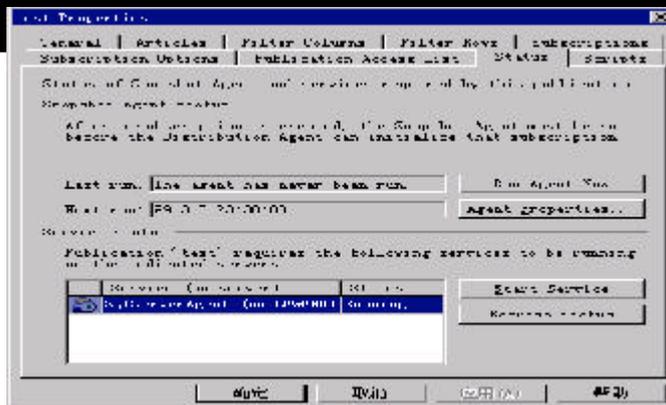


3

在 General 页面中，可以查看和设置出版物的名称、描述和类型，以及 Snapshot 文件格式和同步的时间限制。

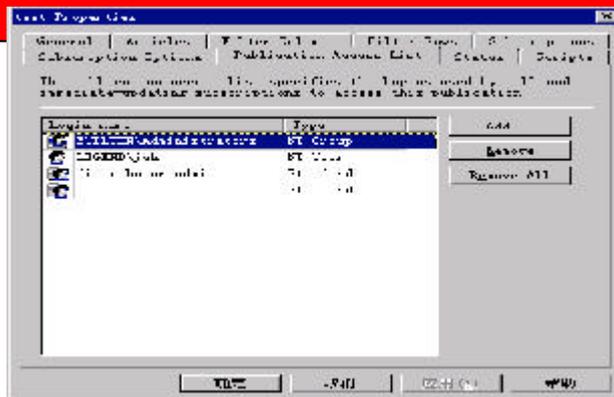


4



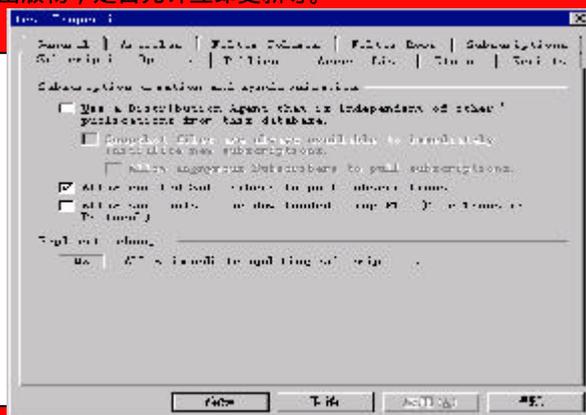
5

在 Publication Access List 中可以查看、添加、删除有访问该出版权限的用户帐号，如下图所示：



6

在 Subscription Option 页面中可以设定是否允许订阅者使用 FTP 协议匿名订阅该出版物，是否允许立即更新等。

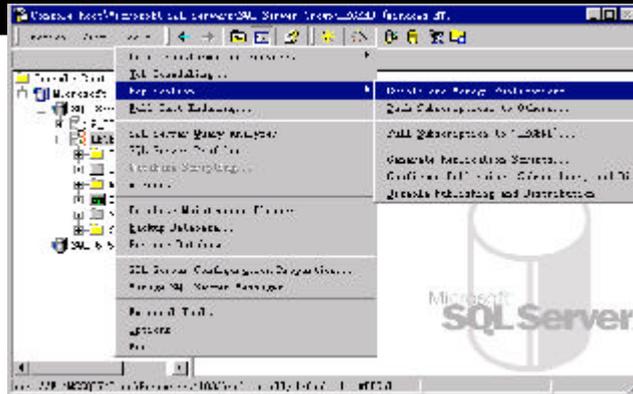


7

通过本例，学会了如何查看和设置出版物的属性。

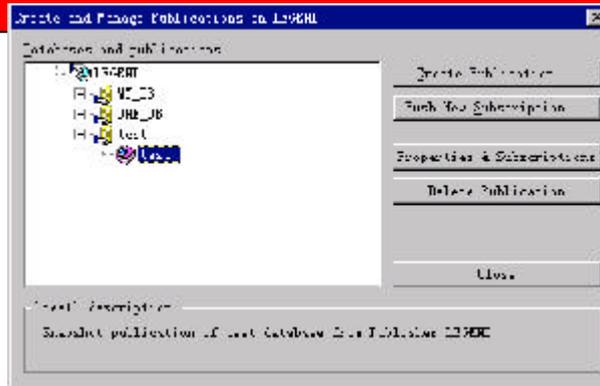
出版物的文章操作 定义、修改和删除文章

1



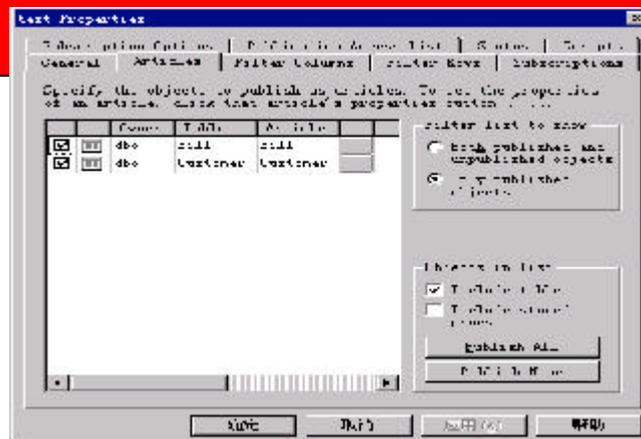
2

展开要操作的数据库, 选定其中的出版物后, 单击 Properties & Subscriptions 按钮继续。如下图:



3

在弹出的对话框中选择 Articles 页面, 如下图所示:



4



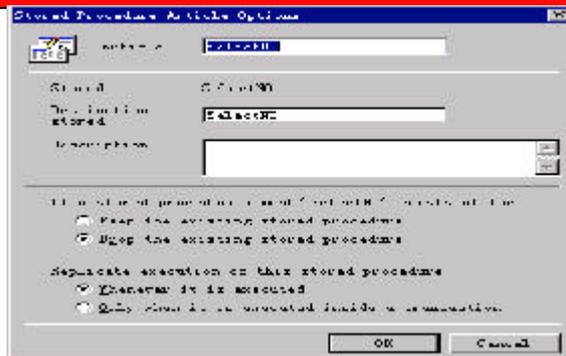
5

选定该对话框中要新建的某个表格或者存储过程名称左边的复选框。也可以单击 Publish All 按钮将列表中所有的数据库对象都添加到出版物中。清除已选中的复选框或者单击 Publish None 可以删除出版物文章。



6

在上图中单击某个被包含的表格或者存储过程名称右边的... 按钮, 可以弹出文章的属性对话框, 如下图所示, 在其中可以设定该文章的名称、描述及目标名称等属性。

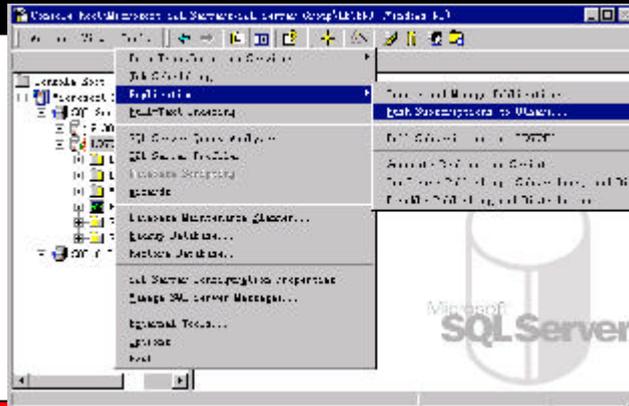


7

通过本例, 学会了如何定义、查看和删除出版物中的数据库表格文章或者存储过程文章。

订阅出版物 使用 Push 向导

1



2

弹出的对话框中将显示允许复制的所有数据库的名称，可以展开其中的某个数据库查看该数据库下是否有出版物。



3

在上图中，单击要展开的某个数据库名称左边的  处。可以展开该数据库选定要 Push 的出版物名称后，单击 Push New Subscription 按钮。如下图：

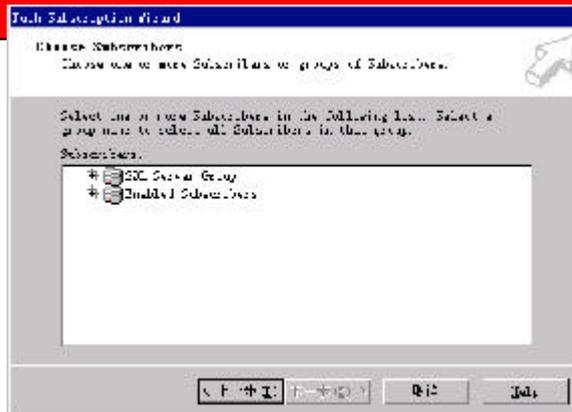


4



5

接着弹出的对话框中，显示了可供选择的服务器组，选择要接收出版物的订阅者所在的服务器组，单击左边的+可以展开该服务器组。



6

在上图中单击+后，展开服务器组，在其中选择要接受出版物的订阅者服务器或者选择某个服务器组（单击组名称即可）。



7

本节未完，下节续。

订阅出版物 使用 Push 向导(续)

7



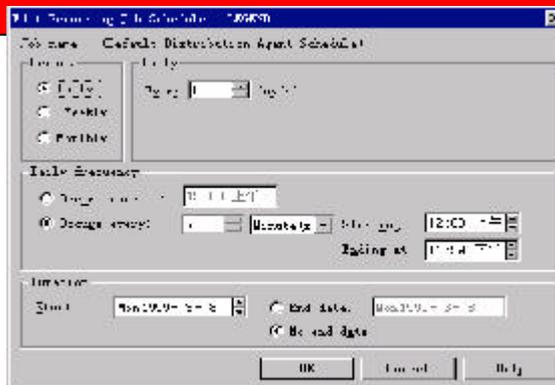
2

在下图的对话框中可以选择 Distribution Agent 的调度时刻，有三种选项：
连续地；可设定调度周期地；缺省地（在创建出版物时定义的）。



3

在上图中单击 Change 按钮，可以在设定调度周期对话框中自己定制 Distribution Agent 的执行时刻。

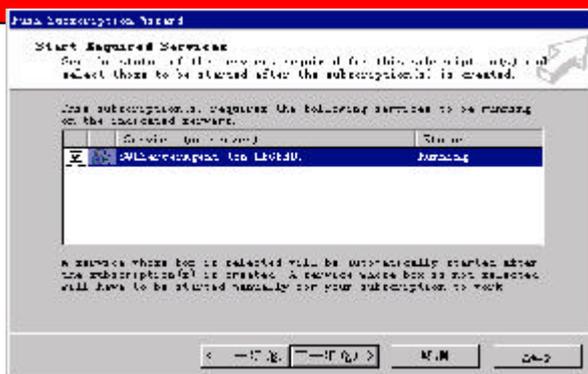


4



5

下图对话框中显示了，执行该 Push 订阅操作需要用到的服务及其当前状态，单击服务名称左边的复选框，可以让服务器在向导结束后启动该服务。



6

最后的对话框中会显示该向导已经完成的设置。确定无误后，单击 Finish 按钮结束。否则，单击上一步回去修改。

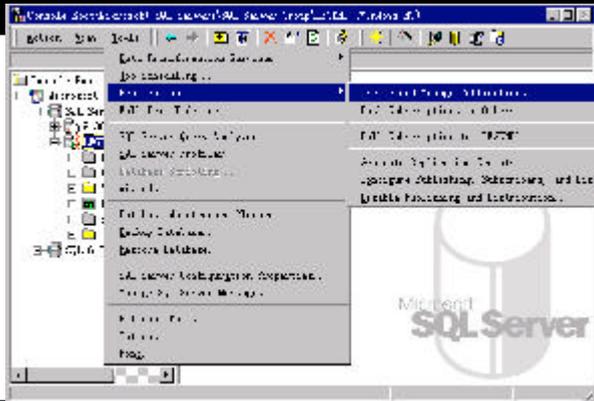


7

通过本例，学会了如何在 Enterprise Manager 中使用 Push Subscription Wizard 发出 Push 订阅。

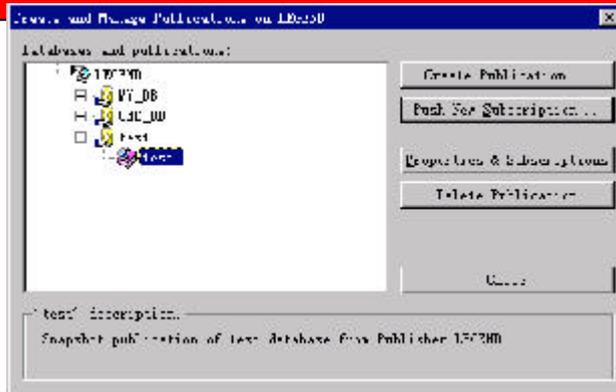
订阅出版物 查看和修改 Push 订阅属性

1



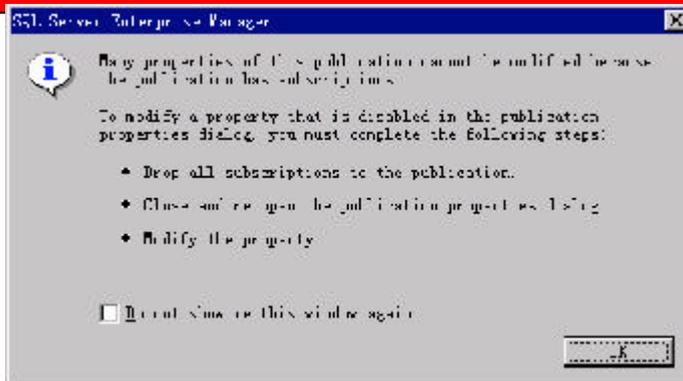
2

在下图对话框中，展开选定的服务器，鼠标单击要查看的出版物名称，再单击 Properties & Subscriptions 按钮继续。

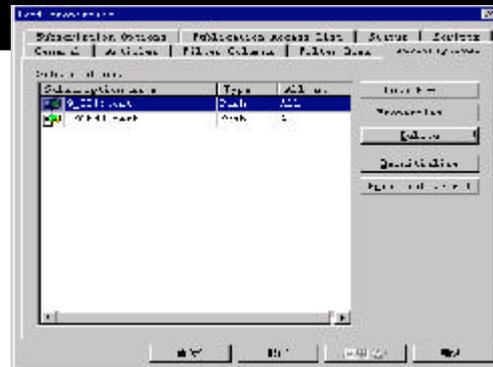


3

如果该出版物中已经有了订阅操作，将会弹出下图对话框，告诉你由于已经有了订阅，所有该出版物的许多属性不能修改。如要修改，可按照该对话框中的提示去做。单击 OK 按钮继续。



4



5

在上图中单击 Properties 按钮，会弹出下图所示的对话框，显示该订阅的属性。



6

在图 4 中单击右边的 Delete 按钮可以删除选定的订阅。单击后会弹出下图所示的对话框，确认是否要删除该订阅。

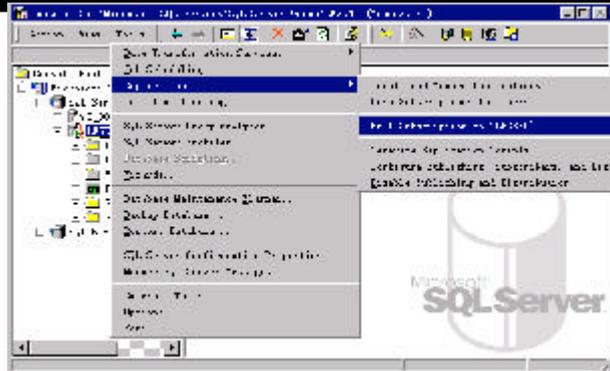


7

通过本例，学会了如何在 Enterprise Manager 中，查看、修改和删除已有的订阅的属性。

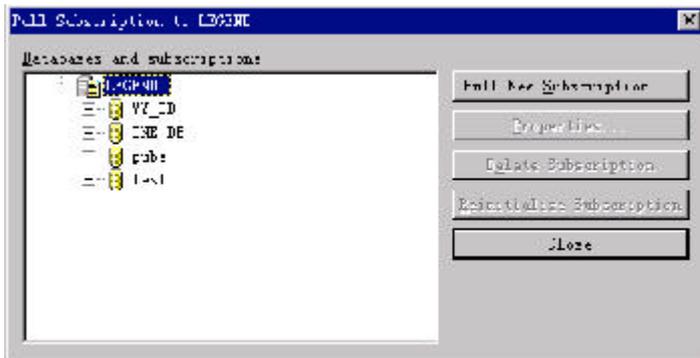
订阅出版物 使用 Pull 向导

1



2

弹出的对话框中，可以显示已有的 pull 订阅，及其所在的数据库。单击 Pull New Subscription 按钮继续。

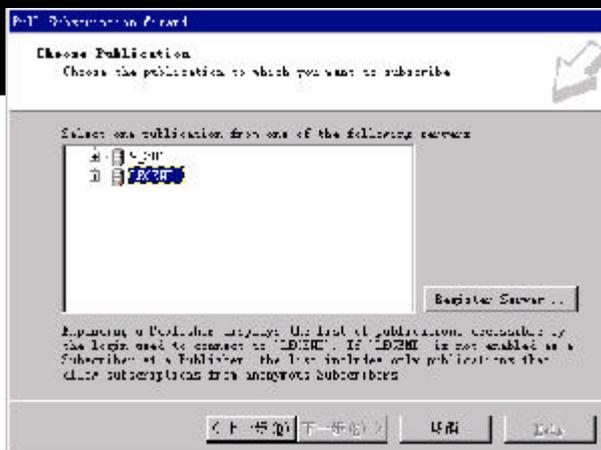


3

下图所示的是 Pull Subscription Wizard 向导的起始对话框，在其中，可以看到该向导所能完成的操作。单击下一步继续。

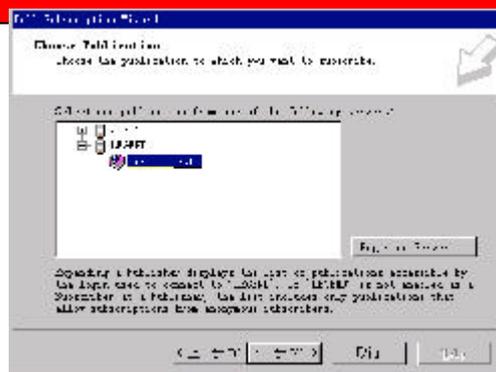


4



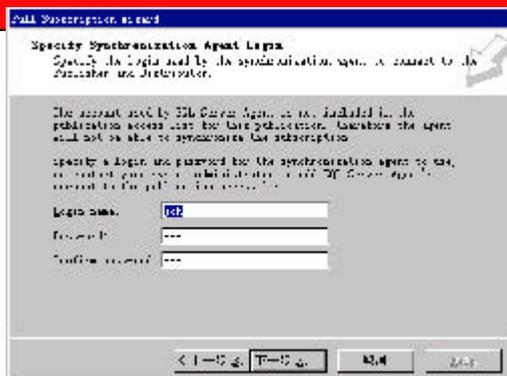
5

在上图中单击选定的出版者服务器名称左边的 **+** 框，展开该服务器，选定要订阅的出版物名称，再单击下一步按钮继续。



6

在下面的对话框中输入登录出版者服务器的用户帐号名称和口令。确定后单击下一步按钮继续。

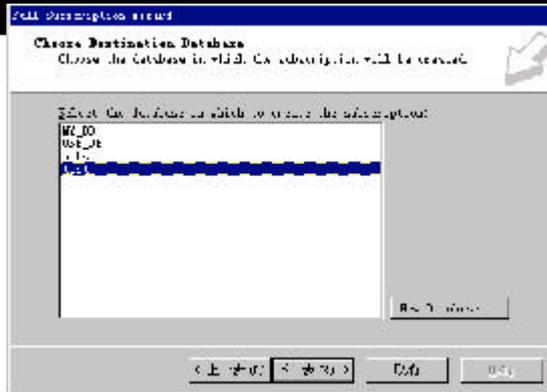


7

本节未完，下节续。

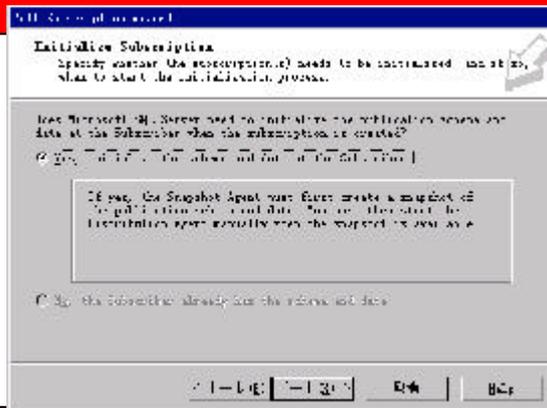
订阅出版物 使用 Pull 向导

7



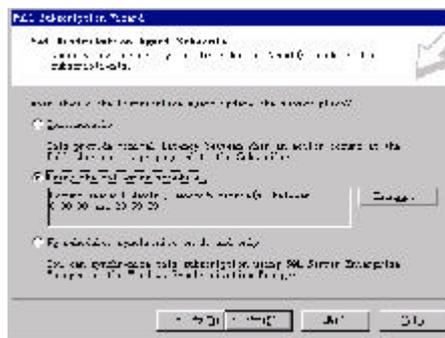
2

在下图对话框中，可以设置是否需要初始化目标数据库中的复制数据。如果是第一次接受复制，必须要进行初始化操作。单击下一步继续。

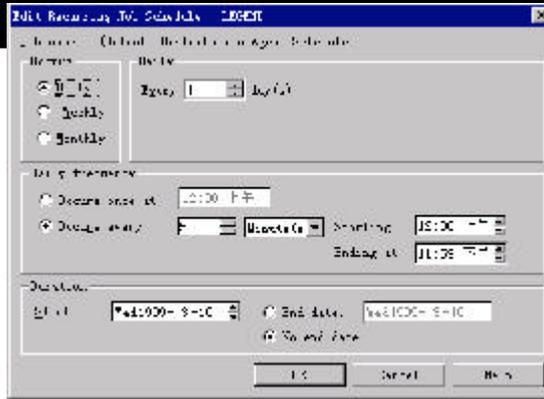


3

在下图中，可以设定执行复制操作的调度时刻。可供选择的有：连续地复制（数据始终与出版物保持一致）、调度（在制定的时刻或者周期执行复制）或者由操作员发出复制命令。

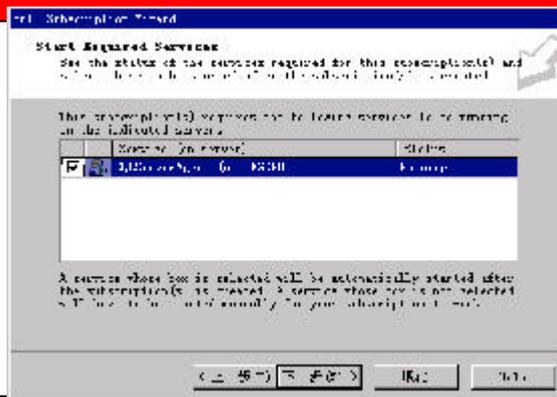


4



5

在下图中，列出了执行该订阅所需要的服务。您可以在其中确认该服务是否已经启动。单击服务名称左边的复选框，可以让该服务在向导结束后立刻启动。



6

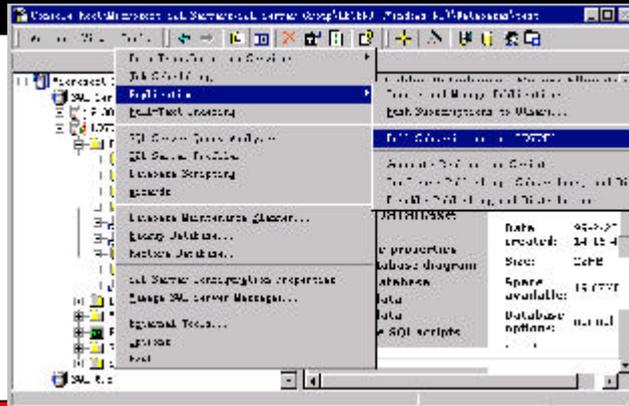
最后，在向导结束的对话框中察看有关设置，确定无误后，单击 Finish 按钮结束。否则，单击上一步按钮返回到设置处修改。



7

通过本例，学会了如何在 Enterprise Manager 中通过 Pull Subscription Wizard 向导，设置 pull 订阅。

1



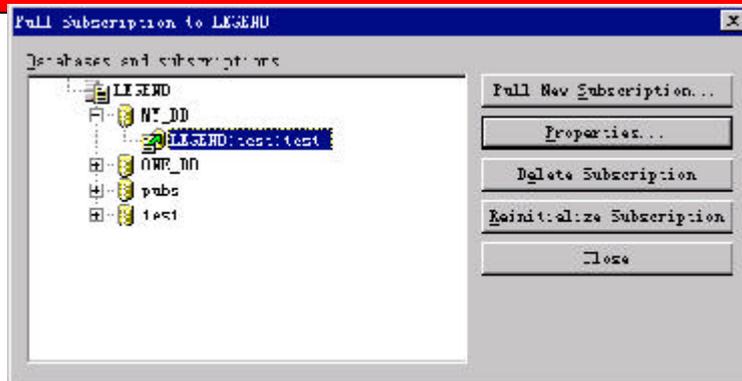
2

在下图对话框中，可以看到所有的用户数据库。以及该数据库内所有的订阅请求。



3

在上图中单击某个数据库名称左边的+框，可以察看该数据库中已有的订阅请求。如下图所示，选定订阅请求的名称后，单击 Properties 按钮继续。



4



5

在下图的 Snapshot Delivery 页面中可以匿名订阅。单击 Use the File Transfer Protocol (FTP) when downloading snapshot from the distributor 复选框。



6

在 Server Address of the Distributor 中输入分布者的名称。端口号为该服务器的 FTP 协议的端口,下面输入匿名登录的登录名和口令。输入 Anonymous 就可以了。

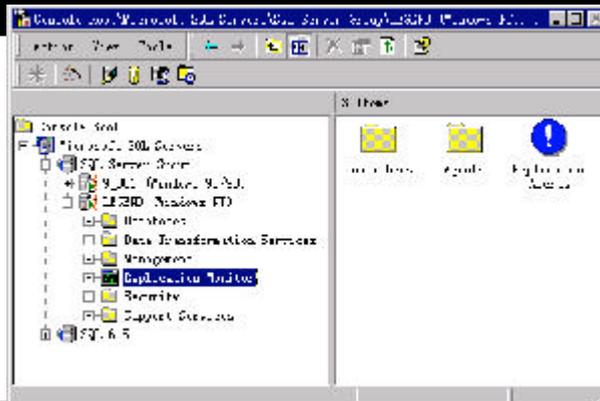


7

通过本例,学会了如何在 Enterprise 中,设定订阅为匿名订阅方式,以及如何查询和修改订阅的属性。

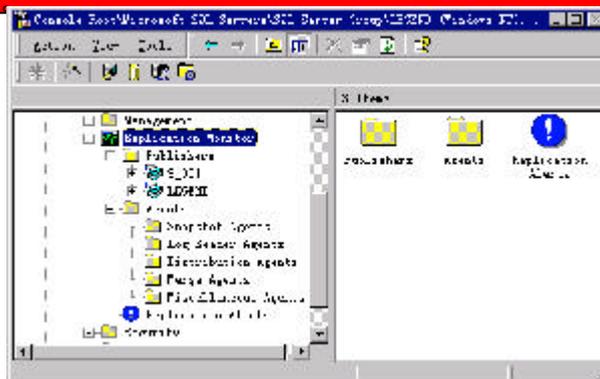
监视复制 使用复制监视器

1



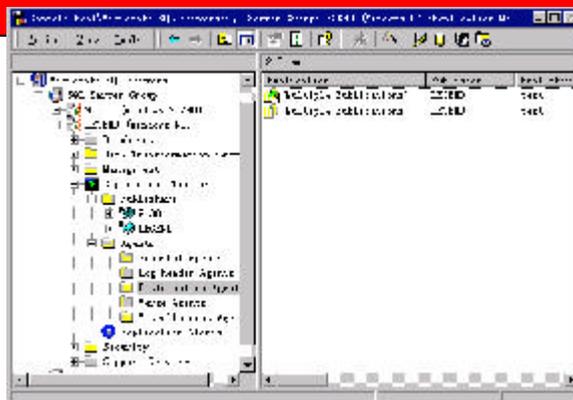
2

展开复制监视器，可以在 Publishers 中看到使用该复制者的出版者服务器名称列表，在 Agents 中看到该分布者上的所有代理，以及定义该分布者的警告。

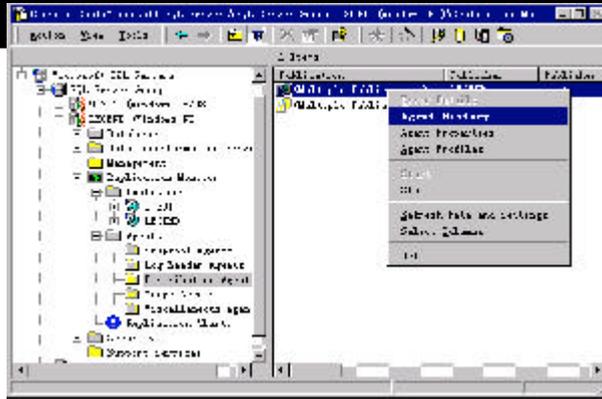


3

展开某个代理，如 Distribution Agent，可以在右窗栏中看到该分布者中注册了的分布代理列表，如下图所示。



4



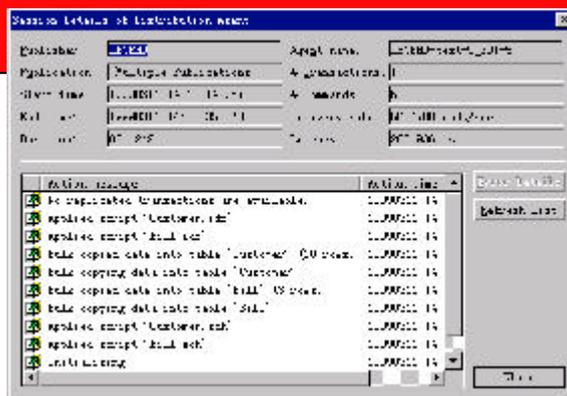
5

在上图中单击 Agent History 后，会弹出如下对话框。其中有该出版者名称，出版物名称，在下面的窗栏中是该代理的事件纪录通过 Filter the list 可以对事件进行过滤。



6

在上图中单击 Session Details 可以察看会话的详细信息，如下图所示。



7

通过本例，学会了如何使用 Enterprise Manager 中复制监视器来察看和监视复制的执行状态和历史纪录。

例

监视复制 复制监视器图标

解

1

		Publisher
		Publisher error
		Publisher retry
		Distribution database
		Subscriber properties

2

下图所示图标自上至下依次表示以下对象：快照型出版、快照型出版出错、快照型出版重试、事务型出版、事务型出版出错、事务型出版重试。

		Snapshot publication
		Snapshot publication error
		Snapshot publication retry
		Transactional publication
		Transactional publication error
		Transactional publication retry

3

下图所示图标自上至下依次表示以下对象：合并型出版、合并型出版出错、合并型出版重试、订阅、Push 订阅、订阅出错、订阅重试。

		Merge publication
		Merge publication error
		Merge publication retry
		Subscription
		Push subscription
		Subscription error
		Subscription retry

4

		Pull subscription
		Pull subscription error
		Pull subscription retry
		Pull subscription with agent running
		Pull subscription with agent not running

5

下图所示图标自上至下依次表示以下对象：代理不在运行的订阅、代理正在运行的订阅、允许出版的数据库、正在运行的 SQL Server Agent、不再运行的 SQL Server Agent、运行状态未知的 SQL Server Agent。

		Subscription (agent not running)
		Subscription (agent running)
		Database is enabled for publishing
		SQL Server Agent running
		SQL Server Agent not running
		Status of SQL Server Agent is unknown

6

下图所示图标自上至下依次表示以下对象：正在运行的 MS DTC、不在运行的 MS DTC、运行状态未知的 MS DTC、作为主码的字段、表格中没有主码、文章、复制监视器、复制监视器出错、复制监视器重试。

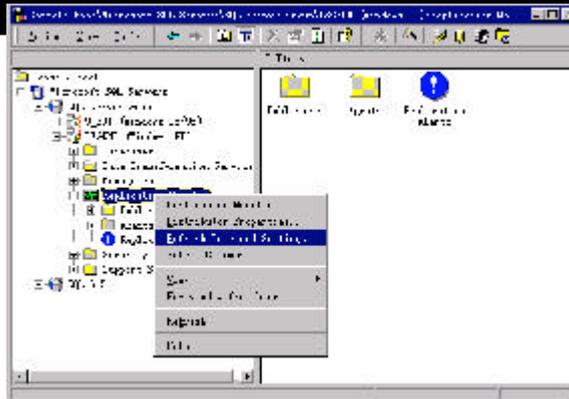
		MS DTC: running
		MS DTC: not running
		Status of MS DTC is unknown
		Column is a primary key
		No primary key in the table
		Article
		Replication Monitor
		Replication Monitor error
		Replication Monitor retry

7

通过本例，我们了解了在 SQL Server 的 Enterprise Manager 中与复制有关的重要图标及其含义。

监视复制 设置复制监视属性

1



2

在下图所示的对话框中可以设置复制监视器的刷新时间间隔，认为代理不响应的时间间隔等。



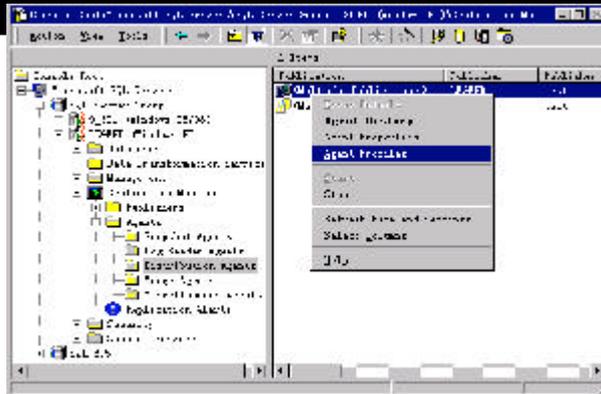
3

在 Performance Monitor 页面中可以设置性能监视器的设置文件的路径名。如下图所示：



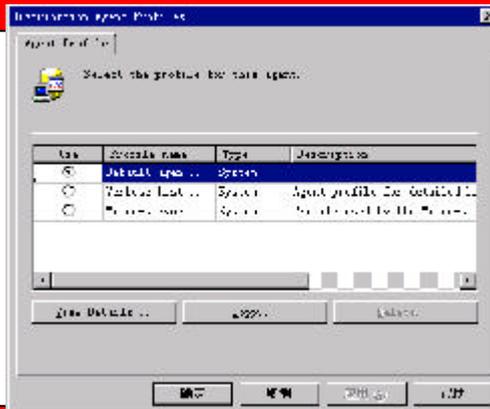
复制代理 设置复制代理参数

1



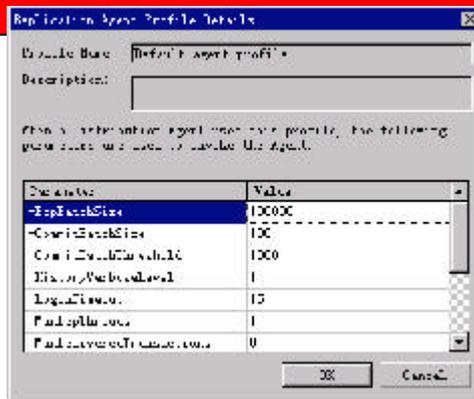
2

在弹出的对话框中，会显示该代理可供选择描述档案。初始情况下，只有三个系统档案可供选择，可以察看档案中的参数设置的细节。

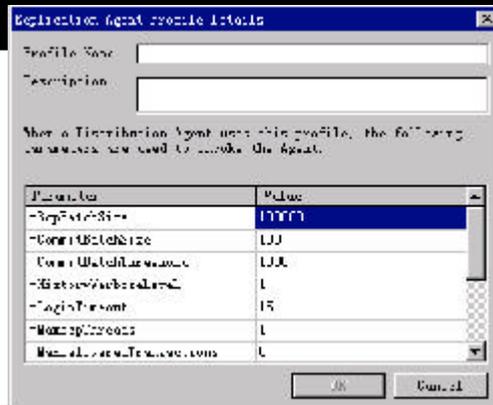


3

在上图中，选定某个档案的名称后，单击 View Details 按钮，可以察看该档案中的参数设置。如下图所示。但是不能在其中直接修改。

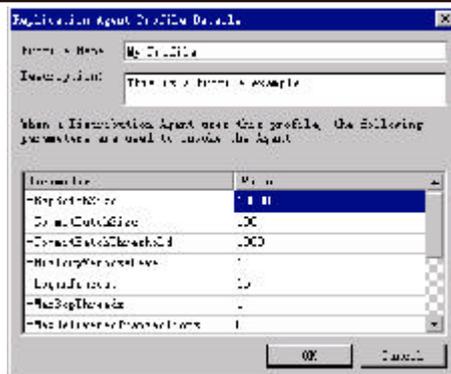


4



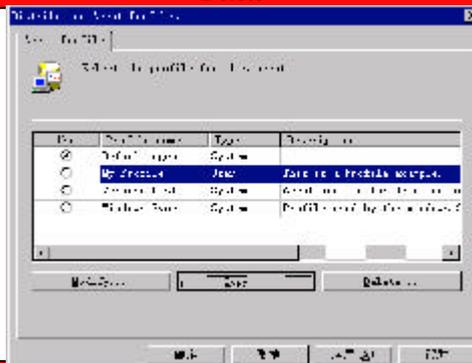
5

在上图对话框中直接修改参数值，输入档案名称、描述等。设置完单击确认。由于篇幅的关系，不可能对每个参数的意义都作解释，读者可以自己察看 SQL Server 的在线帮助获取必要信息。



6

在上图中单击确认键后，在原对话框中会发现多了一个可供选择的档案，该档案是用户档案，要使其生效，只需选中其左端的选择框。如要删除某个档案，选中该档案后，单击 Delete 按钮，确认即可。

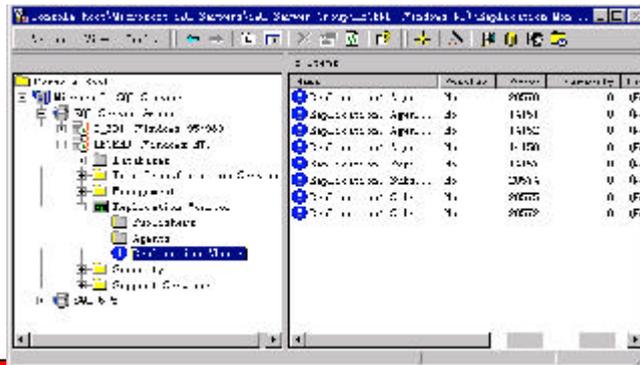


7

通过本例，学会了如何在 Enterprise Manager 中通过设置代理地描述档案，设置代理的启动参数。

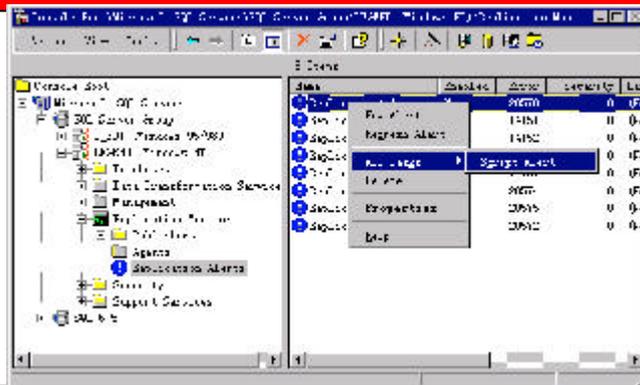
监视复制 监视复制警告

1



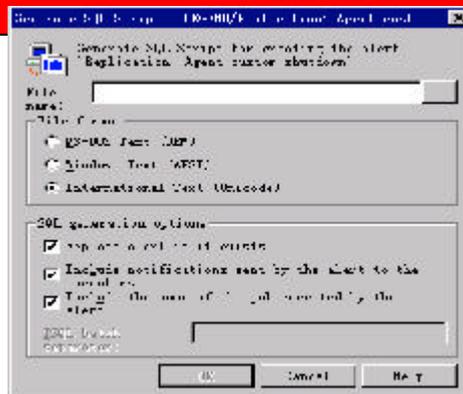
2

在右栏窗格中右键单击某个表格的名称,在弹出菜单中选择 All Tasks/Script Alert..., 可以将定义该警告的 T-SQL 语句存成文件。如下图。

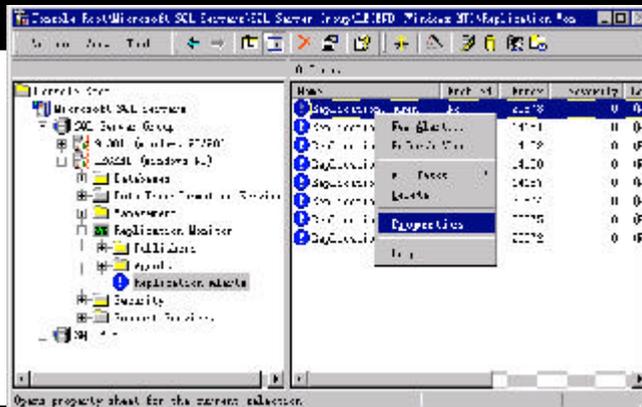


3

在下图对话框中,可以选择将 T-SQL 脚本存成文件的设置,如文件的路径、文件名称、文件格式以及其他一些选项。如下图:

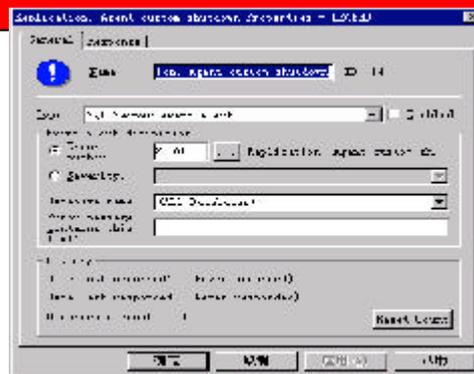


4



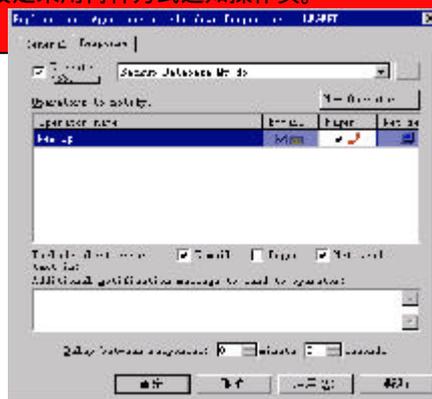
5

弹出如下对话框。其中 General 页面显示该警告的一般信息，如警告名称、类型，以及该警告发生的历史纪录等。



6

在 Response 页面中，可以设定相应的警告的自动执行任务（可以在此定义），也可以设定采用何种方式通知操作员。



7

通过本例，学会了如何在 Enterprise Manager 中察看和设置复制警告的属性。



[返回总目录](#)

第六章

数据转换

本章导读

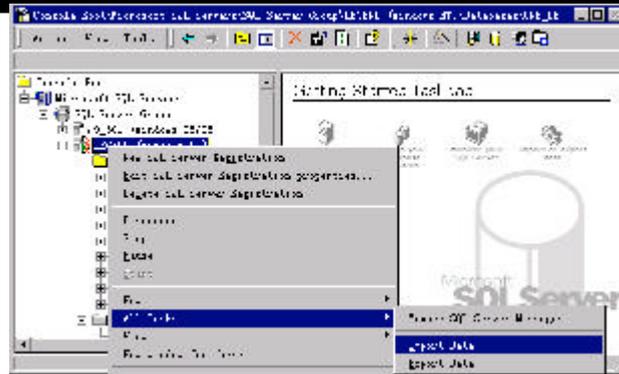
许多组织需要将数据集中管理。但是，通常数据总是按照各式各样的格式存放在不同的地方。使用 SQL Server 的数据转换服务 (Data Transformation Service)，可以非常方便地在多种不同的使用 OLE DB 库的数据源之间导入或者导出数据。

除了表格数据，数据转换服务 (DTS) 还允许在运行 SQL Server 7.0 的服务器之间传送数据库对象，如索引、存储过程等。

本章我们将要介绍 SQL Server 中数据转换服务中的一些基本概念和使用方法。本章内容有：使用数据的导入向导，从别的数据源中导入数据至你的 SQL Server 数据库中；使用数据的导出向导，将 SQL Server 数据库导出至别的数据源中；使用 DTS 设计器设计出执行复杂数据转换任务的 DTS 包。

数据的导入向导 导入 Foxpro 数据库

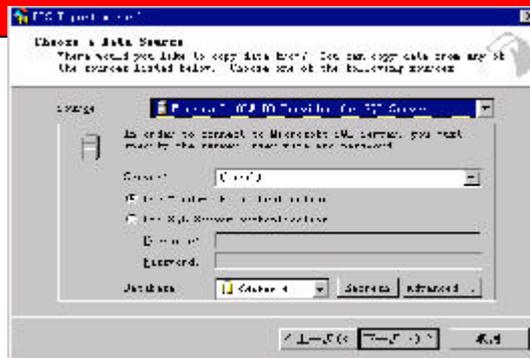
1



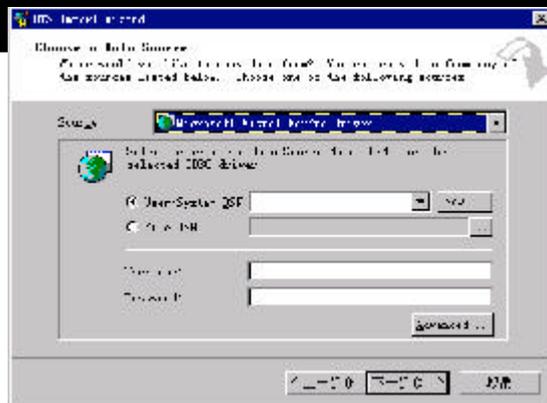
2 接着会弹出下图所示的数据转换服务的导入向导的起始页面如下图，它简要介绍了该向导所能完成的操作。阅读完后单击下一步继续。



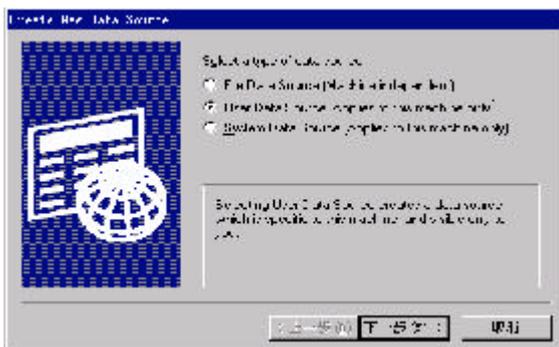
3 接着弹出的对话框，用于设定输入导入的数据源。需要在 Source 栏中选定源数据的类型。本例中，使用 Foxpro 数据源。



4



5 在本例中，使用 Foxpro 的 .mdb 文件作为数据源。需要下面即将介绍的几步操作才能指定数据源。在上图中单击 New 按钮。会弹出如下对话框。在 type of data source 中选择 User Data Source。按下一步。



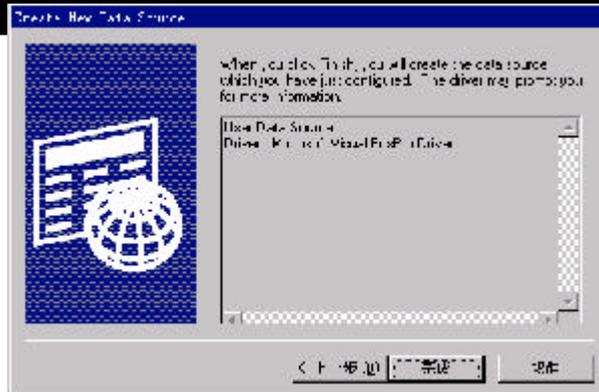
6 在下图所示的对话框中设定数据源的驱动程序为 Microsoft Visual Foxpro Driver 后，单击下一步继续。



7 本节未完，接下节。

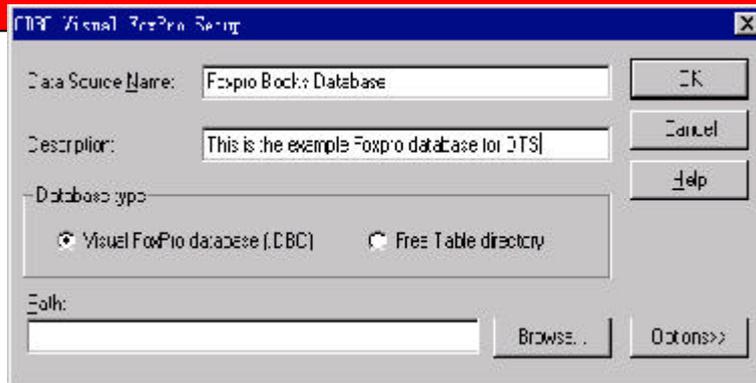
数据的导入向导 导入 Foxpro 数据库 (续)

7



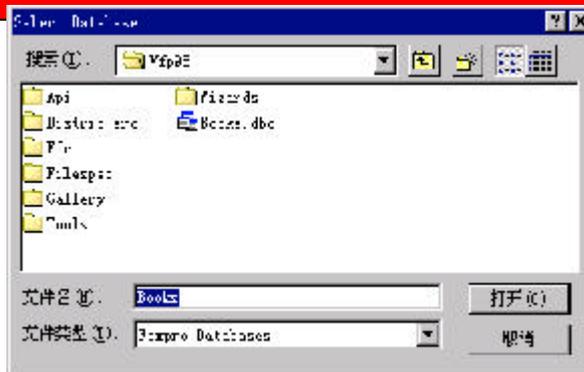
2

在下图的对话框中用具体设定数据源的文件数据库的路径，还可以添加描述信息，以便于今后使用。

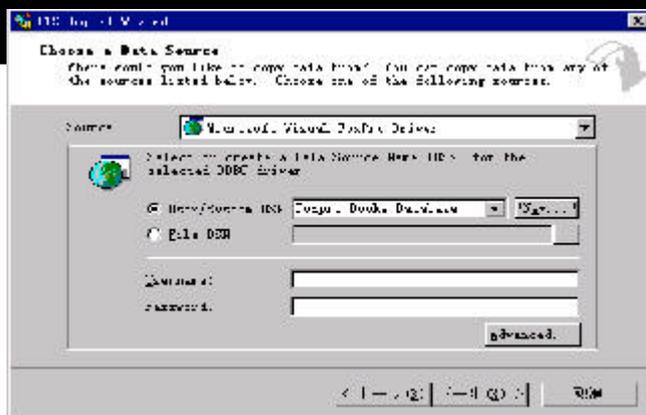


3

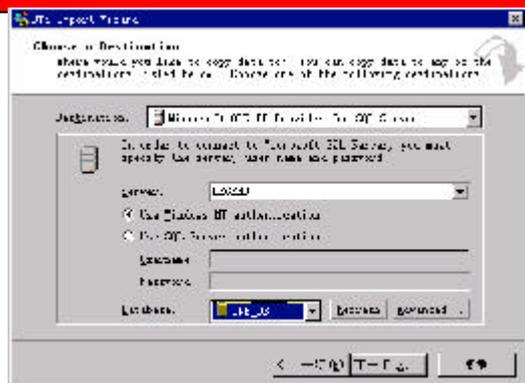
在上图中单击 Browse 按钮弹出如下对话框，在其中输入 Foxpro 文件数据库的路径名。输入完后，单击上图中的 OK 按钮，即可为导入向导设定数据源。



4



5 在如下对话框中，选定导入数据的目标数据库类型。本例中，我们使用 SQL Server 数据库作为目标数据库。在该对话框中直接选择 SQL Server Driver 最下方要设定连接服务器的安全模式。以及目标数据库的名称。

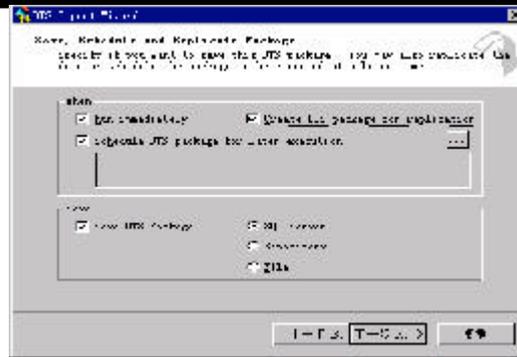


6 在下图中可以选定将源数据库中的表格拷贝到目标数据库，或者通过查询语句限制仅拷贝某些数据。本例中选择表格拷贝。



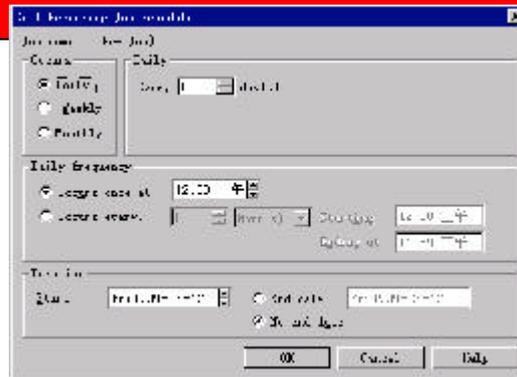
7 本节未完，接下节。

4



5

在上图中单击 **Next >** 按钮，可以设定调度执行复制操作的调度时刻，设置对话框如下图所示：



6

最后，将会显示向导的结束对话框，其中显示了在该向导中进行的设置，如果确定无误后，单击完成。否则，单击上一步返回修改。



7

通过本例，学会了如何使用数据转换服务的导入向导，将 Foxpro 文件数据库导入到 SQL Server 中。

数据的导入向导 导入文本文件数据库

7



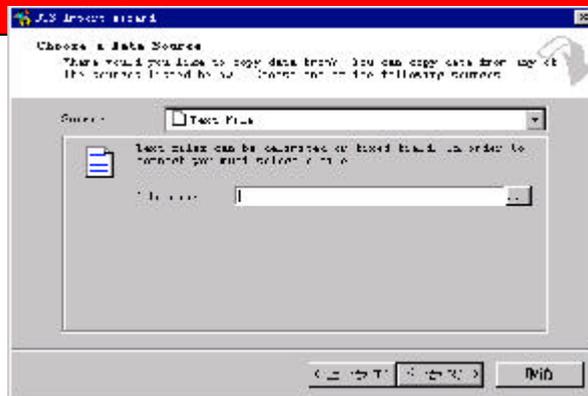
2

在下图所示的数据转化服务导入向导的起始对话框中，会显示导入向导能够完成哪些操作，阅读后单击下一步继续。



3

在数据源对话框中选定数据源为文本文件。在 Source 栏中选择 Text File，如下图所示：



4



5

在下图对话框中选择源文本文件，使用字符分隔字段，还是使用固定字段长度，以及选定文件的类型和纪录之间的间隔等属性。



6

在下面的对话框中，可以尝试着用逗号（Comma），制表符（Tab），分号（Semicolon）或者其他符号（other）来分隔字段，在下方的窗栏中可以预览按照该字符分隔的表格。

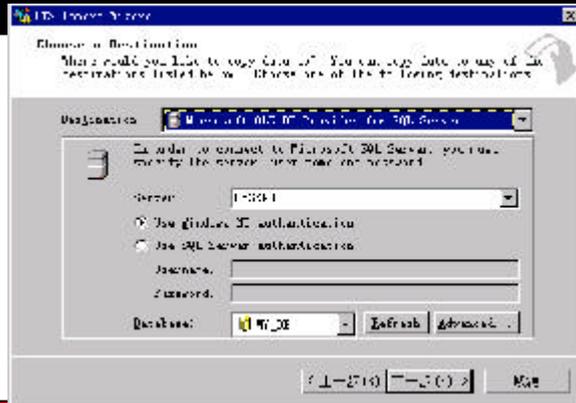


7

本节未完，接下节。

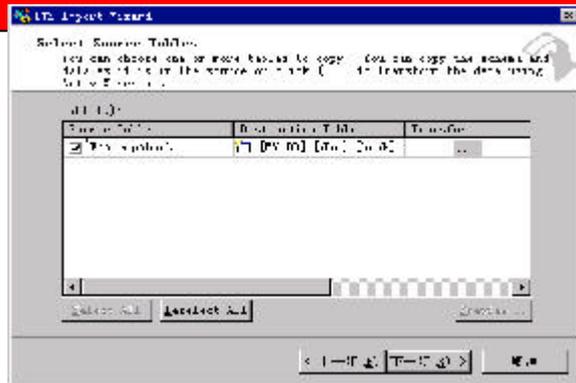
数据的导入向导 导入文本文件数据库（续）

7



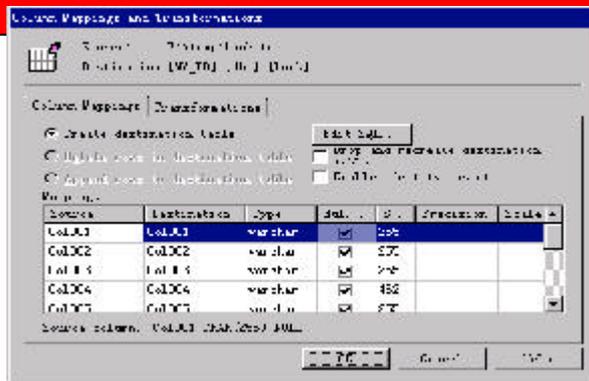
2

在下图所示对话框中可以选定需要拷贝的表格名称，只需单击表格名称左边的复选框即可选定或者取消该表格的复制。

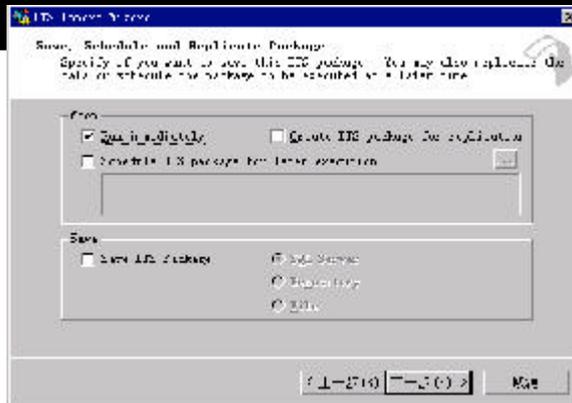


3

在上图中单击...按钮，可以设置对源数据进行转换后再拷贝至目标数据库。弹出对话框如下图所示：



4



5

在向导的结束页面，将会显示通过该向导已经进行的设置。确定无误后单击完成。如果有误，单击上一步返回修改。



6

如果在向导中设定了立刻执行，在向导结束后，会弹出下图对话框执行向导中定义的复制操作。

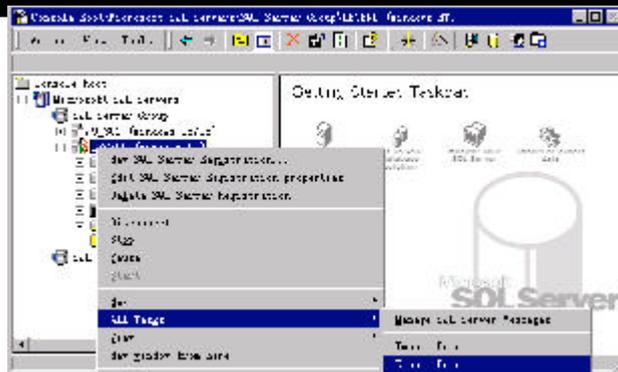


7

通过本例，学会了如何通过 SQL Server 中的数据转换服务的导入向导将文本格式的数据库文件导入到 SQL Server 数据库中。

数据的导出向导 导出数据库至 ACCESS

1



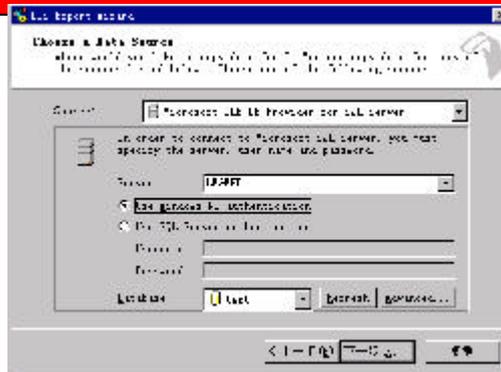
2

下图所示的为数据转换服务的导出向导的起始对话框。它显示了导出向导所能完成的操作。阅读后单击下一步继续。

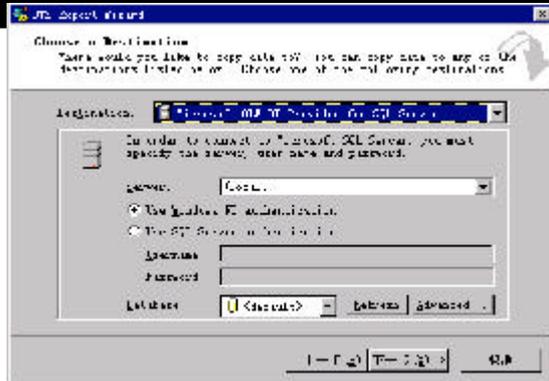


3

在下图所示的对话框中。选择导出数据的数据源。本例中选择 SQL Server。在 Source 栏中选择 Microsoft OLE DB Provider for SQL Server。在 Database 中，选择数据库的名称。

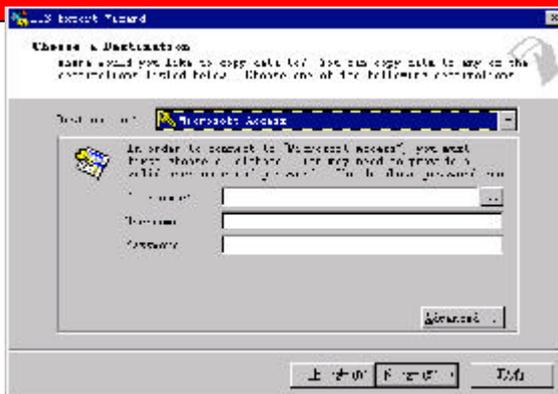


4



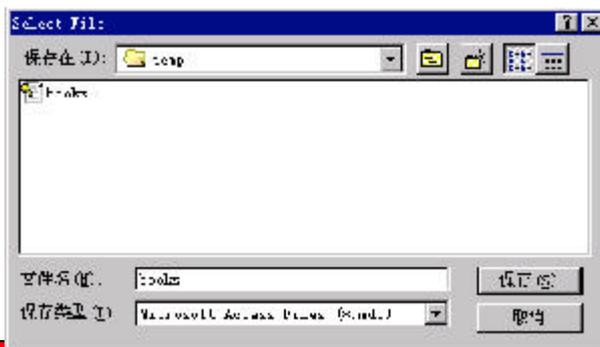
5

在 Destination 栏中，选择 Microsoft Access。下方的窗栏会变成如下所示。其中需要指定目标数据库的文件名。也可以设定打开该数据库的用户帐号和口令。



6

在上图中单击 ... 按钮，弹出如下对话框，可以在其中输入目标数据库的文件名。该文件必须是已经存在的。（注：Foxpro 和 Access 都是文件数据库，即数据库的信息可以都存放在一个文件中）。



7

本节未完，接下节。

数据的导出向导 导出数据库至 ACCESS (续)

7



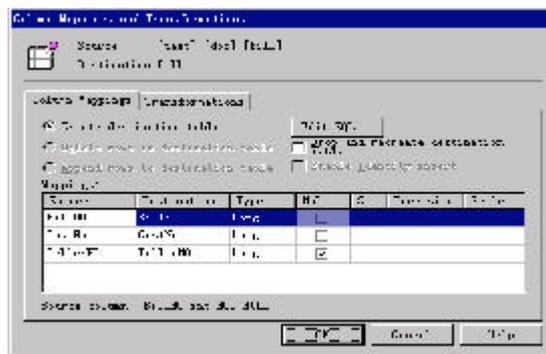
2

在下图所示的对话框中，可以选定将源数据库中哪些表格复制到目标数据库。只需单击表格名称左边的复选框，即可选定或者取消删除复制该表格。

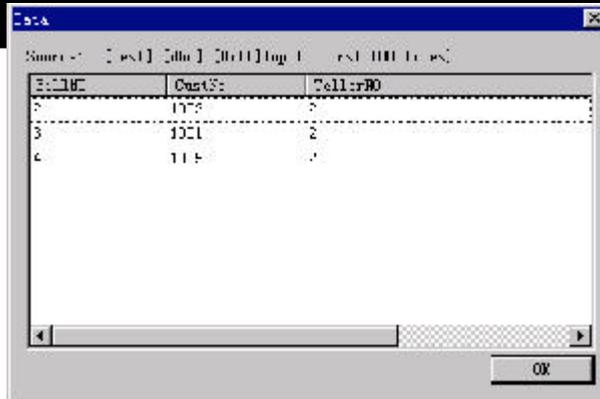


3

在上图中单击...按钮，可以弹出如下对话框。在其中，可以设定是否在进行数据拷贝过程中，对数据进行转换。以及在源表格字段和目标表格字段之间建立何种映射。

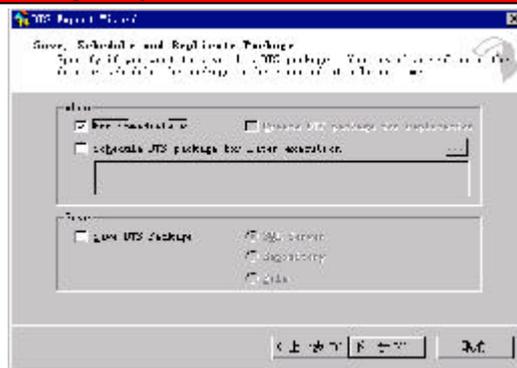


4



5

在下图所示的对话框中，可以设定是否创建 DTS 包，以及何时（立刻执行、调度执行）将该包按照何种方式存放（存在 SQL Server 的 msdb 中，存在 SQL Server 的 Repository 中，存在文件中）。



6

最后是导出向导的结束对话框。其中显示了在该向导进行的设置。确认无误后，单击完成。否则，单击上一步按钮返回修改。

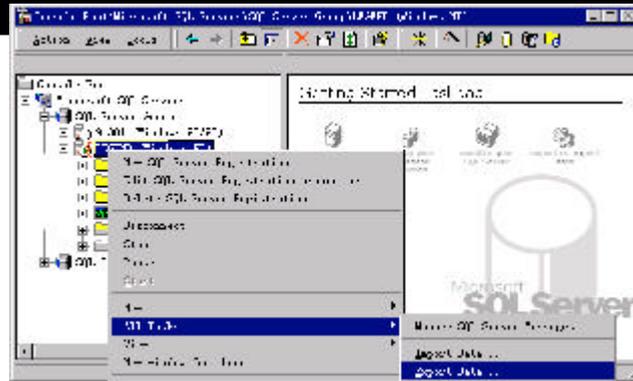


7

通过本例，学会了如何使用数据转换服务的导出向导，完成从 SQL Server 数据库到 Access 数据库的转换复制操作。

数据的导出向导 导出数据库至文本文件

1



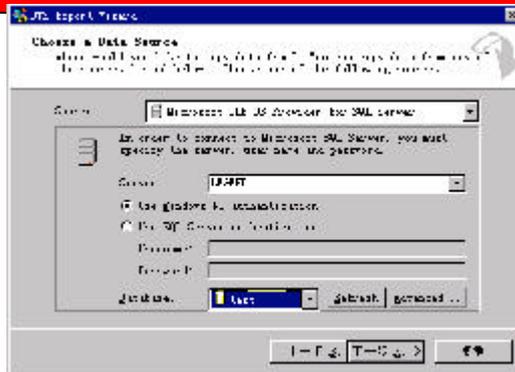
2

下图显示的是数据转换服务的导出向导的起始对话框。其中简要地介绍了该向导所能完成的操作。阅读后单击下一步继续。



3

在下图所示的对话框中，选定数据复制的源数据库类型。本例中，我们使用 Microsoft OLE DB Provider for SQL Server。选定要导出的数据库名称。

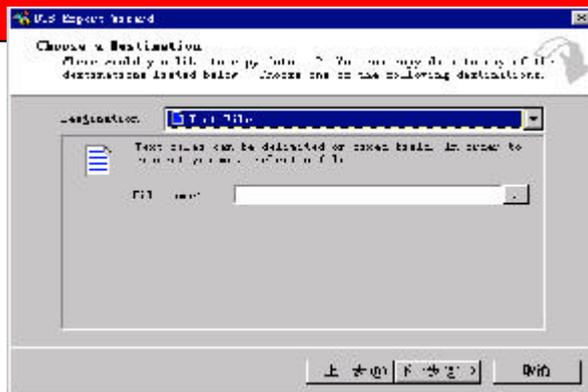


4



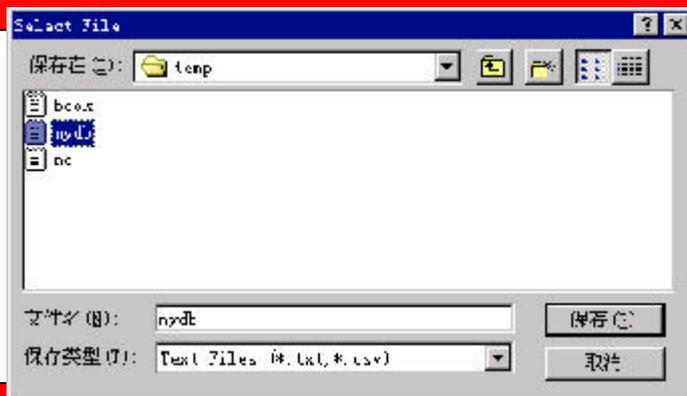
5

接下来的对话框用于选定目标数据库的类型，本例中选定用文本文件。在 Destination 栏中，选择 Text File。



6

在上图中的 File name 栏中可以输入目标文本文件的路径和文件名。单击  按钮，可以弹出如下对话框。在其中，可以设定目标文件的文件名。



7

本节未完，接下节。

数据的导出向导 导出数据库至文本文件（续）

7



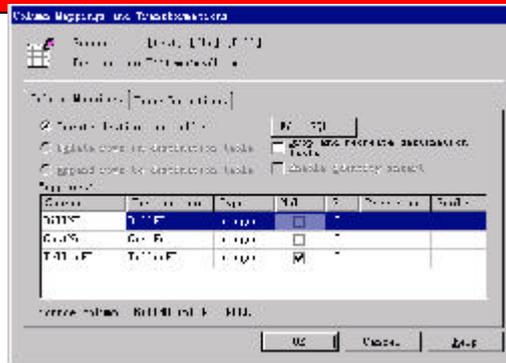
2

在下图中，您可以在 Source Table 中选定将数据库中的哪一个表格拷贝到文本文件中。设置使用字符分隔字段还是固定长度的字段。本例中使用前者。在下方设定分隔字段的字符，以及用什么标点符号来表示文本。

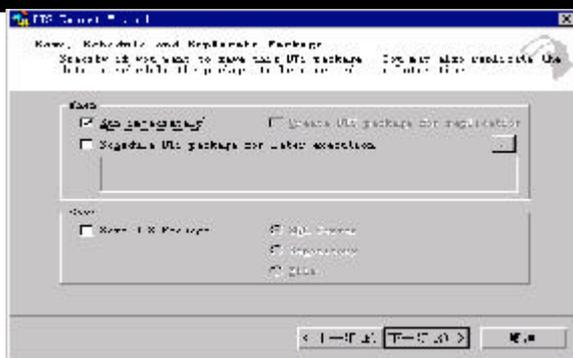


3

在上图中单击 Transform 按钮，可以弹出下图所示的对话框。在其中可以设定源字段至目标字段之间的字段映射以及定义数据转换规则的 ActiveX 脚本语句。



4



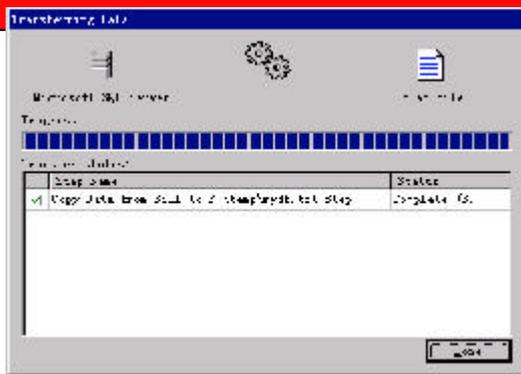
5

下图所示为数据转换服务的导出向导的结束对话框。其中显示了在该向导进行的设置。确认无误后，单击完成按钮。如要修改，单击上一步返回。



6

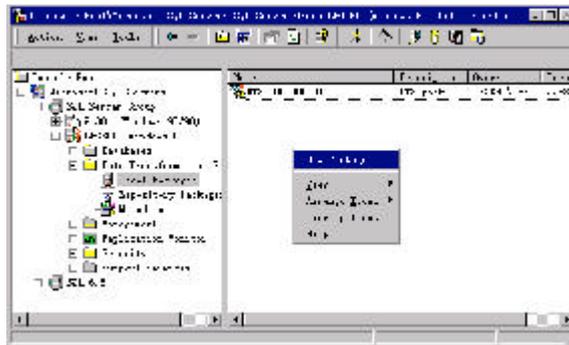
如果在向导中设定了立刻执行。在向导结束后，将会有下图对话框显示复制的执行结果。



7

通过本例，学会了如何使用数据转化服务的导出向导，将 SQL Server 中的数据库复制存储到文本文件中。

7



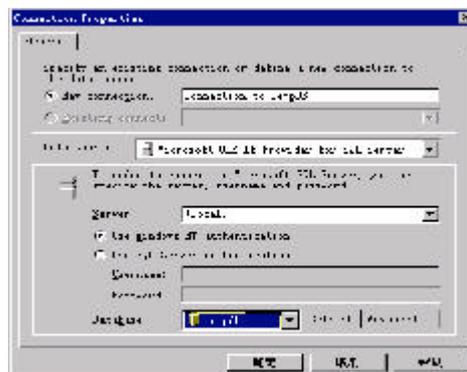
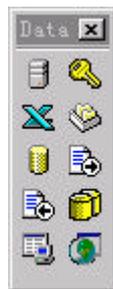
2

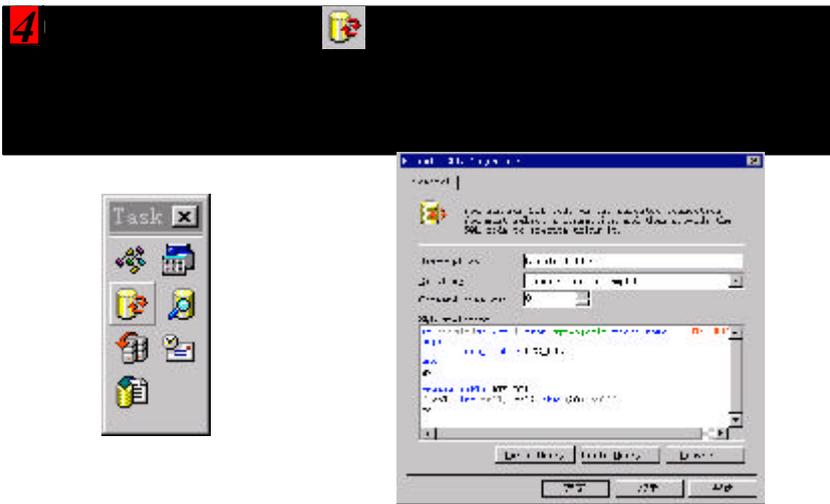
接着在 Enterprise Manager 中将会弹出下图 DTS 设计器窗口，如下图所示。该窗口由工作区、菜单、快捷按钮和浮动按钮组成。其中工作区顾名思义。快捷按钮包括编辑、保存等。浮动窗口包括一些设计的控件。



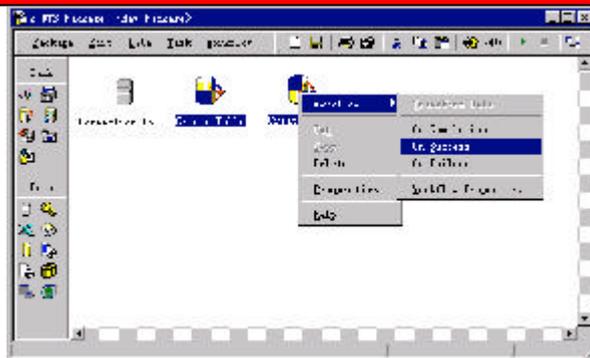
3

在下左图的浮动按钮中单击  按钮，可用于创建 SQL Server 连接。按下该按钮后，会弹出右下图的对话框，设定连接为本地的 SQL Server 的 TempDB 数据库。这里设定的连接方法与前述的向导中的设定相同。

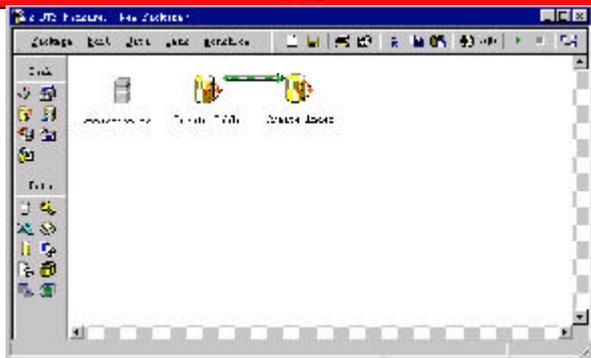




5 仿照上步的方法，再创建一个 T-SQL 任务，在上图新创建的表格中创建索引。接着在工作区中，先选中第一个任务，按住 CTRL 后，再用鼠标选中第二个任务。右键单击第二个任务，在菜单中选择 Workflow/Success。



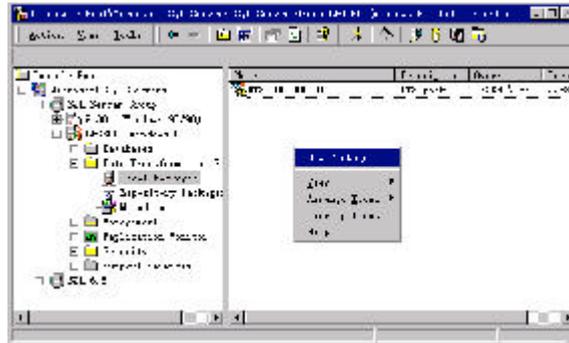
6 在下图中，在两个任务之间，多了一个绿色箭头。表示任务一执行成功后执行任务二。至此，就完成了最简单 DTS 包的设计任务。如下图。单击 ▶ 按钮，可以执行新创建的 DTS 包。



7 通过本例，学会了如何在 Enterprise Manager 中通过 DTS Designer 图形化设计 DTS 包。实际上，DTS Designer 功能非常强大，使用起来也很方便。后面还要介绍更为复杂的 DTS 包设计。

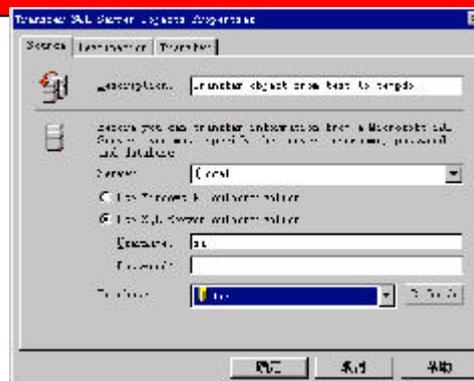
使用 DTS 设计器 在 SQL Server 间传送对象

1



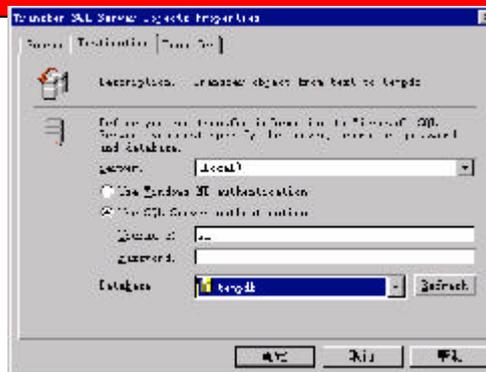
2

在弹出的 Designer 窗口中的 Task 浮动按钮栏中选择  按钮，弹出如下图所示的对话框。由 Source、Destination 和 Transfer 三个页面组成。在 Source 中输入传送对象的来源。



3

在 Destination 栏中，设置传送对象的目标服务器，输入登录该服务器的帐号及口令。选定数据库等。

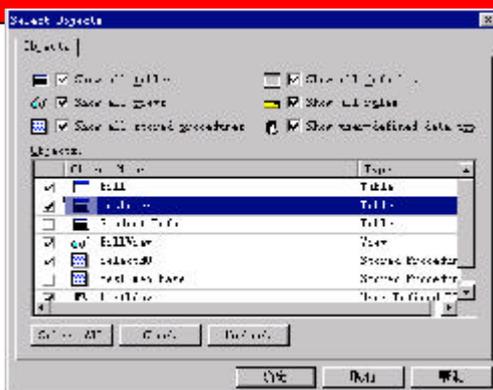


4



5

在上图对话框中,如果没有选中 Transfer All Objects,可以单击 Select Objects 按钮选择要传送源数据库中的哪些数据库对象。在弹出的如下对话框中,可以选择数据库对象。



6

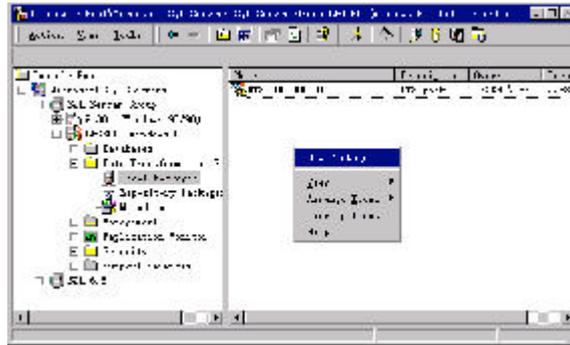
在 Designer 中右键单击工作区的空白处。弹出如下菜单,在其中选择 Add Text Annotation 可以在 Desingner 中添加文本注释。便于察看。



7

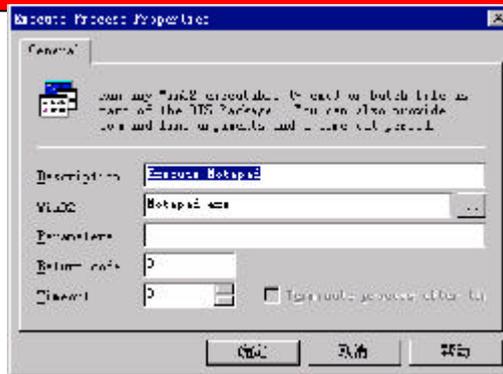
通过本例,学会了如何使用 DTS Designer(设计器)创建完成在 SQL Server 数据库之间拷贝数据库对象的包。以及如何在设计器中添加文本注释。

1



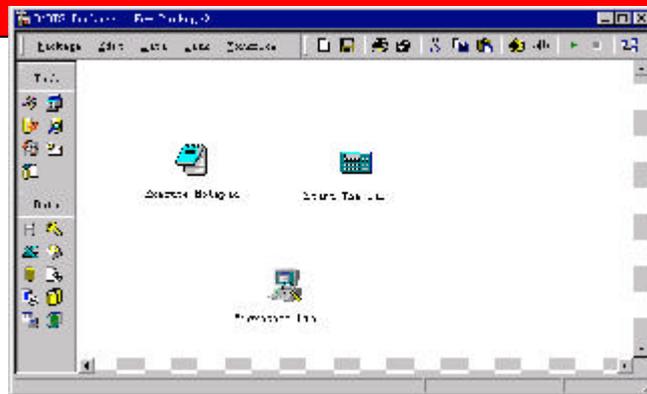
2

在 Designer 窗口中的 Task 浮动按钮栏中, 单击  按钮, 可以创建执行外部程序的任务。在弹出如右下对话框中, 输入外部程序的路径名及可执行文件名。如果需要, 还可输入命令行参数。

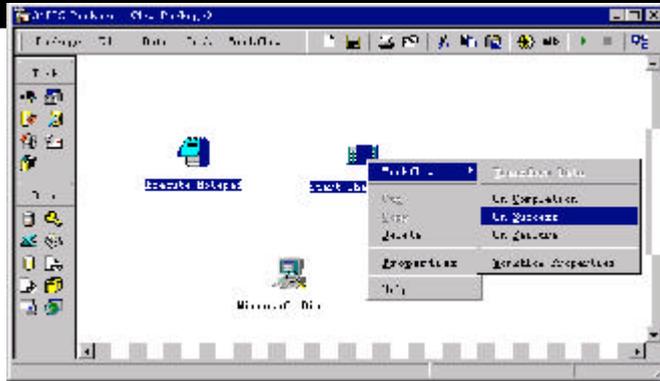


3

连续仿照上一步的方法, 再创建两个执行外部程序的任务。Designer 中的窗栏将会变成下图所示:

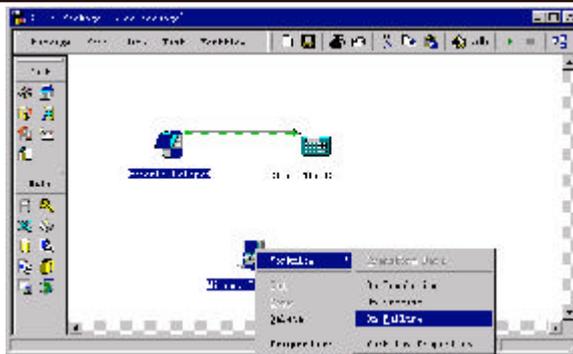


4



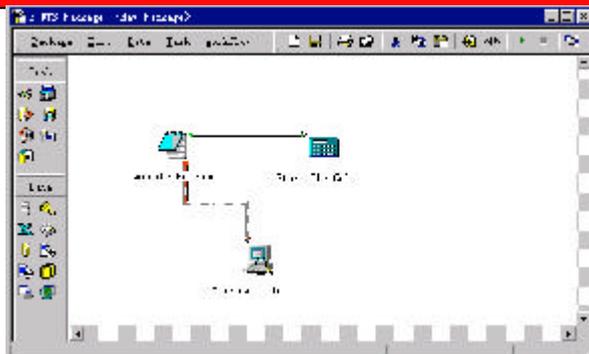
5

先选中任务一。再按住 CTRL 键，鼠标单击任务三。然后，右键单击任务三，在弹出菜单中选择 Workflow/On Failure。这样，当任务一执行失败后将执行任务三。



6

最后，我们将会看到，在 Designer 中，由三个外部程序任务创建了一个 DTS 包。其中的任务执行顺序是我们设计的。任务一成功则执行任务二，否则执行任务三。单击 ▶ 按钮，可以执行该包。



7

通过本例，学会了如何在 Enterprise Manager 的 DTS Designer 中创建一个由外部程序任务组成的 DTS 包。并且控制这些外部任务的执行逻辑顺序。

