

商务活动中 XML Updategram 对关系数据库表的更新

Updating the Relative Database Table Using XML Updategram in Business Affairs

赵俊岚 (内蒙古财经学院 计算机信息管理系 呼和浩特 010070)

摘要:介绍了 XML for SQL Server2000 Web Release1 中包含的 XML Updategram 特性,并着重介绍了 XML Updategram 功能与应用。XML Updategram 允许我们实时、无缝、透明地更新关系数据库,也可以使用接收的 XML 文档中的数据对关系数据库进行更新。这对于忙于数据更新的商务公司来说,无疑提供了一种新的、有效的更新方法。本文具体给出了使用 XML Updategram 对关系数据库表实施更新的 3 种方法。

关键词:SQL Server2000 关系数据库 XML Updategram

1 Updategram 简介

SQL Server2000 中增加了很多强大的 XML 特性,如 SELECT 语句中的 FOR XML 子句、OPENXML、XML 视图等。在 Microsoft 的 XML for SQL Server Web Release1 (WRI) 中,采用 Web Release 格式来升级 SQL Server 内部的 XML 功能,Updategram 则是该 Web Release 中的一个重要功能。

Updategram 由专门的 XML 标签块组成,这些标签块主要用于描述当前的数据状态,以及执行了 Updategram 之后的所希望得到的数据状态。通过 Updategram 可以用 XML 直接修改关系数据库的内容。

在引入 Updategram 之前,使用 XML 文档直接修改关系数据库只有一种方法,即使用 OPENXML 功能。Updategram 为我们提供了一个除 OPENXML 之外的新选择。采用 Updategram 来修改关系数据库,在性能上要高于 OPENXML; Updategram 还提供了使用 schema 并通过 HTTP 来执行关系数据库更新的功能;并且 Updategram 所需的编程量更少,使用起来比 OPENXML 更为容易;此外,使用 OPENXML 需要将整个 XML 文档调入并保留在内存中,对于较大的 XML 文档可能会导致服务器资源的大量占用,致使这种方法的非高效性;使用 OPENXML 需要处理其它的 T-SQL 或者数据库访问机制(如 ADO),因此该方法并非“纯正”的通过 XML 对关系数据库的更新方法。正是由于上述原因,使得 Updategram 成为使用 XML 并通过 HTTP 对数

据库进行直接更新的一种更好的、更完美的选择。

简单描述 Updategram 的语法如下:

```
<ROOT xmlns:updg = "urn:schemas-microsoft-com:xml-updategram" >
  <updg:sync >      <updg:before >
...   </updg:before >  <updg:after >
...   </updg:after >  </updg:sync > </
```

ROOT > 如上 Updategram 中有核心的三大块。其中 <sync > 块包含一个或多个 <before > 和 <after > 对,表示对关系数据库的多处修改,且该块中所有的 <before > 和 <after > 对均被作为一个单独的事务执行; <before > 块表示数据的现有状态; <after > 块表示后期修改以及数据的新状态。

Updategram 所执行的修改(插入、删除或更新)的类型取决于 <before > 和 <after > 块中的内容:如果只给定了 <after > 数据块,而无 <before > 块或 <before > 块置空,则执行插入操作;如果只给定了 <before > 数据块,而无 <after > 块或 <after > 块置空,则执行删除操作;如果同时给定了 <after > 和 <before > 数据块,则执行修改操作。

2 环境配置

执行 Updategram 文件之前,需要进行必要的环境配置。

首先要配置 Web 服务器,用来保存和执行 Updat-

egram 文件。此服务器一般应运行在 Windows 2000 Advaced Server 的 Microsoft IIS 5.0/6.0 之上。并使用“对 SQL Server 的 IIS 虚拟目录管理”工具在 Web 服务器上创建 IIS 虚拟目录。对 IIS 虚拟目录进行配置,使其连接到 SQL Server,并以(有访问权限的)用户身份登录。如:

在 C:\下创建子目录 Updgi,然后再在其下面创建一个子目录 template。

在“对 SQL Server 的 IIS 虚拟目录管理”工具中创建一个虚拟目录 Updgi,并制定其物理路径为 C:\Updgi (见图 1),这里我们为虚拟目录和物理目录选择了相同的名称。并在“虚拟名称”选项卡中,定义虚拟名称和路径(见图 2)。

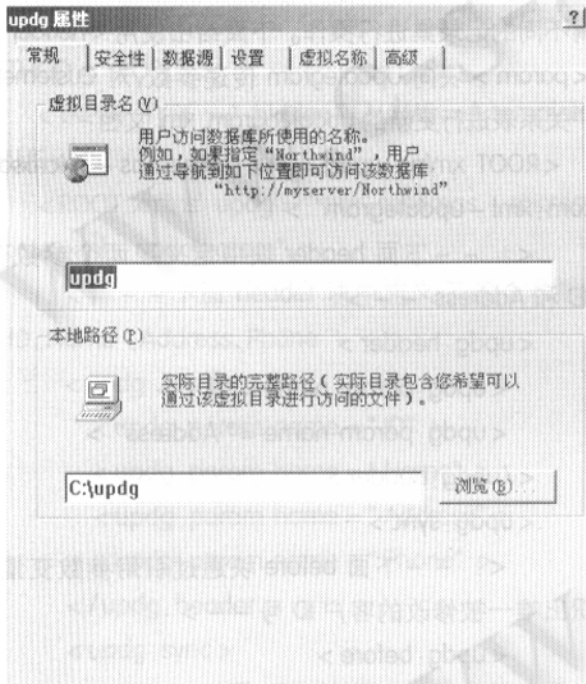


图 1 创建虚拟目录

以有效用户身份登陆 SQL Server。在“数据源”选项卡中,选择 SQL Server 及虚拟目录所使用的数据库,本例中使用的数据库名为“Business”(见图 3)。

在“设置”选项卡中进行选择,所选中的选项将允许把 Updategram 发送给 URL、使用 URL 执行模板文件、对 schema 文件执行 XPath 查询及使用 HTTP POST (见图 4)。

3 Updategram 与商务关系表数据更新

(1) Updategram 与更新。在 Business 数据库的

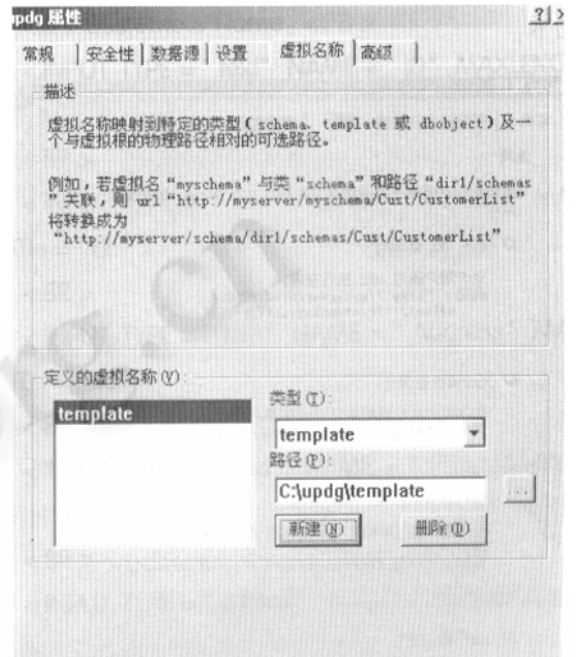


图 2 定义虚拟名称和路径

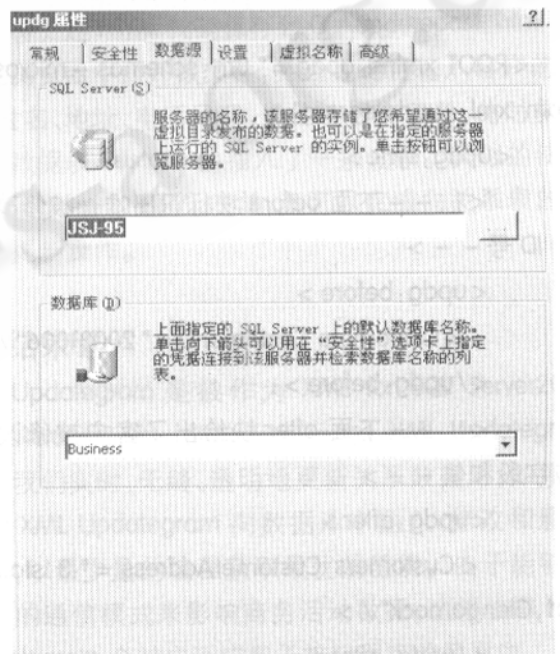


图 3 创建虚拟目录

Customers 客户关系数据表中,有 CustomerID、CustomerName、CustomerAddress、CustomerPhone 四个字

段,其中 CustomerAddress 的值为"25 Hillfoot Dr,How-
wod",若要值变更为"3 station Court,Glengarnock",
则相应 Updategram 中用于修改操作的 update.xml 文
档如下:

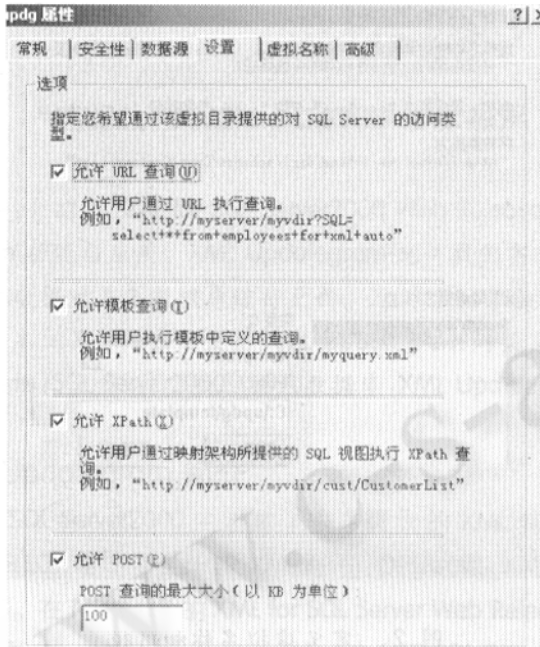


图 4 定义虚拟名称和路径

```
<ROOT xmlns:updg = "urn:schemas-microsoft-com:xml-updategram" >
  <updg:sync >
    <!-- 下面 before 块标识出唯一被修改的
    客户 ID 号 -->
    <updg:before >
      <Customers CustomerID = "20031006" / >
    </updg:before >
    <!-- 下面 after 块给出了客户被修改的
    相应字段和值 -->
    <updg:after >
      <Customers CustomerAddress = "3 station
      Court,Glengarnock" / >
    </updg:after >
  </updg:sync >
</ROOT >
```

此 update.xml 文档中 <before> 块标识出唯一被
修改的客户 ID 号,而 <after> 块则给出了被修改的相

应的字段和值。注意:这里只能有一条与客户 ID 号相
匹配的记录,如果有多条记录或没有记录与 <before
> 块中的客户 ID 号匹配,则将会出错。

将 update.xml 文档保存在 C:\Updg\template 子
目录下面,并通过浏览相应的 URL (http://localhost/
Updg/template/update.xml) 就可以执行 Updategram
了,并且实现了对 Customers 客户关系数据表中指定
了 CustomerID 的 CustomerAddress 值进行自动修改。

上面给出的 Updategram 既有 <before> 块,又有
<after> 块,表示对关系表的记录更新;若只有 <after
> 块,表示对关系表记录插入;若只有 <before> 块,表
示对关系表记录删除。

(2) Updategram 与参数传递。在实际应用中,我们
希望传递参数给 Updategram,以使其能够根据输入
的参数对不同的记录集进行操作。下面给出使用 <header
> 和 <param> 块向 Updategram 传递参数,对 Customers
客户关系表进行更新的 updateparam.xml 文档:

```
<ROOT xmlns:updg = "urn:schemas-microsoft-com:xml-updategram" >
  <!-- 下面 header 块中定义了两个参数变
  量 ID 和 Address -->
  <updg:header >
    <updg:param name = "ID" >
    <updg:param name = "Address" >
  </updg:header >
  <updg:sync >
    <!-- 下面 before 块通过引用参数变量,
    标识出唯一被修改的客户 ID 号 -->
    <updg:before >
      <Customers CustomerID = "$ID" / >
    </updg:before >
    <!-- 下面 after 块通过引用参数变量,给
    出了客户被更新的相应的字段和值 -->
    <updg:after >
      <Customers CustomerAddress = "$Ad-
      dress" / >
    </updg:after >
  </updg:sync >
</ROOT >
```

将 updateparam.xml 文档保存在 C:\Updg\tem-

plate 子目录下面,并通过浏览相应的 URL (http://localhost/Updg/template/updateparam.xml? ID = 20031006&Address = 3 station% Court, Glengarnock) 就可以实现对 Customers 客户关系数据表中给予了客户 ID 号的客户地址进行自动修改了。

(3) Updategram 与 HTML 表单。下面我们将通过 Updategram 与 HTML 提交表单 POST 方法一起使用,实现对 Customers 客户关系表的插入操作,有关 updatepost.html 文档如下:

```
<HTML >
<HEAD >
<TITLE >Insert New Customers Record </TITLE >
</HEAD >
<BODY >
< FORM ACTION = " http://localhost/updg "
METHOD = " POST" >
< INPUT TYPE = " hidden" NAME = " template"
VALUE = '
<ROOT xmlns: updg = " urn: schemas - microsoft
- com: xml - updategram" >
< ! -- 下面 header 块中定义了四个参数变
量 ID、Name、Address、Phone -- >
< updg: header >
< updg: param name = " ID" >
< updg: param name = " Name" >
< updg: param name = " Address" >
< updg: param name = " Phone" >
</ updg: header >
< updg: sync >
< ! -- 下面 after 块通过引用参数变量,给
出了客户所插入的相应的字段和值 -- >
< updg: after >
< Customers CustomerID = " $ ID" / >
< Customers CustomerName = "
$ Name" / >
< Customers CustomerAddress = " $ Ad-
dress" / >
< Customers CustomerPhone = "
$ Phone" / >
</ updg: after >
```

```
</ updg: sync >
</ ROOT >
' >
< B > Customers Name: </ B >
< INPUT TYPE = " text" NAME = " ID" MAXLENGTH
= " 8" >
< BR >
< INPUT TYPE = " text" NAME = " Name" MAX-
LENGTH = " 12" >
< BR >
< INPUT TYPE = " text" NAME = " Address" MAX-
LENGTH = " 40" >
< BR >
< INPUT TYPE = " text" NAME = " Phone" MAX-
LENGTH = " 26" >
< BR >
< INPUT TYPE = " submit" VALUE = " Submit Chan-
ges" >
</ FORM >
</ BODY >
</ HTML >
```

将上述 HTML 代码保存到 C:\updatepost.html,然后在浏览器中打开该文件,并通过键盘输入客户的 ID 号、姓名、地址、电话号码,经检验提交表单,此时便向关系数据表 Customers 插入了一条记录。类似的使用 Updategram 与 HTML 表单,还可以完成向关系表的删除与更新操作。

4 结束语

Updategram 是被作为 XML for SQL Server2000 Web Release1 的一部分而发行的。XML Updategram 允许我们实时、无缝、透明地更新关系数据库,即能够使用 XML Updategram 向数据库中插入、修改和删除数据,以最少量的编程获取最优的性能。由于能够以灵活的通信模式来影响商务活动数据,这将使得 XML Updategram 会被广泛应用于接口程序的开发中。

参考文献

- 1 Microsoft Corporation. SQL Server 2000 体系结构与 (下转第 125 页)

(上接第 87 页)

- XML/Internet 支持[M], 北京:清华大学出版社, 2001. 8.
- 2 Marci Frohock Garcia. SQL Server 2000 系统管理员宝典[M], 北京:清华大学出版社, 2001. 7.
- 3 Paul J. Burke. SQL Server 2000 XML 高级编程[M], 北京:中国电力出版社, 2003. 1.
- 4 Heather Williamson. XML 技术大全[M], 北京:机械工业出版社, 2002. 1.
- 5 H. M. Deitel P. J. Deitel. XML 编程技术大全[M], 北京:清华大学出版社, 2002. 1.
- 6 Mark Graves. XML 数据库设计[M], 北京:机械工业出版社, 2002. 8.