

.NET XML 系列 10 – 利用 DOM 修改 XML 文件

作者：許薰尹

審稿：張智凱

文章編號：N080107201

出刊日期：2007/01/02

上一期介紹利用 DOM 來產生 XML 文件的方式，這一期將延續這個話題，針對如何利用 DOM 物件新增、插入節點到 DOM 物件之中的程式寫法加以深入介紹，並利用事件的機制來偵測文件的異動。

上一期介紹利用 DOM 來產生 XML 文件的方式，這一期將延續這個話題，針對如何利用 DOM 物件新增、插入節點到 DOM 物件之中的程式寫法加以深入介紹，並利用事件的機制來偵測文件的異動。

選取、新增、移除節點

XmlDocument 類別中要選取 XML DOM 結構中的節點，通常會使用 `SelectSingleNode` 方法，或透過 `ChildNodes` 與 `Attributes` 集合。選取某個節點之後，就可以利用 XmlDocument 類別提供的新增語法，插入節點，或利用移除語法從 DOM 移除節點。但是需要注意一件事，新增或移除節點時，不可破壞 XML 文件的語法。

要移除 DOM 中的一個節點，使用 `RemoveChild` 方法。若此欲移除的節點下包含子節點，則子節點所成的子樹會一併被移除。要從 DOM 之中移除多個節點，呼叫 `RemoveAll` 方法，此方法會移除目前節點的所有子代、Attribute 等等。

2008 年 1 月號

本期相關資訊

本文難易程度：

易 ● ○ ○ ○ ○ 難

相關資源與開發工具：

1. Visual Studio 2005
2. Visual C# 2005
3. Visual Basic 2005

.NET Magazine
國際中文電子版

總編：張智凱

主編：許薰尹

社務顧問：邱世萍

技術編輯：張書源 許嘉仁

編輯顧問：鄭淑芬 胡百敬

楊先民 羅慧真

美術設計：魏吉芝 陳淑娟

行銷企劃：曹雅瑩 陳秀慧 許克臻

發行服務：謝佩珊 蔡美雲 莊麗存

發行所：毅達行銷顧問

股份有限公司

客戶服務 service@netmag.com.tw



本電子刊物之所載標誌名稱分屬各該公司所有，非經授權請勿轉載使用，版權所有。如經查證依法律追訴。



The advertisement features a background image of a person's eyes with the letters 'I T I L' superimposed. The main headline is in large, bold, red Chinese characters. Below it, a paragraph of text explains the importance of ITIL for CIOs. To the right, two event boxes provide details for seminars in Taipei and Taichung. At the bottom, logos for SYSTEX and UCOM are displayed.

擴展視野格局，您不能不學ITIL 從MIS到資訊長，您不能不知道ITIL

近年來CIO們面臨很多挑戰。根據iThome 2007年CIO大調查，有44.1%的資訊主管主要目標在提高企業營運效率，其次為改善IT流程、培養IT人員領導與商業相關能力。在2007年CIO的工作重心的調查中，更有80%是屬於ITIL裡面所談到的重要範疇，因此了解ITIL更是CIO不可不知的最佳實作。ITIL可說是IT的管理學，尤其在V3的版本當中加入結合商業流程的作業新觀念，讓CIO站在企業的商業商務的角度，不斷調教IT服務，來滿足企業營運的目標，提供CIO對於整個IT營運流程有全面性的了解，提高企業營運效率。

資訊人員晉升管理職的最佳管道

ITIL教您作對的事，而不是把事做完，對資訊人員而言，能從ITIL當中學習更完整的框架，結合IT與營運目標，了解IT在企業整體運作下所扮演的角色，在ITIL的授課程過程中會以他們所熟悉的IT環境，IT專業名詞，引導他們從純技術的思維，轉型為技術與管理平衡的思維，適合資深的IT技術人員被任命為IT主管，或是生涯規劃中準備轉入管理職的最佳選擇，可以說是“miniMBA focus on IT environment”，取得ITIL Foundation認證對於升遷與進入管理職都能給外界非常強大的說服力。

ITIL V3 課程學習座談會

台北-2008/01/10 (星期四)
PM19:00-20:30
台北市復興北路99號 2樓 / 12樓 / 14樓
(活動當天指示牌告知)
洽詢專線：02-25149191 分機100

台中-2008/01/15 (星期二)
PM19:00-20:30
台中市中港路一段201號2樓
洽詢專線：04-23297722 分機100

SYSTEX making it happen 精誠資訊 | **UCOM** 恒逸資訊 資訊訓練中心 Information Technology Education Center

範例：移除 Xml Element

CreateDOM 範例展示將 books.xml 載入 DOM 之中，然後移除代表第二本圖書 (title) 的節點。

範例中利用 XmlDocument 的 DocumentElement 取得 books 節點，利用 books 節點的 LastChild 屬性取得代表最後一本圖書的 book 節點，然後叫用 RemoveChild 將之移除。

```
Visual Basic
Dim doc As New XmlDocument()
doc.LoadXml(sb.ToString)
Dim r As XmlNode = doc.DocumentElement
r.RemoveChild(r.LastChild)
```

```
C#
XmlNode r = doc.DocumentElement;
r.RemoveChild(r.LastChild);
```

範例：使用 SelectSingleNode 搜尋要移除的節點

你也可先使用 SelectSingleNode 方法，搭配 XPath，找尋想要移除的節點，再利用 RemoveChild 方法移除節點。CreateDOM 範例展示先利用 SelectSingleNode 方法找出 pubid 為 DP 的 book 節點，利用 ParentNode 屬性找到 books 節點，然後叫用 RemoveChild 方法，移除 books 下 pubid 為 DP 的 book 節點：

```
Visual Basic
Dim r As XmlNode = doc.SelectSingleNode("/books/book[@pubid='DP']")
r.ParentNode.RemoveChild(r)
```

```
C#
XmlNode r = doc.SelectSingleNode("/books/book[@pubid='DP']");
r.ParentNode.RemoveChild(r);
```

修改節點內容

XmlComment、XmlText、XmlCDataSection、XmlWhitespace 與 XmlSignificantWhitespace 類別都繼承自 XmlCharacterData 類別，若要操作這幾種類型的節點內容，可以使用 XmlCharacterData 類別的方法。參考表 1。

方法	說明
AppendData	將字元資料附加到指定的文字節點內容之後。
DeleteData	移除節點內的文字。
InsertData	插入字元到指定節點的文字之中。
ReplaceData	取代節點中文字內容。

表 1：XmlCharacterData 類別的方法

範例：修改文字節點內容

CreateDOM 範例展示呼叫 XmlCharacterData 類別的 DeleteData 方法，來修改文字節點中部份的文字內容，將第一本書 title 中的「ADO.NET 2.0 實戰講座」內容，刪掉前三個字，改為「.NET 2.0 實戰講座」。

```
Visual Basic
Dim r As XmlNode = doc.DocumentElement
Dim t As Xml.XmlText = _
r.SelectSingleNode("book/title").FirstChild
Dim cd As System.Xml.XmlCharacterData = t
cd.DeleteData(0, 3)
```

```
C#
XmlNode r = doc.DocumentElement;
System.Xml.XmlText t =(XmlText) r.SelectSingleNode("book/title").FirstChild;
System.Xml.XmlCharacterData cd = t;
cd.DeleteData(0, 3);
```

範例：使用 Value 屬性修文字節點

XmlDocument 與 XmlNode 類別提供一些實用的方法，可利用程式來修改 XmlDocument 中的節點，變更節點中的文字內容。對於 XmlAttribute、XmlCDataSection、XmlComment、XmlDocumentType、XmlProcessingInstruction、XmlSignificantWhitespace、XmlText、XmlWhitespace，與 XmlDeclaration 等類別，都有 Value 屬性，可以直接修改 XmlNode 的 Value 屬性，來變更它們的值。

舉例來說，CreateDOM 範例中，利用 XmlNode 類別的 Value 屬性，將第一本書的書名更改為「ADO.NET 2.0 進階指南」。

```
Visual Basic
Dim r As XmlNode = doc.DocumentElement
Dim t As Xml.XmlText = _
r.SelectSingleNode("book/title").FirstChild
t.Value = "ADO.NET 2.0進階指南"
```

```
C#
XmlNode r = doc.DocumentElement;
XmlText t =
(XmlText)r.SelectSingleNode("book/title").FirstChild;
t.Value = "ADO.NET 2.0進階指南";
```

複製節點

XmlNode 類別的 CloneNode 方法可以用來複製 DOM 中的節點，這個方法有一個選擇性的參數，代表複製動作是否遞迴地進行。若設為 True，代表深層複製，複製的動作會遞迴地進行；若設為 False，代表淺層複製，只複製一層 (含 Attribute)，不複製其下的子節點。

範例：深層複製與淺層複製

CreateDOM 範例展示深層、淺層複製的作法。範例中先取得 DOM 的 DocumentElement 節點，叫用 CloneNode 方法進行複製。當 CloneNode 方法傳入 True 做為參數時，代表進行深層複製；若 CloneNode 方法傳入 False 做為參數時，代表進行淺層複製：

```
Visual Basic
Dim r As XmlNode = doc.DocumentElement
Dim deep As XmlNode = r.CloneNode(True)
MsgBox(deep.OuterXml)

Dim shallow As XmlNode = r.CloneNode(False)
MsgBox(shallow.OuterXml)
```

```
C#
XmlNode r = doc.DocumentElement;
XmlNode deep = r.CloneNode(true);
MessageBox.Show(deep.OuterXml);

XmlNode shallow = r.CloneNode(false);
MessageBox.Show(shallow.OuterXml);
```

深層複製的結果所得到的 XML 節點內容為：

```
<books><book pubid="DP"><title>ADO.NET 2.0實戰講座</title><price>690</price></book><book pubid="Delight Press"><title>ADO.NET 2.0精研講座</title><price>890</price></book></books>
```

淺層複製的結果所得到的 XML 節點內容為：

```
<books></books>
```


取代節點

若要取代某個節點的所有子節點，最簡單的方式是設定 `XmlNode.InnerXml` 屬性。只要將 XML 片段組成一個字串，然後設定給此屬性，就可以取代掉原節點的內容。注意，不要包含原節點的開始與結尾標籤。

範例：使用 `InnerXml` 屬性取代節點

CreateDOM 範例將組成 `title` 與 `price` 的 XML 字串片段指定給 `InnerXml` 屬性，將 `book` 節點下的 `title` 與 `price` 子節點取代掉：

```
Visual Basic
Dim n As XmlNode = doc.DocumentElement.SelectSingleNode("book")
Dim newTitle As String = _
"<title>ADO.NET 2.0進階指南</title><price>880</price>"
n.InnerXml = newTitle
```

```
C#
XmlNode n = doc.DocumentElement.SelectSingleNode("book");
string newTitle =
"<title>ADO.NET 2.0進階指南</title><price>880</price>";
n.InnerXml = newTitle;
```

攔截 `XmlDocument` 事件

`XmlDocument` 類別設計了一些事件，參考表 2，在 DOM 物件中因進行插入、修改，或移除節點的行為，導致模型結構變更時，可利用這些事件，進行適當的處理，以因應節點的變更。

事件	描述
<code>NodeInserting</code>	當一個節點將要新增到 DOM 時觸發。
<code>NodeInserted</code>	當一個節點新增到 DOM 後觸發。
<code>NodeRemoving</code>	當一個節點將要從 DOM 移除時觸發。
<code>NodeRemoved</code>	當移除 DOM 中的節點後觸發。
<code>NodeChanging</code>	當修改節點內容時觸發。
<code>NodeChanged</code>	當節修改點內容後觸發。

表 2：`XmlDocument` 的事件

若要攔截這些事件，可以在程式之中，利用 `XmlNodeChangedEventHandler` 物件，註冊事件發生後，後續要執行的事件處理常式。此事件處理常式中，將取得 `XmlNodeChangedEventArgs` 物件，提供異動的節點相關之資訊，參考表 3。

屬性	說明
<code>Action</code>	指明發生了新增、移除或修改節點的動作。
<code>NewParent</code>	取得動作完成後的父節點。
<code>NewValue</code>	取得動作完成後的節點值。
<code>Node</code>	取得因新增、移除或修改動作而觸發事件之節點。

OldParent	取得動作完成前的父節點。
OldValue	取得動作完成前的節點值。

表 3：XmlNodeChangedEventArgs 屬性

範例：捕捉 DOM 事件

HandleEvent 範例中將 XML 載入 DOM 物件之後，註冊了 NodeRemoved 與 NodeChanged 事件，然後在程式之中，試著將 title 節點的值設定為「ADO.NET 2.0 精研講座」，將 pubid Attribute 移除：

```
Visual Basic
Dim sb As New System.Text.StringBuilder
sb.Append("<books>")
sb.Append("<book pubid=\"DP\" title=\"ADO.NET 2.0實戰講座\"/>")
sb.Append("</books>")
Dim doc As New XmlDocument()
doc.LoadXml(sb.ToString)
AddHandler doc.NodeRemoved, AddressOf DocEventHandler
AddHandler doc.NodeChanged, AddressOf DocEventHandler
For Each node As XmlNode In doc.DocumentElement.ChildNodes
    If node.NodeType = XmlNodeType.Element _
        AndAlso node.Name = "book" Then
        Dim ac As XmlAttributeCollection = node.Attributes
        ac.GetNamedItem("title").Value = "ADO.NET 2.0精研講座"
    End If
    If node.NodeType = XmlNodeType.Element _
        AndAlso node.Name = "book" Then
        Dim ac As XmlAttributeCollection = node.Attributes
        ac.RemoveNamedItem("pubid")
    End If
Next

C#
System.Text.StringBuilder sb = new System.Text.StringBuilder();
sb.Append( "<books>" );
sb.Append("<book pubid=\"DP\" title=\"ADO.NET 2.0實戰講座\"/>");
sb.Append("</books>");
XmlDocument doc = new XmlDocument();
doc.LoadXml( sb.ToString() );
doc.NodeRemoved +=
new XmlNodeChangedEventHandler(DocEventHandler);
doc.NodeChanged +=
new XmlNodeChangedEventHandler(DocEventHandler);
foreach (XmlNode node in doc.DocumentElement.ChildNodes) {
    if (node.NodeType == XmlNodeType.Element &&
        node.Name == "book") {
        XmlAttributeCollection ac = node.Attributes;
        ac.GetNamedItem("title").Value = "ADO.NET 2.0精研講座";
    }
    if (node.NodeType == XmlNodeType.Element &&
        node.Name == "book") {
        XmlAttributeCollection ac = node.Attributes;
        ac.RemoveNamedItem("pubid");
    }
}
```

在 `XmlNodeChanged` 事件處理常式中，利用 `Action` 屬性判斷發生了新增、移除或修改節點的動作，然後將節點的 `OuterXml` 資訊顯示在 `ListBox` 控制項上：

```
Visual Basic
Sub DocEventHandler(ByVal source As Object, _
    ByVal e As XmlNodeChangedEventArgs)
    Select Case e.Action
        Case XmlNodeChangedAction.Remove
            ListBox1.Items.Add("Remove!!" & e.Node.OuterXml())
        Case XmlNodeChangedAction.Change
            ListBox1.Items.Add("Change!!" & e.Node.OuterXml())
        Case XmlNodeChangedAction.Insert
            ListBox1.Items.Add("Insert!!" & e.Node.OuterXml())
    End Select
End Sub
```

```
C#
void DocEventHandler(object source, XmlNodeChangedEventArgs e)
{
    switch (e.Action) {
        case XmlNodeChangedAction.Remove: {
            ListBox1.Items.Add("Remove!!" + e.Node.OuterXml());
            break;
        }
        case XmlNodeChangedAction.Change: {
            ListBox1.Items.Add("Change!!" + e.Node.OuterXml());
            break;
        }
        case XmlNodeChangedAction.Insert: {
            ListBox1.Items.Add("Insert!!" + e.Node.OuterXml());
            break;
        }
    }
}
```

執行的結果參考：

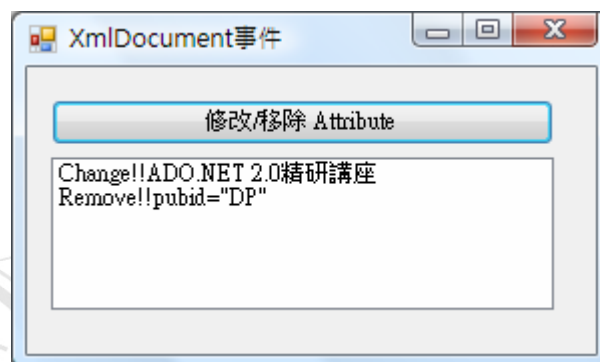


圖 1：事件處理

總結

要移除某節點與其下的子節點可以使用 `RemoveChild` 方法；要移除節點的所有子代呼叫 `RemoveAll` 方法。若要操作 `XmlComment`、`XmlText`、`XmlCDATASection`、`XmlWhitespace` 與 `XmlSignificantWhitespace` 類型的節點內容，可以使用 `XmlCharacterData` 類別的方法。複製 DOM 中的節點可以使用 `XmlNode` 類別的 `CloneNode` 方法。取代某個節點可以利用設定 `XmlNode.InnerXml` 屬性來達成。

關於作者



許薰尹

現任職恆逸資訊教育訓練處系統開發部技術副理，為資深講師與顧問，擁有 MCSD、MCAD、MCT 等認證，有多本譯作，著有《ADO.NET 程式設計開發指南》《C#與.NET Framework 實戰演練》《精通 C# 與 .NET Framework 實戰演練》等書，並為技術雜誌專欄作家，也為 .NET Mag 主編兼網站站長。

