本文将 LINQ to XML 与下面的 XML 技术进行比较：[XmlReader](https://learn.microsoft.com/zh-cn/dotnet/api/system.xml.xmlreader)、XSLT、MSXML 和 XmlLite。 了解这些信息，有助于确定要使用哪种技术。

有关比较 LINQ to XML 与文档对象模型 (DOM) 的信息，请参阅 [LINQ to XML 与 DOM](https://learn.microsoft.com/zh-cn/dotnet/standard/linq/linq-xml-vs-dom)。

**LINQ to XML 与XmlReader**

[XmlReader](https://learn.microsoft.com/zh-cn/dotnet/api/system.xml.xmlreader) 是一种快速的只进非缓存分析器。

LINQ to XML 在 [XmlReader](https://learn.microsoft.com/zh-cn/dotnet/api/system.xml.xmlreader) 基础之上实现，它们紧密集成在一起。 但是，您也可以直接使用 [XmlReader](https://learn.microsoft.com/zh-cn/dotnet/api/system.xml.xmlreader)。

例如，假设要生成一项 Web 服务，该服务每秒将分析几百个 XML 文档，而这些文档具有相同的结构，因此，只需编写一种代码实现即可对 XML 进行分析。 在这种情况下，可能想要直接使用 [XmlReader](https://learn.microsoft.com/zh-cn/dotnet/api/system.xml.xmlreader)。

相反，如果要生成一个系统，用以分析多种小型 XML 文档，而这些文档各不相同，则可能希望利用 LING to XML 提供的生产力改进。

**LINQ to XML 与XSLT**

LINQ to XML 和 XSLT 都提供丰富的 XML 文档转换功能。 XSLT 是基于规则的声明性方法。 XSLT 高级程序员以函数编程方式编写 XSLT，这种方式强调无状态方法。 可以使用实现后无副作用的纯函数编写转换。 许多开发人员还不熟悉这种基于规则的方法（即函数方法），需要付出努力和花费大量时间来学习这项技术。

XSLT 是非常高效的系统，可以生成高性能的应用程序。 例如，一些大型 Web 公司使用 XSLT 从 XML（提取自多种数据存储区）生成 HTML。 托管 XSLT 引擎将 XSLT 编译为公共语言运行时 (CLR) 代码，在某些情况下，其性能甚至比本机 XSLT 引擎还要好。

但是，XSLT 没有利用许多开发人员都具备的 C# 和 Visual Basic 知识。 它要求开发人员用一种不同的复杂编程语言来编写代码。 如果使用两种非集成开发系统，例如 C#（或 Visual Basic）和 XSLT，软件系统的开发和维护会更加困难。

熟悉如何使用 LINQ to XML 查询表达式后，LINQ to XML转换是一种易于使用的强大技术。 本质上，XML 文档是这样形成的：使用函数构造，从各种源提取数据，动态构造 [XElement](https://learn.microsoft.com/zh-cn/dotnet/api/system.xml.linq.xelement) 对象，再将全部内容组合到一个新 XML 树中。 经过这种转换，可以生成一个全新的文档。 相对来说，在 LINQ to XML 中构造转换比较容易、直观，编写出的代码可读性也较强。 这样可以减少开发和维护成本。

LINQ to XML 不用于替换 XSLT。 对于复杂的以文档为中心的 XML 转换，XSLT 仍是很好的工具，如果文档结构的定义不完备，更是如此。

XSLT 的优势在于符合万维网联合会 (W3C) 标准。 如果要求只使用符合标准的技术，XSLT 可能更为合适。

XSLT 是 XML，因此可以以编程方式进行操作。

**LINQ to XML 与MSXML**

MSXML 是基于 COM 的技术，用于处理 Microsoft Windows 提供的 XML。 MSXML 提供 DOM 的本机实现，并且支持 XPath 和 XSLT。 它还包含基于事件的 SAX2 非缓存分析器。

MSXML 性能良好，默认情况下，在大多数情况中都是安全的，并且可在一个浏览器中访问，以便执行 AJAX 式应用程序中的客户端 XML 处理。 在任何支持 COM 的编程语言（包括 C++、JavaScript 和 Visual Basic 6.0）中，都可以使用 MSXML。

建议不要在基于 CLR 的托管代码中使用 MSXML。

**LINQ to XML 与XmlLite**

XmlLite 是一种只进非缓存提取型分析器。 开发人员主要将 XmlLite 用于 C++。 建议开发人员不要将 XmlLite 用于托管代码。

XmlLite 的主要优势在于它是快速的轻量 XML 分析器，在大多数方案中都是安全的。 它可能受到的威胁比较少。 如果必须分析不受信任的文档，并且要预防受到拒绝服务或数据暴露等攻击，则 XmlLite 可能是很好的选择。

XmlLite 未与语言集成查询 (LINQ) 集成。 它不会提高程序员的工作效率，而工作效率提高正是 LINQ 背后的推动力量。