

The LINQ Project

.NET Language Integrated Query - faq

The LINQ Project .NET Language Integrated Query

Para reducir la complejidad para los desarrolladores, y ayudar a fomentar la productividad, Microsoft presentó hoy la solución, el .NET Framework llamado el Language Integrated

Query (LINQ) Project. Un conjunto de extensiones para el lenguaje de programación C# y Visual Basic .NET, que extiende el .NET Framework proveyendo integración de consultas para objetos, base de datos y XML

Usando el LINQ los desarrolladores estarán capacitados para nativamente escribir consultas en C# y Visual Basic sin tener que usar otro lenguaje como el Structured Query Language (SQL) o XQuery, un lenguaje de consulta para acceder a datos XML.



Entrevista de la editorial de Microsoft a Anders Hejlsberg y Paul Vick, arquitectos de C# y VB respectivamente.

Usted habló frecuentemente sobre la incompatibilidad de la impedancia entre los lenguajes de programación usados para escribir aplicaciones y los lugares para almacenar datos. ¿Puede hablar más sobre como esto complica el trabajo de los

desarrolladores?

Hejlsberg: Cuando desarrolladores escriben aplicaciones empresariales hoy en día usando Visual Basic, Java, C# o cualquier idioma basado en objetos, ellos casi siempre necesitan tener acceder a datos. Por ejemplo, un desarrollador que esta creando una página Web de e-commerce necesita recuperar informaciones del producto, por ejemplo el precio y la cantidad actual, directamente de una base de datos relacional. El lenguaje de programación que se necesita para acceder esta data puede ser SQL, XQuery o ambos, pues el desarrollador tiene que cambiarse de escribir por ejemplo en C# a escribir comandos de SQL o XQuery que están basados en comandos string. No solamente en este caso se necesita saber varios lenguajes de programación, sino también buenos conocimientos de aplicar las interfases de programación (APIs) que se conectan en dominios diferentes, como por ejemplo ADO.NET, o Open DataBase Connectivity (ODBC).



Es un trabajo difícil aprender estos diferentes lenguajes de programación y APIs, porque la incompatibilidad semántica entre la programación de datos relacionales y la programación orientada en objetos esta muy grande. Por ejemplo, estos dos mundos tienen sistemas diferentes de tipo, diferentes tipos de datos incorporados, y diferentes funciones estándares. Además, solamente el mundo relacional suporta consultas.

Vick: Los desarrolladores también tienen que vivir con la falta de integración entre estos dominios. Por ejemplo, cuando escriben consultas SQL o XML para una aplicación que esta escrito en la mayoría en lenguaje de programación, no hay control del tipo de tiempo de compilación que les deja saber si su consulta está escrita correctamente. En vez de esto, se enteran de errores de codificación solamente después de mandar la consulta al servidor de la base de datos, si esto no tiene éxito.

Language Integrated Query – **NeuronalTraining.NET**

Hejlsberg: Pues hasta ahora, la construcción de aplicaciones de datos habilitados típicamente cuesta mucho tiempo, tiene mucho peligro de errores y de una experiencia de programación no integrada - un problema que era difícil de solucionar en toda la industria.

¿Qué es el proyecto Language Integrated Query, y como se está ocupando de este problema?

Hejlsberg: El proyecto LINQ está hecho para mejorar la productividad reduciendo la necesidad para los desarrolladores de aprender y usar lenguajes múltiples, diferentes lenguajes cuando construyen aplicaciones de datos habilitados. Lo hace extendiendo el .NET Framework para incluir operaciones de consultas integradas en lenguaje. Eso significa que los desarrolladores pueden construir aplicaciones de datos habilitados completamente en C# o Visual Basic, sin que tienen que cambiar a SQL o XQuery para escribir las porciones de consultas del código.

Vick: LINQ tiene varios componentes, el primero de ellos es un conjunto de especificaciones que definen como las consultas están expresadas en la forma de API. Estas especificaciones son de importancia primordial porque establecen las reglas para añadir soporte de lenguaje para cualquier lenguaje LINQ-enabled, y para añadir soporte de consultas para cualquier objeto LINQ-enabled. Entonces, basado en estas especificaciones, el proyecto LINQ incluye un conjunto de extensiones para lenguajes C# y Visual Basic, y un conjunto de librerías que provee consultas integradas para objetos, base de datos, y datos XML usando sintaxis de lenguajes nativos.

Por el defecto, los operadores de consulta base de LINQ pueden ser aplicados a cualquier array del NET o colección de objetos, que eficazmente provee semejante a SQL y XQuery, como las habilidades para cualquier datos que estén en memoria. LINQ también proporciona dos nuevos APÍs críticos: Dlinq, para tener acceso relacional, base de datos relacionales, y Xlinq, para tener acceso jerárquico, basados en datos XML. Las capacidades de consulta del proyecto LINQ son compositivas, significa que las consultas múltiples pueden ser combinadas fácilmente juntas. Las consultas simples pueden tener acceso a datos de dominios múltiples, y usted puede también especificar la ejecución local y remota.

Hejlsberg: LINQ puede también aumentar la productividad del desarrollador permitiendo a los desarrolladores utilizar todas las características de Visual Studio con las que están familiarizados, como IntelliSense, una característica que proporciona las sugerencias pop-up para el código como se está escribiendo, y verificación de tipo en tiempo de compilación, lo cual proporciona a desarrolladores saber sobre los errores mientras están escribiendo sus consultas, en vez de enterarse sobre ellas más adelante cuando el código no trabaja.

¿Cuánto tiempo el proyecto LINQ ha estado en formación?

Hejlsberg: El proyecto LINQ como esta hasta ahora, es el resultado de la mitad de una década de esfuerzo. Sus múltiples características principales, las nuevas características en el Framework 2.0 para unificar los lenguajes orientados a objetos, y lenguajes de la base de datos en un modelo de programación común. Este proyecto integral y ambicioso ha sido un esfuerzo de colaboración broad-based de parte de Microsoft, implicando el lenguaje, los datos, y los equipos del runtime. Hemos estado trabajando muy duro en él por varios años, y ahora esperamos que ese trabajo continúe.

¿Qué tan innovador es el proyecto LINQ? ¿Hay otros fixes para este problema a nivel industrial?

Hejlsberg: Creemos que el proyecto LINQ provee habilidades muy poderosas que son diferentes a los ofrecidos por otros paradigmas de programación.

Con LINQ, las consultas, operaciones determinadas, y transformaciones son convertidas en conceptos de primera clase en los lenguajes .NET y APIs al modo de las clases y objetos. Las

Language Integrated Query – **NeuronalTraining.NET**

consultas enriquecidas que previamente solo pudieron estar escritas en SQL o XQuery, se pueden expresar ahora nativamente en C# y Visual Basic.

La amplitud de proyecto LINQ también es único. La mayoría de soluciones se han centrado en un solo dominio, tal como XML o SQL, o en un lenguaje particular, mientras que LINQ es diseñado desde cero, para ser accesible y extensible, soportando todas las formas de objetos y datos, usando un lenguaje permitido por LINQ.

Abrimos el camino aquí, pero no hay nada intrínseco sobre LINQ que excluiría a otros lenguajes como C++ o Javascript, o lenguajes de terceros de añadir esta capacidad.

¿Cómo piensas que los desarrolladores respondan al proyecto LINQ?

Vick: Cada vez que hacemos exhibiciones previas del proyecto LINQ para los clientes, hemos recibido respuestas auténticamente entusiastas.

Por que el proyecto LINQ realmente cambia el modelo de programación y lo hace trabajando con toda las clases de datos mucho mas fácil, facilitará cambios positivos en la forma de cómo los desarrolladores trabajan hoy. Las aplicaciones que solieron requerir un montón de código van a ponerse demasiado fáciles y naturales para ellos. Por ejemplo, un desarrollador que está trabajando con una colección de objetos customer y quiere consultar una lista de nombres de esa colección necesitaría escribir 5 o 10 líneas de código para lograr esta tarea simple con las tecnologías de hoy.

Con LINQ esa lista pude ser obtenida escribiendo solamente una línea de código, "Select Name From Customers". Además, las habilidades de consulta previstas por el proyecto LINQ permiten a los desarrolladores combinar operaciones, como campos de filtración de una colección, ordenándolos en nuevas colecciones, usando simplemente una línea de código, en ves de decenas de líneas como típicamente es requerida.

No solo LINQ hará la vida más fácil a los desarrolladores, dejando de escribir decenas de líneas de código, sino que también dejara de hacer eso y mucho más por que la codificación en contra la base de datos será un tanto más fácil.

Una vez que los desarrolladores realicen lo que pueden lograr fácilmente usando array regulares de datos, estarán asombrados de todas las posibilidades. En esencia, LINQ posibilita a los desarrolladores a construir mejores aplicaciones más rápidamente.

¿Cómo pueden los desarrolladores comenzar a usar LINQ hoy? ¿Está diseñado para trabajar con Visual Studio .NET 2005?

Hejlsberg: El proyecto LINQ se enfoca en la futura versión de Visual Studio .NET 2005, sin embargo hacemos una exhibición preliminar del proyecto LINQ disponible en el PDC 05, a fin de que los desarrolladores inmediatamente puedan empezar a tomar ventaja de las capacidades que provee LINQ. La exhibición preliminar contiene versiones preliminares de los diversos componentes del proyecto LINQ, incluyendo un compilador C#, con extensiones para soportar LINQ, así como también enlaces para poder descargar un compilador Visual Basic con soporte a LINQ. Ambos compiladores prelanzador trabajan con el Beta 2 de Visual Studio .NET 2005 y trabajarán con Visual Studio .NET 2005 cuando sea liberado en noviembre.

Jimmy Granados.

MCP ID: 3212643

Microsoft Certified Application Developer C#

Microsoft Certified Professional C#

Website: <http://www.neuronaltraining.net>