Japan Delphi Update

www.codegear.com

Agenda

- General Editor and IDE additions
- Generics
- Moving to RAD Studio 2007

Agenda

- **General Editor and IDE additions**
- Generics
- Moving to RAD Studio 2007

Delphi 2007 for Win32

Fast, Native code applications
- Compatible with Win 2000, XP and **Vista**
- First commercial IDE with native Vista support

Embrace Web 2.0
- New support for AJAX

New DBX4 database architecture
- Streamlines enterprise database connectivity

Simplify complex projects
- MSBuild powerful build/make support

Hundreds of improvements
- Next level of usability and quality

General IDE Demo

- Editor Intelligence
- Search
- Refactorings
- etc

Agenda

- General Editor and IDE additions
- **Generics**
- Moving to RAD Studio 2007

RAD Studio 2007 *(Announcing Soon)*

Support Win32 native and .NET development

Web and desktop client development with .Net

- .Net Framework 2.0 and 3.0 support
  - Including ASP.NET, ADO.NET, etc
- Generics in Delphi for .Net.
- Continued enhancements for supporting the Vista API
- Updated DBX4 architecture with support for ADO.NET 2.0.
- ECO IV – The most productive way to build .NET applications with MDD
- Blackfish SQL – Java and .NET embeddable database

Generics, a definition

"Generics are classes, structures, interfaces, and methods that have placeholders (type parameters) for one or more of the types they store or use."

Consuming Generic Classes

System.Collections.Generic Namespace

- Mscorelib.dll
  - List<T>, IList<T>, ICollection<T>, IEnumerator<T>
  - Dictionary<TKey, TValue>, IDictionary<TKey, TValue>
  - Comparer<T>, IComparer<T> (and more)
- System.dll
  - LinkedList<T>,Queue<T>, Stack<T>
  - SortedList<TKey, TValue>, SortedDictionary<TKey, TValue> (and more)

Demo

Syntactic Sugar?

Typecasting, Boxing and Un-boxing are expensive operations.

- When enumerating lists, the most expensive operation is usually in the body of a loop, exacerbating the problem.
- Generics generates typed code, giving a significant performance gain in the previous common example.

Compile time checking of types, helps to write solid code.

Parameterized Types

A class type where a data type is specified at declaration and instantiation.

```
TTag<T> = class(TObject)
public
 TagValue : T;
end;
```


Parameterized Types

Class

Interface

Record Types

Parameterized Methods

Methods can declare with type parameters.

Parameter and result types can use a type parameter. They are similar to overloaded methods. 2 mechanisms

- Explicitly specifying type arguments
- By automatically inferencing from argument type(s).

Demo

Constraints

- Constraints may be associated with a type parameter of a parameterized type. Constraints declare items that must be supported by any concrete type passed to that parameter in a construction of the generic type.
- Constraints apply to all forms of parameterized types. (Classes, Records and Interfaces) and parameterized methods.
- Constraints are declared in a fashion that resembles type declarations in regular parameter lists:

```
type TFoo<T: ISerializable> = class(TObject)
public
  FField: T;
end;
```

Constraints

Multiple type parameters

- When constraints are specified, multiple type parameters must be separated by a semicolon, as with parameter list declarations:

```
type
TFoo<T: ISerializable; V: IComparable>
```

- Also like parameter declarations, multiple type params can be grouped together in a comma list to bind to the same constraints:

```
type
TFoo<S, U: ISerializable>
```


Constraints

- Interface constraints
- Class type constraints
- Constructor constraint
- Class constraint
- Record constraint

Agenda

- General Editor and IDE additions
- Generics
- **Moving to RAD Studio 2007**

Moving to RAD Studio 2007

- New Project File format
  - Supports MSBuild
  - When you open a .dpr file, it will auto-generate the equivalent .dproj file
- Third Party Components
  - RAD Studio 2007/Delphi 2007/C++Builder 2007 is binary compatible with 2006
  - For earlier versions, you will need either:
    - The source for your components
    - Updated binaries from your component vendor

Moving to RAD Studio 2007

- Database
  - IBExpress and ADOExpress is still part of the product and still supported
  - BDE is still part of the product and still supported for Paradox, Dbase, Access, FoxPro
    - SQLinks (ie. BDE drivers for Oracle, SQL Server, etc) are no longer in the product
  - DBExpress has become DBX and is the future focus of our development efforts

Moving to RAD Studio 2007

- Database cont'd
  - BDE to DBX
    - We have a BDE driver for DBX available
    - Allows DBX access to Paradox, etc
    - Written by one of our consultants, and available
    - Allows a "Form by Form" approach to migration.
    - Might be a good future topic for Japanese Webinar
  - Blackfish SQL
    - Full featured SQL database (triggers, stored procedures, etc)
    - However, can be a good replacement for Paradox, etc even in small applications

Questions?