Getting Started with VelocityGraph (Saturday, January 16, 2021)

This guide compliments the sample programs, the User's Guide and the API reference provided on our site. Developers should review this in order to better understand how to quickly build a VelocityGraph application.

We made a getting started video as well. To see it click here.

Getting assemblies

The simplest way to get started is by using the nuget package.

You can download the entire product package setup from the VelocityGraph website. After you download it, run it to install it on your PC.

Register as User, Create & Download license database (4.odb)

- Register as user
- Create Trial License
- Download License

Add required using Statements

As a minimum you will need the following using statements:

```
using VelocityDb.Session;
using VelocityGraph;
```

Choose a database directory and start an update transaction

```
class QuickStartVelocityGraph
{
  static readonly string systemDir = "QuickStartVelocityGraph"; // appended to SessionBase.BaseDatabasePath

  static void CreateGraph()
  {
    using (SessionNoServer session = new SessionNoServer(systemDir))
    {
      if (Directory.Exists(session.SystemDirectory))
        Directory.Delete(session.SystemDirectory, true); // remove systemDir from prior runs and all its databases.
      // Start an update transaction
      session.BeginUpdate();
```

Create a Graph

```
Graph g = new Graph(session);
```

Persist the graph – give it a persistent id

```
session.Persist(g);
```

Define some Vertex, Edge and Property types

```
VertexType movieType = g.NewVertexType("Movie");
PropertyType movieTitleType = g.NewVertexProperty(movieType, "title", DataType.String, PropertyKind.Indexed);
PropertyType movieYearType = g.NewVertexProperty(movieType, "year", DataType.Integer, PropertyKind.Indexed);

VertexType actorType = g.NewVertexType("Actor");
PropertyType actorNameType = g.NewVertexProperty(actorType, "name", DataType.String, PropertyKind.Indexed);

EdgeType castType = g.NewEdgeType("ACTS_IN", false);
PropertyType castCharacterType = g.NewEdgeProperty(castType, "role", DataType.String, PropertyKind.Indexed);
```

Add some vertices (nodes)

```
// Add some Movies
Vertex matrix1 = movieType.NewVertex();
matrix1.SetProperty(movieTitleType, "The Matrix");
matrix1.SetProperty(movieYearType, (int)1999);

Vertex matrix2 = movieType.NewVertex();
matrix2.SetProperty(movieTitleType, "The Matrix Reloaded");
matrix2.SetProperty(movieYearType, (int)2003);

Vertex matrix3 = movieType.NewVertex();
matrix3.SetProperty(movieTitleType, "The Matrix  Revolutions");
matrix3.SetProperty(movieYearType, (int)2003);

// Add some Actors
Vertex keanu = actorType.NewVertex();
keanu.SetProperty(actorNameType, "Keanu Reeves");

Vertex laurence = actorType.NewVertex();
laurence.SetProperty(actorNameType, "Laurence Fishburne");

Vertex carrieanne = actorType.NewVertex();
carrieanne.SetProperty(actorNameType, "Carrie-Anne Moss");
```

Add some edges (relations)

```
Edge keanuAsNeo = castType.NewEdge(keanu, matrix1);
keanuAsNeo.SetProperty(castCharacterType, "Neo");
keanuAsNeo = castType.NewEdge(keanu, matrix2);
keanuAsNeo.SetProperty(castCharacterType, "Neo");
keanuAsNeo = castType.NewEdge(keanu, matrix3);
keanuAsNeo.SetProperty(castCharacterType, "Neo");

Edge laurenceAsMorpheus = castType.NewEdge(laurence, matrix1);
laurenceAsMorpheus.SetProperty(castCharacterType, "Morpheus");
laurenceAsMorpheus = castType.NewEdge(laurence, matrix2);
laurenceAsMorpheus.SetProperty(castCharacterType, "Morpheus");
laurenceAsMorpheus = castType.NewEdge(laurence, matrix3);
laurenceAsMorpheus.SetProperty(castCharacterType, "Morpheus");

Edge carrieanneAsTrinity = castType.NewEdge(carrieanne, matrix1);
carrieanneAsTrinity.SetProperty(castCharacterType, "Trinity");
carrieanneAsTrinity = castType.NewEdge(carrieanne, matrix2);
carrieanneAsTrinity.SetProperty(castCharacterType, "Trinity");
carrieanneAsTrinity = castType.NewEdge(carrieanne, matrix3);
carrieanneAsTrinity.SetProperty(castCharacterType, "Trinity");
```

Commit the transaction

```
session.Commit();
```

Query the graph

```
static void QueryGraph()
{
  using (SessionNoServer session = new SessionNoServer(systemDir))
  {
    // Start a read only transaction
    session.BeginRead();
    Graph g = Graph.Open(session);
    // Cache SCHEMA
    VertexType movieType = g.FindVertexType("Movie");
    PropertyType movieTitleType = movieType.FindProperty("title");
    VertexType actorType = g.FindVertexType("Actor");
    PropertyType actorNameType = actorType.FindProperty("name");

    // How many vertices do we have?
    Console.WriteLine("Number of Vertices: " + g.CountVertices());

    // Find a movie by name
    Vertex movie = movieTitleType.GetPropertyVertex("The Matrix");

    // Get all actors
    var actors = actorType.GetVertices();

    // Count the actors
    int actorCount = actors.Count();
    Console.WriteLine("Number of Actors: " + actorCount);

    // Get only the actors whose names end with “s”
    foreach (Vertex vertex in actors)
    {
```

```
      string actorName = (string) actorNameType.GetPropertyValue(vertex.VertexId);
      if (actorName.EndsWith("s"))
        Console.WriteLine("Found actor with name ending with \"s\" " + actorName);
    }

    // All vertices and their edges
    var edges = g.GetEdges();
    int edgeCount = edges.Count();

    session.Commit();
  }
}
```

For more examples, download our setup installer and take a look at our many sample projects here or in your installed VelocityDB (see %USERPROFILE%\My Documents\VelocityDB\VelocityDB.sln)