# dotNetRDF Design Document

## dotNetRDF Version 0.4.2

Library: dotNetRDF.dll  
Version: 0.4.2  
Target Date: June 2011  
Author: Rob Vesse  
Proposed Implementer: Rob Vesse  
Last Updated: 04/03/2011

## Required Features

* Thread safe dataset management for Leviathan (if not done in 0.4.1)
* Improved Persistent Graphs (if not done in 0.4.1)
* SPARQL Update commands to become atomic
* SPARQL Support to match 1.1 Last Call specifications which should be released by time development on this branch starts
* Migrate as much of the core Leviathan engine as feasible directly into LeviathanQueryProcessor and LeviathanUpdateProcessor respectively

## Time Permitting Features

* None at present

## Known Issues/Bugs to Fix

* None at present

# Design

## Thread Safe Dataset Management for Leviathan

See dotNetRDF 0.4.1 Design for details

## Improved Persistent Graphs

See dotNetRDF 0.4.1 Design for details

## SPARQL Update commands to become atomic

A SparqlUpdateCommandSet should be applied atomically whereas currently each command is applied in turn and the changes flushed to the dataset. ISparqlDataset will need an additional Discard method (could call it Rollback and rename Flush to Commit)

Will also need to change the update processor implementations so that Flush() is only called at the end of a command set and not after every command. Some modifications will also be needed to ensure that in-memory changes can be discard. This should be achievable using the system described for Improved Persistent Graphs.

## SPARQL Support to 1.1 Last Call Specification

Unable to elaborate on any needed changes at present as Last Call specifications have yet to be published

## Migrate core engine into LeviathanQueryProcessor and LeviathanUpdateProcessor

Move the implementations from the isolated Evaluate() methods on ISparqlAlgebra and SparqlUpdateCommand into the relevant Leviathan processors. This will make the code more manageable and make it possible to make many methods virtual. This will allow people to override specific parts of the engine themselves without having to replicate large chunks of the engine.