

Tech Notes

Backgrounders

[Persisting objects in tables](#)

A general primer on O/R-mapping, independent from re-motion.

[ASP.NET for people who did not bother until now](#)

In order to understand [re-call](#), you should know about the difference between `Server.Transfer` and `Response.Redirect`. This article clarifies the topic.

[What uigen.exe does](#)

`uigen.exe` is a program generator. It uses templates to derive a basic web application from the domain model. This article explains how this works in detail. You can write your own templates for new types of web applications.

[re-strict dictionary and overview](#)

`re-strict` is re-motion's security module. Not much documentation exists at this time, but the article gives you an impression of how `re-strict` works.

[On re-bind's data source controls](#)

re-motion brings its own set of smart web controls. These controls bind to custom *data source controls*. This article explains the capabilities of various types of re-motion's (re-bind's) data source controls.

[Objects in perma-URLs](#)

Per default, re-motion works with `Server.Transfer`, what precludes unique URLs for each domain object instance. This article explains how to program for "perma-URLs", i.e. unique URLs for each domain object instance.

[Hints on how to implement pessimistic locking for re-store](#)

re-store locks domain objects *optimistically*, i.e. the users who *writes* back modifications first have their changes committed to the database. This is in contrast to MS Word: in MS Word, the users who *open* the document first have permission to write back modifications (= pessimistic locking). This article explains how to implement pessimistic locking for (all or just some) classes of domain objects.

[Working with XML queries and the query manager](#)

re-motion provides excellent support for Linq. (In fact, `rubicon` has [advanced the field with re-linq.](#)) However, sometimes you might want to resort to SQL for performance- or other reasons. This article explains how to integrate SQL queries into your domain model.

[Junction objects](#)

re-store supports 1:1 relations and 1:n relations. This article explains how to implement m:n relations using junction objects.

[Globalization remarks](#)

Everything you ever wanted to know about globalization in re-motion – and more!

- [Code generation in re-motion \(dbschema.exe, uigen.exe, wxegen.exe\)](#)
- [Pick location, the page function](#)