



Store Beyond Glorp What's Next



World Headquarters
Cincinnati, Ohio

SIMPLIFICATION THROUGH INNOVATION®

Welcome
August 21, 2011

What is Store ?

- Store is a Version Control System for Cincom Smalltalk
- Uses a Relational Database as the repository
- Versions
 - Bundles and Packages (components)
 - Namespaces
 - Class Definitions
 - Class Extensions (methods only)
 - Shared Variables
 - Methods

Store Capabilities

- Provides Tools for:
 - Publishing Bundles/Packages
 - Merging Bundle/Package versions into the image
 - Loading all version-able entities
 - Listing, Browsing, Comparing all version-able entities
 - Administration
- Supports object queries using Glorp

What is Glorp

- Glorp is an Object Relational Mapping Framework
- Cross Dialect
- Multiple Database Support
- Uses Database Specific Optimizations
 - Arrayed Changes
 - Smart Unions

What Store used to use

- Hand coded (generic) SQL
- Minimal optimized execution
- “Flat” objects
 - Bundle
 - Bundles
 - Packages
 - Methods (package to method)
 - Method
 - PkgClasses (package to class)
 - ClassRecord
- Views into relationships

Glorp Objects

- All in NameSpace : Store.Glorp
- Most subclassed from StoreObject
 - Exceptions:
 - StoreClassExtension
 - StoreMetaclassExtension
 - StoreLoadRecord (obsolete)
 - StoreUserGroup
 - StoreVersionlessPundle
 - StoreAccessPrivilege

Working With Store Objects - Session

- You need a session
 - `session := Store.Glorp.StoreLoginFactory currentStoreSession`
- What is in a session:
 - Accessor (a DatabaseAccessor subclass)
 - System (a StoreDescriptorSystem subclass)
 - Platform (Glorp.DatabasePlatform)
 - session system platform
 - session platform

Working With Store Objects : Selecting

- session readOneOf:
- session read: (readManyOf:)
- session count: X
- Where statements (#read:where:, #count:where:, etc.)
 - [:each | each <someStatmentAnsweringBoolean>]
 - [:each | each xxx AND: (each yyy)]
 - [:each | each xxx OR: (each yyy)]
- Store object 'fields'
 - StoreDescriptorSystem >> classModelFor<XXX>:

Working With Store Objects : Queries

- query := Query where: [:each |]
- query alsoFetch:
 - [:each | each definition comment]
- query orderBy:
 - [:each | each timestamp descending]
- query retrieve:
 - [:each | each count]
 - [:each | each name distinct]

Queries – All versions of a method

```
session := StoreLoginFactory currentStoreSession.  
query := Query  
  read: Store.Glorp.StoreMethodInPackage  
  where: [:each | each definition name = 'fileNew' &  
    (each definition className =  
      'Root.Smalltalk.Tools.Workspace') &  
    (each package name = 'Tools-Workspace')].  
query alsoFetch: #definition.  
query orderBy: [:each | each definition timestamp descending].  
^session execute: query
```

Queries – Latest trunk version of bundle

```
session := StoreLoginFactory currentStoreSession.
```

```
query := Query readOneOf: StoreBundle
```

```
  where: [:each | each name = 'Glorp' &
```

```
    (each currentBlessingLevel >= 15) &
```

```
    (each version similarTo: '7\.[7-9](\.[1-9])* - [0-9]+') ].
```

```
query orderBy: [:each | (each version copyFrom: 1 to: 6) descending ].
```

```
query orderBy: [:each | each timestamp descending ].
```

```
^session execute: query
```

Query gets highest 7.7, 7.7.1, 7.8 or 7.9 version

Administrative Tasks

- Rename a Version
 - packages := session
read: StorePackage
where: [:each | each version = 'Ugly'].
session inUnitOfWorkDo:
[:thisSession |
thisSession register: packages first.
packages first version: 'BetterName']
- Add a SQL Function
 - FunctionExpression class>>createBasicFunctionsFor:
 - DatabasePlatform subclass >> #initializeFunctions
 - at: #length put: (PrefixFunction named: 'LENGTH')
 - session
read: StorePackage
where: [:each | each version length > 50]

Warning, Will Robinson (Traps to avoid)

- Sessions
 - Too Many Hold On To Connection Resources (Cursors)
 - Too Few Grow Huge Caches
- Most StoreObject APIs have two versions:
 - allVersionsWithName: aString
 - allVersionsWithName: aString in: aSessionOrNil
- StorePackage & StoreBundle have session instance variable!
- Don't get full objects if you just want a single attribute

Store Opportunities provided by Glorp

- Schema changes become easier
 - Currently, previous version key is in version record, making tracing history inefficient
 - Text fields that could be large, like method source are stored in Blobs with indirect reference and chaining
- Querying the repository in complex ways becomes easier
 - Possible to run SmallLint on methods in repository, not just when loaded (I think)
- Building Store add-ons becomes easier – composed DescriptorSystem?

Glorp Opportunities provided by Store

- Cincom now has a Glorp application
 - Helps us to understand viewpoing of Glorp users better
 - Likely to drive Glorp feature development a bit faster
 - Creating DBA functionality across platforms/DBs bit win for us
- Users now have a (non-toy) Glorp application to look at in Cincom Smalltalk
 - Want to know how to do something? Browse senders of....

Store Migration Issues - Speed

- Speed has suffered
 - Query speed
 - Old implementation was hand-coded, brittle, relatively fast SQL
 - New implementation uses generated SQL
 - Schema suboptimal
- Likely that we can make it faster than old Store
 - Eliminate unnecessary queries
 - Eliminate redundant processing

Eliminating Redundant Queries

- Redundant queries in session creation
- Queries which don't get quite enough data
 - Proxies need to be resolved
 - 1 query per value – lots of extra round trips
- Data retrieved, used, thrown away, retrieved again

Eliminating Redundant Processing

- Example: ChangeSet
 - VisualWorks has 3 types of ChangeSet
 - Bundle/Package change sets (1 per bundle, per repository)
 - Global change set
 - Named change sets (1 current)
 - Loading a version updates all 3 kinds
 - Unloading package searches and removes changes from all 3
- Reducing the number of change sets being updated **should** significantly improve unloading, loading likely improved to a lesser degree

Enhancements for 7.9

- All old Store objects and tools not loaded by default and moved to:
 - OldStoreDatabase package
 - OldStore Tools package
- Progress in ongoing speed improvement effort
- Store problems that raise Dialogs have been modified to throw exceptions, allowing you to catch them
 - Makes automated testing more robust
 - Supports scripting for build, etc

Enhancements for 7.9, continued

- MergeTool
 - Apply will use RB refactorings to allow merge to be undone if desired
 - Will be able to do automatic merge, taking you directly to the Publish dialog, if no manual merge decisions required
 - Eventually, we would like to make it easy to script Load, Merge and Publish

Enhancements for 7.9, continued

- Gather and Distribute
 - Gather – Allows you to compare multiple bundles or packages and publish all changes as a single package
 - Distribute – After loading a package created by Gather, allows you distribute the contents back where they belong (subject to some limitations)
- Immediate application in product support. There may be others

Questions ? Comments ?

Ask now

--- or ---

Tom Robinson
trobinson1@cincom.com



**© 2011 Cincom Systems, Inc.
All Rights Reserved
Developed in the U.S.A.**

CINCOM, the Quadrant Logo, and Simplification Through Innovation are registered trademarks of Cincom Systems, Inc.

All other trademarks belong to their respective companies.