

CS 580 Client-Server Programming
Fall Semester, 2002
Doc 4 CVS
Contents

Store.....	2
Running Your Own Local Repository	3
Using Store	4
Loading Store into an Image	4
Packages, Parcels & Categories	5
Using Packages.....	9
Browser Level Source Control.....	12
Connecting to the CS580 Store Repository.....	14

References

Source Code Management Guide,
doc/SourceCodeMgmtGuide.pdf in VW 7 installation

Copyright ©, All rights reserved. 2000 SDSU & Roger Whitney, 5500 Campanile Drive,
San Diego, CA 92182-7700 USA. OpenContent (<http://www.opencontent.org/opl.shtml>)
license defines the copyright on this document.

Store

Store is VW source control management system

Store uses a database to store the source code

Store supports

- Oracle
- SQLServer
- ODBC
- PostgreSQL
- DB2
- MS Access

These notes use PostgreSQL a free open source database

The class repository use PostgreSQL

Running Your Own Local Repository

PostgreSQL Repository

Download & documentation

<http://www.postgresql.org/>

For instructions for install on Windows 2000

<http://people.freebsd.org/~kevlo/postgres/portNT.html>

The Store PostgreSQL drivers are part of VW7

MS Access Repository

See instructions at:

<http://wiki.cs.uiuc.edu/VisualWorks/Store+For+Microsoft+Access>

On Sept 14, 2002 StoreForAccess was updated to correct a bug

The more recent version of StoreForAccess can be obtained on VW
Public repository

To gain access to the repository see:

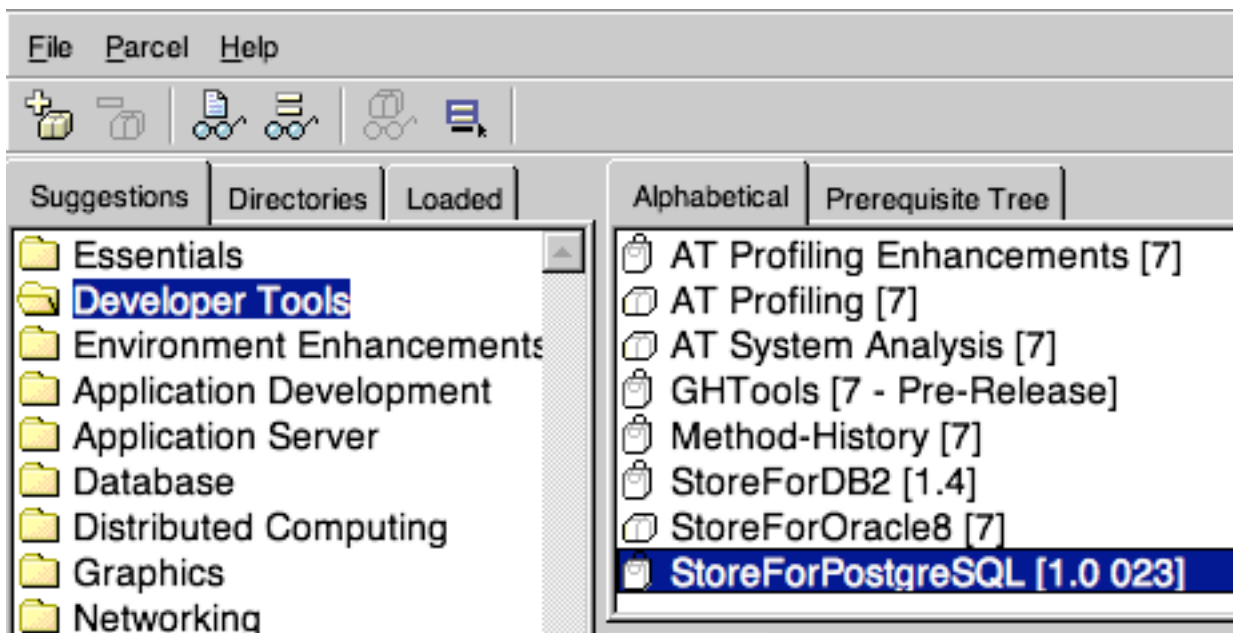
<http://www.cincomsmalltalk.com:8080/CincomSmalltalkWiki/PostgreSQL+Access+Page>

Using Store Loading Store into an Image

Load the StoreForPostgreSQL parcel

Use the Parcel manager

- In the launcher in the System menu select “Parcel Manager”
- In the Parcel Manager Window select Developer Tools
- Scroll the upper right window to find StoreForPostgreSQL
- Select StoreForPostgreSQL



- Click on the load Parcel icon.

The “Load Parcel” icon is the left more icon on the icon bar.

Packages, Parcels & Categories

Open a System browser

In Launcher in Browse menu select “System Browser”

Or select the “System Browsers” icon – the third icon from the left in the Launcher

System browser now supports three ways to organized code:

- Categories
- Package
- Parcels

In the System browser use the “Browser” menu to switch between viewing the code via these methods

Category

A collection of classes

Use to view any class in the system

Package

A package can contain:

- A collection of classes
- Methods added to classes elsewhere in system
- Prerequisite requirements
- Actions to perform when loaded

Development time grouping of related code

Packages can be stored in a repository

Can be grouped into bundles for large projects

Parcel

File based version of a Package

Runtime grouping of related code

Contains binary for fast loading

Can be created without Smalltalk source if you wish to hide source from users

Using Packages

Creating a Package

Open a System browser and set it to view packages

In the System Browser “Browser” menu select “Package” item

In the System Browser “Package” menu select “New” and the submenu item “Package”

You will be prompted for a package name

For this document I will use the name “StoreLectureExamples”

With your new package selected in the System Browser “Package” menu select “Make Current”

This will cause any change you make anywhere in the system to be part of your package when you are dealing with Categories

Now you can set the browser to show categories until you are ready to save your code to the repository

Adding Code to a Package

Assumptions

- You can create a package
- You have made the package the current package
- You are viewing code in Category mode
- You have a System browser open

We will

- Make a change to an existing class
- Add our own class

Changing an Existing Class

Go to the Integer class

Type "Integer" in the find box and hit return

Add the asGrade method to the converting category

asGrade

"Answer the letter grade represented by self"

```
self > 90 ifTrue:['^A'].  
self > 80 ifTrue:['^B'].  
self > 70 ifTrue:['^C'].  
self > 60 ifTrue:['^D'].  
^F'
```

In the bottom of the browser you should see that method is in your package. See below.

The screenshot shows the Smalltalk IDE interface. At the top is a menu bar with options: Browser, Edit, Find, View, Category, Class, Protocol, Method, Tools, Help. Below the menu is a toolbar with various icons. A search box labeled "Find:" contains the text "Integer". The main browser area is divided into four panes: "Category", "Hierarchy", "Instance", and "Shared Variable". The "Category" pane shows a list of categories: Magnitude-Gen, Magnitude-Num, and Net-Namespace. The "Hierarchy" pane shows a list of classes: Float, Fraction, and Integer. The "Instance" pane shows a list of categories: coercing, comparing, and converting. The "Shared Variable" pane shows a list of shared variables: asFloat, asGrade, and asInteger. Below the browser panes are tabs for "Source", "Rewrite", and "Code Critic". The "Source" tab is selected, showing the source code for the asGrade method. The code is:

```
asGrade  
    "Answer the letter grade represented by self"  
    |  
    self > 90 ifTrue:['^A'].
```

 At the bottom of the IDE, there is a status bar with three fields: "Method: #asGrade (converting)", "Parcel: none", and "Package: StoreLectureExample".

Browser Level Source Control

The System browser provides some source control without dealing with Store

To see this functionality

- Modify the asGrade method changing 90 to 95
- Save the method
- Change 80 to 85 and save the method
- Change 70 to 75 and save the method

Now click on the “Undo” icon to undo the last change

You can rollback all the changes to asGrade with the undo icon

Use the redo icon to redo each change



To see the history of all changes to asGrade

While asGrade is selected, in the browser “Method” menu select “Browse Versions”

Creating a New Class

Create a new category for your new class

In the Browser “Category” menu select the “New” menu

Then create a new class

For the notes & class demo I will use the following class.

```
Smalltalk defineClass: #BankAccount
  superclass: #{Core.Object}
  indexedType: #none
  private: false
  instanceVariableNames: 'balance '
  classInstanceVariableNames: ''
  imports: ''
  category: 'Store-LectureNotes'
```

```
BankAccount class methodsFor: 'instance creation'
```

```
new
  ^super new initialize
```

```
BankAccount methodsFor: 'accessing'
```

```
balance
  ^balance
```

```
deposit: aNumber
  balance := balance + aNumber
```

```
withdrawal: aNumber
  balance := balance - aNumber
```

```
initialize
  balance :=0
```

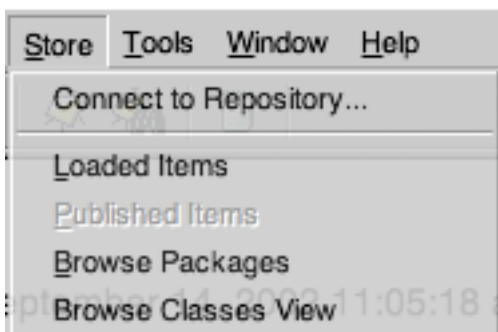
Now to save our package in the repository

Connecting to the CS580 Store Repository

Before saving a package we must connect to the repository

To connect to the CS580 repository on rugby.sdsu.edu your machine needs a connection to the Internet

In the Launcher Store menu select the “Connect to Repository” item



First Time Connection

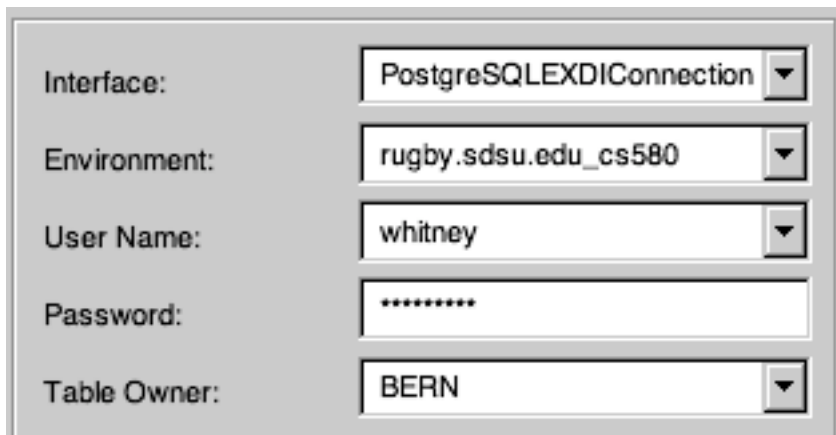
In the “Connect to Database” window:

For Interface select PostgreSQLEXDIDConnection

For Environment type rugby.sdsu.edu_cs580

Contact me for a username and password

Keep BERN as owner of the table.



The screenshot shows a dialog box with the following fields:

Interface:	PostgreSQLEXDIDConnection
Environment:	rugby.sdsu.edu_cs580
User Name:	whitney
Password:	*****
Table Owner:	BERN

If you click on the “Save...” button you can give the settings a name to make it easier to connect later.

Now click on the “OK” button to connect to the repository

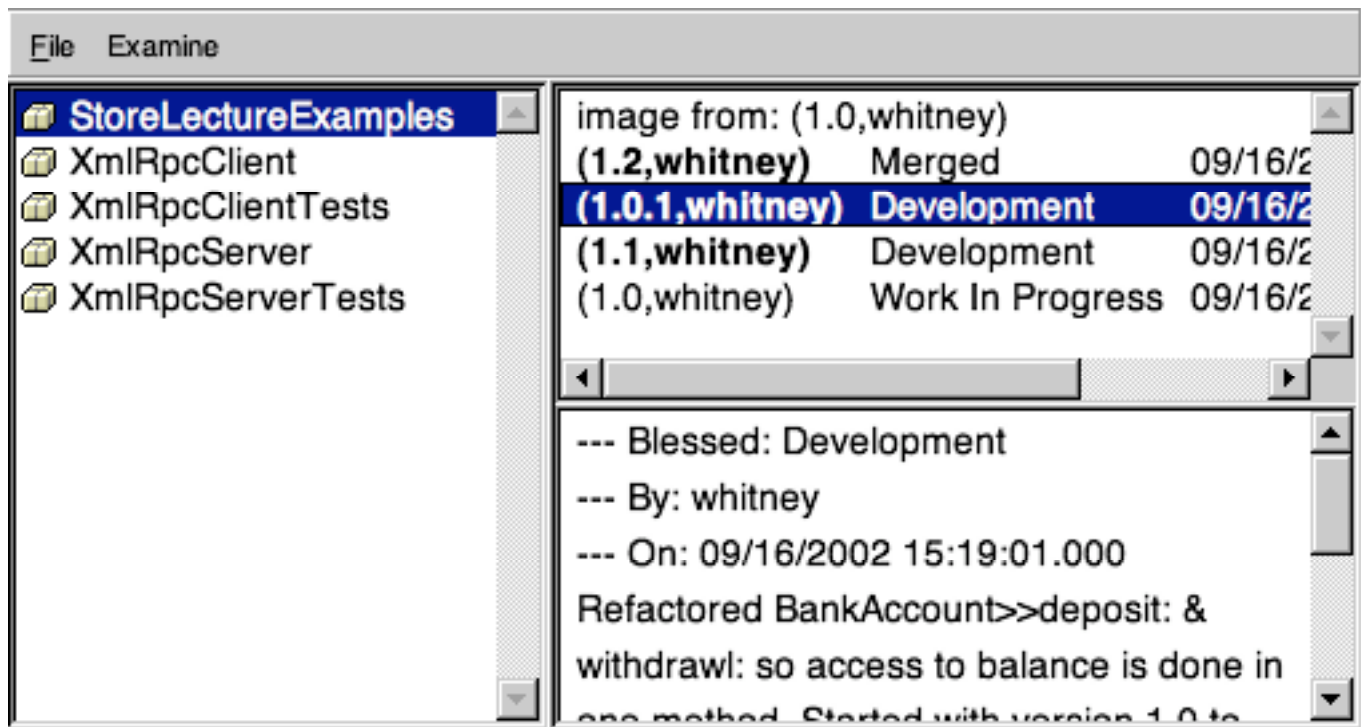
If all went well you are now connected to the repository. If you do not get any errors you will be connected.

Viewing Existing Items in the Repository

Once you are connected to a repository you can view its contents

In the Launcher's "Store" menu select "Published Items"

The resulting window will show you the contents of the repository. When you select a package you will see the different versions of the package. When you select version you see the comment for that version. When a version is selected you can either right click in the upper right window or use the File menu to load that version into your image.



Putting a Package into the Repository

To put a package into a repository

- Open a System browser if one is not currently open
- Use the Package view (Browser menu item “Package”)
- Select the package you wish to publish
- In the browser “Package” menu select the item “Publish...”
- Enter your commit comment in the “Publishing package” window
- Click on the publish button

There are a number of other useful things one can do like merge different versions of a package. However, you should know enough to read the manual to find out how to do that.