

Offline SQLiteSVG database applications with Firefox

SVG Open 2008

Nürnberg – August 26, 2008

Klaus Förster – University of Innsbruck

This talk will ...

- deal with SVG, Javascript, SQLite & Firefox
- present a basic example for client side database storage in Firefox
- examine future chances arising with the upcoming HTML5 standard
- extend the basic example to match HTML5 conformance requirements
- last for 20 minutes and start now ...

History

- Firefox 2, October 2006
- Unified Storage?
 - interface for storing / searching components
 - enable richer interaction with user data
 - eliminate multiple file formats from profiles
- mozStorage
 - SQLite database API
 - C++ / Javascript



SQLite

- *"... a software library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine"* (sqlite.org)
- D. Richard Hipp (Author)
- Source in public domain
- most widely deployed SQL database
- "Consortium" => Mozilla, Symbian, Adobe
- Version 3.5.4.2 as of Firefox 3



mozStorage

- SQLite database API for C++ / Javascript
- available to trusted callers only
 - security restrictions apply
 - file:// and chrome://
- perform tasks like
 - openDatabase
 - createStatement
 - bindParameters
 - execute / executeStep



mozStorage

- used in Firefox 3 for
 - Bookmarks => "Awesombar"
 - History, Cookies
 - Permissions, Passwords
 - Blacklists for phishing sites
- *.sqlite files in profile-directory
- browsable with SQLite Manager Add-on



Example hello.svg

hello.svg

```
netscape.security.PrivilegeManager
    .enablePrivilege('UniversalXPConnect');

db = storageService.openDatabase(nsIFile);

stmt = db.createStatement('INSERT INTO hello VALUES(?1)');
stmt.bindUTF8StringParameter(0,msg);
stmt.execute();
stmt.finalize();

stmt = db.createStatement('SELECT msg FROM hello');
while (stmt.executeStep()) {
    // colName = stmt.getColumnNames(0);
    setText(stmt.getUTF8String(0));
}
stmt.reset();
stmt.finalize();
```


hello.svg

- Example could be easily extended ...
- disadvantages
 - vendor specific
 - security restrictions
 - limited to file:// and chrome://
 - unfrozen API
 - locked to Firefox
- alternate approaches ...



WHATWG / W3C

- Upcoming HTML5 standard
- Ian Hickson (Editor)
- Structured client-side storage (5.10)
 - Storing name/value pairs (5.10.1)
 - Database storage API (5.10.2)
- Offline Web applications (5.7)
 - "ensuring applications are available even when the user is not connected to their network"



Example hello_html5.svg

hello_html5.svg

```
db = openDatabase('hello', '1.0', 'HTML5 Test', 512000);

db.transaction(
  function(tx) {
    tx.executeSql('INSERT INTO hello VALUES(?)', [msg]);
    tx.executeSql('SELECT msg FROM hello', [],
      function(tx, rs) {
        for(var i = 0; i < rs.rows.length; i++) {
          setText(rs.rows.item(i).msg);
        }
      },
      function(tx, err) {
        reportError('sql', err.message);
      }
    );
  }
);
```

HTML5 SQL

- Advantage – „standard conform“
- Firefox lacks implementation
 - mimic it with a wrapper script
 - Html5Db.js for database tasks
 - Io.js for handling of files & directories
 - file:// and chrome:// protocol only
- expect implementation to be available soon ...



HTML5 SQL

- partially implemented by WebKit ...
- examine Client side directory structure
 - subdir "databases" in profile directory
 - Index database with paths, quotas, names, etc.
 - subdir for each "domain" (protocol_host_port)
 - user databases in domain subdirs
 - version attribute and content
 - maximum 5MB per domain



Example tasmania.svg

Tasmania

- Data source Geoscience Australia
- Prepare the database with GDAL / ogr2ogr

```
$ echo "loading geometry from SHAPE file ..."
```

```
$ ogr2ogr -f SQLITE -nln territory \  
        -spat 143.7 -43.8 148.7 -39.4 \  
        -select FEAT_CODE,NAME,UFI \  
        tasmania.sqlite ./aus5fgd_r.shp
```

```
$ echo "loading data from CSV file ..."
```

```
$ ogr2ogr -update -f SQLITE -nln data_energy \  
        tasmania.sqlite ./energy.csv
```



Tasmania

- Fine tune database with sqlite executable
 - add table `__version__`
 - add column for text-anchor in table places
 - create view for power station data
- Initialize client-side database
 - convert database to "Javascript code" (Python)
 - generate HTML-file for initialization
 - open it and populate local database



Tasmania

- Generate Map
 - include Html5Db.js & IO.js
 - include WKTParserLib.js to translate geometry
 - query base layers, style them, add attributes
- Add interactivity
 - show power station data on click
 - allow setting of label-anchor on click

Conclusion

- Client side database storage
 - fascinating new development
 - challenge for the future
 - yet to explore
- (hopefully) soon to be standardized
- This case study
 - more like an experiment
 - changed significantly from first abstract

Thank you for your attention



Questions?