

# Session Tester



# Vision

## **Develop a tool to aid in session-based exploratory testing**

- Tool manages the structure of session-based testing
- Convert painful tasks into tools to help testers perform better
- Create an exploratory testing IDE equivalent: “ITE” integrated test environment
  - Requires different thinking and tools than programming

# History

- Big gaps in the SBT tool space. Bach scan tool deals with information already gathered. Text file format is painful.
- Why not develop a tool to help testers in the same way programmers enjoy using xUnit tools?
  - How do you get the information from the tester's experience into something persisted and useful?
- Tool needs to have a tiny footprint to encourage installation and use.

# History: First Design

- Developed a small, lightweight tool to enter in session information quickly, using very little screen real-estate
- Goal was to aid in information capturing, and to help provide structure to enforce the SBT format
- Decided to persist with XML, since it is so common, and less brittle than text files/scan tool

# History: Development

- Approached SBTM consultants, but didn't get commitment
- Aaron West and I did almost all of the initial design/development work
- Decided to release early, and incrementally, and get community feedback to drive design instead of following the Bach scan tool to the letter
  - *Risky to release a half-baked tool, but it paid off in finding real commitment and contribution*

# History: First Release

- We decided to just put what we had out there, and let the community respond
  - Some consultants complained it didn't meet their vision ,but...
- *Actual users* loved what they saw, and saw the potential in the tool and were excited they could help drive the design
  - Actual users were deviating from the Bach SBTM system in various ways
  - A lot of different community ideas, but a pattern emerged

# Community Response

- We got some things right:
  - Format, supporting features, XML, idea primer, timer
- We got some things wrong:
  - Tool was too small, no ability to edit and view sessions other than XML view, not supporting Bach format as well and scan tool
  - *(no Bach support for the tool as a result)*

# Community Response

## The Most Common response:

- Great potential!

BUT

- “I’m not ready to drop my text editor and replace it with Session Tester, yet.”
- *We realized that text editor is what we are competing with.*



# Community Response

- We now have users from all over the world, from some of the most influential companies in the world
- A handful of senior, very dedicated testers from around the world
- A community of practice has emerged
  - Session Tester has ended up in journal articles
  - Has spawned imitators
- *And we haven't released a v. 1.0 yet!*

# Thoughts So Far

- Given the positive and some negative reactions, I know we are on to something special
- Now have the opportunity to try different technologies and ideas than we might in our day jobs
- Have a dedicated community to provide feedback on the tool
- “Release early, release often,” change and adjust the tool as needed works!

# Changing Direction (slightly)

- We are competing with text editors - let's make something better!
- A lot of feature requests fit into the “tester's IDE” model
- Why not create something as powerful as NetBeans or BBEdit, but smaller, simpler and designed for testers?
- Let the community drive, not the consultants.
  - *Real projects trump theory. Watir demonstrated this.*

## 0.3 Release Vision

- First version of the “tester’s IDE”
  - Direction change is important, and fundamental to the direction of the tool
  - Enhance existing structure, smooth out usability, fix bugs

# Longer Term Vision

- Client version with team integration
  - Share missions, session sheets, project information
  - Ease of debrief
- Reporting for auditors, managers, etc.
  - Tools like XQuery, client/server frameworks will fit in here
  - Still need to sort out details here

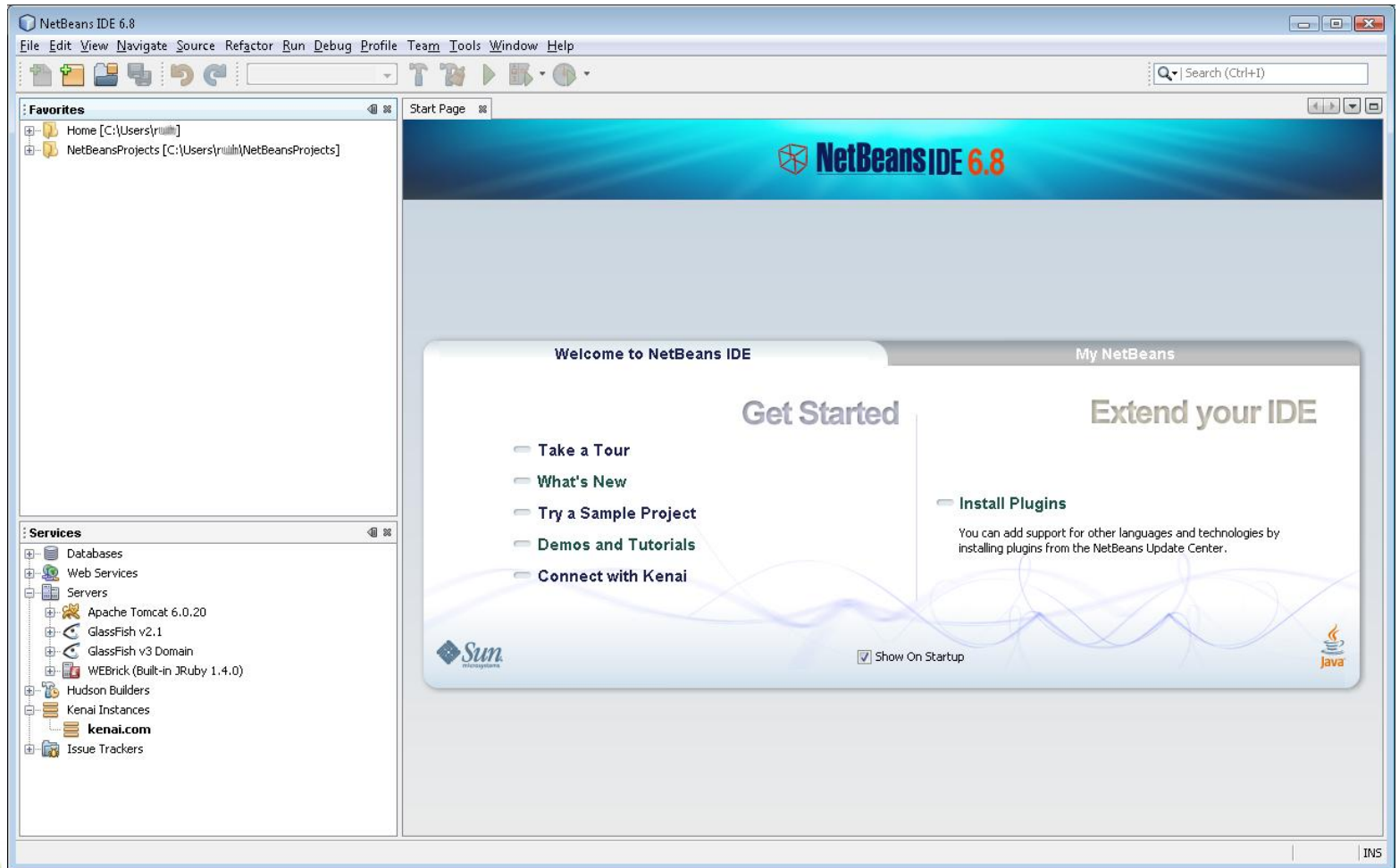
## 0.3 Release Items

- User Interface changes:
  - Start with full-screen, IDE style view – more than just session management
  - Session browser:
    - Find, view, edit sessions in usable fashion
  - Text editor plugin
    - Provide the power of a real text editor
    - Highlight tags
- Reporting
  - Clean up workflow
  - Clean up file system usage

## 0.3 Release

- This release represents a change in the direction of the tool from simple session manager, with information gathering to a simple, powerful, “IDE for exploratory testers”
- Potential to use the tool beyond session-based testing

# IDE Examples: Start Screen

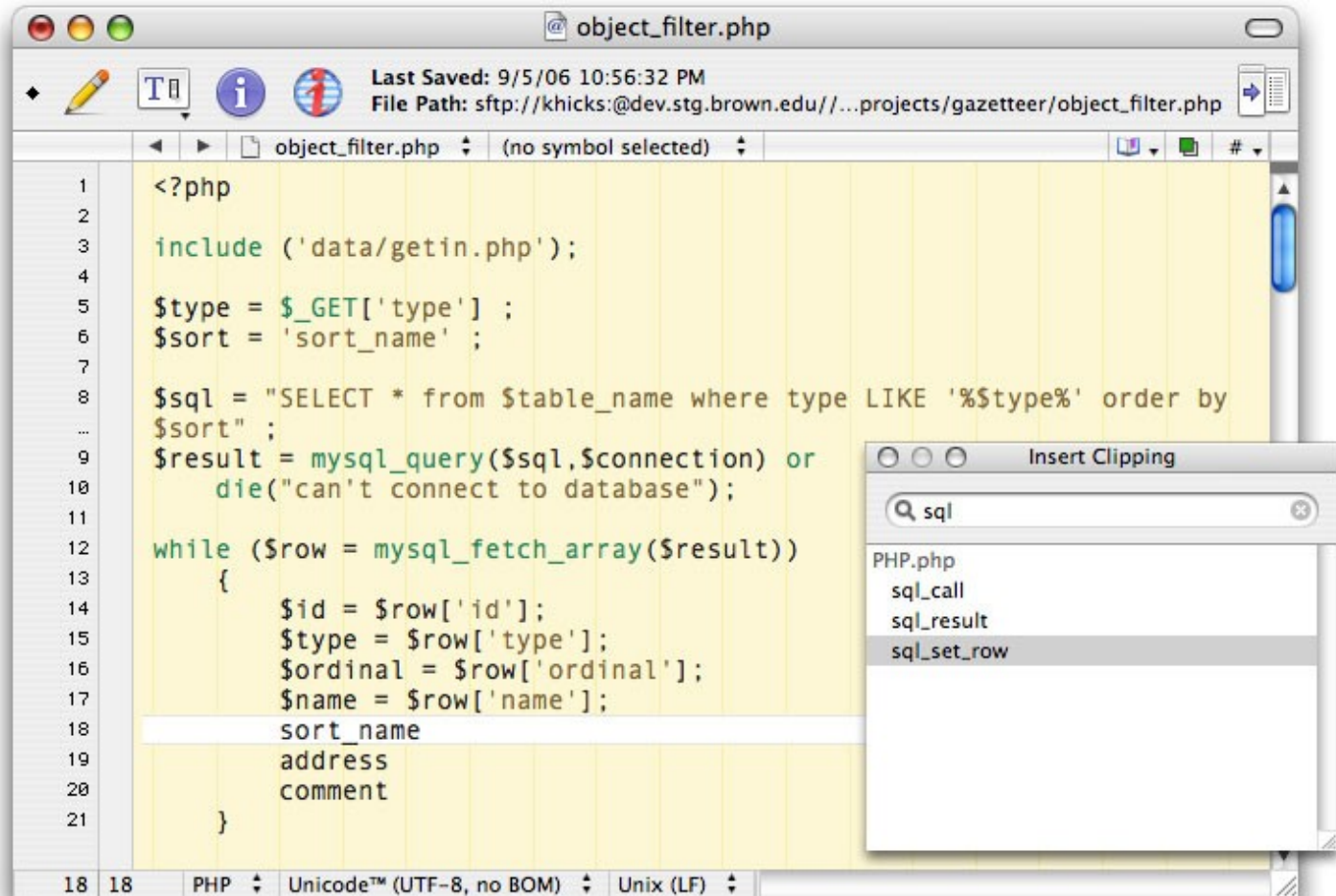




# ST Design Lesson

- An IDE starts up and gives you a lot of places to start to be productive
- A clean project/file browser on the left allows you to quickly navigate and get going
- Developers can configure and wire in their own tools and use an IDE as the central tool for their work
- An IDE is a generic productivity tool

# IDE Examples: Working with Text



The screenshot shows an IDE window titled "object\_filter.php". The status bar at the top indicates "Last Saved: 9/5/06 10:56:32 PM" and "File Path: sftp://khicks:@dev.stg.brown.edu/...projects/gazetteer/object\_filter.php". The main editor area displays the following PHP code:

```
1 <?php
2
3 include ('data/getin.php');
4
5 $type = $_GET['type'] ;
6 $sort = 'sort_name' ;
7
8 $sql = "SELECT * from $table_name where type LIKE '%$type%' order by
... $sort" ;
9 $result = mysql_query($sql,$connection) or
10     die("can't connect to database");
11
12 while ($row = mysql_fetch_array($result))
13 {
14     $id = $row['id'];
15     $type = $row['type'];
16     $ordinal = $row['ordinal'];
17     $name = $row['name'];
18     sort_name
19     address
20     comment
21 }
```

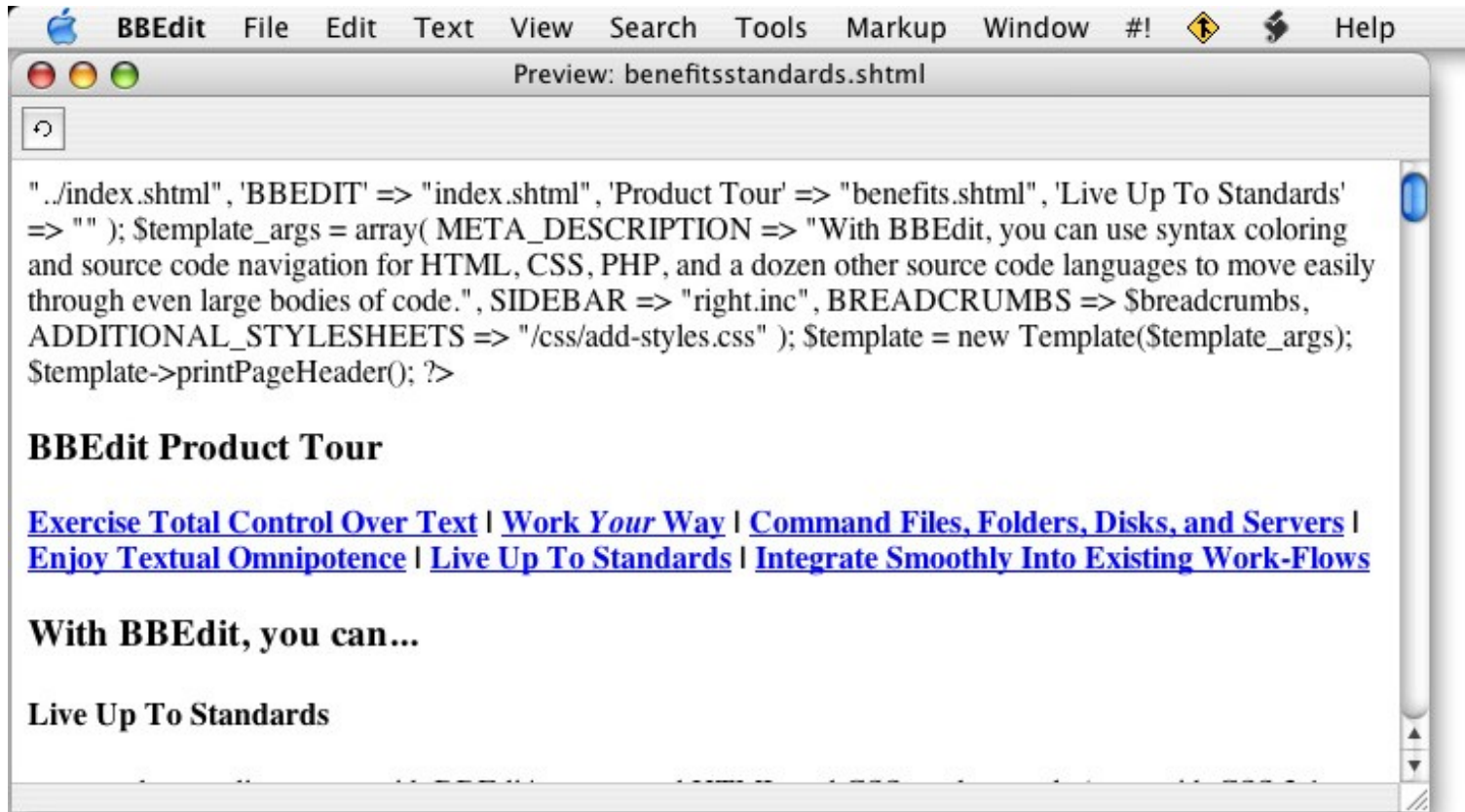
A search bar in the top right corner shows the search term "sql". Below it, a list of results is displayed:

- PHP.php
- sql\_call
- sql\_result
- sql\_set\_row

# ST Design Lesson

- Text highlighting is important
- Working with something pleasant on the eyes makes using the tool more enjoyable
- Helpful tools like intellisense and context-driven options are powerful
  - Will the tool help me as I think?

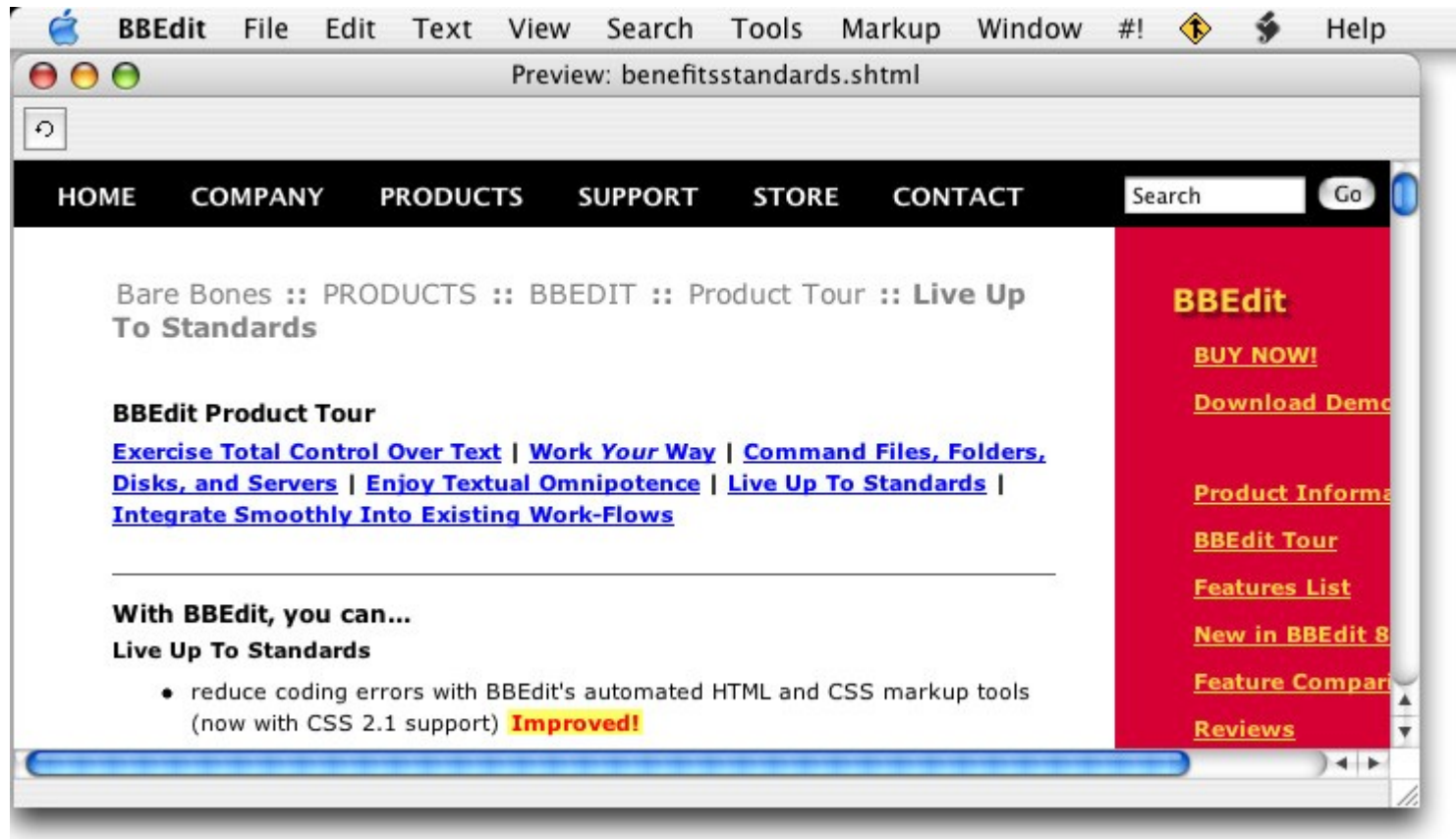
# IDE Examples: Powerful File Menu



# ST Design Lesson

- A simple, useful file menu is powerful
- I like BBEdit for its simplicity (or vi) :P
- “#!” option allows me to wire in my own scripts
  - Eg. Bash script to check out and compile
- Source control integration
- Search, **Edit**, Tools, etc. provide a lot of options
- What options do testers need?

# IDE Examples: WYSIWYG View



# ST Design Lesson

- Reviewing your work in a pleasant format is useful
- Session debriefs are a cornerstone of SBT, so being able to have a nice display of session is important
  - Edit to make sure they look ok for a reviewer/auditor
  - Ease of debrief or audits
  - WYSIWYG editing is also useful

# Getting Community Feedback

- I have a vision for the tool, and if we keep it small and powerful, people will use it
- Ultimately, the community will decide what is useful and what isn't
  - If we get releases to them early and often, it is less painful than a big release once in a while
- This is experimental development – we will get some things wrong
  - Open Source projects tolerate and encourage this



A green decorative shape on the left side of the slide, consisting of a solid green area with a wavy, irregular right edge.

# Questions?