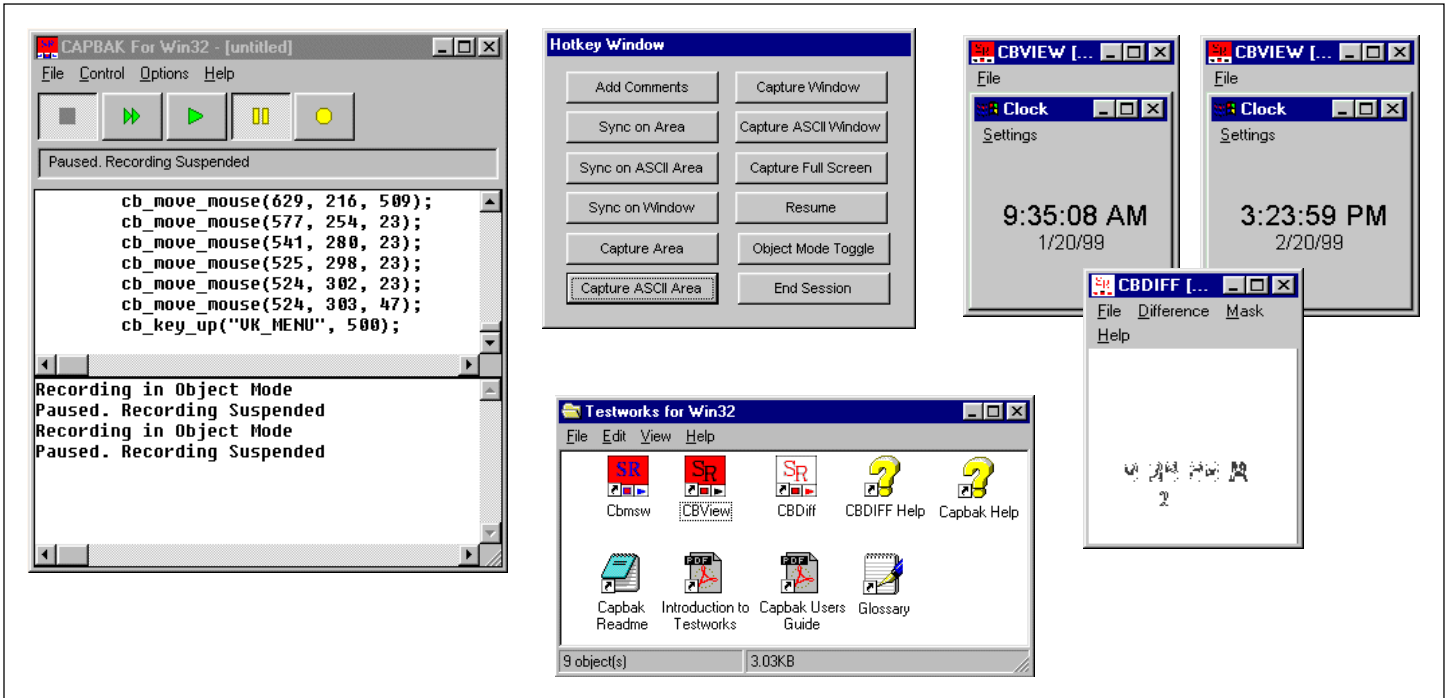




SOFTWARE RESEARCH, INC.

# CAPBAK/MSW

CAPTURE/PLAYBACK TOOL FOR MICROSOFT WINDOWS



CAPBAK/MSW's Record/Play window (at top center) captures images, such as this expected output image on the left. During playback, the screen images are recaptured, as represented by the actual output image on the right. The CBDIFF utility can then be used to compare these output images, as shown in the bottom center window.

## PRODUCT DESCRIPTION

As software applications become more sophisticated, testing efforts must keep up to ensure the software is working correctly. Software testers often only have enough time to verify that reported problems are fixed, but seldom any time to make sure no new problems were introduced in the modification process.

With CAPBAK/MSW™, you can design and develop tests that automate the testing process. It captures all user activities during the testing process including keystrokes, mouse movements, captured bitmap images, and captured ASCII characters into an interpreted C language that is easily understood.

The captured images and characters provide baselines against which future reruns of the tests are compared.

Future tests then entail the playing back of these test sessions. Each test session is

executed as if the user is running the test. CAPBAK/MSW's multiple synchronization modes ensure a reliable playback, so tests can be run unattended. Response images corresponding to baseline images are automatically captured. Comparison of the baseline and response images is done automatically and results are written to log file. The log file allows the tester quickly to identify where tests have failed.

The user has complete control over the tests. Both recording and playback sessions can be aborted, sped up, slowed down, or paused so the user can process other commands.

To verify tests have successfully played back, CAPBAK/MSW's CBVIEW utility displays test sessions captured images. Tests are further verified with the CBDIFF utility which compares baseline and response file images for differences. CBDIFF's masking capability disregards those areas of images that are not necessary

for comparison, such as time or date changes.

With the assistance of its other STW/Regression™ test management tool, SMARTS™, tests can be organized into a hierarchical structure for execution individually or as a part of a test suite and then conditionally evaluated according to the verification method selected.

The verification method can consist of comparing saved images with the CBDIFF utility for small tests or saved log file results of captured images for large tests.

Discrepancies are indicated for further analysis in SMARTS' PASS/FAIL test execution reports and statistics.

This automation can run a full suite of tests on each revision of an application, completely unattended. By running a suite of tests, the tester can detect any problems that may have been introduced on each revision of an application.



**RECORD AND PLAYBACK**

- *True-Time Capture/Playback* — Automatically records and plays back graphical user interface (GUI) application activity.
- *Character Recognition* — Tolerates minor application changes to the graphical user interface (GUI), e.g. button locations, menu list orders, fonts and background colors.
- *C-Interpreted Script* — Translates user activity including keystrokes and mouse movements into an interpreted C language that is easily understood by testers and is representative of user activity.
- *Host-Based Testing* — Tests applications on the host.
- *Client/Server Testing* — Records and plays back tests on the client to test client/server interaction and software on the server.

**IMAGE CAPTURES**

- *Verification Images* — Captures entire screens, windows, partial areas, and menus for playback verification.
- *Synchronization Images* — Saves windows and screen fragments for image synchronization.
- *Regenerated Baseline Images* — One command saves all playback response images as new baseline images.
- *Timing* — Captures do not slow the system during playback.
- *DIB Formats* — Captures images to the standard DIB format, allowing the user to utilize the saved files for other applications.
- *Viewing Images* — Displays saved images with **CBVIEW** utility to verify tests have successfully executed.

**TWO PLAYBACK MODES**

- *General Playback* — Validates test execution and generates or updates a test's expected results.
- *Quick Check* — Compares the behavior of the application-under-test to its behavior during a previous execution by automatically comparing baseline and response images and creating a log file of test results.

**INTERACTIVE PLAYBACK**

- *Playback Interaction* — Pauses, resumes, terminates, speeds up, or slows down a test session.
- *Playback Editing* — Appends new test script information to an existing script.

**MULTIPLE SYNCHRONIZATION MODES**

- *Automatic Event Synchronization* — Playback automatically handles environment event-sensitive situations, such as new windows popping up in varying locations.
- *Character Recognition Verification* — Determines a successful test based on the actual values, extracting the values from bitmap images using OCR technology and translating them into ASCII characters, regardless of font and background color.
- *Image and Window Synchronization* — The system waits for the contents of a screen fragment or a window to display, match with the baseline image and move the window, if necessary.
- *Timing Synchronization* — Playback timing adjusts to different values, allowing playback to be slowed down overall.

**DIFFERENCING UTILITY**

- *Bitmap Image Comparison* — Compares baseline and response file image for discrepancies.
- *Character Recognition Verification* — Determines a successful test based on the actual values, extracting the values from bitmap images using OCR technology and translating them into ASCII characters, regardless of font and background color.
- *Built-in Comparison* — Playback's *Quick Check* mode compares baseline and response file images on the fly during playback.
- *Image Masking* — Masks out inconsequential areas of images (such as dates, times, header, and footers), so that it reports only those differences the user deems significant.

**RECORDED TEST SCRIPT EDITING**

- *Test Script Programming* — Handles multiple test scenarios and responds to test environment variances with condi-

tional expression and loop construct editing.

- *Including Tests* — A test script incorporates other test scripts, allowing for a modular test development approach and the reuse of common test functions.
- *Appending Tests* — Appends new test script information to an existing test during playback.

**SUPPORTED ENVIRONMENTS**

- Tests any MS-Windows 3.1 application version.
- Available for any 386/486 PC or Pentium PC.
- A minimum of 4 MB RAM required.

**TECHNICAL SUPPORT**

- Telephone hot-line assistance for installation and technical questions is available.
- Maintenance contracts provide continuing product support and upgrades.

SR

For more details on CAPBAK/MSW, contact:



SOFTWARE RESEARCH, INC.

1663 MISSION STREET, SUITE 400  
SAN FRANCISCO, CA 94103 USA  
PHONE: (415) 861-2800  
TOLL FREE: (800) 942-SOFT  
FAX: (415) 861-9801  
E-MAIL: info@soft.com  
http://www.soft.com

CAPBAK/MSW, SMARTS, STW/Regression and the SR logo are trademarks of Software Research, Inc. All other systems are either trademarks or registered trademarks of their respective companies.

Software Research, Inc. reserves the right to make changes without notice, and within its own discretion, to any of the information contained herein.