

EVOLVE™ VERIFICATION MANAGEMENT

EVOLVE™ is a tool suite for hardware design and verification teams. EVOLVE™ provides a concept-to-tapeout flow for chip teams including build, test, regression, and release/configuration management. It includes an integrated live test plan, coverage tracking, historical execution tracking, analysis and reporting services

EVOLVE™ seamlessly integrates with existing infrastructure. The tool leverages legacy tests and assists in the transition to a coverage/random based effort. EVOLVE™ reduces debug time, enabling engineers to focus on solving the problem, not in recreating it. It facilitates team collaboration across the aisle and across the globe. EVOLVE™ can recover months of schedule in your chip design program.

RELEASES & CONTINUOUS INTEGRATION

Stop trying to maintain a clean 'Head-Of-Tree'. With the EVOLVE™ Release Server, users can check in files any time and release them to the team when ready. The team always has a quality controlled release to work from.

- Simple and efficient release management - no more maintaining a clean Head-Of-Tree
- Segment release integration into manageable chunks, and distribute it to individual owners
- Update workspace to any release. Retain local changes
- Compare (diff) releases and workspaces
- Break your design into multiple pieces. Divide and conquer the problem.
- Single issue or high performance pipeline mode
- Comprehensive administrator controls
- Enhance collaboration between team members through 'patches'
- Rich, searchable release history: who released which version of a file and when
- Protection from corrupting other user's workspaces

COVERAGE

EVOLVE™ has an integrated coverage management solution that automatically absorbs all the coverage information generated by each test and regression. Track coverage history by individual test, regression and release. Track by groups of these or by date.

- Support for Functional, Assertion, Property, and Line Coverage
- Automatically absorb coverage data generated by tests
- Coverage history linked to test history. Easily see which test generated what coverage
- Link test plan items to cover points
- Generate coverage trends

- Easily display uncovered items. Show items in the test plan that have not been covered
- Optimize regression suites based on coverage

LIVE TEST PLAN

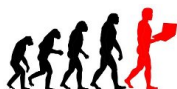
EVOLVE™ introduces an innovative **Live Test Plan**. Link test plan entries to verification elements: tests, regressions or cover points. Instantly see when the entry is complete.

- Capture effort in the test plan: Automatically roll up the number of days remaining
- Define tasks with prebuilt categories or add your own
- Add user defined fields to the test plan
- Test plan entries automatically reflect their current status by querying the Verification Database.
- Easily find orphaned test plan entries
- Easily find uncovered, untested entries, in recent releases or across the history of the project
- Analyze progress by looking into the execution history

BUILDS

EVOLVE™ provides a well defined and proven flow to create models (builds). Builds can be linked to tests allowing automatic or on-demand creation. Advanced features include:

- Smart-builds build only when needed
- Shared builds share executables across tests
- Build libraries share components across builds
- Build Protection: in use builds will not be overwritten
- Customizable paths and target directories
- Build Manifests: Know whether a clean repository checkout, or dirty workspace was used for a build
- Lock builds from edit and automatically notify on build changes
- Rich, searchable build history - who ran what, where, and with what version of the code



TESTS

EVOLVE™ provides a multi-phase test manager that can be used to run tests stand-alone or as part of a regression. EVOLVE™'s test manager manages all of constituent parts you would need to run, validate, and cleanup after a test. Advanced test features include:

- Multiple phases per test: build, prep, run, cleanup
- Independent check step after each phase
- Recreate failed test environment and rerun failed tests
- Link tests to predefined builds. Automatically build necessary binaries if they aren't already present.
- Tests are searchable on multiple criteria including tags, owner, date, and keywords.
- Lock tests from edit, and automatically notify on test changes
- Rich, searchable test history - who ran what, where, and with what version of the code

REGRESSIONS

A regression automatically groups tests and builds into a single, repeatable entity. EVOLVE™ supports many advanced regression features including:

- Resource optimization: shared builds in the regression run only once. Tests don't consume licenses until the pre-requisites are complete.
- Multi-stage regressions support. Run smoke tests first and only run extended tests once they pass.
- Create regressions of regressions
- Automatic rerun of failed tests with debug options
- Failure bucketization
- Monitor running regressions
- Kill all tests in a regression with a single command
- Lock regressions from edit and automatically notify on regression changes
- Rich, searchable regression history - who ran what, where and with what version of the code
- Create dynamic, populated on-the-spot regressions via the command line

TRACKING & REPORTING

Every invocation of EVOLVE™ automatically creates an entry in the **EVOLVE™ Verification Database**. The database provides a complete execution history of the project. This database, combined with complete version control and bug linking allows you to new dimensions of data mining.

- Go back in time to recreate any point in the project

- Easily see what has changed since tests passed or failed. This reduces debug time by pointing at potential bug culprits.
- File bugs and reproduce them from the command line in a single command.
- Rerun any build, test, regression exactly as the original at any time
- Data mine the execution history to profile tests, optimize regressions and increase productivity
- Organize verification by tags. EVOLVE™ allows user defined tags to be applied to each verification element. Search by the tags
- Compare execution statistics across projects
- Export data as XML, CSV, or sift through the raw data using SQL.

INTEGRATED GUI & COMMAND LINE INTERFACE

EVOLVE™ has a comprehensive Web 2.0 GUI. The command line interface provides quick and easy access to all of the commands.

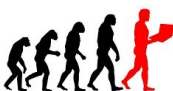
- Generate reports and plots
- List, run, or analyze the history of all available builds, tests and regressions. Edit from within the GUI or CLI
- Dashboard provides easy access to reports.
- Full help available from the command line.
- Compatible command syntax. Works like most tools already in use
- Terse and Full listings available on the CLI, with terminal width awareness
- Full debug mode available, logging comes standard
- Multi-level configuration: site, project, & user

SUPPORTS INDUSTRY STANDARD TOOLS

EVOLVE™ supports most industry standard tools. EVOLVE has a modular architecture that allows rapid deployment of additional support modules as needed.

- Subversion, Perforce, CVS
- Platform LSF, Sun Grid Engine, RTDA, Local, MLT Grid
- Centos, Red Hat Enterprise Linux, Fedora, Ubuntu, SUSE
- MySQL® & PostgreSQL
- Any Verilog/C/C++ Make/build process supported
- Bugzilla & Jira

Contact Missing Link Tools at sales@missinglinktools.com or call 720-246-1139 for more information.



EVOLVE™ COMPARISON CHART

	Cruise Control™	Cadence Incisive®	Atlassian Jira + Bamboo	Electric Cloud	Synopsys Discovery® Pioneer	EVOLVE™
Designed for Hardware		✓			✓	✓
Continuous Integration	✓		✓			✓
Easily Extensible	✓		✓			✓
Grid/Sim Farm Support	✓			✓		✓
Functional Coverage		✓			✓	✓
Failure Analysis		✓				✓
Bug Integration			✓			✓
Advanced Reporting				✓		✓
Legacy/directed test support						✓
Integrated Testplan						✓
Advanced Regression Support						✓
Advanced Tracking						✓

*Surveys conducted 1/12/2009, 3/14/2011

Copyright© 2009-2011, Missing Link Tools.

All Trademarks and copyrights are property of their respective owners.

