CLEANSCAPE
software international

qef

ADVANCED SOFTWARE CONSTRUCTION SYSTEM

Save money	reduce manual work and shorten your release cycle.
Meet deadlines	faulty or incomplete builds that delay production become history.
Prevent errors	production is no longer a fragile or error prone process.
Multi-platform	build scripts that work on any platform without fail.
Make progress	continue software development during a release build.
Comprehend	know the "how" of building your product.

Get *qef* — a replacement architecture for *make* — and make building software a profitable experience.

qef components

Configuration Variable Database

qef scans the entire source tree and builds a database of configurable variables. It also creates a complete dependency recipe for your build environment.

Script Preparation

qef prepares a set of scripts that are language and platform independent and passes them to the back end.

Back End

qef supports multiple back ends. qef supports *make*, Rdist, regression testing, shell and qef's own *make* facility, qmk.

Traceability

qef's log files provide complete information about the build process.Variable definitions, recipes, and dependencies are easily tracked.

"make"-ing it faster, better, more often…

"Siemens Nixdorf Toronto would not exist if it wasn't for ***qef***. It gives us a competitve edge."

… David Macklem, Mgr. Siemens Nixdorf Toronto

qef vs. make	
Promotes consistency across directories, projects and platforms	Does not promote or facilitate consistency between *make* files
Provides a convenient and powerful multi-directory process	Requires users to determine what phases need to be run and when
Provides a gentle nondestructive halt	Does not provide mechanism to gently halt *make* mid-flight
Provides a multi-directory trace	Does not provide mechanism for monitoring multiple *makes*
Provides trailers and headers to facilitate looking at multi-directory construction outputs	Does not provide convenient mechanism for multi-directory builds
Provides facilities for incorporating nonstandard process (even *make*)	Does not provide facilities for using anything but *make*
Completely tracks the build process to determine build problems	No such ability

If you can answer 'yes' to any of these questions, you need *qef*:

- Can you reproduce your product today as it was yesterday using yesterday's sources?
- Are you confident the product you build today is identical to the product you had yesterday?
- Can you attribute direct costs associated with incomplete or faulty builds?
- What confidence does your developers have that the version of the system they built during this development is consistent with the version that would be built as part of the entire project?
- Is it enough to be able to reproduce the sources using UNIX utilities?
- Does the construction system provide an appropriate consistency model?
- Is the construction system automated?
- Do you have complete representation of the product, partial product, or incremental build?
- Do your software developers and/or release engineers spend hours debugging a *make* file?
- Do you become frustrated tracing recipes, dependencies, and variable definitions embedded several levels deep inside your *make* tree?

4.00

1530 Meridian Avenue, Suite 200
San Jose, California 95125
www.cleanscape.net

CLEANSCAPE
software international

(408) 978-7000
Fax: (408) 978-7002
e-mail: sales@cleanscape.net