Alcatel·Lucent

Alcatel-Lucent nmake Product Builder

Enhancing the productivity of your development team

The Alcatel-Lucent nmake Product Builder allows developers to convert source code into a machine-executable product efficiently

Overview

In software development, product building is the process of converting source code into a machine-executable product. Typically, source code is integrated using a build process, which is repeated many times over the course of development. As such, the efficiency of the build process directly affects the productivity of the development team.

The nmake product is designed to provide you with an efficient build process that does the minimum work necessary to construct a new version of a product. It can significantly reduce the project build effort by shortening each phase of your product development cycle. Our product allows your developers to work more efficiently and reduce disk space by a factor of 2 using the Viewpathing feature, which enables each developer to share code while maintaining private development areas. By utilizing the years of expertise in solving product-building problems, nmake can help improve the productivity of your project team and minimize the work required to update a product.

Benefits

- Shortens each phase of the product development cycle, well as making it more robust
- Reduces build times by a factor of 4 to 6 through parall target script execution and remote building in a local ar network
- Decreases large makefile processing time by a factor of 10 when targets are up to date
- Reduces disk space by a factor of 2 due to viewpathing
- Decreases makefile size by a factor of 5 to 10
- Avoids errors so your product is of higher quality
- Reduces the effort for setting up builds and maintaining makefiles by a factor of 1.5 to 2
- High level build specification increases the build portability of your applications across compilers and platforms
- Decrease time to analyze project builds by a factor of 2 to 3 using formatted structured build logs

Features

- Viewpathing significantly decreases build times by sharing a project's common set of up-to-date generated files and allows a clean separation of platform-specific targets
- Provides access to common source files while maintaining private development areas.
- Automatic dependency generation determines implicit prerequisites dynamically, by means of a programmable scanning language
- Distributed and parallel execution supports product builds on both single processor and multiprocessor machines, which can be distributed over a network of workstations
- Efficient shell interface allows nmake to communicate with the shell via pipes. Actions are passed to the shell in a block without intervening backslash and semicolon characters
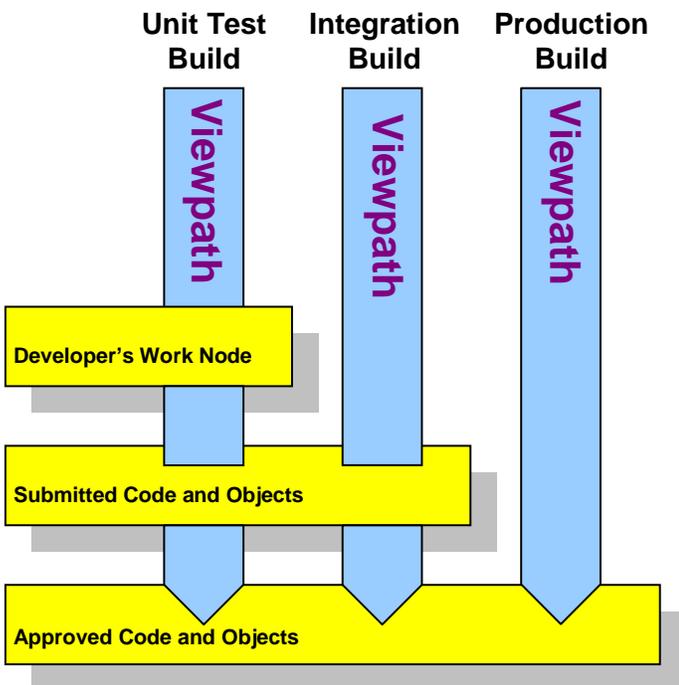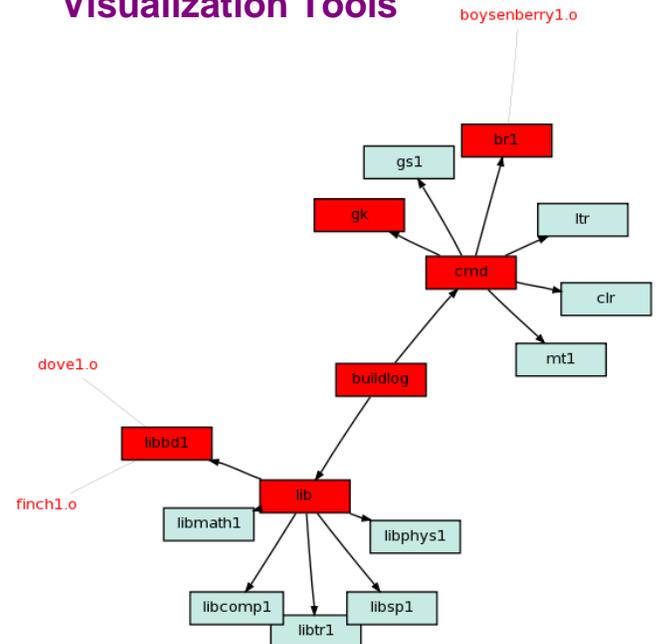
Features continued

- Powerful rule language uses variables, flow control and built-in attributes to allow fine-tuning
- Project-specific and user-specific rules, along with high level user-definable assertions, facilitate the creation of concise, consistent and flexible makefiles
- Efficient makefile processing improves accuracy and minimizes work by compiling makefiles and "remembering" the previous build status
- Language support provided for source files written in various languages (C, C++, JAVA, ESQL/C, IDL, FORTRAN, and others)
- Probe tool automatically detects and records platform and compiler information
- Optional Structured Build Log facilitates automated semantic build log processing and analysis
- Web based build logs provide high level and detail views into build to enhance visibility and diagnostics

To learn more about nmake please contact us at software@alcatel-lucent.com or visit our Web site at: http://nmake.alcatel-lucent.com.

Viewpathing Feature

Unit Test Build **Integration Build** **Production Build**

Viewpath   Viewpath   Viewpath

Developer's Work Node

Submitted Code and Objects

Approved Code and Objects

Visualization Tools

boysenberry1.o

br1

gs1

gk   ltr

cmd

clr

mt1

buildlog

dove1.o

libbd1

finch1.o   lib

libmath1   libphys1

libcomp1   libsp1

libtr1

Visualization tools are available to show file relationships, show job times, highlight build errors, and more. The above graph shows a simple build with failed targets highlighted in red. Rectangle nodes represent makefiles and plain text nodes are targets, and each node links to details. Images on page 1 show an HTML build log, a radial tree of build targets and a sunburst graph showing targets and their build times.

Alcatel·Lucent